LR-DWM: Efficient Watermarking for Diffusion Language Models

Ofek Raban*
Bar-Ilan University

Ethan Fetaya
Bar-Ilan University

Gal Chechik
Bar-Ilan University
NVIDIA

Abstract

Watermarking (WM) is a critical mechanism for detecting and attributing AI-generated content. Current WM methods for Large Language Models (LLMs) are predominantly tailored for autoregressive (AR) models: They rely on tokens being generated sequentially, and embed stable signals within the generated sequence based on the previously sampled text. Diffusion Language Models (DLMs) generate text via non-sequential iterative denoising, which requires significant modification to use WM methods designed for AR models. Recent work proposed to watermark DLMs by inverting the process when needed, but suffers significant computational or memory overhead.

We introduce *Left-Right Diffusion Watermarking* (*LR-DWM*), a scheme that biases the generated token based on both left and right neighbors, when they are available. LR-DWM incurs minimal runtime and memory overhead, remaining close to the non-watermarked baseline DLM while enabling reliable statistical detection under standard evaluation settings. Our results demonstrate that DLMs can be watermarked efficiently, achieving high detectability with negligible computational and memory overhead.

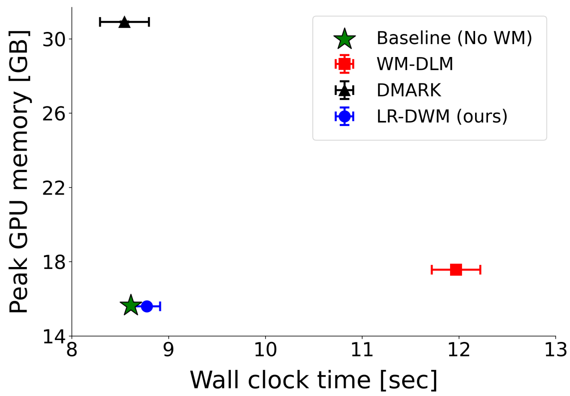

Figure 1: **Computational Efficiency.** Peak GPU memory versus wall-clock generation time on a single NVIDIA H100. The non-watermarked baseline (green star) is shown for reference.

1 Introduction

Watermarking has become a common technique for detection and attribution of text generated by Large Language Models (LLMs). Most existing watermarking schemes are designed for autoregressive (AR) models and critically rely on a simple but powerful property: tokens are generated in a fixed, deterministic left-to-right order. This known ordering allows the watermarking algorithm to use the previously sampled tokens to bias the generation of the next token, using a hash function to favor tokens from a "green" set. A detector can then count the frequency of "green" tokens and flag the text as LLM-generated if they are overrepresented in the text.

Diffusion Language Models (DLMs), receive increasing attention as a low-latency alternative to AR models, but they break the sequential assumption at its core. Instead of exposing tokens in a fixed left-to-right order, they generate text by iteratively denoising an entire sequence in a non-sequential order. At each step, different positions may be updated, and the schedule by which tokens are determined is not known in advance. As a result, one cannot bias a generated token based on tokens on its left, because they may still be undecided when we sample the current token. In this setting, the standard AR watermarking toolkit is invalid without substantial modification. Since DLMs offer unique advantages in parallel decoding, generation speed, and fine-grained contextual control, there is a need for reliable watermarking in this paradigm.

Very recently, (Gloaguen et al., 2025; Wu et al., 2025) adapted AR watermarking to DLMs by trying to invert the hashing function: Be "green" for the previous token, and also make the next token

*Corresponding author: ofekraban@gmail.com

Findings of the Association for Computational Linguistics: ACL 2026, pages 43510–43517

"green". This, however, incurs significant computational or memory overhead. In contrast, we propose a WM scheme that can be implemented with minimal runtime and memory overhead relative to the evaluated baseline.

Our key observation is that since generation order in diffusion models is no longer sequential, watermark constraints need not be restricted to a single causal direction. Instead, a token can be independently influenced by both its left and right neighbors whenever they are available. Based on this observation, we introduce *LR-DWM*: During generation, *LR-DWM* biases the logits in an additive way based on available neighboring tokens from each direction. This design naturally favors tokens that satisfy both left and right constraints, while remaining compatible with the unknown order of diffusion decoding.

The main contributions of this paper are:

(1) A novel two-sided watermarking method for diffusion language models. It embeds watermark signals by independently leveraging both left and right local context in an order-agnostic manner.

(2) We show that *LR-DWM* enables *efficient* watermarking for DLMs, incurring negligible runtime and memory overhead, while maintaining competitive detectability and text quality.

2 Technical Background

Watermarking for Language Models Watermarking for language models embeds a statistical signal by biasing token sampling during decoding using a secret key, typically by hashing previously generated tokens to partition the vocabulary into preferred (green) and non-preferred (red) sets. Then, boosting the green tokens yields a bias that can later be detected (Kirchenbauer et al., 2023; Hu et al., 2023; Zhao et al., 2023). A key objective is to balance detectability against quality degradation (Tu et al., 2024; Pan et al., 2024a).

Diffusion Language Models Diffusion Language Models (DLMs) generate text by iteratively denoising a corrupted token sequence (Ho et al., 2020; Song and Ermon, 2021; Austin et al., 2021; Lou and Ermon, 2024; Sahoo et al., 2024). Unlike AR models, DLM decoding updates tokens in a non-monotonic order, enabling parallel refinement and faster generation, and supporting both stochastic and deterministic decoding (Ye et al., 2025; Kim et al., 2025; Nie et al., 2025). Since generation does not follow a fixed left-to-right factorization, watermarking methods developed for AR decoding are not directly applicable to DLMs.

3 Related Work

Watermarking Diffusion Language Models Recent work has begun extending watermarking to diffusion language models (DLMs).

WM-DLM (Gloaguen et al., 2025) embeds watermarks by marginalizing, in expectation, over missing neighboring tokens during diffusion steps; this incurs substantial per-step computation due to repeated expectation and requires non-zero sampling temperature, as deterministic decoding ($T = 0$) eliminates sampling randomness and prevents watermark application.

DMARK (Wu et al., 2025) proposes an order-agnostic watermarking strategy that adapts AR-style watermark constraints to non-causal decoding via an inverse-style mechanism. When the token to the right of the currently decoded token is known, we bias the tokens that would make the right token green. In order to avoid computational overhead during generation, the method relies on $|\mathcal{V}|^2$ cached lookup tables, trading memory usage for runtime efficiency.

4 Problem Setup

We consider the problem of watermarking text generated by a Diffusion Language Model (DLM). The input to the system is a user-provided prompt c, and the output is a generated text sequence $y = (y_1, \ldots, y_T)$ from a vocabulary $\mathcal{V}$ sampled in an unknown order.

4.1 Watermarking Framework

To address the non-sequential nature of diffusion, we view watermarking as a set of local constraints defined over the final token sequence. At each denoising step t, the model predicts token distributions for positions that have not yet been finalized, and we modify these distributions via lightweight logit biasing. Specifically, let $l_v \in \mathbb{R}^{|\mathcal{V}|}$ denote the logit for token v; we add a bias δ whenever a watermark constraint induced by neighboring tokens is available. Importantly, these constraints are defined solely from the previously determined token. As a result, detection is schedule-agnostic and does not try to invert the hashing process.

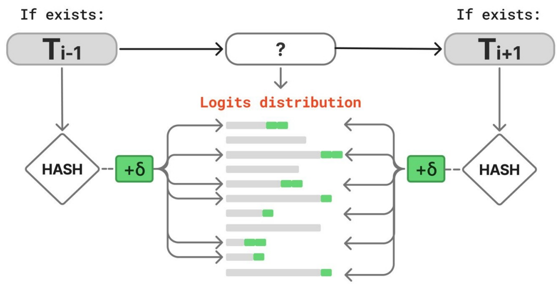

Figure 2: **Schematic of LR-DWM.** For a target token y_i, the model's logits distribution is modified using bidirectional local context. If available, the left neighbor y_{i-1} and the right neighbor y_{i+1} independently induce green-token constraints via hash functions with distinct keys k_L and k_R. Each constraint applies an additive bias δ to its corresponding green set. Boundary tokens fall back to a single-sided constraint when one neighbor is missing.

4.2 Left-Right Diffusion Watermarking

We introduce *LR-DWM*, an algorithm that induces *efficient* watermarking by anchoring signals to the known local neighbors.

Bidirectional Context Hashing. For a token at position i, we treat the left neighbor (y_{i-1}) and right neighbor (y_{i+1}) as independent sources of watermark signal. We hash twice, with secret keys k_L and k_R, to compute the set of green tokens:

$$G_L^{(i)} = \mathcal{H}(y_{i-1}, k_L), \quad G_R^{(i)} = \mathcal{H}(y_{i+1}, k_R) \quad (1)$$

where $\mathcal{H}$ uses a cryptographic hash function (Kirchenbauer et al., 2023). When a neighbor is unavailable, the green set is empty.

Injection Strategy. Since the constraints are applied independently, the total logit modification is the sum of individual biases:

$$l'_v = l_v + \delta \cdot \mathbb{I}(v \in G_L^{(i)}) + \delta \cdot \mathbb{I}(v \in G_R^{(i)}) \quad (2)$$

This additive mechanism naturally favors tokens that satisfy both constraints (when applicable), thereby maximizing the probability of generating tokens that yield a high positive detection score.

4.3 Detection Statistic

As each token has left and right watermarking signals, we unify the signals into a single ternary score s_i. Let $m_L = \mathbb{I}(y_i \in G_L^{(i)})$ and $m_R = \mathbb{I}(y_i \in G_R^{(i)})$ denote whether token y_i matches the green list induced by the left or right neighbor, respectively. We define the score for token y_i as:

$$s_i = m_L + m_R - 1\,, \quad (3)$$

yielding $s_i \in \{+1, 0, -1\}$. A score of $+1$ indicates token matches both neighbors (positive signal), 0 corresponds to a single-sided match (neutral), and -1 indicates no match (negative signal). Under the null hypothesis H_0 of the text being written by a person, token membership in green lists is random. Since the hash function partitions the vocabulary uniformly into two balanced sets, each match indicator is a Bernoulli random variable, yielding an expected score of zero: $\mathbb{E}[s_i] = 0$. It follows that $\mathrm{Var}(s_i) = 1/2$ under the random-hash null. While adjacent scores may be weakly dependent due to shared neighboring tokens, this dependence is local and handled via empirical calibration of σ^2 on human-written text.

The final detection statistic is the standardized sum over a sequence of length T:

$$Z = \frac{1}{\sigma\sqrt{T}} \sum_{i=1}^{T} s_i,$$

where σ^2 is estimated from human-written text. Under H_0, the resulting statistic is empirically well-approximated by a normal distribution, enabling standard Z-score–based thresholding with calibrated false positive rates.

To validate the statistical assumptions underlying our detection algorithm, we empirically evaluate the null hypothesis H_0 using 10,000 human-written texts of length 400 tokens sampled from the C4 corpus (Raffel et al., 2020). We set a threshold on the Z-score that empirically corresponds to 1% FPR and ensure that indeed we get 1% error rate on this human-written corpus.

5 Experiments

5.1 Experimental Setup

Model and Generation Setup. We evaluate our method on two state-of-the-art diffusion language models (DLMs): **LLaDA-8B-Instruct** (Nie et al., 2025) and **DREAM-7B-Instruct** (Ye et al., 2025). For each model, we adhere to the authors' recommended configurations, representing two distinct decoding paradigms: (1) **LLaDA** employs deterministic decoding with greedy refinement and a block length of 25; (2) **DREAM** uses its standard stochastic decoding setup. In both cases, we generate 300-token sequences over 300 diffusion steps. Results on **DREAM** are reported in the Appendix.

Datasets. We used the 600 prompts from **Water-Bench** (Tu et al., 2024). To ensure consistency with prior DLM watermarking evaluations and to mitigate diffusion-specific degeneration effects (e.g., repetitive loops), we adopt the post-generation filtering protocol of Gloaguen et al. (2025).

Metrics. We evaluate watermarking performance along three axes: **(1) Detectability,** measured as true positive rate (TPR) at a 1% false positive rate (FPR), following prior watermarking work, with human-written text treated as negatives; **(2) Text Quality,** measured via perplexity (PPL) using **Qwen2.5-32B** (Alibaba Qwen Team, 2024) as an external oracle

(3) Efficiency, measured by wall-clock generation time and peak GPU memory usage on a single **NVIDIA H100** GPU.

Baselines. We compare **LR-DWM** against the unmodified *vanilla* models and two recent diffusion watermarking methods: **WM-DLM** (Gloaguen et al., 2025), a stochastic expectation-based approach, and **DMARK** (Wu et al., 2025), a deterministic kernel-based method. We note that WM-DLM performance strongly depends on stochastic decoding (with high temperature), however for both LLaDa and DREAM DLMs, this means the base model it watermarks has considerably worse performence.

5.2 Main Results: Computational Efficiency

Figure 1 reports absolute wall-clock generation time and peak GPU memory usage for each method under identical experimental settings. The non-watermarked baseline is included for reference and exhibits identical efficiency under deterministic and stochastic decoding.

LR-DWM introduces minimal additional cost, remaining close to the non-watermarked deterministic baseline in both wall-clock time and memory usage. In contrast, DMARK incurs a substantially larger memory footprint due to caching the entire hash table in advance, nearly doubling peak GPU memory consumption. WM-DLM exhibits higher runtime overhead, reflecting the cost of its expectation-based scoring under stochastic decoding.

Overall, these results indicate that LR-DWM achieves watermarking with negligible computational overhead to its base generation configuration, which is the primary focus of this work.

Table 1: Effectiveness validation: estimated perplexity (PPL $\pm$ SEM) at three fixed detection rates.

Detection rates	**90%**	**99%**	**99.5%**
WM-DLM	5.07 ± 1.40	6.14 ± 1.84	6.33 ± 1.90
DMARK	2.82 ± 0.51	3.28 ± 0.61	3.34 ± 0.63
LR-DWM (OURS)	2.80 ± 0.46	3.32 ± 0.65	3.37 ± 0.66

Detectability vs Naturalness tradeoff We next verify that the efficiency gains of LR-DWM do not come at the expense of watermark effectiveness. Table 1 reports perplexity at fixed detection operating points on LLaDA, showing that LR-DWM maintains competitive detectability and text quality relative to existing diffusion watermarking methods, see full curve in the Appendix. On DREAM-7B, we observe a similar quality-detectability tradeoff, with comparable perplexity at matched detection rates (Appendix Figure 3).

5.3 Watermark Robustness

We evaluate robustness using the MarkLLM benchmark (Pan et al., 2024b) under standard non-adaptive text perturbations. At a fixed operating point achieving 100% detection on clean text, LR-DWM retains high detection rates under most moderate lexical attacks, including 98.8% detection under 10% word deletion and 99.4% under 10% context-aware word substitution. The substantial drop under paraphrasing reflects the disruption of the local bidirectional lexical context exploited by LR-DWM, a known limitation shared by watermarking methods based on local token correlations.

Detailed robustness results across attack types and intensities are reported in Appendix A (Table 2).

6 Conclusion

We introduced LR-DWM, a lightweight watermarking scheme designed for diffusion LMs that does not rely on tokens being generated in sequential order. In contrast with existing methods, LR-DWM is both fast and memory efficient: Across two diffusion models, LR-DWM demonstrates minimal encoding and inference overhead relative to the evaluated baselines, while maintaining detectability and naturalness comparable to existing diffusion watermarking methods under standard non-adaptive evaluation settings.

Limitations

Our method shares limitations with prior watermarking approaches. While the impact on text quality is typically modest, it increases at very high detection rates. Reliable detection also requires sufficiently long text. We evaluate robustness under standard perturbations and an adaptive paraphrasing attack; however, stronger adversarial strategies may further reduce detectability.

References

Alibaba Qwen Team. 2024. Qwen2.5: Technical report. *arXiv preprint arXiv:2412.15115*.

Jacob Austin, Daniel Johnson, Jonathan Ho, Daniel Tarlow, and Aaron van den Oord. 2021. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. 2025. Watermarking diffusion language models. *arXiv preprint arXiv:2509.24368*.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*.

Xinyu Hu, Yuxin Zhang, Jing Li, Zhi Wang, and Zihan Liu. 2023. Variance-reduced watermarking for large language models. In *Advances in Neural Information Processing Systems*.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. 2025. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *Proceedings of the 42nd International Conference on Machine Learning*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*.

Aaron Lou and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratio of the data distribution to the noise distribution. In *Proceedings of the 41st International Conference on Machine Learning*.

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*.

Junjie Pan, Ruiyi Zhang, Zhen Chen, and Ming Li. 2024a. On the reliability of watermarks for large language models. *arXiv preprint*.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, and 1 others. 2024b. Markllm: An open-source toolkit for llm watermarking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Colin Raffel, Noam Shazeer, Adam Roberts, and 1 others. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*.

Yang Song and Stefano Ermon. 2021. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Waterbench: Towards holistic evaluation of watermarks for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Linyu Wu, Linhao Zhong, Wenjie Qu, Yuexin Li, Yue Liu, Shengfang Zhai, Chunhua Shen, and Jiaheng Zhang. 2025. Dmark: Order-agnostic watermarking for diffusion large language models. *arXiv preprint arXiv:2510.02902*.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b. `https://hkunlp.github.io/blog/2025/dream`. Technical report / project page.

Xuandong Zhao, Zhiyuan Wu, Tianyu Xie, and Yuanzhi Zhang. 2023. Provable robust watermarking for large language models. In *Advances in Neural Information Processing Systems*.

7 Appendix

7.1 Algorithm Pseudocode

This section provides additional algorithmic details for LR-DWM, complementing the high-level description in the main paper.

Algorithm 1 LR-DWM Diffusion Decoding with Two-Sided Watermarking (Simplified)

Input: Prompt c, Model $\mathcal{M}$, Keys k_L, k_R, Bias δ
Output: Generated sequence y

Initialize $y^{(S)}$ as a fully masked or corrupted token sequence
for $s = S, \dots, 1$ **do**
  $\mathcal{I}_s \leftarrow$ positions selected for update by the diffusion schedule at step s
  for each position $i \in \mathcal{I}_s$ **do**
    $\ell \leftarrow \mathcal{M}(y, i)$ ▷ *Base logits*
    if y_{i-1} is revealed **then**
      $M_L \leftarrow \textsc{GreenMask}(y_{i-1}, k_L)$
      $\ell \leftarrow \ell + \delta \cdot M_L$
    if y_{i+1} is revealed **then**
      $M_R \leftarrow \textsc{GreenMask}(y_{i+1}, k_R)$
      $\ell \leftarrow \ell + \delta \cdot M_R$
    $y_i \sim \textsc{Decode}(\ell)$
return y

7.2 Detailed Results

7.2.1 Trade-off Curves comparison on LLaDA

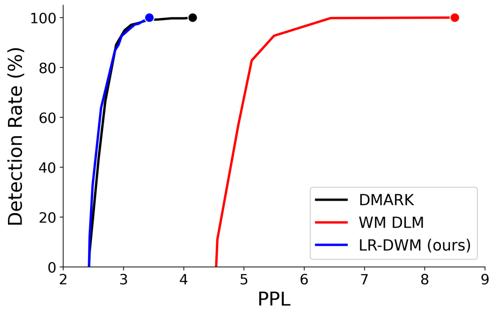

Figure 3: **Quality-Detectability Trade-off.** Detection rate at a FPR of 1% versus perplexity (PPL) across a range of bias strengths δ. LR-DWM exhibits a competitive trade-off relative to DMARK and WM-DLM.

7.2.2 Quality-Detectability Trade-off on DREAM

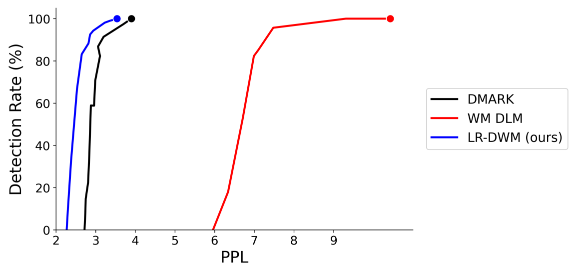

Figure 4: **Quality-Detectability Trade-off on DREAM.** Detection rate versus perplexity (PPL) on the DREAM model for LR-DWM and prior watermarking methods, evaluated under stochastic decoding with low temperature. Each point corresponds to a different watermark strength δ. LR-DWM exhibits a sharp detectability transition at lower PPL values, closely matching the trend observed on LLaDA and indicating consistent generalization across diffusion-based language models.

Generation Setup for DREAM. Results for LR-DWM and DMARK shown in Figure 4 were obtained using the decoding configuration recommended by the DREAM authors. We generate sequences of length 300 using 300 diffusion steps, with entropy-based stochastic decoding and a deliberately low temperature setting. Specifically, we use: `GEN_LENGTH=300`, `STEPS=300`, `TEMPERATURE=0.2`, `ALG=entropy`, `ALG_TEMP=0.0`, `EPS=1e-3`, `TOP_P=0.95`, and no top-k truncation.

Due to the combination of low-temperature decoding and diffusion-based sampling, generation quality is more sensitive to standard quality and length filters, and a substantial fraction of generated texts did not pass these filters. All detection results are therefore computed over the surviving subset. We apply the same generation and filtering procedure consistently across LR-DWM and DMARK on DREAM.

For WM-DLM, the decoding configuration used above did not produce stable generations. We therefore report WM-DLM results using the decoding setup recommended in the original WM-DLM work, while keeping all evaluation metrics and filtering criteria identical.

Finally, we observe a larger gap between LR-DWM and DMARK on DREAM than on LLaDA. We attribute this difference primarily to decoding dynamics and filtering effects under low-temperature diffusion sampling, rather than to changes in the underlying watermark signal.

Table 2: **Robustness of LR-DWM under standard and adversarial attacks.** Detection performance under standard text perturbations at a fixed false positive rate (FPR) of 1%. The baseline corresponds to clean watermarked text generated with $\delta = 3.25$, achieving 100% detection and an average Z-score of 6.653. Reported drops are relative to this baseline. Back-translation uses local OPUS-MT models (EN→ZH→EN).

Attack Type	**Param.**	**Detection (%)**	**Drop (%)**	**Avg. Z**	**Z Drop**
Random token replacement	10%	99.12	0.88	5.35	1.30
	20%	94.15	5.85	4.22	2.43
	30%	76.02	23.98	3.27	3.38
Word deletion	10%	98.83	1.17	5.45	1.20
	20%	96.78	3.22	4.74	1.91
	30%	90.94	9.06	3.91	2.74
	40%	74.85	25.15	3.13	3.52
	50%	59.06	40.94	2.55	4.10
Word substitution	10%	98.25	1.75	5.29	1.36
	20%	95.32	4.68	4.49	2.16
	30%	86.84	13.16	3.73	2.92
	40%	70.47	29.53	3.06	3.59
	50%	51.46	48.54	2.34	4.31
Context-aware substitution (BERT)	10%	99.42	0.58	5.83	0.82
	20%	98.83	1.17	5.48	1.18
	30%	98.54	1.46	5.31	1.34
Back-translation (EN→ZH→EN)	-	55.56	44.44	2.48	4.17
Paraphrasing	standard	50.88	49.12	2.53	4.12
Adversarial paraphrasing	adaptive attacker	15.79	84.21	1.02	5.63

7.3 Robustness to Standard and Adversarial Attacks

We evaluate robustness under both standard (non-adaptive) and adversarial text perturbations using the MARKLLM benchmark. Table 2 reports detection performance at a fixed operating point calibrated to a 1% false positive rate (FPR) on clean text.

We distinguish between two paraphrasing regimes. Standard paraphrasing is performed using a smaller language model with a generic rewriting prompt, and represents a non-adaptive transformation. In contrast, adversarial paraphrasing is generated using Qwen-32B-Instruct, with an explicit prompt that encourages rewriting the text using different words and sentence structures in order to disrupt watermark detection.

Paraphrasing is the most damaging attack, substantially reducing detectability by altering local lexical structure, with a significantly stronger effect under the adversarial setting.

In contrast, *light attacks* such as word deletion, random token replacement, and word substitution preserve high detection rates, indicating a distributed watermark signal. Robustness under *context-aware substitution* is partly explained by a benchmark limitation, as WordNet-based substitutions affect only tokens with available synonyms.

7.4 Distributional Effects Across Diffusion Steps

A key concern in diffusion watermarking is whether the injected bias accumulates across denoising steps and distorts the token distribution. In LR-DWM, this does not occur: each token receives an additive logit bias only once, at the step it is decoded, and is not resampled afterward. Consequently, the perturbation remains local to each decode event and does not compound across diffusion steps.

To quantify the distributional shift, we measure the per-decode KL divergence between the base and watermarked token distributions at $\delta = 2.5$ ($\sim 99\%$ TPR at 1% FPR). The mean KL is 0.245

nats (median: 0.074, 95th percentile: 0.72). Only 2.23% of decoding events exceed 1 nat and fewer than 0.4% exceed 2 nats, indicating that large shifts are rare and localized.

Overall, these results indicate that LR-DWM introduces a small, controlled perturbation that remains local and does not accumulate across diffusion steps.

7.5 Sensitivity to Asymmetric Watermark Strengths

In the main experiments we use a symmetric watermark strength δ for both the left and right contexts. However, the LR-DWM formulation does not require these biases to be equal. To examine the sensitivity of the method to asymmetric configurations, we perform a grid search over pairs of watermark strengths (δ_L, δ_R).

Table 3 reports the resulting detection statistics and text quality. As expected, detectability increases smoothly as the total bias magnitude grows. Several asymmetric configurations achieve high detection rates. For example, $(\delta_L = 2.5, \delta_R = 2.0)$ yields 96.6% TPR, while $(\delta_L = 3.0, \delta_R = 2.0)$ achieves 99.7% TPR.

Perplexity increases predictably with stronger bias values but remains stable across asymmetric settings. We observe no instability or degradation specific to asymmetric configurations. These results indicate that LR-DWM does not require symmetric bias values; the symmetric configuration used in the main experiments was chosen primarily for simplicity.

δ_L	δ_R	Avg Z	TPR (%)	PPL
1.0	2.0	2.583	58.5	2.670
1.0	2.5	2.852	64.7	2.768
1.0	3.0	3.064	70.0	2.848
2.0	1.0	3.827	86.7	2.881
2.0	2.5	4.580	93.8	3.057
2.0	3.0	4.739	94.6	3.159
2.5	1.0	4.509	91.1	3.102
2.5	2.0	5.027	96.6	3.166
2.5	3.0	5.487	98.7	3.347
3.0	1.0	5.260	98.3	3.356
3.0	**2.0**	**5.839**	**99.7**	3.532
3.0	2.5	6.126	99.7	3.621

Table 3: Grid search over asymmetric watermark strengths (δ_L, δ_R). Detection improves smoothly as the total bias magnitude increases, while perplexity grows predictably. High detection rates are achieved under several asymmetric configurations, indicating that LR-DWM does not rely on symmetric bias values.

7.6 Ablation: Left, Right, and Bidirectional Watermarking

We compare left-only, right-only, and bidirectional variants of LR-DWM under two decoding regimes: max-confidence and random unmasking.

Under max-confidence decoding, the diffusion model exhibits a strong left-context bias, resulting in behavior that is close to autoregressive. In this regime, left-only watermarking becomes competitive. However, LR-DWM remains at least as effective at comparable operating points. For example, at $\delta = 1.25$, LR-DWM achieves 63.8% TPR (PPL=2.63), compared to 61.1% for left-only at $\delta = 1.5$ (PPL=2.66). At higher operating points, LR-DWM achieves higher detection; for instance, at $\delta = 2.0$, it reaches 92.6% TPR (PPL=2.97), compared to 90.3% for left-only at $\delta = 2.25$ (PPL=2.92).

Under random unmasking, which removes the left-to-right bias, the advantage of LR-DWM becomes substantial. At $\delta = 2.0$, LR-DWM achieves 80.5% TPR (PPL=4.01), compared to 43.4% for left-only (PPL=3.80) and 30.4% for right-only (PPL=3.78). This gap persists across strengths; for example, at $\delta = 1.5$, LR-DWM reaches 58.1% TPR, while left-only achieves 29.1% and right-only 19.2%. At higher strengths, LR-DWM reaches 94.7% TPR at $\delta = 3.0$ (PPL=4.64), whereas left-only requires $\delta = 5.0$ (PPL=4.91) for comparable detection.

Overall, left-only watermarking is competitive primarily in near-autoregressive regimes induced by max-confidence decoding. In contrast, LR-DWM remains consistently strong across decoding orders, supporting its order-agnostic design.