Beyond Fully Random Masking: Attention-Guided Denoising and Optimization for Diffusion Language Models

Jia Deng[1,5], **Junyi Li**[2*], **Wayne Xin Zhao**[1,5*], **Jinpeng Wang**[3], **Hongyu Lu**[4], **Ji-Rong Wen**[1,5]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Department of Data Science, City University of Hong Kong
[3]Meituan [4]WeChat, Tencent
[5]Beijing Key Laboratory of Research on Large Models and Intelligent Governance
dengjia0510@outlook.com, junyili@cityu.edu.hk, batmanfly@gmail.com

Abstract

Diffusion large language models (dLLMs) offer an efficient alternative to autoregressive models through parallel decoding, yet existing post-training methods largely rely on random masking strategies that overlook intrinsic token dependencies. In this work, we present an empirical analysis of attention in dLLMs and show that tokens attending more strongly to unmasked context exhibit greater generation stability and play a critical role in reasoning. Motivated by these findings, we propose **AGDO**, an attention-guided denoising and optimization framework that aligns both training and optimization with attention-derived dependencies. AGDO determines the denoising order based on attention structure and emphasizes attention-critical tokens during supervised fine-tuning and reinforcement learning. Experiments on mathematical and coding benchmarks demonstrate that AGDO consistently improves reasoning performance, outperforming state-of-the-art post-training methods for dLLMs.

1 Introduction

Diffusion large language models (dLLMs) have recently emerged as a promising alternative to autoregressive (AR) models for language modeling (Li et al., 2025c). Unlike AR models, which generate tokens sequentially from left to right, dLLMs iteratively denoise and decode tokens in parallel, offering substantial efficiency advantages during inference (Labs et al., 2025; Gong et al., 2025). Recent models such as LLaDA (Nie et al., 2025; Zhu et al., 2025) and Dream (Ye et al., 2025) have demonstrated performance competitive with state-of-the-art AR models (AI@Meta, 2024; Team et al., 2024) of comparable scale, highlighting the growing potential of diffusion-based language modeling.

Despite these advances, effectively post-training dLLMs remains challenging. Existing post-training approaches for full-attention dLLMs, including diff-GRPO (Zhao et al., 2025) and wd1 (Tang et al., 2025), typically rely on randomly masking tokens and optimizing the model over the masked positions. While simple and efficient, such strategies fail to align with the actual inference dynamics of dLLMs, leading to a mismatch between training and inference. Several recent works attempt to reduce this discrepancy by introducing different masking strategies. Blockwise supervised fine-tuning (Sun et al., 2025) adopts a semi-autoregressive unmasking order, while other approaches (Wang et al., 2025b) follow the natural left-to-right generation order during training. Although these methods improve training stability and efficiency, they still impose externally defined decoding orders and overlook a critical property of full-attention dLLMs: under bidirectional attention, token dependencies are not strictly determined by positional order, but emerge dynamically through attention interactions.

To better understand these intrinsic dependencies, we conduct an empirical analysis of attention patterns in dLLMs. Our analysis reveals two key observations. First, attention distributions exhibit strong sparsity and temporal consistency across denoising steps, indicating that each token consistently relies on a small and stable set of context tokens. Second, tokens that attend more heavily to already denoised tokens exhibit significantly higher probability stability during generation, suggesting that the denoising order plays a crucial role in maintaining generation reliability. These findings imply that an effective training strategy for dLLMs should explicitly align the denoising trajectory with attention-induced token dependencies.

Motivated by these insights, we propose **AGDO** (**A**ttention-**G**uided **D**enoising and **O**ptimization), a two-stage post-training framework that explicitly aligns the denoising trajectory of dLLMs with intrinsic attention structure. AGDO first derives

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44502–44514

an attention-guided denoising order from valid attention scores, which govern the token sequence to be unmasked during training. By ensuring that tokens are denoised only after sufficient attention-supported context is available, this denoising order forms the foundation of AGDO. On top of this method, AGDO performs supervised fine-tuning (**AGDO-SFT**) and reinforcement learning (**AGDO-RL**), while further emphasizing attention-hub tokens through an attention-based re-weighting strategy.

Extensive experiments on challenging mathematical and coding benchmarks demonstrate that AGDO significantly improves the reasoning capabilities of masked dLLMs, consistently outperforming existing post-training methods. These results validate the importance of aligning training dynamics with intrinsic attention dependencies and highlight attention-guided denoising as a principled design choice for dLLMs.

2 Related Work

Diffusion Large Language Models. DLLMs recently attract attention as an alternative to AR models (Li et al., 2025c). Early attempts to apply diffusion to text employ continuous latent representations (Gong et al., 2022; Li et al., 2022a). Masked diffusion, where masked tokens are iteratively predicted, proves more scalable (Ye et al., 2025; Nie et al., 2025; Cheng et al., 2025a). Recent models such as DiffuLlama (Gong et al., 2024) and Dream (Ye et al., 2025) adapt pre-trained language models with masked diffusion objectives. In contrast, LLaDA (Nie et al., 2025) shows that training large diffusion-based language models from scratch using full-attention achieves performance comparable to AR models like Llama (Dubey et al., 2024). Most recent work focuses on accelerating inference for dLLMs, introducing strategies such as parallel decoding (Gao et al., 2025), speculative decoding (Cheng et al., 2025b) and KV-caching (Wu et al., 2025; Liu et al., 2025). While previous work primarily focuses on pre-training and inference acceleration for dLLMs, we focus on post-training methods to enhance their reasoning capabilities.

DLLMs Reasoning Advancement. Current approaches to enhancing reasoning in dLLMs mainly center around post-training and inference stages. Inference-time methods such as remasking-based optimization (Li et al., 2025b; Bao et al., 2025) and variable-length adaptation (Li et al., 2025a) enable the modeling of longer reasoning chains and utilize the bidirectional context of diffusion models. In the post-training stage, reinforcement learning methods such as group-based advantage estimation and preference optimization are explored (Wang et al., 2025a; Tang et al., 2025; Gong et al., 2025). Additionally, diffu-GRPO (Zhao et al., 2025) applies mean-field approximation to improve computational efficiency. LLaDA 1.5 (Zhu et al., 2025) addresses variance in Evidence Lower Bound (ELBO) based likelihood estimation by employing preference optimization. However, these approaches often overlook the subtle dependencies among tokens that complex reasoning tasks require. Instead, our approach explicitly incorporates attention-derived signals to identify and leverage token dependencies during post-training.

3 Preliminaries

3.1 DLLMs with Full Attention

Unlike autoregressive models that employ causal masking, dLLMs (Nie et al., 2025; Ye et al., 2025) employ bidirectional self-attention, allowing each token to attend to the entire sequence. Specifically, for the l-th layer and the h-th attention head, the attention score matrix $A^{(l,h)}$ is computed as:

$$A^{(l,h)} = \text{softmax}\left(\frac{Q^{(l,h)}K^{(l,h)\top}}{\sqrt{d_k}}\right), \quad (1)$$

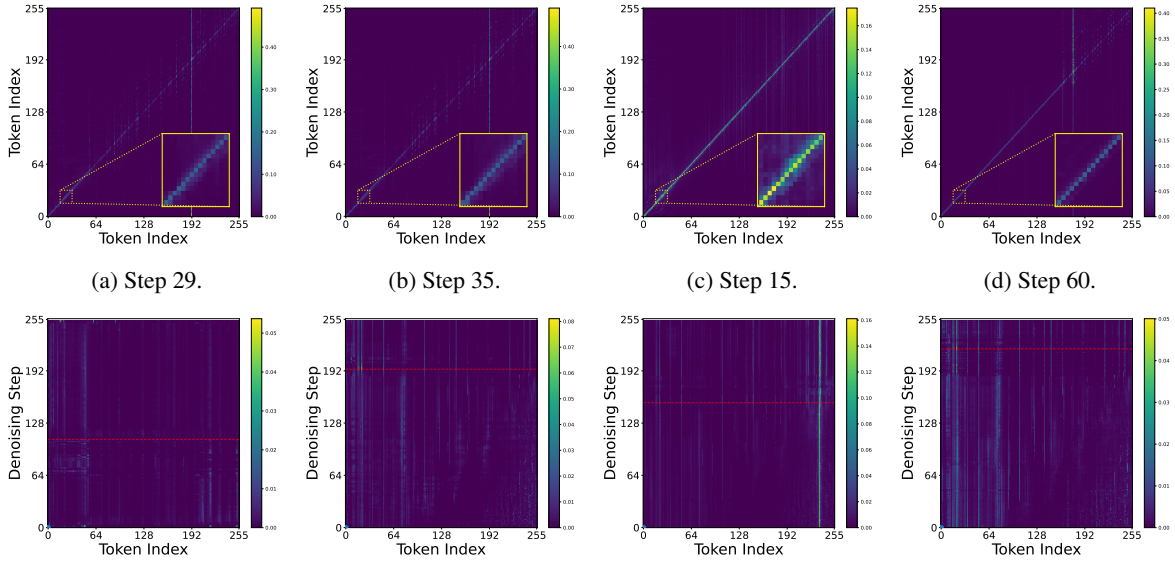
where no attention mask is used, thereby enabling full contextual visibility across all positions.

During the forward process, dLLMs progressively corrupt the original sequence x_0 into a noisy sequence x_t, with the time $t \in [0, 1]$ controlling the corruption level. Each token is independently masked according to the transition distribution:

$$q_{t|0}(x_t^i \mid x_0^i) = \begin{cases} \beta_t, & x_t^i = x_0^i \\ 1-\beta_t, & x_t^i = \texttt{[MASK]} \end{cases}, \quad (2)$$

where β_t decreases with t, and $\beta_1 = 0$ at $t = 1$, indicating the sequence is fully masked. The training objective is to reconstruct the masked tokens conditioned on the partially corrupted sequence x_t. Let $\mathcal{M}_t$ denote the set of token indices that are masked at time step t. The model is trained by minimizing a weighted negative log-likelihood, corresponding to the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{t,x_0,x_t}\left[\frac{1}{t}\sum_{k\in\mathcal{M}_t}\log f_\theta(x_0^k|x_t)\right]. \quad (3)$$



(a) Step 29. (b) Step 35. (c) Step 15. (d) Step 60.

(e) Token 61. (f) Token 195. (g) Token 154. (h) Token 220.

Figure 1: Attention dynamics during the denoising process on Dream-v0-Instruct-7B (Ye et al., 2025). Figure (a) to (d) display the attention distributions among tokens at randomly selected denoising steps. Figure (e) to (h) illustrate how the attention of a randomly chosen token towards other tokens changes throughout the denoising process. The red dashed lines indicate the steps at which the corresponding token is unmasked.

3.2 Policy Optimization for DLLMs

Reinforcement learning (RL) for language models is commonly implemented using Proximal Policy Optimization (PPO) (Schulman et al., 2017), which optimizes a policy π_θ by maximizing the objective:

$$\mathbb{E}_{q,o}\Bigg[\sum_{t=1}^{|o|}\min\Big(r_t\hat{A}_t,\ \text{clip}(r_t,\ 1-\epsilon,\ 1+\epsilon)\hat{A}_t\Big) - \beta\cdot\text{KL}\big[\pi_\theta(\cdot\mid q,o_{<t})\,\|\,\pi_{\text{ref}}(\cdot\mid q,o_{<t})\big]\Bigg], \quad (4)$$

where r_t is the probability ratio between the current policy π_θ and old policy π_{old}, and $\hat{A}_t$ is the estimated advantage.

To reduce the complexity of PPO, recent work (Shao et al., 2024) proposed Group Relative Policy Optimization (GRPO), which eliminates the need for a learned value function by estimating advantages via group-wise normalization. Specifically, given a group of G sampled responses $\{o^1,\ldots,o^G\}$ with corresponding rewards $\{R^1,\ldots,R^G\}$, the advantage assigned to the i-th response is computed as:

$$\hat{A}_t^i = \frac{R^i - \text{mean}(\{R^j\}_{j=1}^G)}{\text{std}(\{R^j\}_{j=1}^G)}. \quad (5)$$

However, applying GRPO to dLLMs presents non-trivial challenges. Unlike AR models (Yang et al., 2025; Dubey et al., 2024), dLLMs do not admit a sequential factorization over tokens, which complicates the computation of token-level likelihood ratios and KL regularization terms required for policy optimization. Recent approaches address this by adopting mean-field approximations, enabling efficient single-pass estimation of both importance weights and KL divergence for policy optimization (Zhao et al., 2025; Tang et al., 2025).

4 Attention Analysis in DLLMs

In this section, we conduct an empirical analysis of the attention mechanism in dLLMs to uncover intrinsic token dependencies during the reasoning process. For clarity, we focus on the final layer, denoted as layer L, where semantic dependencies are typically most pronounced. Our primary analysis is performed on Dream-v0-Instruct-7B. To demonstrate the generality of our findings, we further validate the results on LLaDA in Appendix A.

4.1 Attention Patterns in DLLMs

To gain deeper insights into attention distributions in dLLMs, we utilize queries from the MATH-500 benchmark (Hendrycks et al., 2020) and trace the temporal evolution of attention score maps (as defined in Equation 1) in the final layer across the denoising trajectory. We average the attention scores across all heads and visulize them in Figure 1.

Horizontal sparsity across steps. Figures 1 (a)-(d) show that attention weights in dLLMs exhibit a consistently sparse structure throughout the denoising process. In particular, most tokens primarily attend to themselves and their immediate neighbors, forming prominent diagonal patterns similar to those observed in AR models (Xiao et al., 2023; Hsieh et al., 2024). In addition, we observe distinct vertical structures manifested as bright columns, indicating that a small subset of key tokens attracts concentrated attention from many other positions. Notably, these sparse patterns remain stable across different denoising steps, corroborating prior observations in recent work (Song et al., 2025).

Vertical consistency across steps. Figures 1 (e)-(h) further reveal strong temporal consistency in token-wise attention patterns across denoising steps. Although the query, key, and value representations are recomputed at each inference step, a given token attends to largely the same set of tokens. This behavior appears as persistent vertical lines in the attention maps, suggesting that the most salient contextual dependencies for each token are largely invariant to the masking state. This finding is again consistent with the analysis in Song et al. (2025).

4.2 Impact of Valid Attention

The observations in Section 4.1 motivate us to examine how the denoising level of the attended tokens influences the current token. To quantify this effect, we define the *valid attention score* of token i at its denoising step as:

$$S_i = \sum_{k \in \mathcal{U}} \left(\frac{1}{H} \sum_{h=1}^{H} A_{i,k}^{(L,h)} \right), \tag{6}$$

where $A_{i,k}^{(L,h)}$ denotes the normalized attention score from token i to token k in the layer L and head h, and $\mathcal{U}$ denotes the set of already unmasked tokens. Furthermore, we track the probability evolution of each token from its initial denoising step until the end of the inference process. We define the probability change ΔP_i for token i as:

$$\Delta P_i = P_i^{\text{min}} - P_i^{\text{denoise}}, \tag{7}$$

where P_i^{denoise} is the generation probability of token i at its denoising step, and P_i^{min} is the minimum probability for token i observed during the subsequent inference process.

Figure 2 illustrates the distribution of valid attention scores S and the corresponding average probability change ΔP. We observe a clear positive correlation between S and ΔP. Specifically, tokens that allocate more attention to already unmasked tokens (i.e., higher S_i) exhibit greater stability, in the sense that their probabilities are less likely to decrease as subsequent tokens are generated.

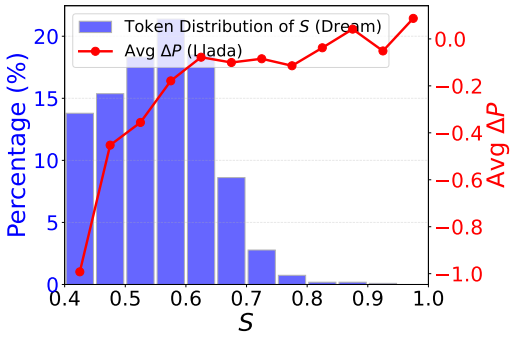


Figure 2: Relationship between the valid attention score S and the average probability change ΔP.

To further explore the influence of S on generation quality, we introduce a hybrid token selection strategy that augments the probability-based baseline with the valid attention score S. At each denoising step, we first construct a candidate set consisting of masked tokens whose probabilities exceed 0.9. Among these candidates, we select the token with the highest S value to be unmasked.

As shown in Table 1, this S-aware denoising strategy consistently outperforms the probability-only baseline across different predefined sequence lengths on MATH-500. These results indicate that incorporating valid attention information leads to a denoising order that better captures intrinsic token dependencies, thereby producing more reliable and higher-quality generations.

Strategy	128	256	512
Max-Prob	16.8	13.8	13.0
Max-S	21.2	16.2	15.2

Table 1: Accuracy on MATH-500 under static sampling with different predefined sequence lengths. Max-Prob selects the token with the highest probability, while Max-S selects the token with the highest valid attention score among high-probability candidates.

5 Approach

Motivated by our attention analysis in Section 4, which reveals strong structural sparsity, temporal

consistency, and stability-aware token dependencies in dLLMs, we propose **AGDO**, an *Attention-Guided Denoising and Optimization* framework. AGDO explicitly aligns training and optimization with intrinsic attention dependencies. Under a shared attention-guided denoising order, AGDO supports two training regimes: **AGDO-SFT**, which performs attention-guided supervised fine-tuning, and **AGDO-RL**, which applies attention-guided denoising optimization.

5.1 Attention-Guided Denoising Order

Our analysis in Section 4.1 shows that tokens allocating more attention to already unmasked tokens exhibit significantly higher probability stability during the denoising process. This observation suggests that the denoising order in dLLMs should respect attention-induced token dependencies.

To operationalize this insight, we construct an attention-guided denoising order based on the valid attention score S defined in Equation 6. For a complete generation trajectory, we perform a single forward pass at the final denoising timestep to obtain the attention score matrix in the last layer. The set of unmasked tokens $\mathcal{U}$ is initialized with all prompt tokens. At each denoising step t, we select the top-n tokens with the highest S scores and add them to $\mathcal{U}$. In the next step, valid attention scores are recomputed for the remaining tokens based on the updated $\mathcal{U}$. This procedure is repeated until all tokens are assigned to a denoising step.

By denoising tokens after sufficient attention-supported context is available, this ordering aligns the generation trajectory with intrinsic dependency structures captured by attention. The detailed algorithm is provided in Appendix B.1.

5.2 Attention-Guided Fine-Tuning

Existing supervised fine-tuning (SFT) methods for dLLMs typically rely on either random masking or fixed semi-autoregressive masking strategies (Zhao et al., 2025; Sun et al., 2025) to compute cross-entropy loss on a preselected subset of tokens. Yet, such approaches ignore the attention-induced dependency structure revealed by our analysis.

In contrast, AGDO-SFT follows the attention-guided denoising order described in Section 5.1. At each timestep t, we randomly mask only the tokens assigned to that denoising step, ensuring that training conditions mirror the intended inference trajectory. Moreover, inspired by findings on attention centrality in AR models (Lin et al., 2024; Li et al., 2025d), we further distinguish tokens by their influence on the rest of the sequence. For each token k, we define an *influence score* I_k as the total attention it receives from other tokens:

$$I_k = \sum_i \left(\frac{1}{H} \sum_{h=1}^{H} A_{i,k}^{(L,h)} \right), \tag{8}$$

where $A_{i,k}^{(L,h)}$ denotes the attention weight from token i to token k in the final layer L.

Tokens with higher I_k function as attention hubs and exert disproportionate influence on the generation of other tokens. To reflect this, we weight the cross-entropy loss at denoising step t as:

$$-\mathbb{E}_{t,x_0,x_t}\left[\frac{1}{|\mathcal{U}_t|}\sum_{k\in\mathcal{U}_t}(1+\gamma_k I_k)\log f_\theta(x_0^k \mid x_t)\right]. \tag{9}$$

This design aligns both the masking schedule and the loss weighting with attention structure, enabling the model to focus on tokens that are most critical for global consistency. The full procedure is described in Appendix B.2.

5.3 Attention-Guided Policy Optimization

We further extend AGDO to reinforcement learning under the GRPO algorithm. As in AGDO-SFT, denoising follows the attention-guided order defined in Section 5.1. Furthermore, to emphasize tokens that garner greater attention during the generation process, we augment the advantage estimation by incorporating I derived in Equation 8. The resulting AGDO-RL objective is formulated as:

$$-\mathbb{E}_{t,q,o}\Bigg[\frac{1}{|\mathcal{U}_t|}\sum_{k\in\mathcal{U}_t}\min\left(r_k\hat{A}'_k,\ \text{clip}(r_k,\ 1-\epsilon,\ 1+\epsilon)\hat{A}'_k\right) - \beta\, D_{\text{KL}}\left(\phi_{\pi_\theta}(\cdot \mid q)\,\|\,\phi_{\pi_{\text{ref}}}(\cdot \mid q)\right)\Bigg], \tag{10}$$

where the attention-adjusted advantage $\hat{A}'_k$ is defined as:

$$\hat{A}'_k = \hat{A}_k + sign(\hat{A}_k)\cdot\delta\cdot I_k, \tag{11}$$

where δ controls the strength of attention guidance. By amplifying policy updates for tokens that are central in the attention graph, AGDO-RL aligns preference optimization with the intrinsic reasoning structure of dLLMs. Implementation details are provided in Appendix B.3.

6 Experiment

6.1 Experimental Setup

Models and Benchmarks. We mainly conduct our experiments on Dream-v0-Instruct-7B (Ye et al.,

	Math			Code		Average
	GSM8K	MATH500	Minerva	LiveBench	LiveCodeBench-v2	
Similar-sized AR LLMs						
Llama3.1-8B-Instruct (AI@Meta, 2024)	84.5	51.9	37.5	19.7	20.0	42.7
Qwen2.5-7B-Instruct (Team et al., 2024)	89.9	74.0	50.4	31.1	26.9	54.5
Masked DLLMs baselines						
Dream-v0-Instruct-7B (Ye et al., 2025)	69.4	38.9	11.6	10.7	10.7	28.3
LLaDA-8B-Instruct (Nie et al., 2025)	81.5	38.3	11.9	4.9	5.9	28.5
SFT from Dream						
SFT	83.5	48.3	14.8	11.3	11.5	33.9
blockwise SFT	**86.0**	51.7	12.3	10.2	11.8	34.4
AGDO-SFT (ours)	85.3	**53.7**	**15.3**	**12.5**	**13.1**	**36.0**
RL from Dream						
Diff-GRPO (Zhao et al., 2025)	85.0	45.5	15.3	15.2	13.9	35.0
Coupled RL (Gong et al., 2025)	86.1	48.8	14.1	13.8	11.8	34.9
TraceRL (Wang et al., 2025b)	86.3	52.8	16.4	14.0	13.0	36.5
AGDO-RL (ours)	**87.7**	53.7	16.1	18.3	14.7	38.1
AGDO(ours)	86.9	**56.2**	**17.0**	**18.4**	**15.6**	**38.8**

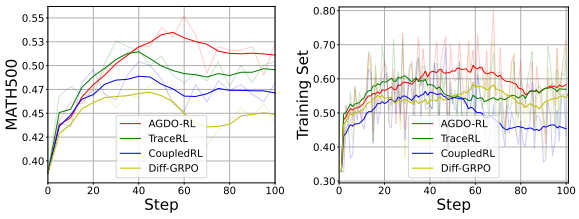
Table 2: The main benchmark results across different math and coding tasks on Dream-v0-Instruct-7B. Best results are in **bold**, and second best are underlined.

2025), and further assess the generalization of our approach on LLaDA-8B-Instruct (Nie et al., 2025). We evaluate model performance across two representative domains: mathematical reasoning and code generation. For mathematical reasoning, we benchmark on MATH-500 (Hendrycks et al., 2020), GSM8K (Cobbe et al., 2021), and Minerva (Lewkowycz et al., 2022), covering problems of varying difficulty and reasoning depth. For code generation, we report results on LiveBench (White et al., 2024) and LiveCodeBench-V2 (Jain et al., 2024), which provide dynamic and execution-based evaluation of real-world coding tasks.

Baselines and Metrics. We compare our approach against several established baselines under both SFT and RL. For SFT, we consider two baselines: (i) **standard SFT** with a fully random masking strategy, and (ii) **blockwise SFT** (Sun et al., 2025). To ensure a fair comparison, we augment the training data for the fully random masking baseline by applying 35 independent random masks per example, resulting in an average of approximately 39 forward passes per data point, which is comparable to that of other methods. For RL, we benchmark against **Diff-GRPO** (Zhao et al., 2025), **Couple-RL** (Gong et al., 2025), and **TraceRL** (Wang et al., 2025b). Similarly, to maintain fairness across methods, we augment the training data for Diff-GRPO and Couple-RL with 15 independent random masks, aligning the average number of forward passes per data point with our approach. For robust evaluation, each experiment is repeated 8 times, and we report the average accuracy across runs. During inference, we adopt a static decoding strategy with a sampling temperature of 0.1, unmasking one token per denoising step, and a maximum response length of 1,024 tokens. More implementation details can be found in Appendix C.

6.2 Main Results

Table 2 presents the quantitative comparison on Dream-v0-Instruct-7B. In the SFT stage, our AGDO-SFT consistently outperforms both standard and blockwise SFT baselines, achieving an average accuracy of 36.0% compared to 34.4% for blockwise SFT. Notably, AGDO-SFT yields a significant 3.0% gain compared to blockwise SFT on the challenging Minerva benchmark, confirming the benefit of aligning training with intrinsic attention dependencies. In the RL stage, AGDO-RL establishes new state-of-the-art results among baselines. Specifically, AGDO-RL achieves 18.3% on LiveBench, substantially exceeding diff-GRPO (15.2%). Furthermore, as illustrated in Figure 3, our method demonstrates more sustained accuracy growth and superior stability during training on both training and testing sets, indicating that attention-guided optimization effectively mitigates learning difficulties in dLLMs.



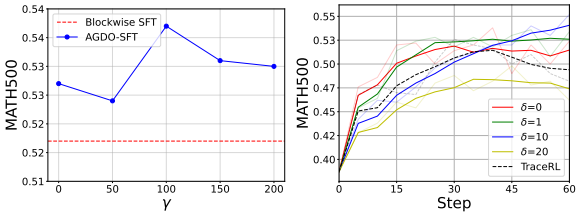
(a) Accuracy on MATH500. (b) Accuracy on MATH and GSM8K training sets.

Figure 3: Accuracy changes on training and testing sets during reinforcement training on Dream-v0-Instruct-7B.

6.3 Ablation Studies

6.3.1 Ablation on γ and δ

We investigate the impact of hyperparameters γ and δ on attention-based weighting in AGDO-SFT and AGDO-RL through ablation studies. As illustrated in Figure 4, tuning γ during the fine-tuning stage consistently improves accuracy over the blockwise SFT baseline, with $\gamma = 100$ yielding the best result on MATH500. Notably, AGDO-SFT outperforms blockwise SFT by approximately 2% even when $\gamma = 0$, a setting that implies the absence of I-enhanced training weights. This result demonstrates the isolated effectiveness of aligning the denoising order with attention based dependence.



(a) Accuracy changes on MATH500 under different γ. (b) Accuracy changes on MATH500 under different δ.

Figure 4: Ablication results on γ and δ.

In the RL phase, the accuracy curve for $\delta = 0$ consistently remains above that of TraceRL, further validating the proposed strategy of aligning the denoising trajectory with attention. We also observe that setting $\delta < 10$ leads to additional gains in training accuracy on top of the attention-guided denoising strategy. Conversely, performance deteriorates when $\delta = 20$. We hypothesize that an excessively large δ induces drastic gradient updates, which violates the trust region constraints fundamental to Proximal Policy Optimization.

Methods	Block Size					Average
	8	16	32	64	128	
SFT	48.4	48.2	48.7	49.9	50.5	49.1
Blockwise SFT	43.9	46.3	46.3	45.5	46.9	45.8
AGDO-SFT	49.1	48.5	50.2	50.5	49.7	49.6
Diff-GRPO	43.9	43.5	45.5	46.9	45.9	45.1
Coupled RL	47.1	48.4	49.2	49.6	51.5	49.2
TraceRL	47.5	50.5	50.3	52.9	51.9	50.6
AGDO-RL	48.0	50.6	53.3	53.7	52.3	51.6
AGDO	51.5	52.8	53.4	53.9	53.6	53.0

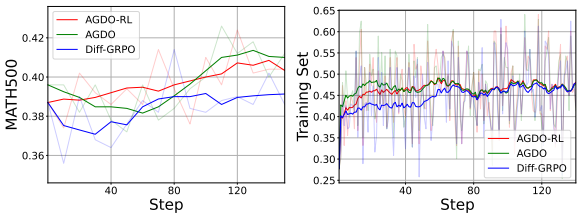
Table 3: Ablation results on MATH500 with different block sizes under a predefined length of 512.

6.3.2 Ablation on Sampling Configs

To evaluate the robustness of our method under stricter context constraints, we conduct ablation studies using a reduced sequence length of $L = 512$, in contrast to the $L = 1024$ setting employed in the main experiments. Table 3 details the performance across block sizes ranging from 8 to 128. In the fine-tuning stage, naive Blockwise SFT exhibits a marked performance decline (averaging 45.8%) compared to standard SFT (49.1%). However, AGDO-SFT effectively mitigates this degradation, achieving an average accuracy of 49.6% and surpassing standard SFT. This result underscores the efficacy of our proposed method. In the subsequent RL stage, AGDO-RL consistently outperforms all RL baselines across varying block sizes. Notably, AGDO-RL attains the highest average accuracy of 51.6%, peaking at 53.7% with a block size of 64. AGDO gains the best results on all block sizes, which further validates the effectiveness of aligning training with attention.

6.3.3 Application on LLaDA

To further validate the effectiveness of our approach, we evaluate our algorithms on LLaDA-8B-Instruct (Nie et al., 2025) using the GSM8K and MATH500 benchmarks, with the training set kept same with Dream. For the SFT stage, we adopt standard SFT with fully random masking and blockwise SFT as baselines. For the full training pipeline, we benchmark against d1-LLaDA (Zhao et al., 2025), wd1 (Tang et al., 2025), and LLaDA-1.5 (Zhu et al., 2025). As shown in Table 4, our methods outperform the baselines. The accuracy curves in Figure 5 further show that our methods achieve faster improvement than Diff-GRPO (Zhao et al., 2025), demonstrating their effectiveness.



(a) Accuracy on MATH500. (b) Accuracy on MATH and GSM8K training sets.

Figure 5: Accuracy changes during reinforcement training on LLaDA-8B-Instruct (Nie et al., 2025).

Methods	GSM8K	MATH500
Baseline		
LLaDA (Nie et al., 2025)	81.5	38.7
SFT from LLaDA		
SFT	82.1	38.3
Blockwise SFT	82.3	38.4
AGDO-SFT (ours)	**83.3**	**39.6**
SFT+RL from LLaDA		
d1-LLaDA (Zhao et al., 2025)	82.1	40.2
wd1 (Tang et al., 2025)	82.3	39.0
LLaDA-1.5 (Zhu et al., 2025)	83.3	42.6
AGDO (ours)	**85.3**	**42.8**

Table 4: Results on GSM8K and MATH500 benchmarks for LLaDA-8B-Instruct (Nie et al., 2025).

6.3.4 Ablation on Layer and Head Selection

We investigate the impact of layer selection and head aggregation strategy on AGDO-SFT performance. Table 5 compares attention signals extracted from different layers. Using the last layer yields the highest accuracy, consistent with prior findings that deeper layers capture higher-level semantic dependencies (Clark et al., 2019).

Layer Selection	MATH500 Acc (%)
First Layer	48.8
Middle Layer	52.5
Last Layer (Ours)	**53.7**

Table 5: Ablation on attention layer selection.

Table 6 compares different head aggregation strategies. Following Li et al. (2025d), we identify local-focused and global-focused heads and compare them against averaging all heads. The all-heads strategy achieves the best result, indicating that both local and global attention patterns contribute to effective denoising guidance.

Head Aggregation	MATH500 Acc (%)
Local-focused heads	53.0
Global-focused heads	52.2
All heads (Ours)	**53.7**

Table 6: Ablation on attention head aggregation for AGDO-SFT.

6.3.5 Ablation on Order vs. Weighting

To disentangle the contributions of the attention-guided denoising order and the influence-based loss weighting, we conduct a controlled ablation starting from the blockwise SFT baseline. As shown in Table 7, applying the attention-guided order alone (*i.e.*, $\gamma = 0$) improves accuracy by 1.0% over blockwise SFT, while adding influence-based weighting alone also yields gains. Combining both components achieves the best performance. Notably, replacing the influence score I with random weights leads to a smaller improvement, confirming that the gains stem from the attention-derived signal rather than stochastic regularization.

Method	MATH500 Acc (%)
Blockwise SFT (Baseline)	51.7
Order Only ($\gamma = 0$)	52.7
Weight Only (random order)	52.4
Random Weight (random order)	51.9
Order + Weight (Ours)	**53.7**

Table 7: Ablation on the individual contributions of the attention-guided denoising order and influence-based weighting in AGDO-SFT.

6.4 Further Analysis

6.4.1 Internalization of Attention Alignment

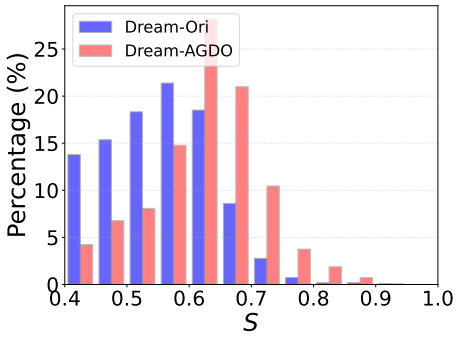


Figure 6: Comparison of average ΔP and S.

To investigate the impact of our training framework on internal reasoning mechanisms, we com-

pare the distribution of valid attention score S on the MATH500 benchmark between the original Dream model and the AGDO-trained model. As illustrated in Figure 6, the distribution of S for the AGDO-trained model exhibits a distinct rightward shift compared to the baseline. This trend indicates that the model has internalized the attention-guided denoising strategy, learning to prioritize the generation of tokens that possess strong dependencies on the already unmasked context. By maximizing valid attention during inference, the model effectively reduces generation uncertainty and establishes more robust logical chains, thereby enhancing the quality of generated responses. This further demonstrates the effectiveness of our method.

6.4.2 Computational Cost Analysis

A potential concern regarding AGDO-RL is the overhead incurred by the online attention analysis during RL. However, generating a response requires T denoising steps (*i.e.,* T forward passes), whereas our method adds only **one** single forward pass to analyze the full sequence.

We empirically test the overhead of online attention analysis during RL on Dream-v0-Instruct-7B using 8 NVIDIA H20 GPUs and $T = 1024$ denoising steps. Under the same RL settings as the main experiments, Table 8 shows that *Attention Analysis* accounts for only **3%** of the total rollout time, indicating that AGDO-RL improves reasoning with negligible impact on training efficiency.

Stage	Forward Passes	Total Time (s)
Rollout Stage	262,144	390
Attention Analysis	256	12
Total	262,400	402

Table 8: Time breakdown of a single RL iteration. The overhead from our Attention Analysis is marginal compared to the Rollout Stage.

6.4.3 Generalization to General NLP Tasks

To verify that AGDO does not introduce regressions on non-reasoning tasks, we evaluate on HellaSwag (Zellers et al., 2019) and CommonsenseQA (Talmor et al., 2019) using the same inference configuration. As shown in Table 9, AGDO consistently outperforms all baselines across both SFT and RL stages. Notably, AGDO-SFT surpasses both standard SFT and blockwise SFT on CommonsenseQA by 2.6% and 5.2% respectively, confirming that aligning denoising with attention dependencies enhances linguistic coherence and commonsense reasoning without introducing domain-specific bias.

Methods	HellaSwag	CSQA
Dream-v0-Instruct-7B	45.4	34.3
SFT		
SFT	60.9	75.5
Blockwise SFT	57.1	72.9
AGDO-SFT (ours)	**61.8**	**78.1**
RL		
Diff-GRPO	53.0	26.0
Coupled RL	40.0	27.3
TraceRL	45.7	27.6
AGDO-RL (ours)	58.0	35.0
AGDO (ours)	**64.7**	**78.3**

Table 9: Results on general NLP benchmarks. CSQA denotes CommonsenseQA.

7 Conclusion

In this paper, we revisit post-training for diffusion large language models (dLLMs) and show that existing random masking strategies fail to fully leverage intrinsic token dependencies. To address this limitation, we propose AGDO, a unified framework that integrates attention-guided denoising with supervised and reinforcement learning. By aligning the denoising order with attention-derived dependencies and emphasizing attention-critical tokens during optimization, AGDO better matches training dynamics with the model's internal reasoning structure. Experiments on mathematical and coding benchmarks demonstrate consistent improvements over state-of-the-art baselines, highlighting the effectiveness of attention-guided denoising and optimization for dLLMs.

Limitation

Our experiments mainly focus on dLLMs with full attention, utilizing its properties in dLLMs to optimize algorithms. Therefore, we do not discuss the performance on block attention-based dLLMs. In future work, we plan to investigate the characteristics of block attention in dLLMs and design tailored algorithmic improvements.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China

No. 92470205 and Beijing Major Science and Technology Project under Contract No. Z251100008425002.

References

AI@Meta. 2024. Llama 3 model card.

Wenrui Bao, Zhiben Chen, Dan Xu, and Yuzhang Shang. 2025. Learning to parallel: Accelerating diffusion large language models via learnable parallel decoding. *arXiv preprint arXiv:2509.25188*.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and 1 others. 2025a. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*.

Zicong Cheng, Guo-Wei Yang, Jia Li, Zhijie Deng, Meng-Hao Guo, and Shi-Min Hu. 2025b. Deer: Draft with diffusion, verify with autoregressive models. *arXiv preprint arXiv:2512.15176*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*, pages 276–286.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yifeng Gao, Ziang Ji, Yuxuan Wang, Biqing Qi, Hanlin Xu, and Linfeng Zhang. 2025. Self speculative decoding for diffusion large language models. *arXiv preprint arXiv:2510.04147*.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, and 1 others. 2024. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. 2025. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and 1 others. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14982–14995.

Sami Jaghouar, Jack Min Ong, Manveer Basra, Fares Obeid, Jannik Straube, Michael Keiblinger, Elie Bakouch, Lucas Atkins, Maziyar Panahi, Charles Goddard, and 1 others. 2024. Intellect-1 technical report. *arXiv preprint arXiv:2412.01152*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. 2025. Mercury: Ultra-fast language models based on diffusion. *Preprint*, arXiv:2506.17298.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. 2025a. Beyond fixed: Training-free variable-length denoising for diffusion large language models. *Preprint*, arXiv:2508.00819.

Pengxiang Li, Shilin Yan, Joey Tsai, Renrui Zhang, Ruichuan An, Ziyu Guo, and Xiaowei Gao. 2025b. Adaptive classifier-free guidance via dynamic low-confidence masking. *arXiv preprint arXiv:2505.20199*.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. 2025c. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022a. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, and 1 others. 2025d. Attention illuminates llm reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization. *arXiv preprint arXiv:2510.13554*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, and 1 others. 2022b. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. 2024. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. *arXiv preprint arXiv:2411.19943*.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. 2025. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*.

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*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Yuerong Song, Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. 2025. Sparse-dllm: Accelerating diffusion llms with dynamic cache eviction. *arXiv preprint arXiv:2508.02558*.

Bowen Sun, Yujun Cai, Ming-Hsuan Yang, and Yiwei Wang. 2025. Blockwise sft for diffusion language models: Reconciling bidirectional attention and autoregressive decoding. *arXiv preprint arXiv:2508.19529*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. 2025. wd1: Weighted policy optimization for reasoning in diffusion language models. *arXiv preprint arXiv:2507.08838*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Chengyu Wang, Paria Rashidinejad, DiJia Su, Song Jiang, Sid Wang, Siyan Zhao, Cai Zhou, Shannon Zejiang Shen, Feiyu Chen, Tommi Jaakkola, and 1 others. 2025a. Spg: Sandwiched policy gradient for masked diffusion language models. *arXiv preprint arXiv:2510.09541*.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. 2025b. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, and 1 others. 2024. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.

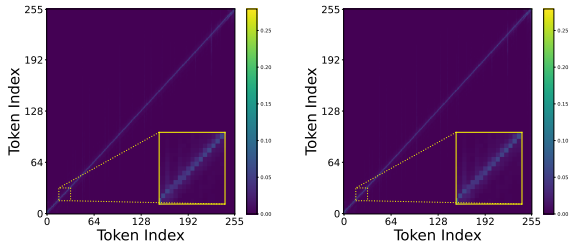
Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2025. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and 1 others. 2025. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*.

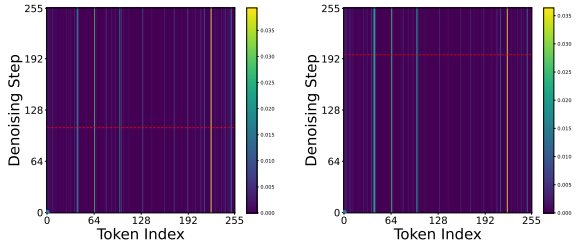
A Attention Analysis on LLaDA

To assess the generalizability of our findings beyond the Dream model, we extend our empirical analysis to LLaDA-8B-Instruct (Nie et al., 2025).

As illustrated in Figure 7, LLaDA displays attention patterns comparable to those observed in the Dream model, notably exhibiting horizontal sparsity and vertical consistency. Furthermore, the results in Table 10 suggest that aligning the decoding order with intrinsic attention dependencies could serve as a generalized principle for enhancing generation quality in dLLMs.



(a) Step 136 attention map. (b) Step 237 attention map.



(c) Token 81 attention change. (d) Token 20 attention change.

Figure 7: Analysis of attention patterns on LLaDA. (a) and (b) visualize the attention distributions among tokens at two randomly selected denoising steps. (c) and (d) illustrate the temporal attention dynamics of specific tokens towards others throughout the denoising process.

Strategy	**128**	**256**	**512**
Max-Prob	19.9	23.8	16.1
Max-S	28.5	30.3	32.7

Table 10: Comparison of decoding accuracy on MATH500 using static sampling across different predefined sequence lengths on LLaDA. For the Max-S strategy, the probability threshold is set to 0.

B Detailed Algorithms

Here we provide the pseudocode for the three core components of the AGDO framework.

B.1 Attention-Guided Denoising Order

Algorithm 1 Attention-Guided Denoising Order

Input: Sequence X; Model π_θ; Step size n; Block size m
Output: Denoising Traces $T \in \mathbb{Z}^{|X|}$
Init: $\mathcal{U} \leftarrow \text{Prompt}(X)$; $\mathcal{M} \leftarrow \text{Masked}(X)$; $T \leftarrow \mathbf{0}$; rank $\leftarrow 1$
$A \leftarrow \text{Forward}(X, \pi_\theta)$
while $\mathcal{M} \neq \emptyset$ **do**
  $\forall i \in \mathcal{M} : S_i \leftarrow \sum_{k \in \mathcal{U}} \text{MeanHead}(A_{i,k})$
  $\mathcal{K} \leftarrow \text{TopK}(\{S_i\}, n)$
  $\forall k \in \mathcal{K} : T_k \leftarrow \text{rank}; \quad \mathcal{U} \leftarrow \mathcal{U} \cup \mathcal{K}; \quad \mathcal{M} \leftarrow \mathcal{M} \setminus \mathcal{K}$
  rank $\leftarrow$ rank $+ 1$
end while
Constant $\mathcal{C} \leftarrow |X|$
for $idx = 0 \ldots |X| - 1$ **do**
  batch_num $\leftarrow \lfloor idx/m \rfloor$
  $T_{idx} \leftarrow T_{idx} + (\text{batch\_num} + 1) \cdot \mathcal{C}$
end for
Return T

B.2 AGDO-SFT

Algorithm 2 AGDO-SFT

Input: Dataset $\mathcal{D}$; Model π_θ; Rate η; Coeff γ

for batch $B \in \mathcal{D}$ **do**
  $\mathcal{L} \leftarrow 0$
  for $X \in B$ **do**
    $T \leftarrow \text{Alg.1}(X); \quad A \leftarrow \text{Forward}(X, \pi_\theta)$
    $\forall k : I_k \leftarrow \sum_i \text{MeanHead}(A_{i,k})$ ▷ Eq. 8
    for $t = 0 \ldots \max(T)$ **do**
      $\mathcal{K}_t \leftarrow \{k \mid D_k = t\}$
      $\mathcal{M}_{sub} \leftarrow \text{RandomMask}(\mathcal{K}_t)$
      $\mathcal{M}_{in} \leftarrow \{k \mid D_k > t\} \cup \mathcal{M}_{sub}$
      $W \leftarrow \{1 + \gamma I_k \mid k \in \mathcal{M}_{sub}\}$
      $\mathcal{L} \leftarrow \mathcal{L} + \text{WeightedNLL}(\pi_\theta(X_{\mathcal{M}_{in}}), \mathcal{M}_{sub}, W)$ ▷ Eq. 9
    end for
  end for
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for

B.3 AGDO-RL

Algorithm 3 AGDO-RL

Input: Prompts $\mathcal{Q}$; Ref π_{ref}; Group G; Coeff δ

for iteration $1 \dots$ MaxIter **do**
  $q \sim \mathcal{Q}$
  $\{o_1 \dots o_G\} \leftarrow \text{Gen}(q, \pi_\theta)$
  $R \leftarrow \text{Reward}(q, \{o_j\})$; $\hat{A} \leftarrow \text{Norm}(R)$
  $\mathcal{L} \leftarrow 0$
  for $j = 1 \dots G$ **do**
    $T \leftarrow \text{Alg.1}([q, o_j], \pi_\theta)$; $A \leftarrow \text{Forward}([q, o_j], \pi_\theta)$
    $\forall k : I_k \leftarrow \sum_i \text{MeanHead}(A_{i,k})$ ▷ Eq. 8
    $\forall k : \hat{A}'_k \leftarrow \hat{A}_j + \text{sign}(\hat{A}_j) \cdot \delta \cdot I_k$ ▷ Eq. 11
    for $t = 0 \dots \max(T)$ **do**
      $\mathcal{K}_t \leftarrow \{k \mid T_k = t\}$
      $\mathcal{M}_{sub} \leftarrow \text{RandomMask}(\mathcal{K}_t)$
      $\mathcal{M}_{in} \leftarrow \{k \mid T_k > t\} \cup \mathcal{M}_{sub}$
      $\pi_{curr} \leftarrow \pi_\theta(o_j[\mathcal{M}_{sub}] \mid [q, o_j]_{\mathcal{M}_{in}})$
      $\pi_{old} \leftarrow \pi_{ref}(o_j[\mathcal{M}_{sub}] \mid [q, o_j]_{\mathcal{M}_{in}})$
      $\mathcal{L} \leftarrow \mathcal{L} + \text{GRPO\_Obj}(\pi_{curr}, \pi_{old}, \hat{A}'_{\mathcal{M}_{sub}})$ ▷ Eq. 10
    end for
  end for
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for

C Implementation Details

All experiments are conducted on NVIDIA H20 GPUs. **Supervised Fine-Tuning (SFT):** For mathematical reasoning, we use 2,000 training samples from Wang et al. (2025b). For code generation, we distill 2,000 samples from CodeContest (Li et al., 2022b) using Qwen2.5-32B-Instruct (Team et al., 2024). Models are trained for one epoch with a learning rate of 1×10^{-6} and a total batch size of 128. We set `post_num` to 16 and γ to 100. **Reinforcement Learning (RL):** We adopt the training framework from Wang et al. (2025b). The training data consists of the training splits of GSM8K and MATH for mathematical reasoning, and PrimeIntellect (Jaghouar et al., 2024) for code generation. For coding tasks, the reward is defined as the fraction of unit tests passed. During rollout, we sample 32 prompts per step and generate 8 responses for each prompt. We employ a static denoising strategy with a temperature of 1.0, unmasking one token per denoising step, and limit the maximum sequence length to 1,024 tokens. The total batch size is set to 512, with a learning rate of 1×10^{-6}. We set δ to 1. The `post_num` is set to 16 for mathematical tasks and 0 for code generation.