The Flexibility Trap: Rethinking the Value of Arbitrary Order in Diffusion Language Models

Zanlin Ni[1], **Shenzhi Wang**[1], **Yang Yue**[1], **Tianyu Yu**[2], **Weilin Zhao**[2], **Yeguo Hua**[2], **Tianyi Chen**[2], **Jun Song**[3], **Cheng Yu**[3], **Bo Zheng**[3], **Gao Huang**[1]✉

[1] LeapLab, Tsinghua University [2] Tsinghua University [3] Alibaba Group

Diffusion Large Language Models (dLLMs) break the rigid left-to-right constraint of traditional LLMs, enabling token generation in arbitrary orders. Intuitively, this flexibility implies a solution space that strictly supersets the fixed autoregressive trajectory, theoretically unlocking superior reasoning potential for general tasks like mathematics and coding. Consequently, numerous works have leveraged reinforcement learning (RL)[1] to elicit the reasoning capability of dLLMs. In this paper, we reveal a counter-intuitive reality: arbitrary order generation, in its current form, *narrows* rather than expands reasoning boundary. We find that dLLMs tend to exploit this flexibility to bypass high-uncertainty tokens that are crucial for exploration, leading to a premature collapse of the solution space. This observation challenges the premise of existing RL approaches for dLLMs, where considerable complexities, such as handling combinatorial trajectories and intractable likelihoods, are often devoted to preserving this flexibility. We demonstrate that effective reasoning is better elicited by intentionally forgoing arbitrary order and applying just standard Group Relative Policy Optimization (GRPO) instead. Our approach, **JustGRPO**, is minimalist yet surprisingly effective (*e.g.*, 89.1% accuracy on GSM8K) while fully retaining the parallel decoding ability of dLLMs.

Code and pretrained models are available at https://github.com/LeapLabTHU/JustGRPO

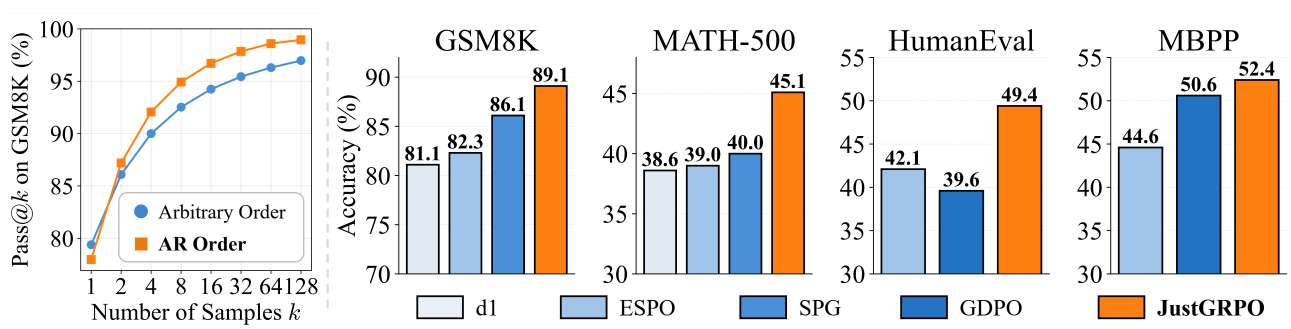

Figure 1: **Less flexibility unlocks better reasoning potential.** *Left*: We observe a counter-intuitive phenomenon where restricting dLLMs to standard Autoregressive (AR) order expands the reasoning solution space. *Right*: Motivated by this, we propose "**JustGRPO**". By foregoing complex arbitrary-order adaptations and adopting standard GRPO, we effectively elicit the reasoning capability of dLLMs.

[1] In this paper, unless otherwise specified, we use the term RL to refer specifically to Reinforcement Learning with Verifiable Rewards (RLVR), as it is the predominant paradigm for enhancing the reasoning capabilities of dLLMs.

1. Introduction

Recent research has witnessed a surge in Diffusion Large Language Models (dLLMs) (Nie et al., 2025; Ye et al., 2025; Zhu et al., 2025; Zhao et al., 2025), which challenge the dominant autoregressive (AR) paradigm (Brown et al., 2020; Achiam et al., 2023) by treating sequence generation as a discrete denoising process. Central to the appeal of dLLMs is their theoretical flexibility, which offers two distinct advantages over the strict left-to-right causal chain of AR models: efficient parallel decoding and the capability for arbitrary-order generation.

While the efficiency gains of parallel decoding have been well-established (Wu et al., 2025b; Labs et al., 2025; DeepMind, 2025; Song et al., 2025; Wu et al., 2025a), the implications of arbitrary-order generation remain less explored (Ye et al., 2024; Kim et al., 2025). Theoretically, the unconstrained generation order constitutes a superset of the fixed autoregressive trajectory. This flexibility naturally suggests a potential for superior reasoning: in general reasoning tasks like mathematics and coding, such freedom could unlock non-sequential problem-solving paths inaccessible to standard left-to-right models. As a result, recent works have increasingly adopted RL to elicit reasoning capabilities of dLLMs (Zhao et al., 2025; Gong et al., 2025; Wang et al., 2025a; Ou et al., 2025).

In this paper, we present a counter-intuitive observation: arbitrary-order generation, in its current form, *narrows* rather than expands the reasoning potential elicitable by RL. To rigorously assess this, we employ Pass@k (Chen, 2021), which measures the coverage of solution space. Recent studies suggest that RL primarily acts to sharpen the base model's distribution; consequently, the Pass@k performance of the base model effectively sets the upper bound for the reasoning capability of the model after RL training (Yue et al., 2025; Liu et al., 2025; Zhang et al., 2025). Under this metric, we compare the reasoning potential of LLaDA (Nie et al., 2025) with arbitrary-order generation against standard AR decoding. As shown in Figure 1 (Left), restricting a dLLM to standard AR order in fact yields a higher Pass@k, and consequently a higher reasoning boundary, than its flexible counterpart.

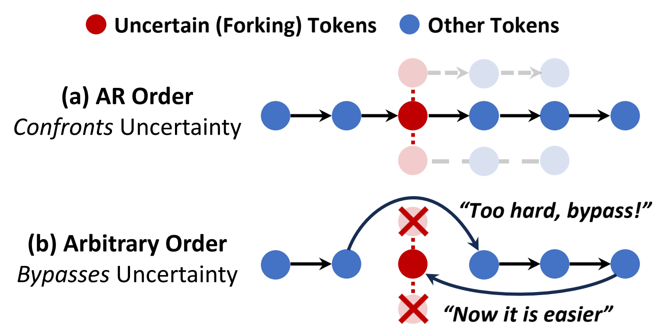

Figure 2: **Confronting *vs.* bypassing uncertainty.** (a) **AR order** preserves reasoning space by forcing decisions at uncertain tokens. (b) **Arbitrary order** bypasses uncertainty and resolves easier tokens first. Once future context is established, the original forks collapse, prematurely narrowing the solution space.

We attribute this counter-intuitive phenomenon to the way the model handles uncertainty. Reasoning process is inherently non-uniform, it typically hinges on sparse "forking tokens", *i.e.*, connectives like "Therefore" or "Since" which do not merely continue a sentence but fundamentally steer the logical trajectory into distinct branches (Wang et al., 2025c; Cheng et al., 2025; Huang et al., 2025a). At these forks, the reasoning path diverges, naturally manifesting as localized spikes in entropy (Wang et al., 2025c). Standard AR decoding compels the model to *confront* this uncertainty (Figure 2a). By sampling exactly at the fork, the model is able to explore different reasoning paths, thereby preserving the diversity of the generated rationales. Arbitrary-order generation, however, allows the model to *bypass* these hard decisions (Figure 2b). It prioritizes low-entropy completions first. By the time the model returns to infill the bypassed forks, the established bidirectional context has already severely constrained the potential branches. The ambiguity is prematurely resolved, and the original high entropy is suppressed. We term this phenomenon *entropy degradation*. Effectively, the model trades the exploration of diverse reasoning paths for the greedy optimization of local consistency.

The above observations motivate a rethink of RL for dLLMs. Current methods operate under the assumption that preserving arbitrary-order flexibility is essential. This commitment incurs a heavy tax: algorithms must grapple with a combinatorial explosion of denoising trajectories (Zhao et al., 2025; Yang et al., 2025; Gong et al., 2025) and intractable marginal likelihoods (Ou et al., 2025), forcing reliance on unstable approximations (Wang et al., 2025a; Ou et al., 2025; Rojas et al., 2025). However, if arbitrary order is non-essential, or even detrimental for eliciting reasoning potential, this complexity is unjustified.

To this end, we propose a return to simplicity with **Just GRPO**. We demonstrate that eliciting reasoning potential for general reasoning tasks does not require complex, diffusion-specific RL adaptations. Instead, it is best achieved by simply treating the dLLM as an AR model during RL training. This allows us to apply standard GRPO (Shao et al., 2024) without bells and whistles, turning an otherwise intractable optimization with unstable approximations into a well-defined task.

JustGRPO is surprisingly effective. On complex reasoning benchmarks, it achieves competitive results (*e.g.*, 89.1% accuracy on GSM8K, 45.4% on MATH), surpassing methods that rely on complex diffusion-specific RL. Crucially, while we train with AR constraints to maximize reasoning potential, the model retains dLLMs' ability of efficient parallel decoding at inference time. By returning to basic left-to-right ordering, this work encourages a re-evaluation of arbitrary versus AR order in the development of next-generation language models.

2. Preliminaries

2.1. Diffusion Large Language Models

Diffusion Large Language Models (dLLMs), particularly Masked Diffusion Models (MDMs), generate sequences by iteratively denoising a masked state x_t initialized from fully masked tokens. The process is indexed by a continuous time variable $t \in [0, 1]$, representing the masking ratio. Given a clean sequence x_0, the forward process independently masks each token with probability t:

$$q(x_t^k \mid x_0^k) = \begin{cases} \texttt{[MASK]}, & \text{with prob } t, \\ x_0^k, & \text{with prob } 1-t. \end{cases}$$

Unlike Gaussian diffusion, MDMs directly predict the clean token. A neural network $p_\theta(x_0 \mid x_t)$ estimates the original token distribution at masked positions. During inference, generation starts from x_1 (all `[MASK]`) and iteratively unmasks a subset of tokens based on heuristics such as confidence scores, updating $x_t \to x_{t-\Delta t}$ until completion. As a special case, autoregressive generation can be recovered by always unmasking the leftmost token. The model is trained by minimizing the Negative Evidence Lower Bound, which reduces to a weighted cross entropy loss over masked tokens:

$$\mathcal{L}_{\text{MDM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_t\sim q(x_t|x_0)} \left[\frac{1}{t} \sum_{k=1}^{L} \mathbf{1}[x_t^k = \texttt{[MASK]}] \log p_\theta(x_0^k \mid x_t) \right].$$

2.2. Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) is a reinforcement learning algorithm that avoids value function estimation by using group level reward normalization. It is typically applied to autoregressive policies $\pi_\theta(o_k \mid o_{<k}, q)$. For each query q, GRPO samples a group of G outputs $\{o_1, \dots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$. An advantage A_i is computed by standardizing the reward $r(o_i)$ against the group statistics: $A_i = (r(o_i) - \mu_G)/\sigma_G$, where μ_G and σ_G are the group mean and standard deviation. The GRPO objective maximizes a clipped surrogate function with a KL regularization term:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{L_i} \sum_{k=1}^{L_i} \left(\min\left(\rho_{i,k} A_i, \text{clip}(\rho_{i,k}, 1-\epsilon, 1+\epsilon) A_i\right) - \beta D_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \right) \right] \tag{1}$$

with the token level importance ratio $\rho_{i,k} = \frac{\pi_\theta(o_{i,k}|o_{i,<k},q)}{\pi_{\theta_{\text{old}}}(o_{i,k}|o_{i,<k},q)}$.

2.3. Pass@k as a Proxy for Reasoning Potential

To rigorously quantify the reasoning capability boundaries of dLLMs, we adopt the Pass@k metric (Chen, 2021; Yue et al., 2025). In the context of Reinforcement Learning with Verifiable Rewards (RLVR), which

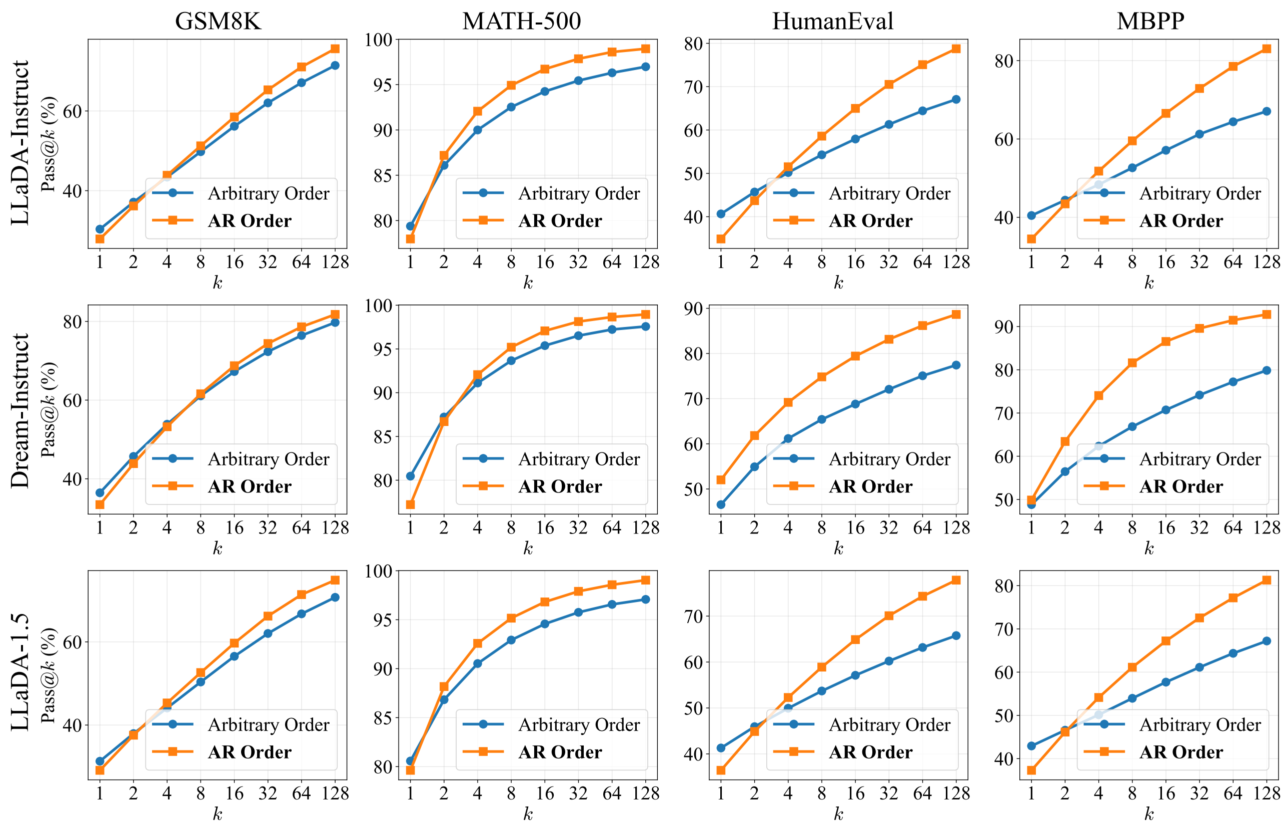

Figure 3: **Reasoning potential measured by Pass@k.** While arbitrary order is competitive in single-shot settings ($k = 1$), it exhibits notably flatter scaling curves compared to AR Order.

is currently the dominant paradigm for enhancing reasoning capabilities, exploration is a prerequisite for improvement. An RL agent can only reinforce correct reasoning paths if it is capable of sampling them during the exploration phase to obtain a positive reward signal.

Accordingly, Pass@k has been widely established as a standard measure of a model's *reasoning potential* (Yue et al., 2025; Liu et al., 2025; Zhang et al., 2025). It measures the probability that at least one correct solution is generated within k independent samples, effectively delineating the upper bound of the solution space accessible to the RL optimizer. Formally, following the unbiased estimator formulation (Chen, 2021; Yue et al., 2025), given n samples where c are correct, Pass@k is calculated as:

$$\text{Pass@}k = \mathbb{E}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right] \tag{2}$$

A high Pass@k indicates that the correct reasoning trajectory lies within the model's sampling distribution and is thus learnable via RL optimization. Conversely, if a model consistently fails to yield a solution despite a vast sampling budget, it suggests the problem lies beyond its intrinsic reasoning boundary. In such scenarios, standard RLVR methodologies are fundamentally limited by the absence of positive exploration signals (Yue et al., 2025).

3. The Flexibility Trap

In this section, we rigorously test whether the theoretical flexibility of arbitrary-order generation translates into higher reasoning potential. We compare two decoding modes: *Arbitrary Order*, which follows standard diffusion decoding low-confidence remasking (Nie et al., 2025; Zhao et al., 2025; Wu et al., 2025b), and *AR Order*, where arbitrary-order flexibility is disabled and generation is strictly constrained to left-to-right decoding. We adopt the commonly used experimental setup in prior diffusion LLM work: a maximum of 256 tokens decoded over 256 steps, a semi-autoregressive block size of 32. Sampling temperature is set to 0.6 as in (Yue et al., 2025). The prompt template follows (Zhao et al., 2025). Results under alternative temperatures and sampling configurations are deferred to Appendix B.

3.1. Arbitrary Order Limits Reasoning Potential

Pass@k analysis. We first evaluate the reasoning potential using the Pass@k metric on three representative dLLMs: LLaDA-Instruct (Nie et al., 2025), Dream-Instruct (Ye et al., 2025), and LLaDA 1.5 (Zhu et al., 2025) on four reasoning benchmarks: GSM8K, MATH500, HumanEval, and MBPP. As shown in Figure 3, while arbitrary order often achieves competitive performance at $k = 1$, it exhibits a notably flatter scaling curve compared to the AR mode. As k increases, the AR mode demonstrates a stronger capacity to uncover correct solutions.

Solution space coverage. One might hypothesize that arbitrary order explores a different solution space, albeit less efficiently, which could account for its lower reasoning potential. We test this by analyzing solution coverage at $k = 1024$ using LLaDA-Instruct. Figure 4 presents a stark reality: the reasoning traces generated by arbitrary order are largely a strict subset of those generated by AR. On HumanEval, AR solves 21.3% of the problems that arbitrary order misses, whereas the reverse is negligible (0.6%). This indicates that the flexible decoding process rarely unlocks genuinely new solutions. Instead, it appears to retreat into a more conservative subset of the AR solution space.

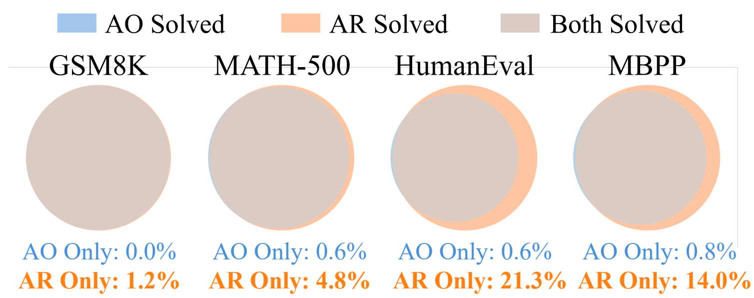

Figure 4: **Solution space coverage** measured by Pass@1024. The reasoning traces generated by arbitrary order are largely a strict subset of those generated by AR Order.

3.2. Mechanism: The Entropy Degradation

Adaptive decoding bypasses logical forks. To understand why the theoretically superior solution space of dLLMs collapses in practice, we examine more closely how two decoding modes handle uncertainty. In AR order, the model is constrained to strictly resolve the left-most unresolved token at each step, forcing the model to confront uncertainty as it arises. In contrast, arbitrary order *adaptively* selects tokens to update based on model confidence, preferentially generating "easy" tokens with high certainty while bypassing "hard" ones. Inspecting the frequently bypassed tokens reveals a clear pattern: As shown in Figure 5, the diffusion sampler disproportionately defers logical connectives and transition markers such as "Therefore", "Thus", and "Since". Prior work has shown that such tokens are often with high entropy (which also holds true in dLLMs, see Figure 6), and as "reasoning sparks" or "logical forks", functioning as forks that determine subsequent reasoning directions (Wang et al., 2025c; Cheng et al., 2025; Wang et al., 2025b; Huang et al., 2025a). In conventional language models, keeping these tokens in high-entropy state is critical for effective exploration of the reasoning space (Wang et al., 2025c).

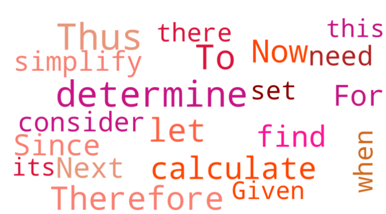

Figure 5: **Frequently bypassed tokens** in arbitrary order, measured on MATH-500, are typically logical connectors and transition words.

The "entropy degrade" phenomenon. This adaptive behavior comes at a cost: the premature collapse of reasoning possibilities. We measure the entropy of these pivotal connectors at decoding. In AR order, these tokens maintain high entropy, reflecting a genuine branching point where multiple logical paths remain viable. In contrast, arbitrary order exhibits a sharp decrease in entropy (Figure 6, with more results in Appendix B). By deferring the logical connector, the model commits to a specific future outcome that is generated based on its inherent inductive biases *before* deciding the logic that leads there. When the model eventually returns to fill in the bypassed connector, it is no longer making a navigational decision at a fork; it is more like selecting the connector that grammatically bridges the gap to its pre-generated conclusion. The decoding process thus implicitly shifts from reasoning exploration into semantic pattern matching. We term this phenomenon *entropy degradation*.

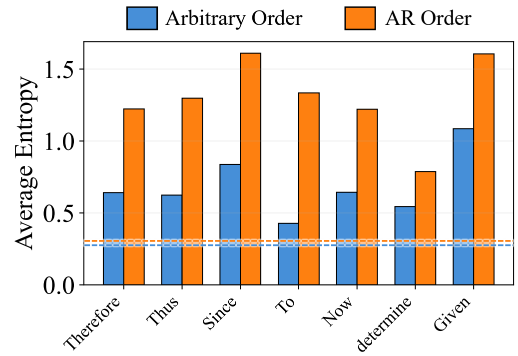

Figure 6: **Entropy degradation.** While the global average entropy of Arbitrary Order remains comparable to AR (dashed lines), the entropy at logical forks drops significantly (blue bars).

Conclusion. In summary, the flexibility of arbitrary order serves as a mechanism for inference-time exploitation rather than reasoning exploration. By bypassing high-uncertainty tokens, the model effectively collapses the solution space to a safe, low-entropy path, squeezing out slightly better single-shot coherence at the expense of reasoning potential. Autoregressive models, by contrast, lack this bypassing capability and are therefore forced to sample directly from high-entropy distributions at logical forks. It is precisely this inability to circumvent critical decision points that prevents the premature narrowing of the search space and preserves the reasoning potential.

4. "Just GRPO" for Diffusion Language Models

The findings in Section 3 suggest that arbitrary order actually limits the reasoning potential accessible to RL. Despite this, current RL methods for dLLMs remain heavily burdened by the need to preserve this specific flexibility. In this section, we uncover the heavy "tax" imposed by this flexibility (Section 4.1) and show that discarding it enables a minimalist yet surprisingly effective solution: **JustGRPO** (Section 4.2).

4.1. The Flexibility Tax in dLLMs' RL

Existing diffusion RL methods operate under the premise that the policy must optimize over the full combinatorial space of denoising trajectories $\mathcal{T}$ to preserve the flexibility of arbitrary order generation. This design choice, while conceptually general, introduces several fundamental challenges.

Ambiguity in token-level decomposition. In dLLMs, a generation state s_t is a noisy sequence conditioned on a stochastic unmasking trajectory τ. Unlike autoregressive models, dLLMs do not admit a unique, index-aligned conditional probability of the form $\pi(o_t \mid s_t)$, making token-level credit assignment ambiguous and rendering the standard importance ratio $\rho_t = \frac{\pi_\theta(o_t|s_t)}{\pi_{\text{old}}(o_t|s_t)}$ hard to define.

Intractable sequence likelihood. Autoregressive models factorize the sequence likelihood as $\log \pi(o) = \sum_t \log \pi(o_t \mid o_{<t}, q)$, whereas dLLMs require marginalization over all valid denoising trajectories, $\pi_\theta(o \mid q) = \sum_{\tau \in \mathcal{T}} \pi_\theta(o, \tau \mid q)$. For a sequence of length N, the trajectory space grows as $|\mathcal{T}| = O(N!)$, rendering exact likelihood computation infeasible and forcing existing methods to rely on ELBO-based surrogates rather than the true objective (Wang et al., 2025a; Ou et al., 2025).

Sampler-learner mismatch. Even with an accurate likelihood approximation, a more subtle issue persists. In practice, rollout samples are produced by heuristic-guided policies $o \sim \pi_\theta^{\text{heur}}(o \mid q)$ to explore the combinatorial space. However, the ELBO objective still targets the likelihood of the original model distribution $\pi_\theta(o \mid q)$, rather than that of the heuristic-guided sampler $\pi_\theta^{\text{heur}}(o \mid q)$, leading to a critical mismatch between sampling and optimization that can degrade performance (Schulman et al., 2015).

4.2. JustGRPO

We propose a return to simplicity. Since pure autoregressive order yields better reasoning potential (Section 3), we explicitly forgo arbitrary-order generation during the RL stage. This constraint transforms the dLLM from a chaotic sequence denoiser into a well-defined autoregressive policy π_θ^{AR}.

Formulation. Standard GRPO assumes a policy $\pi(o_t|o_{<t}, q)$ accepting a *partial* sequence $o_{<t}$ with all tokens observed and predicting *one* token o_t at a time, where q is the query. Diffusion language models, however, are architected as sequence-level denoisers that accept a *full* sequence with mixed observed and masked tokens and predict the original values for *all* masked positions simultaneously.

By forgoing arbitrary-order generation, we are able to bridge the above gap and rigorously define an AR policy π_θ^{AR} for dLLMs. To obtain the probability of the next token o_t given history $o_{<t}$, we construct an input state $\tilde{x}_t$ where the past is observed and the future is masked:

$$\tilde{x}_t = [\underbrace{o_1, \dots, o_{t-1}}_{\text{Observed}}, \underbrace{\texttt{[MASK]}, \dots, \texttt{[MASK]}}_{\text{Masked}}]. \tag{3}$$

Although the diffusion language model outputs predictions for all masked positions, the autoregressive policy concerns only the next token o_t. We thus define $\pi_\theta^{\mathrm{AR}}(o_t|o_{<t}, q)$ as the probability distribution of o_t:

$$\pi_\theta^{\mathrm{AR}}(o_t|o_{<t}, q) \triangleq \mathrm{Softmax}(f_\theta(\tilde{x}_t))_t, \tag{4}$$

where f_θ is the model logit. Consequently, the likelihood of a complete reasoning chain o is exactly computable as:

$$\pi_\theta^{\mathrm{AR}}(o|q) = \prod_{t=1}^{L} \pi_\theta^{\mathrm{AR}}(o_t|o_{<t}, q). \tag{5}$$

Optimization. The above formulation enables the direct application of standard GRPO to diffusion language models. For each query q, we sample a group of outputs $\{o^i\}_{i=1}^G$ using the old policy $\pi_{\theta_{\mathrm{old}}}^{\mathrm{AR}}$. The objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim P(Q), \{o^i\}_{i=1}^G \sim \pi_{\theta_{\mathrm{old}}}^{\mathrm{AR}}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o^i|} \sum_{t=1}^{|o^i|} \left(\min\left(\rho_{i,t} \hat{A}_{i,t}, \mathrm{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon) \hat{A}_{i,t} \right) - \beta \mathbb{D}_{\mathrm{KL}} \right) \right], \tag{6}$$

where $\rho_{i,t} = \frac{\pi_\theta^{\mathrm{AR}}(o_t^i|o_{<t}^i, q)}{\pi_{\theta_{\mathrm{old}}}^{\mathrm{AR}}(o_t^i|o_{<t}^i, q)}$ is the probability ratio between the current and old policies and $\hat{A}_{i,t}$ denotes the group-standardized advantage.

Remarks. One might worry if training in AR mode reduces the diffusion language model into a standard autoregressive model. This is not the case. The AR constraint is applied *only during training* to correctly assign credit. It refines the model's joint distribution $p(o)$ without altering the underlying architecture. At inference time, the model retains its conditional independence properties, still allowing us to employ parallel samplers (Ben-Hamu et al., 2025) to accelerate decoding. JustGRPO thus achieves the reasoning depth of autoregressive models while preserving the inference speed of dLLMs (see in Section 5.2).

5. Experiments

We evaluate JustGRPO on standard mathematical reasoning and coding benchmarks. Our experimental design aims to verify two hypotheses: (i) that enforcing autoregressive (AR) order during RL training elicits superior reasoning capabilities compared to complex arbitrary-order approximations, and (ii) that this constraint applies only to the optimization objective, leaving the diffusion model's parallel decoding capabilities intact at inference.

Experimental Setups. We apply JustGRPO on LLaDA-Instruct (Nie et al., 2025) and evaluate its effectiveness on four standard reasoning and coding benchmarks: GSM8K, MATH, HumanEval, and MBPP. For mathematical tasks, we train on the official training split of each dataset following (Zhao

Table 1: **System-level comparison** RL post-training approaches on LLaDA-Instruct. JustGRPO consistently achieves state-of-the-art performance across all tasks and sequence lengths. LLaDA-1.5 and LLADOU are excluded from the comparison as LLaDA-1.5 is trained on a privately collected dataset at a significantly larger scale, while LLADOU modifies the base architecture with an auxiliary module.

	GSM8K			**MATH-500**			**HumanEval**			**MBPP**		
Model / Seq Len	**128**	**256**	**512**	**128**	**256**	**512**	**128**	**256**	**512**	**128**	**256**	**512**
LLADA-1.5	-	83.3	-	-	-	-	29.3	39.6	51.9	39.6	39.9	38.8
LLADOU	-	88.1	-	-	41.1	-	-	59.1	-	-	51.6	-
D1	73.2	81.1	82.1	33.8	38.6	40.2	-	-	-	-	-	-
WD1	77.2	80.8	82.3	33.3	34.4	39.0	-	-	-	-	-	-
d-TreeRPO	-	81.2	82.6	-	37.7	38.9	-	-	-	-	-	-
ESPO	80.0	82.3	83.7	36.0	39.0	43.4	28.1	42.1	**50.0**	47.4	44.6	44.2
GDPO	78.4	82.8	84.5	33.2	39.6	41.4	26.2	39.6	39.0	43.6	50.6	47.1
SPG	78.5	86.1	84.5	33.4	40.0	41.8	-	-	-	-	-	-
JustGRPO (Ours)	**83.8**	**89.1**	**89.8**	**39.0**	**45.1**	**45.2**	**37.8**	**49.4**	48.7	**50.6**	**52.4**	**49.0**

et al., 2025; Ou et al., 2025). For coding tasks, we train on a subset of AceCoder-87K (Zeng et al., 2025) following (Gong et al., 2025; Ou et al., 2025). We apply JustGRPO directly to the models without additional task-specific SFT. Our training recipe largely follows (Huang et al., 2025b). To evaluate the trained models, we follow the standard LLaDA evaluation protocol (Nie et al., 2025), which applies low-confidence remasking together with semi-autoregressive decoding in blocks of 32 tokens, using a sampling temperature of 0. We evaluate all benchmarks at generation lengths of 128, 256, and 512 following (Zhao et al., 2025; Ou et al., 2025). More details on training are provided in Appendix A.

5.1. Main Results

Table 1 reports the system-level comparison. We observe that simplifying the training objective to a standard autoregressive formulation yields consistent improvements over methods specifically designed for dLLMs.

Performance on reasoning tasks. On GSM8K, JustGRPO achieves 89.1% accuracy (seq len 256), improving upon the previous best method, SPG, by a non-trivial margin (3.0%). This trend generalizes to the more challenging MATH-500 benchmark, where our approach outperforms ESPO by 6.1%. These results challenge the prevailing assumption that RL for diffusion models requires optimizing over the full combinatorial space of denoising trajectories. Instead, treating the dLLM as a sequential generator during training appears more effective for credit assignment in logic-heavy tasks.

Robustness across generation budgets. A potential concern is that AR constraints might overfit the model to specific trajectory lengths. However, we observe robust performance gains across varying sequence lengths (128, 256, 512). This stability suggests that the policy has improved its fundamental reasoning capability, *i.e.*, the ability to navigate logical branches, rather than merely memorizing fixed-length patterns.

5.2. JustGRPO Preserves Parallel Decoding

We further investigate whether the AR training constraint compromises the model's inherent diffusion capabilities. We employ the training-free Entropy Bounded (EB) Sampler (Ben-Hamu et al., 2025) to evaluate inference performance under varying degrees of parallelism (tokens per step).

Figure 7 reveals that our model not only retains full compatibility with parallel decoding but exhibits a strictly superior trade-off between speed and accuracy. Surprisingly, the performance gain becomes more pronounced as parallelism increases. As shown in the MBPP and HumanEval results, while the baseline's

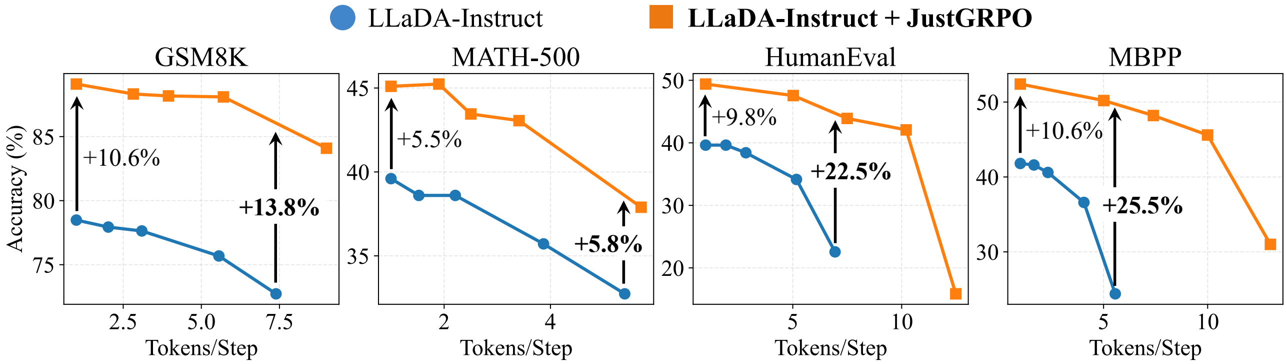

Figure 7: **JustGRPO preserves the parallel decoding capability of dLLMs.** Interestingly, when compared to the original instruct model, accuracy gain is larger with more parallel tokens, likely due to more robust reasoning capabilities after JustGRPO training. We adopt training-free EB-sampler (Ben-Hamu et al., 2025) for parallel decoding.

performance degrades sharply with more aggressive parallel steps, our model maintains stability. For instance, on MBPP, the accuracy gap expands from $+10.6\%$ at conservative settings (1 tokens/step) to $+25.5\%$ at aggressive settings ($\sim$5 tokens/step).

This observation suggests that JustGRPO does not merely fit a specific decoding path; rather, it learns a more robust reasoning manifold that is resilient to the approximation errors inherent in parallel sampling. The AR formulation thus acts as an effective training scaffold that refines the joint distribution $p(o)$ during optimization, providing a more stable foundation for parallel samplers to exploit at inference.

6. Related Work

Diffusion language models. Inspired by the success of diffusion models in continuous image domains (Ho et al., 2020; Rombach et al., 2022), recent work has extended diffusion to discrete text generation. Early approaches operating in continuous embedding spaces (Li et al., 2022; Gong et al., 2022; Han et al., 2022) suffered from optimization and discretization issues. In contrast, masked diffusion models (Lou et al., 2023; Sahoo et al., 2024; Shi et al., 2024; Ou et al., 2024), which define the diffusion process directly in the discrete token space via random masking, have become the dominant paradigm. Notably, recent large-scale models such as LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025) demonstrate scalability and performance competitive with autoregressive (AR) models. The emergence of such large-scale diffusion language models has spurred a growing body of follow-up studies exploring their distinctive modeling and inference properties. Among various advantages discussed in the literature, two aspects have received particular attention. First, dLLMs naturally support parallel decoding, enabling significant inference acceleration (Wu et al., 2025b; Ben-Hamu et al., 2025; Labs et al., 2025; DeepMind, 2025; Song et al., 2025). Second, their non-autoregressive formulation allows for arbitrary-order token generation, which has been hypothesized to benefit complex reasoning by relaxing strict left-to-right constraints (Ye et al., 2024; Kim et al., 2025).

The value of order arbitrariness. While early studies validated the utility of arbitrary order generation in constrained tasks like Sudoku and Zebra Puzzles (Ye et al., 2024; Nie et al., 2025; Kim et al., 2025), recent research has begun to investigate its value in standard reasoning domains. One line of research, exemplified by Dream-Coder (Xie et al., 2025) and DiffuCoder (Gong et al., 2025), shows that diffusion models can naturally exhibit non-standard, human-like decoding behaviors (*e.g.*, sketch-first reasoning) without any explicit supervision on generation order. DiffuCoder (Gong et al., 2025) further quantifies this phenomenon by demonstrating that higher sampling temperatures reduce AR-ness, with increased order randomness correlating with improved output diversity. Another line of research, exemplified by P2 (Peng et al., 2025) and LLaDOU (Huang et al., 2025b), explicitly optimizes the decoding strategy for better generation order, thus better performance. Despite these advances, it remains unclear whether the observed gains primarily arise from better exploitation of existing solution patterns encoded in the data and model, or whether order arbitrariness itself enables qualitatively new reasoning strategies that are unattainable under a purely autoregressive decoding regime.

Reinforcement learning for diffusion language models. Reinforcement learning for dLLMs faces structural optimization hurdles distinct from the autoregressive paradigm, primarily stemming from the combinatorial explosion of denoising trajectories. While early attempts (Zhao et al., 2025; Yang et al., 2025; Gong et al., 2025) sought to adapt token-level formulations directly, they were fundamentally limited by ill-defined state transitions, necessitating reliance on unstable mean-field approximations. Consequently, the field has shifted toward sequence-level perspectives (Zhu et al., 2025; Wang et al., 2025a; Rojas et al., 2025; Ou et al., 2025), employing various surrogates to approximate the intractable marginal likelihood. Yet, a critical off-policy misalignment persists across these methods: the heuristic-guided sampling required for efficient exploration diverges from the underlying diffusion prior, rendering gradients biased without principled correction (Schulman et al., 2015). Two notable exceptions partially address these issues. LLaDOU (Huang et al., 2025b) explicitly models token position selection via an auxiliary policy, enabling direct estimation of trajectory likelihoods, while TraceRL (Wang et al., 2025d) aligns optimization with inference traces through a shrinkage-based step-wise MDP formulation. Nevertheless, these approaches remain committed to preserving the full arbitrary-order generation mechanism, implicitly treating its structural complexity as indispensable, whereas we question whether effective RL training can be achieved more simply by reexamining the necessity of arbitrary order itself.

7. Conclusion

The intuitive appeal of diffusion language models (dLLMs) lies in their order arbitrariness, often perceived as a superior mechanism for navigating complex reasoning paths. Our study reveals a counter-intuitive reality: this unrestricted flexibility in fact narrows the reasoning potential. By allowing the model to bypass high-entropy tokens, effectively skipping the most demanding logical branches, arbitrary-order generation acts as an exploitation mechanism that prioritizes greedy optimization of individual trajectories at the expense of broader solution coverage.

Therefore, the elicitation of the reasoning capability of dLLMs can be easier. By operating dLLMs in a standard autoregressive way, we enable the direct application of Group Relative Policy Optimization (GRPO) without any complex adaptations tailored for order arbitrariness. This intentional constraint paradoxically yields a significant upgrade in reasoning performance, while fully preserving the parallel decoding capabilities of dLLMs. By returning to the basic, natural left-to-right order of language modeling, we hope to encourage a re-examination of its real value in the training of next-generation diffusion models.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

Ben-Hamu, H., Gat, I., Severo, D., Nolte, N., and Karrer, B. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025. 7, 8, 9

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

Chen, M. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 2, 3, 4

Cheng, D., Huang, S., Zhu, X., Dai, B., Zhao, W. X., Zhang, Z., and Wei, F. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025. 2, 5

DeepMind. Gemini diffusion, 2025. URL https://deepmind.google/models/gemini-diffusion/. 2, 9

Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022. 9

Gong, S., Zhang, R., Zheng, H., Gu, J., Jaitly, N., Kong, L., and Zhang, Y. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025. 2, 8, 9, 10, 14

Han, X., Kumar, S., and Tsvetkov, Y. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*, 2022. 9

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 9

Huang, G., Xu, T., Wang, M., Yi, Q., Gong, X., Li, S., Xiong, R., Li, K., Jiang, Y., and Zhou, B. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2510.03222*, 2025a. 2, 5

Huang, Z., Chen, Z., Wang, Z., Li, T., and Qi, G.-J. Reinforcing the diffusion chain of lateral thought with diffusion language models. *arXiv preprint arXiv:2505.10446*, 2025b. 8, 9, 10, 14

Kim, J., Shah, K., Kontonis, V., Kakade, S., and Chen, S. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025. 2, 9, 15

Labs, I., Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025. 2, 9

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022. 9

Liu, M., Diao, S., Lu, X., Hu, J., Dong, X., Choi, Y., Kautz, J., and Dong, Y. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025. 2, 4

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023. 9

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. 2, 5, 7, 8, 9, 15

Ou, J., Nie, S., Xue, K., Zhu, F., Sun, J., Li, Z., and Li, C. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024. 9

Ou, J., Han, J., Xu, M., Xu, S., Xie, J., Ermon, S., Wu, Y., and Li, C. Principled rl for diffusion llms emerges from a sequence-level perspective. *arXiv preprint arXiv:2512.03759*, 2025. 2, 3, 6, 8, 10, 14

Peng, Z., Bezemek, Z., Patel, S., Rector-Brooks, J., Yao, S., Tong, A., and Chatterjee, P. Path planning for masked diffusion model sampling. *ArXiv*, abs/2502.03540, 2025. URL https://api.semanticscholar.org/CorpusID:276161145. 9

Rojas, K., Lin, J., Rasul, K., Schneider, A., Nevmyvaka, Y., Tao, M., and Deng, W. Improving reasoning for diffusion language models via group diffusion policy optimization. *arXiv preprint arXiv:2510.08554*, 2025. 3, 10

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 9

Sahoo, S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024. 9

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015. 6, 10

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Xiao, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. 3

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024. 9

Song, Y., Zhang, Z., Luo, C., Gao, P., Xia, F., Luo, H., Li, Z., Yang, Y., Yu, H., Qu, X., et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025. 2, 9

Wang, C., Rashidinejad, P., Su, D., Jiang, S., Wang, S., Zhao, S., Zhou, C., Shen, S. Z., Chen, F., Jaakkola, T., et al. Spg: Sandwiched policy gradient for masked diffusion language models. *arXiv preprint arXiv:2510.09541*, 2025a. 2, 3, 6, 10

Wang, J., Liu, R., Zhang, F., Li, X., and Zhou, G. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*, 2025b. 5

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X., Yang, J., Zhang, Z., et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025c. 2, 5, 15

Wang, Y., Yang, L., Li, B., Tian, Y., Shen, K., and Wang, M. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*, 2025d. 10

Wu, C., Zhang, H., Xue, S., Diao, S., Fu, Y., Liu, Z., Molchanov, P., Luo, P., Han, S., and Xie, E. Fast-dllm v2: Efficient block-diffusion llm. *arXiv preprint arXiv:2509.26328*, 2025a. 2

Wu, C., Zhang, H., Xue, S., Liu, Z., Diao, S., Zhu, L., Luo, P., Han, S., and Xie, E. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025b. 2, 5, 9

Xie, Z., Ye, J., Zheng, L., Gao, J., Dong, J., Wu, Z., Zhao, X., Gong, S., Jiang, X., Li, Z., et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025. 9

Yang, L., Tian, Y., Li, B., Zhang, X., Shen, K., Tong, Y., and Wang, M. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025. 2, 10

Ye, J., Gao, J., Gong, S., Zheng, L., Jiang, X., Li, Z., and Kong, L. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024. 2, 9

Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025. 2, 5, 9, 15

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. 2, 3, 4, 5

Zeng, H., Jiang, D., Wang, H., Nie, P., Chen, X., and Chen, W. Acecoder: Acing coder rl via automated test-case synthesis. *arXiv preprint arXiv:2502.01718*, 2025. 8, 14

Zhang, C., Neubig, G., and Yue, X. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025. 2, 4

Zhao, S., Gupta, D., Zheng, Q., and Grover, A. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*, 2025. 2, 5, 7, 8, 10, 14

Zhu, F., Wang, R., Nie, S., Zhang, X., Wu, C., Hu, J., Zhou, J., Chen, J., Lin, Y., Wen, J.-R., et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025. 2, 5, 10

Appendix

A. Experimental Details

A.1. Data Preparation

For mathematical reasoning tasks, we train on the official training split of each dataset, following the standard protocol in (Zhao et al., 2025; Ou et al., 2025). For code generation tasks, we adopt the **AceCoder-87K** dataset (Zeng et al., 2025). Following the data processing pipeline of DiffuCoder (Gong et al., 2025), we further select 21K challenging samples from AceCoder-87K that are equipped with verifiable unit tests.

A.2. Training Configuration

Our training setup largely follows (Huang et al., 2025b). Different from (Huang et al., 2025b), we perform reinforcement learning directly on the dLLM without introducing training modules. During the rollout phase, we adopt exact autoregressive sampling, which allows direct log-probability computation under the standard GRPO formulation. We also reduce the total number of training steps, as JustGRPO exhibits fast and stable convergence. All experiments are conducted on 16× NVIDIA H100 GPUs. Training on GSM8K takes approximately three days. Detailed hyperparameters are reported in Table 2.

Table 2: Training hyperparameters for JustGRPO.

Hyperparameter	**Value**
Base Model	LLaDA 8B Instruct
RL Algorithm	GRPO
Optimizer	AdamW
Learning Rate	5×10^{-6}
LR Scheduler	Constant
Weight Decay	0.0
Optimizer Betas (β_1, β_2)	$(0.9, 0.999)$
Global Batch Size	64
Group Size (G)	16
Policy Update Steps	1
Training Steps	125
Max Completion Length	256
Sampling Temperature	1.0
KL Penalty Coefficient	0.0

A.3. Reward Function

Mathematical Reasoning Tasks. We employ a binary reward scheme. Each completion receives a reward of 1 if and only if the final answer is mathematically equivalent to the ground-truth solution, and 0 otherwise following (Huang et al., 2025b).

Code Generation Tasks. For code generation, the total reward r is defined as a weighted sum of a correctness reward r_{code} and a format reward r_{format}:

$$r = r_{\text{code}} + r_{\text{format}}. \tag{7}$$

- **Correctness reward (r_{code})**: Defined as the pass rate (ranging from 0 to 1) of the generated code on the provided unit tests. This term is only evaluated when $r_{\text{format}} = 1$.
- **Format reward (r_{format})**: A heuristic reward designed to encourage syntactically valid outputs.
  - 1.0: A valid Markdown code block with correct Python syntax.
  - 0.5: A valid Markdown code block that contains syntax errors.
  - 0.0: Failure to generate a valid Markdown code block.

B. More Analysis Results

To further validate the robustness of our findings in Section 3, we conduct extended analyses on the HumanEval benchmark using the LLaDA-Instruct model (Nie et al., 2025). We investigate the impact of temperature, sampling strategies, and decoding block sizes on reasoning performance.

B.1. Temperature analysis.

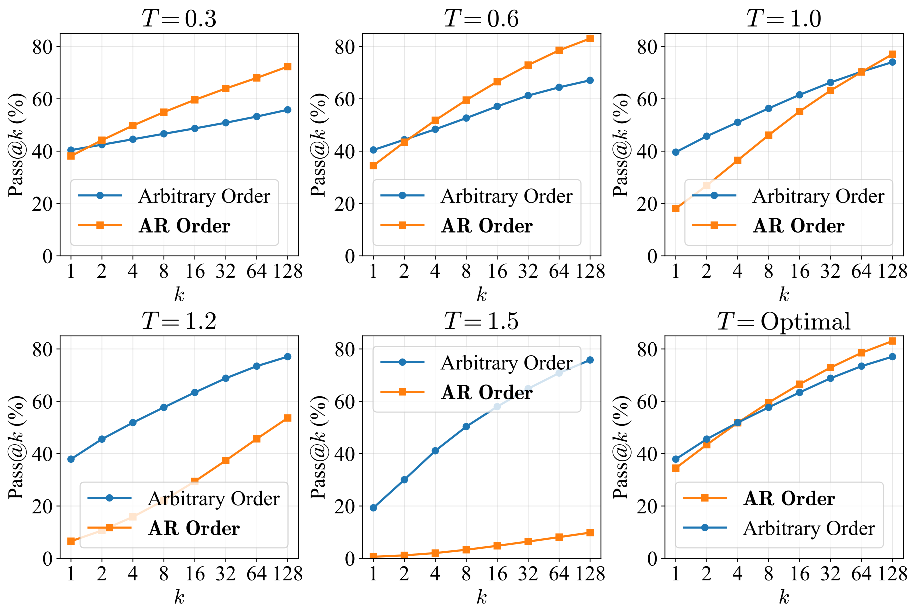

Figure 8: **Pass@K comparison with different temperatures**.

As shown in Figure 8, the AR mode exhibits a standard pattern: performance peaks at moderate temperatures ($T \approx 0.6$) and degrades when $T > 1.0$, whereas the Arbitrary Order mode attains its peak performance at higher temperatures.

This observation aligns with the *entropy degradation* mechanism discussed in Section 3: the diffusion sampler inherently suppresses uncertainty at critical branching points, thereby requiring higher temperatures to induce sufficient exploration. Crucially, even under these optimized settings, the arbitrary order mode fails to match the reasoning potential of the AR mode. As illustrated in the "Optimal" comparison setting (bottom right of Figure 8), the best-performing AR configuration still outperforms the optimal arbitrary order baseline in scaling behavior. A plausible explanation is that excessively high temperatures in arbitrary order decoding inject noise into tokens that require high determinism (*e.g.*, code syntax or mathematical suffixes), leading to nonsensical outputs and degraded results (Wang et al., 2025c).

B.2. Different sampling algorithms.

We also experiment with different sampling algorithms, such as negative entropy sampling (Neg-Entropy) (Ye et al., 2025) and top-k margin sampling (Margin) (Kim et al., 2025) in Figure 9(a). The results show that although a more considered sampling algorithm can achieve better pass@k performance than default confidence-based sampling, it still cannot catch up with AR order. Meanwhile, these better pass@k algorithms also show slightly worse pass@1 performance compared to confidence-based sampling, making the whole pass@k curve more close to AR order's pass@k curve. To investigate this similarity, we calculate the per-problem accuracy correlation between different sampling algorithms and the AR mode (Figure 9(b)). We observe that algorithms with higher scaling potential (higher pass@128) consistently show stronger correlation with AR, with the most effective method (Neg-Entropy) achieving a correlation

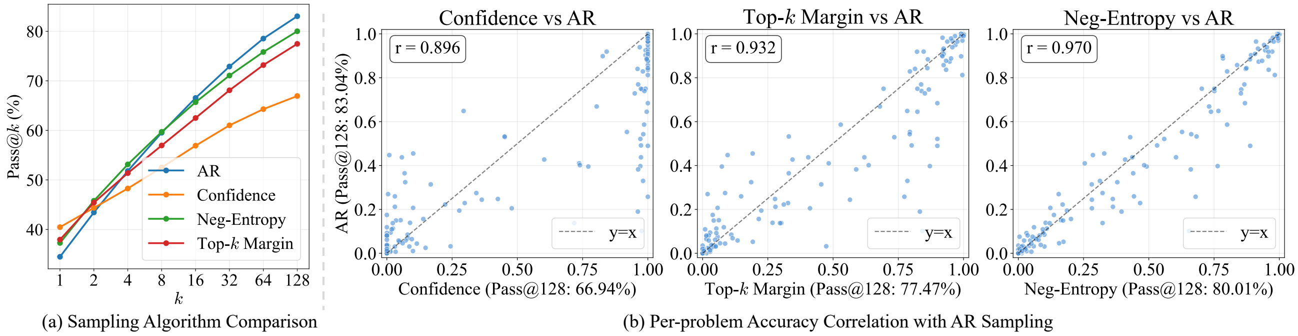

Figure 9: **(a) Sampling algorithm comparison**. **(b) Correlation between different sampling algorithms and AR in per-problem accuracy**. We compare different sampling algorithms in pass@k performance and correlation with AR in per-problem accuracy.

of 0.970. This suggests that sampling algorithms with better pass@k tend to behave more like AR in terms of task-level performance characteristics.

B.3. Block size analysis.

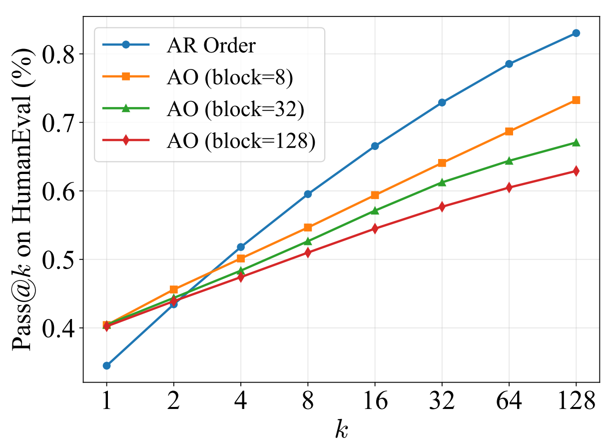

Figure 10: **Pass@k comparison with different semi-autoregressive block sizes** of arbitrary order (AO) generation. Smaller block sizes explicitly restrict the model's order flexibility, leading to a more AR-like behavior and better pass@k performance.

We analyze the effect of the semi-autoregressive block size in Figure 10. The AR Order (which is effectively block size of 1) maintains a clear advantage across all tested block sizes. Furthermore, we observe a trend of improvement in pass@k as the block size decreases (from 128 to 8). Since smaller block sizes explicitly restrict the model's order flexibility, this trend is aligned with our finding that less order flexibility leads to better reasoning potential.

B.4. Entropy Comparison Results on More Forking Tokens

To validate the robustness of the "entropy degrade" phenomenon observed in Section 3, we extended our analysis to a wider range of logical connectors. We conducted experiments on a comprehensive set of common logical connectors that typically serve as pivotal decision points in reasoning chains.

The results, shown in Figure 11, demonstrate that the phenomenon is consistent across these diverse tokens. Similar to the primary findings, the AR order maintains higher average entropy at these forks,

Figure 11: **Entropy comparison results on more forking tokens**.

indicating active reasoning and decision-making. Conversely, the Arbitrary Order consistently results in lower entropy, confirming the premature collapse of branching possibilities.

The specific forking tokens evaluated in this extended experiment include: “Therefore”, “Thus”, “So”, “Since”, “When”, “Given”, “However”, “Let”, “First”, “Then”, “Next”, “Finally”, “Now”, “Similarly”, “Calculate”, “Solving”, “Notice”, “Specifically”, “Follows”, “Because”, “But”, “Or”, “Consider”, “Also”, “Express”, and “Write”.