Single-Step Initialization for Exploratory Parallel Rollouts in Diffusion LLMs

Dongjae Jeon [*1] **Bumjun Kim** [*2] **Mingyu Kim** [3] **Albert No** [2]

Abstract

We propose training-free parallel decoding for rollout generation in diffusion large language model (dLLM) policy optimization, reducing rollout cost without auxiliary models or policy modification. We find, however, that confidence-based decoding suffers from delayed branching, and parallel decoding largely inherits this characteristic. Rollouts agree on both unmasked tokens and positions for much of generation, leading to a lack of exploration that weakens the group-relative learning signal. We address this with a minimal initialization step in which each rollout independently unmasks one uniformly random position after which the original sampler resumes unchanged. The intervention is drop-in compatible with any sampling strategies. Combined with Fast-dLLM on LLaDA-8B-Instruct, it improves rollout diversity and yields stronger downstream RL performance on GSM8K and MATH-500.

1. Introduction

Diffusion large language models (dLLMs) have recently emerged as a promising alternative to autoregressive (AR) language models (Austin et al., 2021; Sahoo et al., 2024; Lou et al., 2024; Nie et al., 2025). Unlike AR models, dLLMs generate text by iteratively denoising a fully masked sequence, allowing flexible generation order and parallel unmasking of multiple positions (Wu et al., 2025; Kim et al., 2025b; Luo et al., 2026).

Recent policy optimization methods for dLLMs adapt GRPO by sampling multiple rollouts per prompt and updating the policy with group relative rewards (Wang et al., 2025a; Zhao et al., 2025; Tang et al., 2025). However, rollout generation remains a major bottleneck, since full sequence attention prevents standard AR accelerations such as exact KV caching. Although several methods accelerate rollout generation in AR models, they often require additional modules or distribution mismatch correction, which is difficult to apply to dLLMs due to the intractability of policy likelihoods (Liu et al., 2025b; Huang et al., 2025; Li et al., 2026). Training free parallel decoding therefore provides a natural solution: it reduces rollout cost without modifying the policy or introducing extra model components.

However, fast and accurate parallel decoding alone is not sufficient. RL training also requires exploratory rollouts that cover diverse reasoning paths, since group-relative advantages vanish when samples follow similar trajectories and receive similar rewards. Recent work shows that confidence-based sampling often lacks such diversity (Ni et al., 2026; Shen et al., 2026). We observe that parallel decoding inherits and amplifies this limitation. By prioritizing high-confidence positions to suppress the joint-marginal mismatch, these methods produce rollouts with nearly identical early trajectories, and the dependence is in fact stronger than in step-by-step decoding because unmasking multiple less-confident tokens together severely degrades sample quality. As a result, parallel decoding strategies yield lower lexical diversity, solution-level diversity, and reasoning potential.

We trace this diversity collapse to delayed branching. Tracking rollouts under confidence-based decoding reveals that samples remain in lockstep through much of the generation, agreeing on both the tokens unmasked and the positions chosen. By the time they finally diverge, a large fraction of the sequence has already been unmasked, and the resulting variation is confined to surface details rather than core reasoning paths for meaningful exploration.

Motivated by this finding, we propose a simple initialization strategy that forces early branching. At the first generation step, each rollout independently unmasks one uniformly random position, after which the original decoding strategy resumes unchanged. The intervention is simple and compatible with any step-by-step or parallel decoding strategy. By breaking the shared fully-masked starting state, it induces branching before confidence-based selection takes over, recovering rollout diversity without modifying the sampler.

[1]Department of Computer Science, Yonsei University, Seoul, Korea [2]Department of Artificial Intelligence, Yonsei University, Seoul, Korea [3]School of Artificial Intelligence, Kookmin University, Seoul, Korea. Correspondence to: Mingyu Kim <mgyukim@kookmin.ac.kr>, Albert No <albertno@yonsei.ac.kr>.

Proceedings of the 43rd International Conference on Machine Learning, RLxF: Reinforcement Learning from World Feedback Workshop, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

Experiments on math tasks (GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2023)) show that this small intervention substantially improves rollout diversity in parallel decoding and yields better downstream RL performance than the baselines. These gains are achieved without modifying the underlying decoding strategy, suggesting that the diversity bottleneck can be addressed by initialization alone.

Our contributions are summarized as follows.

- We propose using training-free parallel decoding for rollout generation in dLLM policy optimization, reducing rollout cost without auxiliary models or policy modification.
- We identify delayed branching as a key source of diversity collapse in confidence-based decoding. We further show that parallel decoding inherits and amplifies this limitation, since suppressing the joint-marginal mismatch forces these methods to select confident tokens.
- We introduce a simple initialization strategy in which each rollout independently unmasks one random position at the first step, after which the original sampler resumes unchanged. The method is drop-in compatible with any sampling algorithms.
- We show that a single initialization step substantially improves rollout diversity, reasoning potential, and downstream RL performance. Combined with Fast-dLLM, our method yields consistent gains on mathematical reasoning benchmarks.

2. Preliminaries

2.1. Masked Diffusion Language Modeling

Masked Diffusion Models (MDMs) define the diffusion process by corrupting the original sequence through token masking (Austin et al., 2021; Lou et al., 2024; Sahoo et al., 2024). Given a clean sequence $\mathbf{x}_0 = x_0^1 \dots x_0^L \in \mathcal{V}^L$, a time variable $t \in [0, 1]$ is sampled uniformly, and each token is independently replaced with the special mask token $[\mathbf{M}]$ with probability t. This produces a masked sequence $\mathbf{x}_t$. The model is trained to estimate $p_\theta(x_0^i|\mathbf{x}_t)$, thereby *predicting each masked token from the observed partial sequence*. The training objective is a weighted cross entropy loss over the masked tokens:

$$\mathcal{L}_{\text{MDM}}(\mathbf{x}_0; \theta) = -\mathbb{E}_{t,\mathbf{x}_t}\left[\frac{1}{t}\sum_{i:x_t^i=[\mathbf{M}]} \log p_\theta(x_0^i|\mathbf{x}_t)\right],$$

which serves as a variational upper bound on the negative log likelihood $-\log p_\theta(\mathbf{x}_0)$ (Shi et al., 2024; Ou et al., 2025).

Once trained, the model provides a token wise conditional distribution $p_\theta(x^i|\mathbf{x})$ for each position i under any partially masked sequence $\mathbf{x} \in \overline{\mathcal{V}}^L$, where $\overline{\mathcal{V}} = \mathcal{V} \cup \{[\mathbf{M}]\}$. Generation begins from the fully masked sequence $\mathbf{x}_1 = [\mathbf{M}] \cdots [\mathbf{M}]$ and proceeds by gradually filling in masked positions. At the distributional level, this process does not require a unique order, since any permutation π induces the factorization $p(\mathbf{x}_0) = \prod_{i=1}^L p(x^{\pi(i)}|\mathbf{x}^{\pi(<i)})$, making any unmasking order theoretically valid. In practice, however, the chosen order substantially affects generation quality (Zhong et al., 2026; Zheng et al., 2024). A widely used strategy is confidence based sampling (Kim et al., 2025a; Nie et al., 2025; Chang et al., 2022), which first selects the masked position with the highest maximum predicted probability. By prioritizing low uncertainty tokens, this strategy first fills positions whose predictions are relatively stable, thereby providing more trustworthy partial context for subsequent denoising steps.

Building on the MDM framework, recent work has extended masked diffusion to large scale language modeling, leading to Diffusion Large Language Models (dLLMs). Representative models include LLaDA (Nie et al., 2025; Zhu et al., 2025) and Dream (Ye et al., 2025).

2.2. Parallel Decoding in Diffusion Language Models

A key advantage of dLLMs is their ability to decode in parallel, where multiple masked positions can be filled during a single denoising step. Since the main inference cost is dominated by repeated full sequence forward passes, or function evaluations (NFE), fewer denoising steps translate directly into faster generation. This makes parallel decoding particularly attractive for policy optimization, where rollouts require repeated sampling of many sequences.

This efficiency, however, comes with a limitation. dLLMs are trained to predict conditional marginal distributions at individual masked positions, whereas unmasking multiple tokens at once requires modeling their joint distribution. This leads to errors when the true joint distribution differs from the product of the corresponding marginals (Ben-Hamu et al., 2025; Wu et al., 2025):

$$p(x^{i_1}, \dots, x^{i_k}|\mathbf{x}) \neq \prod_{j=1}^{k} p(x^{i_j}|\mathbf{x}). \tag{1}$$

This mismatch disappears only when the tokens unmasked in the same step are conditionally independent given the current partially masked sequence. Therefore, many parallel decoding strategies aim to identify sets of weakly dependent tokens at each decoding step, allowing them to increase parallelism while preventing joint marginal errors from accumulating significantly (Ben-Hamu et al., 2025; Wu et al., 2025; Kim et al., 2025b; Luo et al., 2026).

Training-free parallel decoding for dLLMs. One line of work addresses this problem using additional training, auxiliary models, or modified decoding objectives (Israel et al., 2025; Bao et al., 2025; Chen et al., 2025b). These approaches tackle inter token coupling more explicitly, but they introduce extra cost and may deviate from the original policy through surrogate objectives, auxiliary targets, or training inference mismatches. Since our goal is to reduce the cost of rollout generation while preserving the original policy, we focus on training free methods that can be applied directly without additional training or policy modification.

Training-free methods must select tokens that minimize the *joint-marginal mismatch* in Equation (1) using only the conditional marginals provided by the model. Existing approaches mainly rely on confidence-based signals, prioritizing the parallel decoding of high-confidence tokens. The simplest strategy ranks masked positions by marginal confidence and decodes the top-k positions together (Nie et al., 2025). **Fast-dLLM** (Wu et al., 2025) performs selective parallel decoding by unmasking only positions whose confidence exceeds a fixed threshold (e.g., 0.9).

These criteria are well motivated in high-certainty regimes: when each selected token has probability close to one, its prediction is nearly deterministic, so the gap between the true joint and the product of marginals shrinks. Thus, high confidence serves as a practical proxy for selecting tokens that can be decoded together safely.

2.3. Policy Optimization in Diffusion Language Models

Recent methods adapt GRPO (Shao et al., 2024) to dLLMs by preserving its group-relative rewards while modifying the policy update to address diffusion-specific challenges (Zhao et al., 2025; Tang et al., 2025; Wang et al., 2025a). Given a prompt q, GRPO samples a group of G rollouts $\{o_1, \ldots, o_G\} \sim \pi_{\text{old}}(\cdot \mid q)$, and computes the group-normalized advantage

$$\hat{A}_i = \frac{R(q, o_i) - \text{mean}\big(\{R(q, o_j)\}_{j=1}^{G}\big)}{\text{std}\big(\{R(q, o_j)\}_{j=1}^{G}\big)}. \tag{2}$$

This advantage captures the relative quality of o_i within the group and is broadcast to all rollout tokens during the update. Thus, diverse rollouts are important: when samples lead to different rewards, they provide stronger reward contrast and more informative advantages for policy learning. Applying GRPO to dLLMs is challenging because the policy likelihoods required by the GRPO ratio are computationally intractable (Zhao et al., 2025; Tang et al., 2025; Wang et al., 2025a). Existing methods address this through different likelihood surrogates. **SPG** (Wang et al., 2025a) replaces the likelihood with a sign dependent sandwiched surrogate, using a lower bound for positive advantage samples and an upper bound for negative advantage samples.

All methods require G rollouts per prompt to compute $\hat{A}_i$. This is particularly costly for dLLMs because full sequence attention prevents exact KV caching across denoising steps, forcing each rollout to perform a full forward pass at every step and increasing both latency and memory usage. While rollout acceleration techniques have been developed for autoregressive LLMs, such as speculative decoding and quantized rollout models (Liu et al., 2025b;a), they do not directly transfer to dLLMs. These methods either require additional model copies and memory overhead, or rely on importance sampling to correct policy mismatch, which again requires likelihood ratios that are intractable for dLLMs. This motivates the use of parallel decoding for rollout generation, as it reduces the number of denoising steps without modifying the policy or introducing auxiliary models.

3. Fast and Accurate Parallel Decoding Is Not Enough for Rollouts

3.1. Rollout Diversity in Policy Optimization

Rollout diversity is central to policy optimization in LLM reasoning and general RL, and becomes more critical under sparse rewards such as RLVR (Shao et al., 2024; Wang et al., 2025b; Yu et al., 2026). Diverse rollouts cover a broader range of reasoning trajectories and outcomes, providing richer exploratory signals, and reducing premature concentration on narrow solution modes (Wang et al., 2022; Trott et al., 2019; Haarnoja et al., 2018; Ahmed et al., 2019; Yao et al., 2025). GRPO depends on this property directly: the group-relative advantage $\hat{A}_i$ (Equation (2)) collapses to zero whenever the G rollouts receive the same reward, erasing the learning signal entirely.

The Diversity Cost of Confidence-based Sampling. Recent work has shown that confidence-based sampling in dLLMs often suffers from diversity collapse (Ni et al., 2026; Chen et al., 2025a; Shen et al., 2026). These works share a common diagnosis: confidence-based sampling improves single-sample quality (greedy accuracy) by unmasking the model's most certain predictions, but performance saturates quickly under repeated sampling. Ni et al. (2026) makes this point directly, arguing that even forcing AR-style generation is preferable in RL training.

Training-free parallel decoding strategies naturally inherits this limitation. As discussed in Section 2.2, existing methods prioritize unmasking high-confidence tokens together to avoid the *joint–marginal* mismatch (See Equation (1)) and prevent error accumulation. In the parallel decoding regime, this dependence on confidence is even more restrictive: while uncertain tokens may still be explored sequentially, unmasking several uncertain tokens simultaneously can quickly corrupt the entire generation.

Setup. To quantify the diversity degradation in parallel decoding, we evaluate parallel decoding method (Fast-dLLM) with two step-by-step baselines (AR, Confidence). AR uses forced left-to-right generation, while Confidence selects the highest-confidence position at each step. For each setting, we draw 16 samples per question on a random 100-question subset of GSM8K. Following Yao et al. (2025), we report two classes of metrics.

(1) *Sample diversity* is measured by **Div-N-gram** and **Div-Self-BLEU** at the lexical level and by **Div-Eq** at the solution level, where Div-Eq is the fraction of distinct equations across samples; higher values indicate greater diversity.

(2) *Reasoning potential* is measured by **Potential@K (Yao et al., 2025)**, a Pass@K variant evaluated on problems that greedy decoding fails, which correlates strongly with post-training reasoning performance. We use temperature 0.9 by default. For more details in setup, refer to Appendix A.

Method	Div-1	Div-2	Div-4	Div-SB	Div-Equ
AR	**11.10**	**30.10**	**56.40**	**21.30**	**38.70**
Confidence	6.90	15.80	27.80	7.30	22.20
Fast-dLLM	6.93	15.89	27.84	7.26	20.09

Table 1. Average diversity across 16 samples per question. Div-N denotes distinct N gram, and Div-SB denotes Div-Self-BLEU. Higher values indicate greater lexical and solution-level diversity.

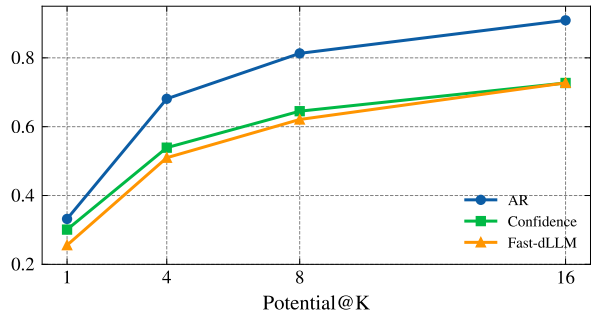


Figure 1. Potential@K results for $K \in \{1, 4, 8, 16\}$.

Empirical Evidence of Diversity Collapse. As shown in Table 1, Fast-dLLM exhibits low diversity across every metric and closely match the step-by-step Confidence baseline. The degradation appears at both the lexical and solution levels, meaning these methods repeatedly generate samples with similar surface forms and reasoning paths. AR sampling sits above the entire confidence-based family on every diversity metric, and the same ordering reappears in Potential@K (See Figure 1): every confidence-based sampler, parallel or step-by-step, falls below AR. Lost diversity thus comes paired with lost reasoning potential.

Temperature Does Not Recover Diversity. To check whether the diversity gap closes when we simply sample at a higher temperature, we re-run the experiment at $T = 1.2$. As shown in Table 2, the effect on every confidence-based sampler is marginal. Compared with the $T = 0.9$ setup (Table 1), the differences are small across all metrics, and Div-Eq—the solution-level measure—is essentially unchanged. AR at $T = 0.9$ still upper-bounds every confidence-based sampler at $T = 1.2$, confirming that the diversity collapse is structural to confidence-based selection.

Method	Div-1	Div-2	Div-4	Div-SB	Div-Equ
AR ($T = 0.9$)	**11.10**	**30.10**	**56.40**	**21.30**	**38.70**
Confidence	7.40	17.90	32.60	9.20	20.10
Fast-dLLM	7.48	17.91	32.51	9.11	21.24

Table 2. Average diversity across 16 samples per question at sampling temperature $T = 1.2$. The AR result at $T = 1.2$ is omitted due to severe quality degradation under high temperature sampling. Full results, including Potential@K, in Appendix A.

Although AR style sampling achieves higher diversity in the step by step regime, this remedy does not directly carry over to parallel decoding. AR fixes the unmasking order to be left-to-right, one token per step, whereas parallel decoding treats the choice of *which positions to unmask jointly* as the central algorithmic problem. Any practical remedy must therefore be minimal yet effective: preserving the selection rule of existing parallel strategies while still increasing exploration.

3.2. Confidence-based Sampling Delays Branching

To identify the cause of diversity collapse, we further track the sampling trajectories of AR style and confidence based decoding throughout the denoising process. For analytical clarity, we restrict this comparison to step by step samplers.

Setup. Specifically, for each question we identify when the $K = 16$ samples initially diverge during generation. We define and track two branch points: the **Vocab Branch Point** marks the first step at which samples stop unmasking the same token, and the **Position Branch Point** marks the first step at which they stop unmasking at the same position.

Let $v_t^{(r)}$ and $i_t^{(r)}$ denote the token and position unmasked by sample r at step t, respectively. The vocab consensus fraction at step t is defined as the fraction of samples occupied by the most frequently unmasked token,

$$C_{\text{vocab}}(t) = \max_{v \in \mathcal{V}} \frac{\left|\{r \in [K] : v_t^{(r)} = v\}\right|}{K},$$

and the Vocab Branch Point τ_{vocab} is the first step at which this fraction drops below a threshold ρ:

$$\tau_{\text{vocab}} = \min \{t : C_{\text{vocab}}(t) \leq \rho\}.$$

The Position Branch Point τ_{pos} is defined analogously by

replacing the unmasked vocab $v_t^{(r)}$ with the unmasked position $i_t^{(r)}$ and taking the maximum over positions $i \in [L]$.

Intuitively, a smaller τ means the K samples diverge earlier; a larger τ means they continue to unmask the same tokens (or at the same positions) for longer. The threshold ρ controls how much dispersion triggers a branch point: smaller ρ is stricter—the most frequent tokens or position must cover a smaller share of samples. At the extreme, $\rho = 1/K$ triggers a branch point only when all K samples unmask unique tokens (or select different positions).

Method	$\rho = 0.5$		$\rho = 0.3$	
	τ_{vocab}	τ_{pos}	τ_{vocab}	τ_{pos}
AR	4.46	N/A	14.35	N/A
Confidence	14.58	14.96	51.34	55.90

Table 3. Average Branch Points (τ_{vocab}, τ_{pos}) for two step-by-step strategies on GSM8K, out of 256 total generation steps. τ_{pos} for AR is marked *N/A*: AR uses the same unmasking position across samples at every step, so $C_{\text{pos}}(t) = 1$ throughout.

Results. As shown in Table 3, AR branches early while Confidence sampling barely branches at all. At $\rho = 0.5$, AR reaches its vocab branch point around step 4—samples begin to diverge within a handful of generation steps. Confidence sampling, by contrast, holds over-half agreement on both token and position until step 14, and at the looser threshold $\rho = 0.3$ the divergence is delayed even further: roughly 5 of the $K = 16$ samples ($\approx 30\%$) agree on both the same token and the same position for nearly 50 steps—close to a fifth of the total 256-step budget—effectively following the identical trajectory throughout that span.

In short, the apparent flexibility of confidence-based sampling backfires. Although the procedure can freely choose which positions to unmask—unlike AR's fixed left-to-right order—always reaching for the most-confident tokens funnels large fraction of samples onto a dominant trajectory, pushing the branch point far into generation. This is unfortunate: by the time samples diverge, the core problem-solving logic may already be largely fixed, leaving little room for meaningful exploration.

4. Single-token Initialization Forces Branching

Initialization is a well-established source of diversity in diffusion models (Jeon et al., 2024; Guo et al., 2024; Li et al., 2025; Ban et al., 2025). In image diffusion, even deterministic samplers produce diverse outputs simply by varying the random seed, which gives each sample a different Gaussian noise to start from. Interventions at the early sampling steps are also known to have the strongest effect on the final output, making the choice of initialization particularly important (Kim et al., 2026; Kynkäänniemi et al., 2024).

dLLMs are different: every sample begins from the same fully-masked terminal state, so diversity must emerge entirely from the sampling algorithm. As we discussed in the previous section, however, confidence-based sampling fails to inject diversity early: the moment of branching arrives only after a large fraction of tokens have already been unmasked. In other words, the samples remain near-duplicates of one another for much of generation.

Init. Method	JSD_{init}	$\text{JSD}_{\text{sample}}$
Confidence	0.0048	0.0015
Random	0.1104	0.1675

Table 4. Branching effect of the initialization step on GSM8K. JSD_{init} measures the distributional shift from the fully masked state to the initialized state, while $\text{JSD}_{\text{sample}}$ measures divergence across $K = 16$ initialized samples.

Simple Initialization Procedure. To mitigate this delayed branching, we propose a minimal intervention: at the first generation step, each of the K samples unmasks a position chosen uniformly at random, independently across samples. This forces branching at the very first step, regardless of the sampler. From the second step onward, we resume the original decoding strategy unchanged, whether Confidence sampling, Fast-dLLM, or any other parallel method.

Branching Effect. To examine whether single token initialization creates meaningful branching, we compare the model distributions at the fully masked initial state and at the second state, where one random token has been unmasked by the initialization intervention.

We consider two complementary measurements. First, we measure the per-position shift in probability mass induced by the initialization, capturing how strongly the intervention changes the prediction landscape. Second, we measure the variation across the K initialized samples, capturing whether different random initializations induce distinct predictive states. Together, these analyses test whether the initialization both perturbs the common starting state and differentiates each sample early in generation. Both quantities are computed using Jensen–Shannon divergence (JSD).

As shown in Table 4, random token initialization yields substantially larger divergence in both measures compared with confidence based initialization. Here, confidence based initialization corresponds to letting the original confidence sampling algorithm determine the first unmasking step. The much smaller divergence in this counterpart shows that confidence sampling leaves the early sample states highly similar. This agrees with the observation in Table 3: meaningful branching under confidence sampling is delayed until many tokens have already been unmasked.

Recovered Diversity. As shown in Table 5, the early branching induced by random initialization yields substantial diversity gains. Even a single random token at the first step markedly increases diversity, and extending the random initialization to the first three steps amplifies the effect fur-

Method	Div-1	Div-2	Div-4	Div-SB	Div-Equ
Fast-dLLM	7.48	17.91	32.51	9.11	21.24
+ 1 Rand. Init.	9.70	24.50	45.40	17.10	31.90
+ 3 Rand. Init.	11.60	30.30	55.70	23.70	32.50

Table 5. Average diversity across 16 samples per question on GSM8K at sampling temperature $T = 1.2$. Single token initialization substantially increases diversity, with further gains from multiple initialization steps.

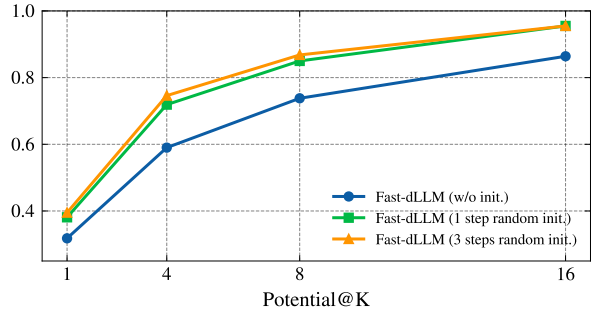


Figure 2. Potential@K results for $K \in \{1, 4, 8, 16\}$. Initialization substantially improves reasoning potential across samples.

ther. The same pattern carries over to Potential@K (See Figure 2), where the single-token intervention already opens a substantial gap over plain Fast-dLLM.

Overall, these results show that the diversity collapse of confidence-based sampling can be largely mitigated with a simple intervention. A single-step random initialization, applied independently across samples, suffices to separate samples early in generation without introducing a whole new sampler. The intervention is compatible with any decoding strategy, since the original sampler resumes unchanged after the first step. Just as different Gaussian noise induce diverse trajectories in image diffusion, a single random unmasking step diversifies dLLM samples even when the remaining steps continue to unmask the most confident tokens.

5. Experiments

In this section, we apply the proposed initialization strategy to parallel decoding and examine whether the increased rollout diversity translates into improved RL performance.

Setup. We use LLaDA-8B-Instruct (Nie et al., 2025) and evaluate mathematical reasoning performance on GSM8K and MATH-500. For policy optimization, we adopt SPG w/ Mixture, the stronger of the two variants reported in Wang et al. (2025a). Rollouts are generated with Fast-dLLM for the parallel-decoding setting. Unless otherwise specified, all methods are trained for 1,500 steps with $G = 8$ rollouts per prompt. Training rollouts are sampled at temperature $T = 1.2$ for all methods.

For final evaluation, all methods use deterministic step by step greedy Confidence decoding, so performance differences stem only from rollout generation. For each method, we evaluate checkpoints saved every 100 training steps and report the best performing checkpoint.

Table 6. Results on SPG w/ Mixture with Fast-dLLM and add-on variants. Acc. denotes final-answer accuracy (%), and NFE denotes the average number of function evaluations.

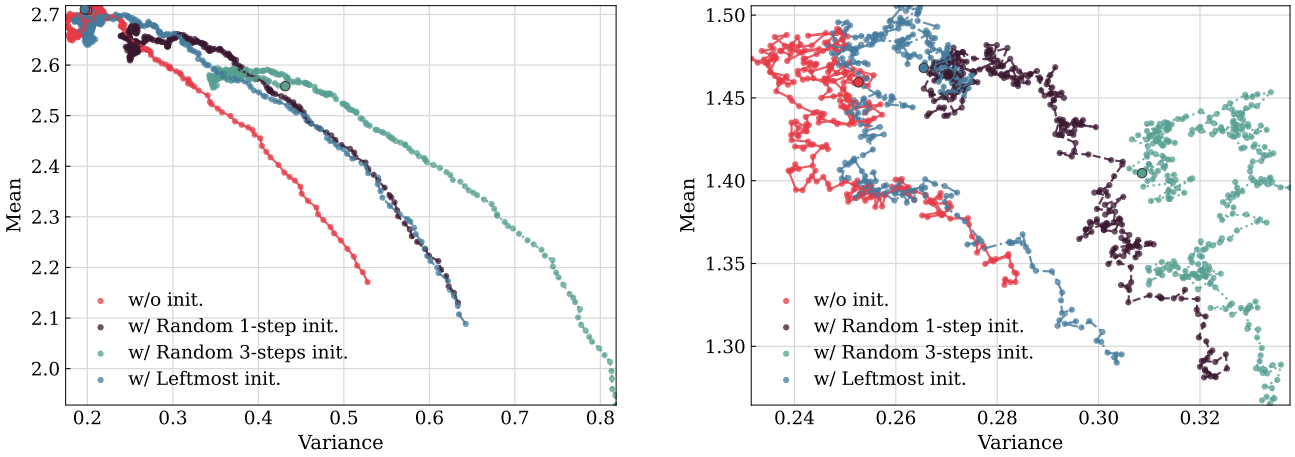
Method	GSM8K		MATH-500	
	Acc. ↑	NFE ↓	Acc. ↑	NFE ↓
Fast-dLLM	82.18	68.53	37.40	98.01
+ Random 1 step	**83.02**	70.81	**40.00**	100.23
+ Random 3 steps	81.80	72.55	36.60	104.71
+ Leftmost 1 step	82.64	66.93	37.60	95.31

Results. We compare Fast-dLLM with three initialization variants. Random 1-step initialization selects one masked answer position uniformly at random for each rollout at the first denoising step before switching to Fast-dLLM, while Random 3-step initialization repeats this procedure for the first three steps. Leftmost initialization instead selects the same leftmost masked answer position for all rollouts, serving as a fixed-position control. After initialization, all variants resume the same Fast-dLLM decoding rule; only the early position-selection policy differs.

As shown in Table 6, Random 1-step initialization improves Fast-dLLM on both GSM8K and MATH-500. This suggests that a small amount of rollout-specific early branching can make rollout generation more effective for policy optimization. Leftmost initialization yields only modest gains, indicating that the improvement is not simply due to adding an initialization step, but to introducing variation across rollouts at the beginning of generation.

However, more random initialization is not always better. Random 3-step initialization performs worse than the one-step variant and falls below the original Fast-dLLM baseline on both datasets. This suggests a quality–diversity trade-off: stronger random initialization can further increase rollout diversity, but excessive randomness in the early denoising steps may degrade rollout quality and weaken the reward signal used for policy optimization. Overall, these results support our main claim that effective rollout generation requires controlled early branching—enough to diversify rollouts, but not so much that it disrupts high-quality reasoning trajectories.

To analyze rollout behavior during training, we trace the reward mean–variance trajectories as shown in Figure 3. For each logged training snapshot, we use $G = 8$ rollouts per prompt, compute the reward mean and reward variance across rollouts for each prompt, and then average these statistics over the mini-batch. We then apply a rolling win-



(a) GSM8K *(b)* MATH-500

Figure 3. Reward mean–variance trajectories of Fast-dLLM variants on GSM8K and MATH-500.

dow over 100 snapshots and plot the smoothed trajectories. The trajectories are shown in the mean–variance plane: upward movement indicates higher average rollout reward, while rightward movement indicates larger within-prompt reward variance. As training progresses, the trajectories generally move from the lower-right region to the upper-left region, indicating that reward increases while rollout reward variance decreases.

The original Fast-dLLM baseline achieves a relatively high reward mean but remains in a low-variance region. Since the variance is computed within each prompt's rollout group, this indicates that Fast-dLLM often produces rollouts with similar rewards for the same problem. In other words, although its average rollout quality remains reasonable, each rollout group has limited reward-level diversity, providing only weak reward contrast for the group-relative advantage. Leftmost initialization slightly increases reward variance, but its trajectory remains close to the original Fast-dLLM baseline. This is expected because Leftmost initialization uses a fixed initial position for all rollouts and therefore does not introduce rollout-specific positional branching. In contrast, random initialization substantially increases reward variance by allowing different rollouts to branch from different early positions.

However, higher variance alone is not sufficient. Random 3-step initialization moves toward a higher-variance but lower-mean region, suggesting that excessive early randomness degrades rollout quality and makes the additional diversity less useful for policy optimization. Random 1-step initialization provides a better balance: it increases within-prompt reward variance while preserving a relatively high reward mean. This supports our interpretation that a controlled amount of early branching can improve downstream RL performance.

6. Conclusion

We apply training-free parallel decoding to rollout generation in diffusion LLM policy optimization, reducing rollout cost without auxiliary models or policy modification. We find that confidence-based decoding suffers from delayed branching, and parallel decoding largely inherits this characteristic, leading to a lack of exploration that weakens the group-relative learning signal. We address this with a simple initialization step in which each rollout independently unmasks one uniformly random position, after which the original sampler resumes unchanged. Combined with Fast-dLLM on LLaDA-8B-Instruct, it improves rollout diversity and downstream RL performance on GSM8K and MATH-500, showing that the diversity bottleneck in parallel-decoded rollouts can be resolved by initialization alone.

7. Limitation

Due to limited computational resources, we could not fully reproduce the large-scale RL training budget used in prior SPG experiments. Our experiments use fewer training steps and a smaller effective batch size. Thus, the absolute performance numbers may be affected by the constrained training setup. However, all methods are compared under the same budget, so the relative trends remain informative.

Impact Statement

This work proposes a decoding-time intervention to improve rollout diversity in diffusion language models. Since the method is training-free and does not require new data collection or user-facing deployment, its direct societal impact is limited. However, as it may indirectly support stronger policy optimization, downstream use should follow appropriate safety safeguard

References

Ahmed, Z., Le Roux, N., Norouzi, M., and Schuurmans, D. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pp. 151–160. PMLR, 2019.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021.

Ban, Y., Wang, R., Zhou, T., Gong, B., Hsieh, C.-J., and Cheng, M. The crystal ball hypothesis in diffusion models: Anticipating object positions from initial noise. In *International Conference on Learning Representations*, volume 2025, pp. 34384–34408, 2025.

Bao, W., Chen, Z., Xu, D., and Shang, Y. Learning to parallel: Accelerating diffusion large language models via learnable parallel decoding. *arXiv preprint arXiv:2509.25188*, 2025.

Ben-Hamu, H., Gat, I., Severo, D., Nolte, N., and Karrer, B. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *NeurIPS*, 2025.

Chang, H., Zhang, H., Jiang, L., Liu, C., and Freeman, W. T. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

Chen, Q., Li, H., Qin, L., Peng, D., Liu, J., Wang, J., Wu, C., Chen, X., Du, Y., and Che, W. Beyond surface reasoning: Unveiling the true long chain-of-thought capacity of diffusion large language models. *arXiv preprint arXiv:2510.09544*, 2025a.

Chen, Z., Fang, G., Ma, X., Yu, R., and Wang, X. dparallel: Learnable parallel decoding for dllms. *arXiv preprint arXiv:2509.26488*, 2025b.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Guo, X., Liu, J., Cui, M., Li, J., Yang, H., and Huang, D. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9380–9389, 2024.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Huang, W., Ge, Y., Yang, S., Xiao, Y., Mao, H., Lin, Y., Ye, H., Liu, S., Cheung, K. C., Yin, H., et al. Qerl: Beyond efficiency–quantization-enhanced reinforcement learning for llms. *arXiv preprint arXiv:2510.11696*, 2025.

Israel, D., Broeck, G. V. d., and Grover, A. Accelerating diffusion llms via adaptive parallel decoding. In *NeurIPS*, 2025.

Jeon, D., Kim, D., and No, A. Understanding and mitigating memorization in generative models via sharpness of probability landscapes. *arXiv preprint arXiv:2412.04140*, 2024.

Kim, J., Shah, K., Kontonis, V., Kakade, S. M., and Chen, S. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *ICML*, 2025a.

Kim, M., Kim, Y.-H., and Park, M. Safety-guided flow (sgf): A unified framework for negative guidance in safe generation. *arXiv preprint arXiv:2603.13300*, 2026.

Kim, S. H., Hong, S., Jung, H., Park, Y., and Yun, S.-Y. Klass: Kl-guided fast inference in masked diffusion models. In *NeurIPS*, 2025b.

Kynkäänniemi, T., Aittala, M., Karras, T., Laine, S., Aila, T., and Lehtinen, J. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

Li, S., Le, H., Xu, J., and Salzmann, M. Enhancing compositional text-to-image generation with reliable random seeds. In *International Conference on Learning Representations*, volume 2025, pp. 45444–45465, 2025.

Li, Y., Elangovan, R., Dong, X., Panda, P., and Khailany, B. Qurl: Efficient reinforcement learning with quantized rollout. *arXiv preprint arXiv:2602.13953*, 2026.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *ICLR*, 2023.

Liu, B., Wang, A., Min, Z., Yao, L., Zhang, H., Liu, Y., Zeng, A., and Su, J. Spec-rl: Accelerating on-policy reinforcement learning via speculative rollouts. *arXiv preprint arXiv:2509.23232*, 2025a.

Liu, L., Yao, F., Zhang, D., Dong, C., Shang, J., and Gao, J. Flashrl: 8bit rollouts, full power rl, August 2025b. URL https://fengyao.notion.site/flash-rl.

Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *ICML*, 2024.

Luo, L., Shi, Z., Luo, J., Wang, Z., Ren, S., Wang, W., and Zhang, T. Dawn: Dependency-aware fast inference for diffusion llms. *arXiv preprint arXiv:2602.06953*, 2026.

Ni, Z., Wang, S., Yue, Y., Yu, T., Zhao, W., Hua, Y., Chen, T., Song, J., Yu, C., Zheng, B., et al. The flexibility trap: Why arbitrary order limits reasoning potential in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

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. In *NeurIPS*, 2025.

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. In *ICLR*, 2025.

Sahoo, S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. In *NeurIPS*, 2024.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shen, Y., Feng, T., Han, J., Wang, W., Chen, T., Shen, C., Leskovec, J., and Ermon, S. Improving diffusion language model decoding through joint search in generation order and token space. *arXiv preprint arXiv:2601.20339*, 2026.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. In *NeurIPS*, 2024.

Tang, X., Dolga, R., Yoon, S., and Bogunovic, I. wd1: Weighted policy optimization for reasoning in diffusion language models. *arXiv preprint arXiv:2507.08838*, 2025.

Trott, A., Zheng, S., Xiong, C., and Socher, R. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. *Advances in neural information processing systems*, 32, 2019.

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.

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*, 2025b.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

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*, 2025.

Yao, J., Cheng, R., Wu, X., Wu, J., and Tan, K. C. Diversity-aware policy optimization for large language model reasoning. *arXiv preprint arXiv:2505.23433*, 2025.

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.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

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.

Zheng, L., Yuan, J., Yu, L., and Kong, L. A reparameterized discrete diffusion model for text generation. In *COLM*, 2024.

Zhong, Y., Gu, Y., Zang, Z., Li, X., Ding, Y., Jia, X., Shen, Y., Lan, Z., Zhu, L., Liu, W., et al. Parallelism and generation order in masked diffusion language models: Limits today, potential tomorrow. *arXiv preprint arXiv:2601.15593*, 2026.

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.

A. Details on Diversity Experiments

Diversity metrics. Following the metric definitions of Yao et al. (2025), we evaluate sample diversity and reasoning potential from $K = 16$ sampled responses per question. For confidence-based and parallel decoding methods, we use sampling temperature $T = 1.2$. For AR decoding, we use $T = 0.9$, since AR sampling at $T = 1.2$ substantially degraded output quality. We report three types of diversity metrics. Div-N-gram measures lexical diversity as the percentage of distinct n-grams among all generated n-grams, and we report Div-1, Div-2, and Div-4. Div-Self-BLEU measures inter-sample diversity by computing $100 - \text{SelfBLEU}$ among the K responses for the same question. Div-Equ measures solution-level diversity for mathematical reasoning. Following Yao et al. (2025), let $\mathcal{A}_i$ be the multiset of all equations extracted from the K responses for question q_i, and let $\mathcal{U}_i$ be the set of unique equations. We compute

$$\text{Div-Equ} = 100 \times \frac{1}{N} \sum_{i=1}^{N} \frac{|\mathcal{U}_i|}{|\mathcal{A}_i|}.$$

Higher values indicate greater diversity for all diversity metrics.

Potential@K. Potential@K measures whether sampling can recover a correct answer on problems that greedy decoding fails. Let $\text{Pass@1}(q_i)$ denote the correctness of deterministic greedy decoding for question q_i, and let $\text{Pass@K}(q_i)$ indicate whether at least one of the K sampled responses is correct. Following Yao et al. (2025), we compute

$$\text{Potential@K} = \frac{\sum_{i=1}^{N} \text{Pass@K}(q_i)\,(1 - \text{Pass@1}(q_i))}{\sum_{i=1}^{N} (1 - \text{Pass@1}(q_i))}.$$

Thus, Potential@K measures Pass@K only over questions not solved by deterministic greedy decoding. We use it as a proxy for reasoning potential under repeated sampling, not as a direct measure of final RL performance.

Additional Results. We additionally evaluate AR decoding at temperature $T = 1.2$. As shown in Table 7, AR decoding at $T = 1.2$ yields high diversity scores, but its Potential@K remains substantially lower than the AR result at $T = 0.9$ used in the main experiments. This suggests that the increased diversity mainly comes from noisy or malformed generations rather than useful reasoning alternatives. Therefore, we use AR decoding with $T = 0.9$ in the main experiments.

Table 7. Additional results of AR decoding at temperature $T = 1.2$.

Method	**Potential@K**				**Diversity**				
	@1	**@4**	**@8**	**@16**	**Div-1**	**Div-2**	**Div-4**	**Div-SB**	**Div-Equ**
AR	10.8	34.1	52.4	72.7	39.1	63.8	83.2	59.1	56.6