Efficient and Stable Reinforcement Learning for Diffusion Language Models

Anonymous ACL submission

Abstract

Reinforcement Learning (RL) is crucial for unlocking the complex reasoning capabilities of Diffusion-based Large Language Models (dLLMs). However, applying RL to dLLMs faces unique challenges in efficiency and stability. To address these challenges, we propose Spatio-Temporal Pruning (STP), a framework designed to simultaneously improve the efficiency and stability of RL for dLLMs. STP compresses the redundancy in the generative process through: (1) *spatial pruning*, which constrains the exploration space using static priors; and (2) *temporal pruning*, which bypasses redundant late-stage refinement steps. Our theoretical analysis demonstrates that STP strictly reduces the variance of the log-likelihood estimation, thereby ensuring more stable policy updates. Extensive experiments demonstrate that STP surpasses state-of-the-art baselines in both efficiency and accuracy. Our code is available at `https://anonymous.4open.science/r/STP-757E`.

1 Introduction

Diffusion-based large language models (dLLMs) have emerged as an alternative paradigm for generative modeling (Nie et al., 2025; Ou et al., 2025; Yang et al., 2025). By iteratively refining sequences through a denoising process, dLLMs offer a non-autoregressive alternative that breaks the left-to-right sequential generation constraint inherent to traditional autoregressive (AR) models (Lou et al., 2024). Open-weight dLLMs like LLaDA (Nie et al., 2025) now demonstrate performance competitive with AR models of comparable scale, while closed-source models such as Mercury (Khanna et al., 2025) and Gemini Diffusion (DeepMind., 2025) achieve more than an order-of-magnitude generation speed-up over AR models.

Applying reinforcement learning (RL) to dLLMs is essential for improving their reasoning abilities and ensuring safety (Guo et al., 2025; Wang et al., 2024). However, the integration of RL with diffusion-based language models differs fundamentally from the autoregressive (AR) setting. In AR models, token probabilities are factorized along a single left-to-right generation order, enabling exact log-likelihood computation via the chain rule (Radford et al., 2018). In contrast, diffusion language models generate sequences with bidirectional context, where tokens can be updated in multiple orders. This makes the exact likelihood computation – which marginalizes over a combinatorial set of generation trajectories – intractable and typically approximated via multi-step Monte Carlo sampling (Zhao et al., 2025; Zhu et al., 2025). This reliance on estimation has led to two challenges for RL training:

- **C1. Efficiency.** Likelihood estimation relies on multi-step Monte Carlo sampling over diffusion timesteps, increasing the computational cost of trajectory generation.
- **C2. Stability.** Stochastic estimation over both diffusion time and token states introduces variance into the policy gradient estimator, which destabilizes the RL training process (Zhao et al., 2025; Wang et al., 2025).

As summarized in Table 1, existing RL methods

Table 1: Comparison of RL algorithms for dLLMs. STP (Ours) achieves the best efficiency and accuracy empirically with a unique theoretical guarantee on stability. Symbols ✗, ✓, and ✓✓ denote *limited*, *good*, and *state-of-the-art* performance, respectively.

Method	Efficiency	Stability	Accuracy
Standard GRPO	✗	✓	✓
Diffu-GRPO	✓	✗	✗
UniGRPO	✓	✗	✗
STP (Ours)	✓✓[1]	✓✓[2]	✓✓[3]

[1] **+13.1%** training speedup over the fastest baseline.
[2] Theoretically provable variance reduction (see Section 4).
[3] Up to **81.7%** relative improvement on logic reasoning tasks.

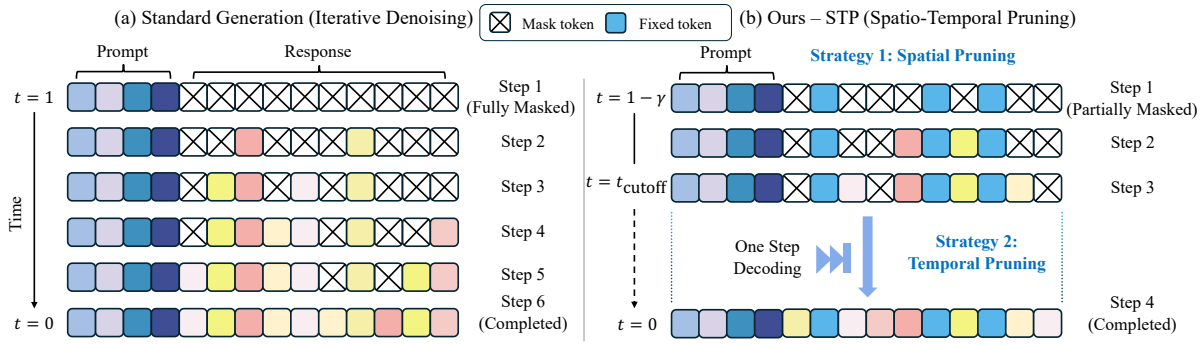


Figure 1: Comparing (a) standard sequence generation with iterative denoising with (b) our generation process with spatio-temporal purning, which reduces computational cost and ELBO variance theoretically while improving empirical accuracy.

for dLLMs fail to simultaneously achieve high computational efficiency and training stability. Standard GRPO is computationally expensive due to multi-step Evidence Lower Bound (ELBO) estimation. To mitigate this burden, efficiency-oriented approaches like Diffu-GRPO (Zhao et al., 2025) and UniGRPO (Yang et al., 2025) employ a one-step denoising strategy to approximate token likelihoods. Although these methods improve efficiency, they introduce estimation variance, which leads to unstable gradient updates and suboptimal accuracy (see experimental results in Table 2).

We propose an alternate method that improves the efficiency by reducing sampling redundancy across *Spatial* (sequence length) and *Temporal* (diffusion steps) dimensions. As shown in Table 1, our method achieves the best efficiency and has a theoretically guaranteed variance reduction, leading to better stability and state-of-the-art accuracy. Moreover, our method is complementary to current RL improvements and can be combined with them to achieve greater improvements (see experimental results in Table 3). More specifically, our framework, **Spatio-Temporal Pruning (STP)**, consists of two pruning strategies as shown in Figure 1: (1) *spatial pruning*, which incorporates the *semi-offline* strategy (Chen et al., 2023) by mixing offline data with online generation, thereby constraining the search space to prevent inefficient exploration; and (2) *temporal pruning*, which leverages the non-autoregressive nature of dLLMs to compress redundant refinement steps in the late stage. This is based on the insight that later steps are primarily dedicated to local syntactic refinement (Huang et al., 2025), allowing for redundancy reduction without compromising generation quality.

Our main contributions are summarized as follows:

- We propose Spatio-Temporal Pruning (STP), which is, to the best of our knowledge, the first work to simultaneously optimize the computational efficiency and training stability of RL training for dLLMs.
- We provide a theoretical analysis demonstrating that STP not only reduces computational cost of trajectory sampling but also lowers the variance of the log-likelihood estimation, thereby enabling more stable and efficient reinforcement learning for dLLMs.
- Extensive experiments demonstrate the effectiveness of STP: it reduces training time by 13.1% vs. the fastest baseline (*Diffu-GRPO*) while consistently outperforming the strongest baseline (*GRPO w/ ELBO*), achieving steady gains on math benchmarks and up to 81.7% relative improvement on logic benchmarks.

2 Preliminaries

2.1 GRPO for dLLMs

Although our framework is suitable for various RL algorithms, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025) as our base RL algorithm for its memory efficiency and its prevalence in RL for dLLMs (Zhao et al., 2025; Yang et al., 2025).

Objective Formulation. GRPO samples a group of G outputs $\{o_1, \ldots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ for each prompt. For each output o_i, a group-relative advantage is computed as $\hat{A}_i = (R(o_i) - \mu_R)/\sigma_R$, where R is the reward function, μ_R and σ_R are the mean and standard deviation of rewards

within the group. The objective maximizes:

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

where $\rho_i^k = \frac{\pi_\theta(o_i^k|c, o_i^{<k})}{\pi_{\theta_{\text{old}}}(o_i^k|c, o_i^{<k})}$ is the importance sampling ratio for the k-th token of output i, ϵ and β are hyper-parameters, and π_{ref} is a fixed reference policy. This objective encourages the policy to increase the probability of tokens that yield high advantages A_i^k, while the clipping mechanism and KL penalty jointly ensure training stability by constraining the policy update.

2.2 Masked Diffusion Large Language Models

Masked dLLMs (Austin et al., 2021; Shi et al., 2024; Nie et al., 2025) generate text $x_0 = (x_0^1, \ldots, x_0^L)$ of length L through an iterative denoising process indexed by continuous time $t \in [0, 1]$. The forward process $q(x_t|x_0)$ independently masks tokens in the clean sequence x_0. Following the standard linear schedule (Nie et al., 2025; Sahoo et al., 2024), the conditional distribution for each token at time t is given by:

$$q(x_t^i|x_0^i) = \begin{cases} 1-t, & x_t^i = x_0^i \\ t, & x_t^i = \texttt{[MASK]} \end{cases}. \tag{2}$$

Thus, t equals to the proportion of masks in the sequence, x_1 is a fully masked sequence, and x_0 is the clean data. The reverse generative process recovers x_0 from x_1 by iteratively sampling x_s from x_t $(s < t)$ using a mask predictor $\pi_\theta(x_0|x_t)$.

Intractability and ELBO. Unlike AR models where likelihood is factorizable, the exact log-likelihood $\log \pi_\theta(x_0)$ of Masked dLLMs requires marginalizing over all possible high-dimensional latent trajectories $x_{1:T}$, which is computationally intractable (Zhao et al., 2025; Wang et al., 2025). Therefore, optimization is performed on the Evidence Lower Bound (ELBO), which serves as a tractable surrogate. For a given sequence x_0, the

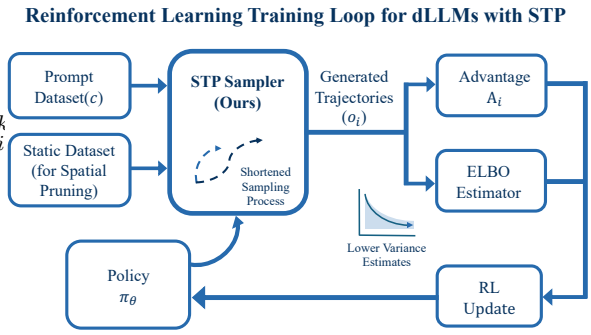


Figure 2: Replacing standard sampling with STP accelerates trajectory generation for exploration. Furthermore, by constraining the sampling space, STP yields lower-variance ELBO estimates, facilitating stable policy updates.

ELBO is defined as:

$$\mathcal{B}_{\pi_\theta}(x_0) \triangleq \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1)\\ x_t\sim q(x_t|x_0)}}\Bigg[\frac{1}{t}\sum_{i=1}^{L}\mathbb{I}(x_t^i = \texttt{[MASK]}) \cdot \log \pi_\theta(x_0^i|x_t)\Bigg]. \tag{3}$$

The token-wise ELBO derived from Eq. (3) is:

$$\mathcal{B}_{\pi_\theta}(x_0^k) \triangleq \mathbb{E}_{\substack{t\sim\mathcal{U}(0,1)\\ x_t\sim q(x_t|x_0)}}\Bigg[\frac{1}{t}\cdot\mathbb{I}(x_t^k = \texttt{[MASK]}) \cdot \log \pi_\theta(x_0^k|x_t)\Bigg]. \tag{4}$$

In practice, we expend considerable computation to mitigate the high variance inherent in its Monte Carlo estimation (Nie et al., 2025).

Applying GRPO to dLLMs For dLLMs, the final clean data x_0 obtained after the denoising process serves as the trajectory o for the subsequent RL training. The intractable token likelihoods in ρ_i^k is approximated using the exponential of the token-wise ELBO in Eq. (4):

$$\rho_i^k(\theta) \approx \exp\left(\mathcal{B}_{\pi_\theta}(o_i^k) - \mathcal{B}_{\pi_{\theta_{\text{old}}}}(o_i^k)\right). \tag{5}$$

As shown in Figure 2, the ELBO estimator enables the application of policy gradient methods to diffusion models. However, this approximation suffers from high variance, which destabilizes RL training. This motivates us to modify the trajectory generation process to reduce variance by STP Sampler, leading to more stable and efficient policy learning.

3 Spatio-Temporal Pruning

The generative process of a dLLM can be formulated as a trajectory within a spatio-temporal lattice $\mathcal{L} \times \mathcal{T}$, where $\mathcal{L} = \{1, \ldots, L\}$ denotes spatial token indices and $\mathcal{T} = [0, 1]$ represents diffusion time. To accelerate this process and reduce variance, we propose **Spatio-Temporal Pruning (STP)**. STP unifies two pruning strategies corresponding to the two dimensions of the lattice: *Spatial Pruning* (reducing the effective action space) and *Temporal Pruning* (reducing the integration steps).

3.1 Spatial Pruning

Spatial pruning constrains the search space to prevent inefficient exploration. This is theoretically motivated by the *Semi-Offline RL* paradigm (Chen et al., 2023), which theoretically demonstrates that mixing offline data with online generation can balance exploration efficiency and generation quality. Compared to AR models, this paradigm is particularly well-suited for dLLMs due to their inherent ability to handle internal masks.

We reformulate the denoising process in dLLMs from a single trajectory into a two-stage procedure, parameterized by the spatial pruning ratio γ . Unlike the conventional schedule where timestep t decreases uniformly from 1 to 0 in steps of Δt, the transition between our two stages is determined by $((1-\gamma)$.

Stage 1 (Spatially-pruned Initialization, $1 \geq t \geq \gamma$):

$$x_t^i = \begin{cases} \texttt{[MASK]}, & \text{if } e_i = 1 \\ x_{\text{fixed}}^i, & \text{if } e_i = 0 \end{cases} \tag{6}$$

where $e_i \sim \text{Bernoulli}(\gamma)$ controls whether exploration is performed for the i-th token, and x_{fixed}^i denotes an anchor sequence from a static dataset, which may come from reference solutions or prior model generations.

Stage 2 (Spatially-pruned Denoising, $\gamma > t \geq 0$):

$$x_t^i = \begin{cases} \hat{x}_t^i, & \text{if } e_i = 1 \text{ and } m_i = 0 \\ \texttt{[MASK]}, & \text{if } e_i = 1 \text{ and } m_i = 1 \\ x_{\text{fixed}}^i, & \text{if } e_i = 0 \end{cases} \tag{7}$$

where $\hat{x}_t^i \sim \pi_\theta(\cdot|x_{t+\Delta t})$, $m_i = \mathbb{I}(i \in S_{maxconf})$, and $S_{maxconf} = argmax(\pi_\theta(\cdot|x_{t+\Delta t}))$ corresponds to the low-confidence decoding strategy (Nie et al., 2025). In the corresponding forward process used by the ELBO estimator, the conditional distribution for each token at time t is defined as:

$$q_{\text{SP}}(x_t^i|x_0^i) = \begin{cases} 1-t, & \text{if } i \notin S_{\text{fixed}} \text{ and } x_t^i = x_0^i \\ t, & \text{if } i \notin S_{\text{fixed}} \text{ and } x_t^i = \texttt{[MASK]} \\ 1, & \text{if } i \in S_{\text{fixed}} \text{ and } x_t^i = x_{\text{fixed}}^i \end{cases}. \tag{8}$$

where $S_{\text{fixed}} = \{i | e_i = 0\}$. Thus $|S_{\text{fixed}}| = \gamma L$.

By fixing these tokens, we reduce the space of the policy π_θ from $\mathcal{V}^L$ to $\mathcal{V}^{L-|S_{\text{fixed}}|}$ and shorten the sequence length for ELBO estimation from L to $L - |S_{\text{fixed}}|$. This accelerates the sampling process and **lowers the variance** of the estimator as derived in Theorem 4.1 and Theorem 4.3.

3.2 Temporal Pruning

While spatial pruning operates on the spatial dimension, temporal pruning operates on the temporal dimension. Standard diffusion is inherently a *fully observable* Markovian process: the transition to state $x_{t-\Delta t}$ requires explicit observation of the immediate predecessor x_t. This necessitates a rigorous, step-by-step diffusion until $t = 0$.

We propose accelerating late-stage denoising via a *partially observable* process (Chen et al., 2023) since the later steps are primarily dedicated to local syntactic refinement (Huang et al., 2025). The intermediate states between a cutoff time t_{cutoff} and 0 are treated as unobserved variables, allowing the model to bypass the iterative chain and directly predict the final outcome.

Specially, we decompose the **Stage 2** in Section 3.1 into two sub-stages, controlled by a temporal cutoff $t_{\text{cutoff}} > 0$.

Stage 2a (Spatially-pruned Denoising, $\gamma > t \geq t_{\text{cutoff}}$):

$$x_t^i = \begin{cases} \hat{x}_t^i, & \text{if } e_i = 1 \text{ and } m_i = 0 \\ \texttt{[MASK]}, & \text{if } e_i = 1 \text{ and } m_i = 1 \\ x_{\text{fixed}}^i, & \text{if } e_i = 0 \end{cases} \tag{9}$$

is equivalent to the undivided Stage 2.

Stage 2b (Temporally-pruned Denoising, $t_{\text{cutoff}} > t \geq 0$):

$$x_t^i = \begin{cases} \tilde{x}_t^i, & \text{if } e_i = 1 \\ x_{\text{fixed}}^i, & \text{if } e_i = 0 \end{cases} \tag{10}$$

where $\tilde{x}_t^i = argmax(\pi_\theta(\cdot|x_{t_{\text{cutoff}}}))$.

This design leverages the inherent advantage of the masked dLLM architecture, which enables the simultaneous decoding of tokens at all positions in

one single forward pass. By pruning the redundant steps $t \in (t_{\text{cut}}, 0]$, we reduce the computational cost of the trajectory sampling from T steps to $T \times (1 - t_{\text{cut}})$ steps.

4 Theoretical Analysis

In this section, we provide a theoretical grounding for Spatio-Temporal Pruning (STP). We analyze its impact on sampling efficiency and the variance of the Evidence Lower Bound (ELBO) estimator. Furthermore, we explicitly connect these properties to the stability of GRPO training.

4.1 Sampling Efficiency

The standard sampling process for a dLLM requires N iterative denoising steps to transition from $t = 1$ to $t = 0$. Let $\mathcal{C}_{\text{step}}$ denote the computational cost of a single forward pass (which is roughly constant).

Theorem 4.1. *(Computational Complexity Reduction) Let N be the total number of diffusion steps, γ be the spatial pruning ratio, and $t_{cutoff} \in (0, 1)$ be the temporal pruning cutoff. The standard sampling cost is $\mathcal{C}_{std} = N \cdot \mathcal{C}_{step}$. Under STP, the sampling cost is reduced to:*

$$\mathcal{C}_{STP} \approx ((1 - \gamma - t_{cutoff}) \cdot N + 1) \cdot \mathcal{C}_{step}. \quad (11)$$

This linear reduction in temporal complexity directly translates to wall-clock speedup, as sampling time is dominated by the number of forward passes.

4.2 Variance Reduction of the ELBO Estimator

A core contribution of STP is the stabilization of the RL training signal. The ELBO estimator $\hat{\mathcal{B}}_{\pi_\theta}(y)$ used in optimization is stochastic due to the sampling of time steps t and the noise in the diffusion process.

Assumption 4.2. (Boundedness of ELBO Estimator) We assume the ELBO estimator difference $\Delta\hat{\mathcal{B}} = \hat{\mathcal{B}}_{\pi_\theta}(y) - \hat{\mathcal{B}}_{\pi_{\text{old}}}(y)$ is bounded. Specifically, there exists a constant $K > 0$ such that $|\Delta\hat{\mathcal{B}}| \leq K$.

This assumption is reasonable because π_{old} and π_θ are separated by only a few training steps, and constraints like clipping and KL regularization explicitly ensure their closeness. Based on this assumption, we analyze the variance reduction of ELBO estimator via STP.

Theorem 4.3. *(Variance Reduction in ELBO) Let $\hat{\mathcal{B}}^{STP}_{\pi_\theta}(y)$ denote the ELBO estimator computed under the STP framework. The variance of the estimator is strictly bounded by the variance of the standard estimator:*

$$\mathbb{V}[\hat{\mathcal{B}}^{STP}_{\pi_\theta}(y)] < \mathbb{V}[\hat{\mathcal{B}}^{standard}_{\pi_\theta}(y)]. \quad (12)$$

This theorem guarantees that STP reduces the variance of ELBO estimation, thereby stabilizing RL training. The proof is provided in Appendix A.1.

4.3 Impact on GRPO Stability

GRPO relies on importance sampling to estimate policy gradients off-policy. In Masked dLLMs, the likelihood ratio is approximated via ELBOs: $\rho_i \approx \exp(\mathcal{B}_{\pi_\theta}(o_i) - \mathcal{B}_{\pi_{\theta_{old}}}(o_i))$. Let $\hat{\mathcal{L}}_{\text{GRPO-E}}$ be the empirical GRPO loss using ELBO estimates. Let $\Delta\hat{\mathcal{B}} = \hat{\mathcal{B}}_{\pi_\theta} - \hat{\mathcal{B}}_{\pi_{\text{old}}}$ be the random variable representing the estimator difference, and $\Delta\mathcal{B}$ be the true ELBO difference.

Theorem 4.4. *(Bias and Variance Reduction in GRPO) Under Assumption 4.2, the bias and variance induced by the ELBO estimation are strictly bounded by the variance of the ELBO difference estimator $\mathbb{V}[\Delta\hat{\mathcal{B}}]$:*

$$\left|\mathbb{E}[\hat{\mathcal{L}}_{GRPO\text{-}E}] - \mathcal{L}_{GRPO\text{-}E}\right| \leq C_1 \cdot \mathbb{E}_{data}[\mathbb{V}[\Delta\hat{\mathcal{B}}]] + C_2 \cdot \mathbb{E}_{data}[\sqrt{\mathbb{V}[\Delta\hat{\mathcal{B}}]}] \quad (13)$$

$$\mathbb{V}[\hat{\mathcal{L}}_{GRPO\text{-}E}] \leq V_{data} + C_3 \cdot \mathbb{E}_{data}[\mathbb{V}[\Delta\hat{\mathcal{B}}]]$$

where C_1, C_2, C_3 are constants depending on the advantage A and the bound K.

The full proof is provided in Appendix A.2. This confirms that reducing the variance of the ELBO estimator ($\mathbb{V}[\Delta\hat{\mathcal{B}}]$) directly tightens the bounds on both bias and variance of the GRPO objective.

High variance in ELBO estimation leads to an overestimation of the importance weights, potentially causing exploding gradients and unstable updates in GRPO. By reducing $\mathbb{V}[\hat{\mathcal{B}}]$ via Theorem 4.3, STP minimizes this bias and the variance of the gradient estimator, thereby enabling more stable and efficient reinforcement training.

5 Experiments

5.1 Experimental Setup

We conduct our experiments based on LLaDA-8B-Instruct (Nie et al., 2025) model. To comprehensively assess reasoning capabilities, we

Table 2: Main results on reasoning benchmarks. We report test accuracy (Acc., %) and total training wall-clock time (Time, s) using the LLaDA-8B-Instruct base model. **Bold** indicates the best performance. † denotes training collapse. The last row indicates the absolute accuracy gain and relative time reduction of STP against the best baseline.

Methods	**MATH**		**GSM8K**		**Countdown**	
	Acc.	**Time (s)**	**Acc.**	**Time (s)**	**Acc.**	**Time (s)**
LLaDA-8B-Instruct	32.80%	-	77.79%	-	16.80%	-
GRPO w/ ELBO	34.20%	126,045	80.52%	163,035	36.33%	164,657
Diffu-GRPO	32.80%	113,358	80.21%	146,641	25.39%	147,100
SPG	34.00%	119,664	76.12%	163,401	0.00%†	166,603
STP (Ours)	**37.20%**	**95,361**	**80.97%**	**144,536**	**66.02%**	**142,055**
Improv. (vs. Best Baseline)	+3.00%	-15.88%	+0.45%	-1.44%	+29.69%	-3.43%

evaluate performance on four challenging benchmarks: GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) for mathematical reasoning, Countdown (Pan et al., 2025) for logical reasoning.

Our proposed framework is benchmarked against three representative RL baselines for dLLMs: (1) **GRPO w/ ELBO**: A direct adaptation of GRPO (Shao et al., 2024) for dLLMs that employs the Evidence Lower Bound (ELBO) estimator as a policy gradient surrogate. (2) **Diffu-GRPO** (Zhao et al., 2025): An efficiency-oriented approach that approximates likelihood via a one-step unmasking strategy to reduce computation. (3) **SPG** (Wang et al., 2025): An advanced RL algorithm utilizing joint upper and lower likelihood bounds to mitigate gradient bias; we implement its token-level variant for a fair comparison.

5.2 Implementation Details

Training Configuration. We conduct all experiments on NVIDIA A800 GPUs (80GB). Following the baselines, we employ Low-Rank Adaptation (LoRA) for parameter-efficient fine-tuning with rank $r = 128$ and scaling factor $\alpha = 64$. We optimize the model using AdamW with a learning rate of 3×10^{-6}. Detailed hyperparameters are listed in Table 8 in Appendix D.

STP Configuration. For our Spatio-Temporal Pruning (STP), we utilize the model's own pre-generations as the source for the static dataset. To prevent reward hacking, tokens enclosed within specific answer delimiters (e.g., `<answer>`... `</answer>`) are excluded from the fixed set. We set the temporal pruning cutoff to $t_{\text{cutoff}} = 0.05$ across all tasks. For the spatial pruning ratio, we set $\gamma = 0.15$ on the MATH dataset and $\gamma = 0.05$ on GSM8K and Countdown based on hyperparameter tuning.

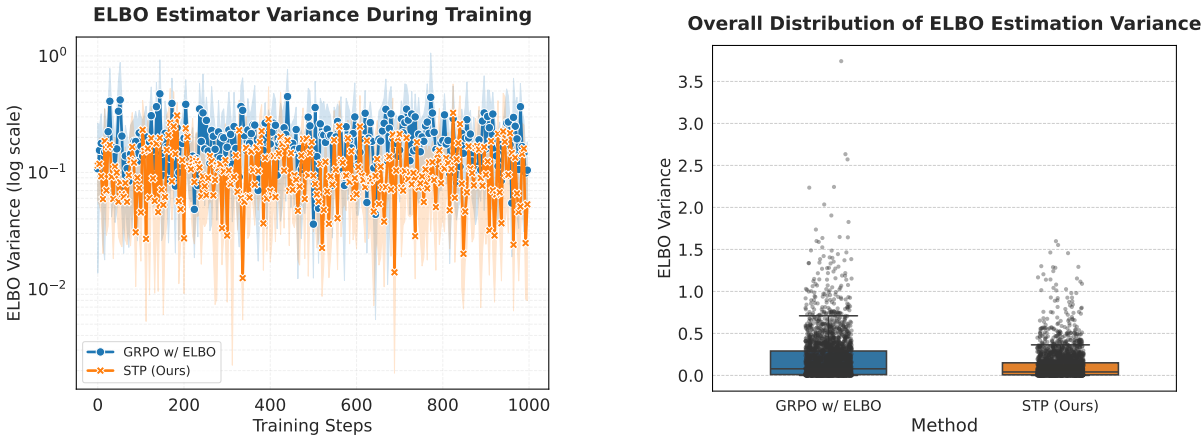
Optimization and Evaluation. Our training objective adopts the reinforcement learning formulation from Diffu-GRPO (Zhao et al., 2025), incorporating KL divergence penalty and gradient clipping for stability. We do not apply prompt masking during training. To ensure consistency, the generation sequence length is fixed to $L = 256$ for both training and evaluation. We report performance on the final checkpoint.

5.3 Main Results

Main evaluation results are summarized in Table 2. The empirical findings demonstrate that STP achieves a favorable trade-off, simultaneous enhancing optimization quality and throughput.

Computational Efficiency. As shown in Table 2, STP consistently minimizes training latency across benchmarks, outperforming even the efficiency-driven Diffu-GRPO baseline. Specifically, STP achieves relative training time reductions of **15.88%** on MATH and **3.43%** on Countdown compared to the fastest baseline, confirming that the spatio-temporal pruning trajectory effectively discards redundant calculations without optimization degradation.

Reasoning Accuracy. In terms of task performance, STP outperforms all evaluated baselines. On the challenging MATH benchmark, STP obtains an accuracy of **37.20%**, yielding a **3.00%** absolute improvement over the strong GRPO baseline. On GSM8K, STP maintains a steady margin, reaching **80.97%**. Notably, on Countdown, STP achieves an accuracy of **66.02%**, whereas SPG



(a) Variance over Training Steps

(b) Variance Distribution

Figure 3: **Empirical Verification of Variance Reduction.** (a) ELBO estimation variance during training dynamics, where STP (Orange) consistently exhibits lower variance than GRPO w/ ELBO (Blue). (b) Aggregate variance distribution demonstrating that STP effectively suppresses median variance and extreme outliers, aligning with Theorem 4.3.

suffers from severe optimization instability and collapses early (0.00%)[1].

5.4 Analysis of ELBO Variance Reduction

Theorem 4.3 and Theorem 4.4 theoretically establish that STP minimizes the variance of the ELBO estimator, thereby stabilizing policy updates. To empirically validate this, we monitor the estimation variance during training on the MATH dataset by recording the ELBO values across 1,000 training steps for sampled trajectories.

As illustrated in Fig. 3a, the standard ELBO estimator exhibits substantial fluctuations throughout training, whereas STP maintains a low, stable variance boundary. This mitigation is vital for importance-sampling methods like GRPO, where extreme likelihood variance often causes erratic gradient steps. Furthermore, the aggregate distribution in Fig. 3b indicates that the variance reduction is statistically significant, suppressing both the median spread and outlying variances. These empirical observations corroborate the stability and performance gains discussed in Sec. 5.3.

5.5 Compatibility with Other RL Algorithms

Because STP operates orthogonally on the trajectory sampling dimension, it can serve as a plug-and-play acceleration framework compatible with other dLLM reinforcement learning objectives. To verify this, we integrate STP with Diffu-GRPO (Zhao et al., 2025) and SPG (Wang et al., 2025) on the MATH dataset.

Table 3: Compatibility of STP with advanced RL algorithms on MATH. Results show absolute accuracy and total training wall-clock time.

Method	Accuracy	Time (s)	Speedup
Diffu-GRPO	32.80%	113,358	-
+ STP	**34.40%**	**79,529**	**+29.8%**
SPG	34.00%	119,664	-
+ STP	**34.60%**	**101,207**	**+15.4%**

As shown in Table 3, STP consistently delivers dual benefits in accuracy and wall-clock acceleration. Incorporating STP into Diffu-GRPO provides a +1.60% accuracy boost alongside a **29.8%** speedup. Similarly, when paired with SPG, STP yields an acceleration of **15.4%** and an absolute accuracy gain of +0.60%. These results show that STP successfully reduces sampling overhead and estimation noise without conflicting with specialized policy gradient formulations.

5.6 STP as Guided Exploration: Leveraging Static Solutions

Beyond efficiency, STP mitigates the "cold start" exploration challenge in challenging tasks by injecting high-quality priors. Unlike standard RL which decodes from scratch, STP utilizes *Spatial Pruning* to fix a subset of tokens based on static data, effectively acting as guided exploration akin to Supervised Fine-Tuning (SFT).

To validate this, we compare STP using self-generated priors versus ground-truth (GT) solutions

[1]This training collapse is a known issue, which has also been reported in Issue #5 of the official SPG repository.

Table 4: Comparison of exploration sources on the MATH dataset. Utilizing model pre-generations yields the best overall performance, validating our variance reduction mechanism.

Method	Source	Acc.	Time (s)
GRPO w/ ELBO	-	34.20%	126,045
STP (Ours)	GT Solutions	34.80%	98,846
STP (Ours)	Model Gens	**37.20%**	**95,361**

Table 5: Ablation of STP components on MATH. The integration of both spatial and temporal pruning yields optimal efficiency and performance.

Method	SP	TP	Acc.	Time (s)
GRPO w/ ELBO	×	×	34.20%	126,045
Spatial Only	✓	×	35.60%	101,200
Temporal Only	×	✓	36.60%	102,190
STP (Ours)	✓	✓	**37.20%**	**95,361**

in Table 4. STP with model generations achieves the highest performance (37.20%), outperforming the baseline by 3.00%, which underscores the potency of our variance-reduction strategy. Anchoring exploration with GT solutions yields a slightly lower accuracy (34.80%) compared to model generations, suggesting that self-generated trajectories provide a more organic and accessible distribution alignment for the model's policy improvement.

5.7 Ablation Study

We isolate the effects of Spatial Pruning (SP) and Temporal Pruning (TP) to evaluate their distinct roles in the trade-off between efficiency and performance. Table 5 presents the results. Applying Spatial Pruning alone improves accuracy over the baseline ($34.20\% \rightarrow 35.60\%$) and reduces training time by approximately 19.7%, confirming that anchoring the generation with static tokens successfully guides exploration. Temporal Pruning alone yields an accuracy of 36.60%, suggesting that the early-exit mechanism effectively filters out the noisy tail of the diffusion process. Crucially, the combined STP framework achieves the best accuracy (37.20%) and the lowest training time (95,361s), showcasing a powerful synergistic effect where SP constrains exploration space and TP avoids late-stage temporal redundancy. Due to page limitations, detailed analyses of token selection strategies and hyperparameter sensitivity ($\gamma, t_{\text{cutoff}}$) are deferred to Appendix B and Appendix C, respectively.

6 Related Work

Diffusion Models for Text Generation. Diffusion models have recently expanded from continuous domains to discrete text generation. Early approaches modeled text in continuous embedding spaces (Li et al., 2022; Gong et al., 2022). Recently, Masked Diffusion Models (MDMs) that operate directly on discrete tokens such as LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025), demonstrate generation capabilities competitive with autoregressive models with superior efficiency.

RL for Diffusion Language Models. While RL algorithms like PPO and GRPO are standard for enhancing autoregressive models (Schulman et al., 2017; Shao et al., 2024), applying them to dLLMs is non-trivial due to the intractability of the exact likelihood. Recent advancements focus on adapting GRPO by approximating likelihoods concentrating on large time comsumption and high variance of the estimation: both DiffuGRPO (Zhao et al., 2025) and UniGRPO (Yang et al., 2025) employ a one-step unmasking approximation to improve the efficiency of likelihoods estimation. To mitigate the high variance of the estimation, methods like wd1 (Tang et al., 2025) and DiffuCoder (Gong et al., 2025) propose weighted objectives or coupled sampling schemes. Unlike these approaches which primarily focus on estimator formulation, our STP constrains the sampling trajectory in both spatial and temporal dimensions, reducing the variance of the estimation fundamentally while simultaneously lowering the computational cost.

7 Conclusion

We propose STP, a principled framework that tackles the critical bottlenecks of efficiency and stability in Reinforcement Learning for dLLMs. By effectively identifying and pruning redundancy across both spatial and temporal dimensions, STP significantly reduces the computational overhead of trajectory sampling. More importantly, we theoretically and empirically demonstrated that STP lowers the variance of the ELBO estimator, providing a high-quality training signal that is essential for stable policy optimization. Our results on reasoning tasks confirm that STP not only accelerates training but also serves as a guided exploration mechanism, enabling the model to achieve state-of-the-art performance. Furthermore, STP is orthogonal to other algorithmic advancements, making it a versatile plug-and-play module for future dLLM research.

Limitations

Although Spatio-Temporal Pruning (STP) substantially enhances training efficiency and stability for diffusion language models, certain limitations remain. The optimal spatial pruning ratio γ exhibits task dependency, requiring manual calibration rather than being dynamically or adaptively scheduled based on problem difficulty.While the orthogonal compatibility with Diffu-GRPO and SPG demonstrates the versatility of STP, our validation is primarily conducted on LLaDA-8B, leaving its generalization capabilities to larger parameter scales as a subject for future work.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.

Changyu Chen, Xiting Wang, Yiqiao Jin, Victor Ye Dong, Li Dong, Jie Cao, Yi Liu, and Rui Yan. 2023. Semi-offline reinforcement learning for optimized text generation. In *International Conference on Machine Learning*, pages 5087–5103. PMLR.

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

Google DeepMind. 2025. Gemini diffusion is our new experimental research model. Technical report, Google DeepMind, https://blog.google/innovation-and-ai/models-and-research/google-deepmind/gemini-diffusion/.

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

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. 2025. Reinforcing the diffusion chain of lateral thought with diffusion language models. *arXiv preprint arXiv:2505.10446*.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, and 1 others. 2025. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 1.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32819–32848.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *The Thirteenth International Conference on Learning Representations*.

Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. 2025. Tinyzero. https://github.com/Jiayi-Pan/TinyZero. Accessed: 2025-01-24.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical

reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. 2024. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167.

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

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

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, and 1 others. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. 2025. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.

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

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

A Proofs of Theoretical Analysis

A.1 Proof of Theorem 4.3

Proof. We analyze the variance of the ELBO estimator. Recall the standard estimator definition:

$$\hat{\mathcal{B}}(y) = \frac{1}{K}\sum_{k=1}^{K} \ell(\theta, y, t_k, \epsilon_k) \tag{14}$$

where $t_k \sim \mathcal{U}[0,1]$ and ℓ is the reweighted loss at step t. By the Law of Total Variance, we can decompose the variance into two terms: variance due to time step sampling ($\mathcal{V}_t$) and variance due to noise sampling conditioned on time ($\mathcal{V}_{noise}$):

$$\mathbb{V}[\hat{\mathcal{B}}] = \mathbb{V}_t[\mathbb{E}_\epsilon[\ell|t]] + \mathbb{E}_t[\mathbb{V}_\epsilon[\ell|t]]. \tag{15}$$

1. Effect of Spatial Pruning: In STP, we fix a set of tokens S_{fixed}. The loss function becomes a sum over only the non-fixed tokens $i \notin S_{fixed}$. Let L be the sequence length and $L' = L - |S_{fixed}|$. Assuming token-wise losses have bounded covariance, the variance of the sum scales with the number of terms. Let $\mathcal{L}_{full} = \sum_{i=1}^{L} \ell_i$ and $\mathcal{L}_{pruned} = \sum_{i \notin S_{fixed}} \ell_i$,

$$\mathbb{V}_\epsilon[\mathcal{L}_{pruned}|t] \approx \frac{L'}{L}\mathbb{V}_\epsilon[\mathcal{L}_{full}|t] < \mathbb{V}_\epsilon[\mathcal{L}_{full}|t]. \tag{16}$$

Thus, spatial pruning directly reduces the second term $\mathbb{E}_t[\mathbb{V}_\epsilon[\ell|t]]$.

2. Effect of Temporal Pruning: Tokens generated via temporal pruning are still treated as "generated" tokens. During the ELBO calculation for the RL update, these tokens follow the standard probabilistic formulation. Therefore, the decrease in variance strictly originates from Spatial Pruning.

Combining both effects, $\mathbb{V}[\hat{\mathcal{B}}^{STP}] < \mathbb{V}[\hat{\mathcal{B}}^{standard}]$. □

A.2 Proof of Theorem 4.4

Lemma A.1. *(Bias and Variance of Exponential Transformation) Let X be a random variable satisfying $|X| \le K$ with mean $\mu = \mathbb{E}[X]$ and variance $\mathbb{V}[X]$. Let $f(x) = e^x$. Then the transformed variable $Y = f(X)$ satisfies:*

$$|\mathbb{E}[f(X)] - f(\mu)| \le \frac{C_{\exp}}{2}\mathbb{V}[X] \tag{17}$$

$$\mathbb{V}[f(X)] \le C_{\exp}^2 \mathbb{V}[X] \tag{18}$$

where $C_{\exp} = e^K$ is the Lipschitz constant of e^x on $[-K, K]$.

Proof. Consider the second-order Taylor expansion of $f(X)$ around μ:

$$f(X) = f(\mu) + f'(\mu)(X-\mu) + \frac{1}{2}f''(\xi)(X-\mu)^2 \tag{19}$$

where ξ lies between X and μ. Taking the expectation on both sides:

$$\begin{aligned}\mathbb{E}[f(X)] = f(\mu) + f'(\mu)\underbrace{\mathbb{E}[X-\mu]}_{0} \\ + \frac{1}{2}\mathbb{E}[f''(\xi)(X-\mu)^2].\end{aligned} \tag{20}$$

Since $f(x) = e^x$, we have $f''(x) = e^x$. Given the boundedness assumption $|X| \leq K$, it implies $|\xi| \leq K$, thus $|f''(\xi)| \leq e^K = C_{\text{exp}}$.

$$|\mathbb{E}[f(X)] - f(\mu)| = \left|\frac{1}{2}\mathbb{E}[f''(\xi)(X-\mu)^2]\right| \leq \frac{C_{\text{exp}}}{2}\mathbb{E}[(X-\mu)^2] = \frac{C_{\text{exp}}}{2}\mathbb{V}[X]. \tag{21}$$

Since $f(x) = e^x$ is continuously differentiable and bounded on $[-K, K]$, it is Lipschitz continuous with constant $L = \sup_{z\in[-K,K]} |f'(z)| = e^K = C_{\text{exp}}$. Using the standard property of Lipschitz functions on variance:

$$\begin{aligned}\mathbb{V}[f(X)] &= \mathbb{E}[(f(X) - \mathbb{E}[f(X)])^2] \\ &\leq \mathbb{E}[(f(X) - f(\mu))^2].\end{aligned} \tag{22}$$

Since f is C_{exp}-Lipschitz:

$$|f(X) - f(\mu)| \leq C_{\text{exp}}|X - \mu|, \tag{23}$$

squaring both sides and taking expectations:

$$\mathbb{E}[(f(X)-f(\mu))^2] \leq C_{\text{exp}}^2\mathbb{E}[(X-\mu)^2] = C_{\text{exp}}^2\mathbb{V}[X]. \tag{24}$$

Thus, $\mathbb{V}[f(X)] \leq C_{\text{exp}}^2\mathbb{V}[X]$. □

Lemma A.2. *(Lipschitz Continuity of GRPO Objective) Let r be the importance ratio and A be the advantage. The per-sample GRPO objective function $g(r; A) = \min(rA, clip(r, 1-\epsilon, 1+\epsilon)A)$ is Lipschitz continuous with respect to r. Specifically, for a fixed advantage A, there exists a constant $C_{clip} = |A|$ such that:*

$$|g(r_1; A) - g(r_2; A)| \leq C_{clip}|r_1 - r_2|. \tag{25}$$

Proof. The function $g(r; A)$ is a composition of linear scaling, clipping, and the minimum operator.

- The function $h_1(r) = rA$ is Lipschitz with constant $|A|$.
- he function $h_2(r) = \text{clip}(r, 1-\epsilon, 1+\epsilon)A$ is Lipschitz with constant 0 (in clipped regions) or $|A|$ (in unclipped regions), hence bounded by $|A|$.
- The minimum of two Lipschitz functions is Lipschitz with a constant equal to the maximum of their Lipschitz constants.

Thus, $g(r; A)$ is $|A|$-Lipschitz continuous w.r.t r. □

Proof. (Theorem 4.4: Bias and Variance in GRPO Estimates)

For a single sample tuple (x, y) with advantage A, let $\hat{r} = e^{\Delta\hat{\mathcal{B}}}$ and $r^* = e^{\Delta\mathcal{B}}$. The loss is $L(\hat{r}) = g(\hat{r}; A)$.

1. Bias Analysis: We aim to bound $|\mathbb{E}[L(\hat{r})] - L(r^*)|$. By triangle inequality:

$$\begin{aligned}|\mathbb{E}[L(\hat{r})] - L(r^*)| \leq &\underbrace{|\mathbb{E}[L(\hat{r})] - L(\mathbb{E}[\hat{r}])|}_{\text{Jensen gap of } L} \\ &+ \underbrace{|L(\mathbb{E}[\hat{r}]) - L(r^*)|}_{\text{Bias from exponential}}.\end{aligned} \tag{26}$$

Using Lemma A.2, the first term is bounded: $|\mathbb{E}[L(\hat{r})] - L(\mathbb{E}[\hat{r}])| \leq C_{\text{clip}}\mathbb{E}[|\hat{r} - \mathbb{E}[\hat{r}]|] \leq C_{\text{clip}}\sqrt{\mathbb{V}[\hat{r}]}$. From Lemma A.1, $\mathbb{V}[\hat{r}] \leq C_{\text{exp}}^2\mathbb{V}[\Delta\hat{\mathcal{B}}]$. So, Term 1 $\leq C_{\text{clip}}C_{\text{exp}}\sqrt{\mathbb{V}[\Delta\hat{\mathcal{B}}]}$.

For the second term, using the Lipschitz property of L: $|L(\mathbb{E}[\hat{r}]) - L(r^*)| \leq C_{\text{clip}}|\mathbb{E}[\hat{r}] - r^*|$. Recall $r^* = e^{\Delta\mathcal{B}} = e^{\mathbb{E}[\Delta\hat{\mathcal{B}}]}$ (since $\Delta\hat{\mathcal{B}}$ is unbiased (Zhu et al., 2025)). This is exactly the bias of the exponential function bounded in Lemma A.1: $|\mathbb{E}[e^{\Delta\hat{\mathcal{B}}}] - e^{\mathbb{E}[\Delta\hat{\mathcal{B}}]}| \leq \frac{C_{\text{exp}}}{2}\mathbb{V}[\Delta\hat{\mathcal{B}}]$. So, Term 2 $\leq \frac{C_{\text{clip}}C_{\text{exp}}}{2}\mathbb{V}[\Delta\hat{\mathcal{B}}]$.

Combining both terms gives the bias bound dependent on $\mathbb{V}[\Delta\hat{\mathcal{B}}]$ and $\sqrt{\mathbb{V}[\Delta\hat{\mathcal{B}}]}$.

2. Variance Analysis: Using the law of total variance:

$$\mathbb{V}[\hat{L}] = \mathbb{V}_{\text{data}}[\mathbb{E}_{\text{est}}[\hat{L}]] + \mathbb{E}_{\text{data}}[\mathbb{V}_{\text{est}}[\hat{L}]]. \tag{27}$$

Focusing on the estimation variance term $\mathbb{V}_{\text{est}}[\hat{L}]$: Since L is C_{clip}-Lipschitz w.r.t r:

$$\mathbb{V}_{\text{est}}[L(\hat{r})] \leq C_{\text{clip}}^2\mathbb{V}_{\text{est}}[\hat{r}]. \tag{28}$$

From Lemma A.1, $\mathbb{V}_{\text{est}}[\hat{r}] \leq C_{\text{exp}}^2\mathbb{V}[\Delta\hat{\mathcal{B}}]$. Thus, the total variance is bounded by the intrinsic data variance plus a term proportional to $\mathbb{V}[\Delta\hat{\mathcal{B}}]$. This confirms that reducing the variance of the ELBO estimator ($\mathbb{V}[\Delta\hat{\mathcal{B}}]$) directly tightens the bounds on both bias and variance of the GRPO objective. □

B Extended Ablation Study

B.1 Spatial Pruning Strategy

We examine the token selection strategy for Spatial Pruning by comparing *Random Selection* against *Low Confidence Retention*. In the latter strategy, we use tokens with high-confidence as fixed tokens. Table 6 shows that Low Confidence Retention significantly outperforms Random Selection (35.60%

vs. 33.00%). Although Random Selection exhibits lower wall-clock time due to arbitrary token fixing, it incurs severe performance degradation. The superiority of our strategy stems from its ability to intelligently allocate the exploration budget: by fixing the high-confidence tokens that the model has already mastered, it forces the model to focus its optimization efforts on steps with higher uncertainty, thereby maximizing learning efficiency.

Table 6: Comparison of Spatial Pruning strategies ($\gamma = 0.15$). Retaining low-confidence tokens for active exploration yields superior accuracy.

Strategy	**Acc.**	**Time (s)**
Random	33.00%	**87,336**
Low Confidence (Ours)	**35.60%**	101,200

C Hyperparameter Sensitivity

We investigate the sensitivity of STP to two key hyperparameters: the *Spatial pruning ratio* (γ) and the *Temporal pruning cutoff* (t_{cutoff}). To systematically identify the optimal configuration, we adopt a two-phase sequential tuning strategy on the MATH dataset, as detailed in Table 7.

In the first phase, we isolate the impact of temporal pruning by disabling spatial pruning ($\gamma = 0.00$) and searching across different values of t_{cutoff} ($t_{\text{cutoff}} \in \{0.00, 0.05, 0.10\}$). As shown in the top panel of Table 7, incorporating temporal truncation significantly reduces wall-clock time compared to the baseline. In terms of reasoning accuracy, a conservative cutoff ($t_{\text{cutoff}} = 0.05$) yields the highest performance (36.60%), outperforming both the baseline (34.20%) and a more aggressive truncation ($t_{\text{cutoff}} = 0.10$, 35.20%). This indicates that an early-exit threshold of 0.05 effectively preserves text quality while mitigating late-stage temporal redundancy.

In the second phase, we fix the optimized temporal cutoff to $t_{\text{cutoff}} = 0.05$ and conduct a fine-grained grid sweep over the spatial pruning ratio γ from 0.00 to 0.25. The bottom panel of Table 7 demonstrates that training time monotonically decreases as more spatial tokens are pinned, validating the efficiency gain brought by spatial structural anchoring. However, accuracy exhibits a non-monotonic trajectory with respect to γ. Pinned token ratios that are either too conservative ($\gamma = 0.05$) or overly aggressive ($\gamma \geq 0.10$) lead to sub-optimal optimization or a lack of sequence diversity, respectively. Remarkably, setting $\gamma = 0.15$ delivers the peak accuracy of 37.20% while further reducing training time to 95,361s. These sequential findings justify our choice of adopting $\gamma = 0.15$ and $t_{\text{cutoff}} = 0.05$ as the default configuration for the MATH benchmark, showcasing that static spatial guidance works synergistically with temporal truncation.

Table 7: Hyperparameter sensitivity analysis on the MATH dataset. Top: Sequential search of the Temporal Pruning Cutoff t_{cutoff} (with fixed $\gamma = 0.00$, where $\gamma = 0, t_{\text{cutoff}} = 0$ corresponds to the standard GRPO w/ ELBO baseline). Bottom: Fine-grained sweep of the Spatial Pruning Parameter γ (under the optimized fixed $t_{\text{cutoff}} = 0.05$).

γ	t_{CUTOFF}	ACC.	TIME (S)
Phase 1: Search of Temporal Cutoff t_{cutoff} (Fixed $\gamma = 0.00$)			
0.00	0.00	34.20%	126,045
0.00	0.05	**36.60%**	102,190
0.00	0.10	35.20%	**102,165**
Phase 2: Sweep of Spatial Ratio γ (Fixed $t_{cutoff} = 0.05$)			
0.00	0.05	36.60%	102,190
0.05	0.05	36.20%	98,537
0.10	0.05	32.80%	96,156
0.15	0.05	**37.20%**	**95,361**
0.20	0.05	34.20%	94,991
0.25	0.05	34.00%	95,290

D Detailed Hyperparameters

Table 8 lists the detailed hyperparameters used in our experiments. We largely align our settings with SPG (Wang et al., 2025), while adopting the KL penalty coefficients from Diffu-GRPO (Zhao et al., 2025).

Table 8: Hyperparameters for Training.

Hyperparameter	**Value**
Base Model	LLaDA-8B-Instruct
LoRA Rank (r)	128
LoRA Alpha (α)	64
Temperature	0.9
Learning Rate	3×10^{-6}
Optimizer	AdamW
β_1	0.9
β_2	0.99
Weight Decay	0.1
Batch Size	6
Gradient Accumulation Steps	2
Inner Updates (μ)	4
Clip Ratio (ϵ)	0.2
KL Coefficient (β)	0.04
Prompt Masking (p_{mask})	0.0
Number of Completions per Prompt	6
Number of Monte Carlo Estimation Samples	3
Sequence Length (L)	256
Static Source	Model Pre-generations