STP: EFFICIENT REINFORCEMENT LEARNING FOR DIFFUSION LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

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 that removes redundancy along both dimensions of the diffusion process. STP employs static priors, anchors reliable positions, and concentrates exploration, while leveraging bidirectional masked generation to bypass redundant late-stage refinement. Theoretically, we show that STP reduces rollout sampling cost and characterize the estimation bias–variance trade-off of spatial pruning and the output perturbation induced by temporal pruning. We further relate the bias and variance of paired ELBO-difference estimates to error and variance bounds for GRPO objective. Extensive experiments demonstrate that STP outperforms the evaluated baselines in accuracy and RL training efficiency. 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 such as LLaDA Nie et al. (2025) now demonstrate performance competitive with AR models of comparable scale, while closed-source systems such as Mercury Khanna et al. (2025) and Gemini Diffusion DeepMind. (2025) report substantial generation speedups 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 language models differs fundamentally from the autoregressive 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.** Rollout generation and likelihood estimation require repeated denoising passes and stochastic evaluation over diffusion time, making each policy update substantially more expensive than in the AR setting.
- **C2. Stability.** Stochastic estimation over both diffusion time and token states introduces variance into the policy gradient estimator, which destabilizes policy updates Zhao et al. (2025); Wang et al. (2025).

Standard GRPO with full ELBO estimation retains the complete multi-step rollout and is consequently expensive. Efficiency-oriented methods such as Diffu-GRPO Zhao et al. (2025) and Uni-GRPO Yang et al. (2025) reduce this cost via one-step denoising to approximate token likelihoods. While efficient, these approximations introduce estimation variance, leading to unstable gradient

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

updates and suboptimal accuracy. Moreover, they do not eliminate redundancy in the rollout procedure. This motivates a complementary approach that changes the sampling trajectory to remove redundant computation from the rollout process, thereby improving efficiency while trading a controlled amount of bias for variance reduction and greater training stability.

We propose **Spatio-Temporal Pruning (STP)**, a framework that views dLLM generation as a trajectory on a lattice of token positions and diffusion times, as described in Section 3. STP combines two pruning operations: (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 improve exploration efficiency; and (2) *temporal pruning*, which exploits the bidirectional masked-dLLM architecture to skip a potentially redundant refinement segment. This is motivated by the observation that later steps are primarily dedicated to local syntactic refinement Huang et al. (2025a), making it possible to reduce redundant computation while controlling the resulting output perturbation.

We formalize the resulting efficiency–stability trade-off theoretically. Our analysis establishes that STP reduces rollout sampling cost, while spatial pruning trades the bias for a lower upper bound on ELBO-estimator variance. We also derive a pathwise output-perturbation bound for temporal pruning and bound the estimation error and variance of paired ELBO differences and their induced error and variance in GRPO objective. These results explain how STP can improve the efficiency and stability of GRPO training when the pruning-induced bias is controlled. We evaluate STP with LLaDA-8B-Instruct on MATH, GSM8K, and Countdown. STP achieves the best accuracy and shortest training time on all three benchmarks, reduces the measured paired ELBO-difference variance by 79.1% on MATH, and is compatible with several existing RL objectives and ELBO estimators.

Our main contributions are summarized as follows:

- We propose **Spatio-Temporal Pruning (STP)**, which combines anchor-conditioned spatial pruning with temporal early completion to reduce redundant rollout computation and improve the efficiency and stability of RL for masked dLLMs.
- We provide a theoretical characterization of STP covering rollout sampling-cost reduction, the variance upper bound and bias–variance trade-off of spatially pruned ELBO estimation, the output perturbation induced by temporal pruning, and bounds on the estimation error and variance of paired ELBO differences and their impact on GRPO objective.
- Experiments on MATH, GSM8K, and Countdown show that STP improves accuracy and training efficiency, reduces paired ELBO-difference variance, and is compatible with several existing RL algorithms and ELBO estimators.

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 standard ELBO is defined as:

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

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

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

In practice, we spend 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 are approximated using the exponential of the token-wise ELBO in Eq. 4:

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

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)**. The overall design is illustrated in Figure 1. 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

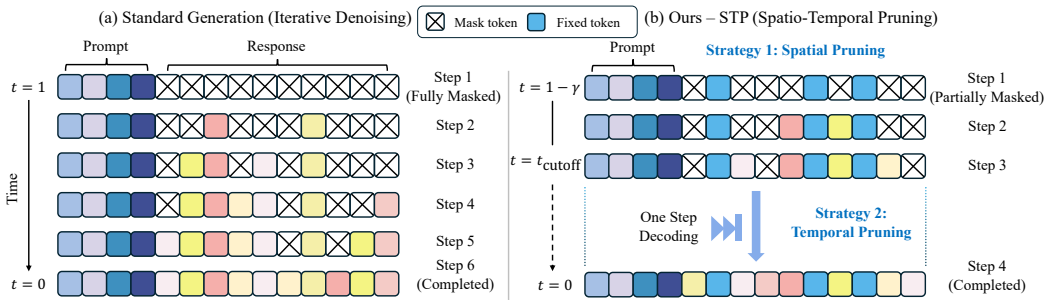


Figure 1: Comparing (a) standard sequence generation with iterative denoising with (b) our generation process with spatio-temporal pruning, which reduces computational cost and the variance upper bound of conditional ELBO estimation while improving empirical accuracy.

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 into a two-stage procedure, parameterized by the spatial pruning ratio γ. Let S_{fixed} be the selected set and $\bar{\gamma} = |S_{\text{fixed}}|/L$ its realized fraction over the full response. The initialization leaves a mask fraction $t_{\text{start}} = 1 - \bar{\gamma}$.

Stage 1 (Spatially-pruned Initialization): Let x_{fixed} be an anchor sequence from a static dataset, such as reference solutions or prior model generations, and let $e_i = \mathbb{I}\{i \notin S_{\text{fixed}}\}$ indicate whether position i is explored.

$$x^i_{t_{\text{start}}} = \begin{cases} \texttt{[MASK]}, & e_i = 1, \\ x^i_{\text{fixed}}, & e_i = 0. \end{cases} \tag{6}$$

For random selection, eligible positions have $e_i \sim \text{Bernoulli}(1-\gamma)$; confidence-based selection fixes high-confidence anchor tokens instead.

Stage 2 (Spatially-pruned Denoising, $t_{\text{start}} > t \geq 0$): Let m_i indicate whether position i has been revealed by the current step, and let $\hat{x}^i_t$ be its assigned token value.

$$x^i_t = \begin{cases} \hat{x}^i_t, & e_i = 1 \text{ and } m_i = 1, \\ \texttt{[MASK]}, & e_i = 1 \text{ and } m_i = 0, \\ x^i_{\text{fixed}}, & e_i = 0. \end{cases} \tag{7}$$

For newly revealed positions, $\hat{x}^i_t$ is sampled from the preceding state, with step size $\Delta t > 0$, using $\pi_\theta(\cdot \mid x_{t+\Delta t})$; previously revealed tokens retain their values. At each step, the scheduled number of highest-confidence candidates among the remaining masks is revealed, following the low-confidence remasking strategy Nie et al. (2025).

In the corresponding forward process used by the ELBO estimator, let $u \in [0,1]$ be the probability of independently masking each non-fixed position, and let x_u denote the resulting corrupted response. The conditional distribution for each token at time t is defined as:

$$q_{\text{SP}}(x^i_u \mid x^i_0) = \begin{cases} 1-u, & i \notin S_{\text{fixed}} \text{ and } x^i_u = x^i_0, \\ u, & i \notin S_{\text{fixed}} \text{ and } x^i_u = \texttt{[MASK]}, \\ 1, & i \in S_{\text{fixed}} \text{ and } x^i_u = x^i_{\text{fixed}}. \end{cases} \tag{8}$$

Spatial pruning reduces the number of free token assignments and the number of token contributions evaluated in the conditional ELBO. This accelerates the sampling process and provides a variance upper-bound reduction of the estimator as derived in Theorem 4.1 and Theorem 4.2.

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), motivated by the possibility of diminishing late-stage changes Huang et al. (2025a). 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. This is an approximation to the full sampling process; its output disagreement is analyzed in Section 4.3.

Specifically, we decompose the **Stage 2** in Section 3 into two sub-stages, controlled by a cutoff $0 \le t_{\text{cutoff}} \le t_{\texttt{start}}$.

Stage 2a (Spatially-pruned Denoising, $t_{\texttt{start}} > t \ge t_{\text{cutoff}}$): The process follows Eq. 7, revealing high-confidence candidates and preserving all previously revealed tokens, until the remaining mask count is $h = \lfloor t_{\text{cutoff}} L \rfloor$.

Stage 2b (Temporally-pruned Completion): Define D_θ as the deterministic parallel-completion operator that fills each masked position with its highest-probability token, preserving all visible tokens. Applied to the cutoff state, it gives $x_0 = D_\theta(x_{t_{\text{cutoff}}})$, with

$$x_0^i = \begin{cases} \arg\max_{v \in \mathcal{V}} \pi_\theta(v \mid x_{t_{\text{cutoff}}}), & x^i_{t_{\text{cutoff}}} = \texttt{[MASK]}, \\ x^i_{t_{\text{cutoff}}}, & \text{otherwise.} \end{cases} \tag{9}$$

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. The steps below t_{cutoff} are replaced by this completion pass, with the combined sampling cost given in Theorem 4.1.

4 Theoretical Analysis

In this section, we provide a theoretical grounding for Spatio-Temporal Pruning (STP). Unless stated otherwise, the prompt and model parameters are fixed. All proofs are in Appendix A.

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:*

$$\mathcal{C}_{STP} \approx ((1 - \gamma - t_{cutoff}) \cdot N + 1) \cdot \mathcal{C}_{step}. \tag{10}$$

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

Spatial pruning reduces the number of stochastic token contributions entering the ELBO estimate. This creates an opportunity to improve estimation accuracy by trading variance savings against the scoring bias induced by conditional evaluation.

Fix a response y matching the anchors and write $S = S_{\text{fixed}}$, $U = \{1, \dots, L\} \setminus S$, and $n = |U|$. Let $\mathcal{B}_{\pi_\theta}(y)$ be the full ELBO and $\mathcal{B}^S_{\pi_\theta}(y)$ its conditional counterpart from Eq. 8. For $n > 0$, draw a mask size m uniformly from $\{1, \dots, n\}$ and a uniform m-element subset $V \subseteq U$. Let z_V be y masked exactly at V. A single conditional-score draw is

$$Z_{\text{P}} = \frac{n}{m} \sum_{i \in V} \log p_\theta(y^i \mid z_V). \tag{11}$$

Using all L positions as eligible positions defines the full draw Z_{F}. These draws are unbiased for their respective ELBO targets, as shown in Appendix A.1. Set $Z_{\mathrm{P}} = 0$ if $n = 0$. Let $\widehat{\mathcal{B}}^{S}_{\pi_\theta}(y)$ and $\widehat{\mathcal{B}}_{\pi_\theta}(y)$ average $M \geq 1$ independent draws of each type.

For this fixed response and model, let $\Lambda_\theta(y)$ be the largest token negative log-probability over all masked versions of y. It is finite for finite softmax logits because there are finitely many positions and masks.

Theorem 4.2 (ELBO Estimation Variance Bound)**.** *For the finite-mask estimators above,*

$$\mathrm{Var}(\widehat{\mathcal{B}}^{S}_{\pi_\theta}(y)) \leq \frac{n^2 \Lambda_\theta(y)^2}{4M}, \qquad \mathrm{Var}(\widehat{\mathcal{B}}_{\pi_\theta}(y)) \leq \frac{L^2 \Lambda_\theta(y)^2}{4M}. \tag{12}$$

The proof is provided in Appendix A.1. Thus spatial pruning reduces the upper bound of the ELBO estimation variance by the factor $(n/L)^2 = (1 - \bar{\gamma})^2$.

MSE trade-off. Define the target bias $b_S(y) = \mathcal{B}^{S}_{\pi_\theta}(y) - \mathcal{B}_{\pi_\theta}(y)$. Then

$$\mathbb{E}\widehat{\mathcal{B}}^{S}_{\pi_\theta}(y) - \mathcal{B}_{\pi_\theta}(y) = b_S(y). \tag{13}$$

Let $v_{\mathrm{F}} = \mathrm{Var}(Z_{\mathrm{F}})$ and $v_{\mathrm{P}} = \mathrm{Var}(Z_{\mathrm{P}})$. With mean squared error (MSE) measured against the same full ELBO,

$$\mathrm{MSE}(\widehat{\mathcal{B}}^{S}_{\pi_\theta}(y)) = b_S(y)^2 + \frac{v_{\mathrm{P}}}{M} \leq b_S(y)^2 + \frac{n^2 \Lambda_\theta(y)^2}{4M}, \tag{14}$$

$$\mathrm{MSE}(\widehat{\mathcal{B}}^{S}_{\pi_\theta}(y)) - \mathrm{MSE}(\widehat{\mathcal{B}}_{\pi_\theta}(y)) = b_S(y)^2 - \frac{v_{\mathrm{F}} - v_{\mathrm{P}}}{M}. \tag{15}$$

The MSE decomposition is derived in Appendix A.2. When $b_S(y)^2 < (v_{\mathrm{F}} - v_{\mathrm{P}})/M$, spatial pruning improves actual ELBO estimation accuracy under the same score-estimation budget.

4.3 Output Error Bound for Temporal Pruning

Write $\tau = t_{\text{cutoff}}$, let $z_\tau = x_{t_{\text{cutoff}}}$ be the cutoff state, and let $\mathcal{M}_\tau = \{i : z^i_\tau = \texttt{[MASK]}\}$ be its mask set, of size $h = |\mathcal{M}_\tau| = \lfloor \tau L \rfloor$. Let $Y_{\text{early}} = D_\theta(z_\tau)$ be the parallel completion and Y_{full} the result of continuing the same anchored prefix to completion. For two length-L responses define normalized Hamming distance by

$$d_H(y, y') = \frac{1}{L} \sum_{i=1}^{L} \mathbb{I}\{y^i \neq y'^i\}. \tag{16}$$

Theorem 4.3 (Temporal Output-Perturbation Bound)**.** *For the decoder in Section 3.2, the paired endpoints satisfy*

$$d_H(Y_{\text{early}}, Y_{\text{full}}) \leq \frac{h}{L} \leq \tau \quad \textit{almost surely}. \tag{17}$$

The proof is provided in Appendix A.3. The disagreement is bounded by the fraction of masked positions at the cutoff while replacing the remaining refinement passes by a single completion pass, establishing an efficiency–output-fidelity trade-off.

4.4 Impact on GRPO Stability

Fix a response y and its group-relative advantage A. Define the full ELBO difference $\Delta\mathcal{B}(y) = \mathcal{B}_{\pi_\theta}(y) - \mathcal{B}_{\pi_{\text{old}}}(y)$ and the paired estimator $\Delta\widehat{\mathcal{B}}(y) = \widehat{\mathcal{B}}_{\pi_\theta}(y) - \widehat{\mathcal{B}}_{\pi_{\text{old}}}(y)$, with current and old scores evaluated using the same corruption randomness.

Assumption 4.4 (Boundedness of ELBO Estimator)**.** For the full and pruned ELBO-difference estimators under consideration, there exists $K > 0$ such that $|\Delta\widehat{\mathcal{B}}(y)| \leq K$ and $|\Delta\mathcal{B}(y)| \leq K$ almost surely.

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.

Let $b_\Delta = \mathbb{E}[\Delta\widehat{\mathcal{B}}(y)] - \Delta\mathcal{B}(y)$ and $v_\Delta = \mathrm{Var}(\Delta\widehat{\mathcal{B}}(y))$, with moments taken over corruption randomness at fixed y, A. Denote the clipped contribution in Eq. 1 by $g(r; A) = \min\{rA, A\,\mathrm{clip}(r, 1-\varepsilon, 1+\varepsilon)\}$. Set $\widehat{L} = g(e^{\Delta\widehat{\mathcal{B}}(y)}; A)$, $L_0 = g(e^{\Delta\mathcal{B}(y)}; A)$, and $C_A = |A|e^K$.

Theorem 4.5 (Bias and Variance Bounds for GRPO)**.** *Under Assumption 4.4, the bias and variance induced by the ELBO estimation are bounded by the variance and bias of the ELBO difference estimator:*

$$|\mathbb{E}\widehat{L} - L_0| \le C_A \left(|b_\Delta| + \sqrt{v_\Delta} + \tfrac{1}{2}v_\Delta\right), \tag{18}$$

$$\mathrm{Var}(\widehat{L}) \le C_A^2 v_\Delta. \tag{19}$$

Appendix A.4 gives the proof. When spatial pruning reduces v_Δ while keeping $|b_\Delta|$ controlled, it tightens the error and variance bounds of this surrogate. Together with the sampling-cost reduction in Theorem 4.1, this characterizes the stability–efficiency benefit of STP for GRPO 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 evaluate performance on three 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. Unless otherwise specified, all experiments are conducted on NVIDIA A800 GPUs (80 GB); a small subset of experiments is conducted on NVIDIA H100 GPUs. 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 9 in Appendix C. **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. Sensitivity analyses are provided in Appendix B.1. **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, with two tokens decoded at each diffusion step. We report performance on the final checkpoint.

5.3 Main Results

The main evaluation results are summarized in Table 1. Overall, STP achieves the best performance across all three reasoning benchmarks while also requiring the shortest RL training time, demonstrating a favorable trade-off between optimization quality and computational efficiency. The model-generated static traces used by STP are constructed once before RL training and are reused throughout the corresponding experiments. Their construction costs are 5,429s, 5,964s, and 6,244s

Table 1: Main results on reasoning benchmarks. We report test accuracy (Acc., %) and RL training wall-clock time (Time, s) using the LLaDA-8B-Instruct base model. **Bold** indicates the best performance. The reported RL training times exclude the one-time offline construction costs of the static traces: 5,429s, 5,964s, and 6,244s for MATH, GSM8K, and Countdown, respectively.

Methods	MATH		GSM8K		Countdown	
	Acc.	Time (s)	Acc.	Time (s)	Acc.	Time (s)
LLaDA-8B-Instruct	32.8%	-	77.8%	-	16.8%	-
GRPO w/ ELBO	34.2%	126,045	80.5%	163,035	36.3%	164,657
Diffu-GRPO	32.8%	113,358	80.2%	146,641	25.4%	147,100
SPG	34.0%	119,664	76.1%	163,401	0.0%	166,603
STP (Ours)	**36.6%**	**95,361**	**81.0%**	**144,536**	**66.0%**	**142,055**

Table 2: Comparison of accuracy, sampling efficiency, and the variance of the paired current–old ELBO difference over the first 2,000 training steps on the MATH benchmark. NFE denotes the Number of Function Evaluations, i.e., the average number of model forward evaluations per rollout batch during denoising generation.

Method	Acc.	NFE	$\mathrm{Var}[\Delta\widehat{\mathcal{B}}(y)]$	Relative variance
GRPO w/ ELBO	34.0%	128.0	1.97×10^{-4}	1.000
STP (Ours)	36.0%	108.3	4.11×10^{-5}	0.209 (-79.1%)

on MATH, GSM8K, and Countdown, respectively. These one-time offline preprocessing costs are excluded from the RL training times reported in Table 1. Including static-trace construction, the end-to-end STP costs are 100,790s, 150,500s, and 148,299s on MATH, GSM8K, and Countdown, respectively. Due to page limitations, detailed analyses of compatibility with other RL algorithms, token selection strategies and ablation study, are deferred to Appendix B.2, Appendix B.3 and Appendix B.4, respectively.

Computational Efficiency. As shown in Table 1, STP consistently minimizes RL training latency across benchmarks, outperforming even the efficiency-driven Diffu-GRPO baseline. Compared with the fastest baseline, STP reduces the RL training time by **15.9%** on MATH, **1.4%** on GSM8K, and **3.4%** on Countdown. These results indicate that the spatio-temporal pruning trajectory effectively reduces online rollout computation without degrading optimization performance.

Reasoning Accuracy. STP achieves the highest accuracy on all three benchmarks. On the challenging MATH benchmark, STP obtains an accuracy of 36.6%, corresponding to a **2.4%** absolute gain over the strongest baseline. On GSM8K, STP reaches 81.0%, yielding a **+0.5%** absolute gain over the best baseline. The largest improvement is observed on Countdown, where STP achieves 66.0% accuracy, corresponding to a **+29.7%** absolute gain over GRPO w/ ELBO. In contrast, SPG collapses during training on this benchmark and obtains 0.0% accuracy,[1] highlighting the improved optimization stability of STP.

5.4 Analysis of ELBO Variance Reduction

To examine estimation noise during training, we measure the variance of the paired current–old ELBO difference, $\mathrm{Var}[\Delta\widehat{\mathcal{B}}(y)]$, over the first 2,000 training steps on MATH. As shown in Table 2, STP reduces the measured variance from 1.97×10^{-4} to 4.11×10^{-5}, corresponding to a 79.1% reduction. Meanwhile, the average number of function evaluations (NFE) per rollout batch decreases from 128.0 to 108.3 (15.4%), while accuracy improves from 34.0% to 36.0%. These results demonstrate lower ELBO-difference estimation variance alongside fewer generation-time forward evaluations and higher accuracy in this experiment. The variance reduction directly concerns the

[1] SPG collapsed in all three repeated runs on the Countdown benchmark, consistent with the training instability reported in Issue #5 of the official SPG repository.

Table 3: Comparison of anchored rollout variants on the MATH dataset. The performance gap demonstrates the additional benefit of spatial pruning beyond anchor-guided exploration.

Method	Acc.
GRPO w/ ELBO	34.2%
Anchor-full	34.4%
STP (Ours)	**36.6%**

Table 4: Comparison of exploration sources on the MATH dataset. STP with model-generated solutions achieves the best performance compared with ground-truth solutions.

Method	Source	Acc.	Time (s)
GRPO w/ ELBO	-	34.2%	126,045
STP (Ours)	GT Solutions	34.8%	98,846
STP (Ours)	Model Gens	**36.6%**	**95,361**

v_Δ term in Theorem 4.5, providing empirical support for a lower-noise GRPO training. Complementary evidence from the training-dynamics analysis in Appendix B.5 shows that STP generally maintains a lower single-policy ELBO-estimation variance than GRPO w/ ELBO throughout the recorded training period.

5.5 STP as Guided Exploration: Leveraging Static Solutions

STP uses static solutions to guide rollout generation and applies spatial pruning to update only a subset of anchor-conditioned tokens. To separate the effects of anchor guidance and spatial pruning, we compare STP with an *Anchor-full* variant that uses the same model-generated anchors but performs full-sequence remasking and loss computation. As shown in Table 3, Anchor-full achieves 34.4% accuracy, only 0.2 percentage points above the GRPO w/ ELBO baseline, whereas STP reaches 36.6%. The resulting **2.2%** absolute gain shows that the spatial-pruning update rule provides a distinct contribution beyond anchor-guided exploration alone, consistent with the bound of Eq. 19. Table 4 further shows that model-generated anchors outperform ground-truth solutions when used with STP (36.6% vs. 34.8%), indicating better alignment with the policy being optimized.

6 Related Work

Diffusion language models. Text diffusion models have evolved from continuous representations Li et al. (2022); Gong et al. (2022) to discrete and masked formulations Austin et al. (2021); Shi et al. (2024); Sahoo et al. (2024). Recent masked diffusion language models, including LLaDA Nie et al. (2025) and Dream Ye et al. (2025), achieve competitive language-modeling and reasoning performance while supporting bidirectional generation and inpainting. **Reinforcement learning for diffusion language models.** RL for dLLMs is challenging because sequence likelihoods require marginalizing over denoising trajectories. Existing methods address this through mean-field or one-step approximations (Diffu-GRPO Zhao et al. (2025) and UniGRPO Yang et al. (2025)), coupled masking or sampling strategies Gong et al. (2025b), preference optimization Zhu et al. (2025), weighted objectives Tang et al. (2025), likelihood bounds Wang et al. (2025), and semi-deterministic ELBO estimation in GDPO Rojas et al. (2026). STP is compatible with these approaches by changing the rollout and conditional-scoring trajectory. More detailed related works are discussed in Appendix D.

7 Conclusion

We propose Spatio-Temporal Pruning (STP) for efficient and stable reinforcement learning with masked diffusion language models. STP combines static-token anchoring with early parallel completion to reduce rollout computation. Our analysis establishes a lower variance upper bound for the conditional ELBO estimator, characterizes its bias–variance trade-off relative to the full ELBO, bounds temporal output perturbation, and relates paired ELBO-difference estimation errors to GRPO surrogate stability. Experiments on reasoning tasks show improved accuracy and shorter RL training time than the evaluated baselines. Furthermore, STP is orthogonal to other algorithmic advancements, suggesting that it can serve as a modular component for RL training with masked dLLMs.

AI USE STATEMENT

In this work, we used generative AI tools to improve the grammar, fluency, clarity, and idiomaticity of the manuscript; to shorten and reorganize passages to meet the page limit; to inspect mathematical proofs for possible gaps, logical inconsistencies, and notation mismatches; and to suggest revisions to proof steps. We also used generative AI tools to assist with modifications and debugging of experimental code and to organize and present experimental results. The tools were used as supplementary aids and were not used as a substitute for the authors' research reasoning or judgment.

We reviewed all AI-assisted text, proof-related suggestions, code modifications, and organization of the experimental results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies an algorithmic method for improving the efficiency and stability of reinforcement learning for diffusion-based language models. Our experiments use publicly available mathematical and logical reasoning benchmarks, namely MATH, GSM8K, and Countdown, and do not involve human subjects, interaction with users, or the collection or release of new personal or sensitive data. The work does not develop or evaluate a system for making decisions about individuals, nor does it target applications involving surveillance, persuasion, or other high-risk domains. Although improvements in language-model capabilities may have broader downstream implications, the method proposed here is evaluated only as a computational training technique on established reasoning benchmarks and is not presented as a safety-critical or autonomous decision-making system. We followed standard research practices, report the experimental methodology and limitations of the method, and provide code to support reproducibility. To the best of our knowledge, this work does not raise additional concerns regarding privacy, discrimination, human-subjects research, or legal compliance, and the authors have no conflicts of interest or external sponsorship to declare.

REPRODUCIBILITY STATEMENT

The proposed Spatio-Temporal Pruning (STP) algorithm and its implementation details are described in Section 3 and Section 5, while the assumptions and complete proofs of the theoretical results are provided in Section 4 and Appendix A. The experimental setup, benchmark tasks, model configuration, training procedure, and evaluation protocol are described in Section 5; extended diagnostics, ablations, token selection strategies, and hyperparameter sensitivity analyses are reported in Appendix B. The complete training hyperparameters are listed in Appendix C, including the optimizer, learning rate, LoRA configuration, batch size, and pruning parameters. We evaluate on the publicly available MATH, GSM8K, and Countdown benchmarks. The implementation and scripts used for training and evaluation are available at the anonymous repository `https://anonymous.4open.science/r/STP-757E`.

REFERENCES

Marianne Arriola, Aaron Gokaslan, Justin Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, volume 2025, pp. 50726–50753, 2025.

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

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

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

Google DeepMind. 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/, 2025.

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

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, volume 2025, pp. 5046–5073, 2025a.

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

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

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

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

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling. *arXiv preprint arXiv:2507.01679*, 2025b.

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

Vishnu Teja Kunde, Fatemeh Doudi, Mahdi Farahbakhsh, Dileep Kalathil, Krishna Narayanan, and Jean-Francois Chamberland. Reinforcement learning for diffusion llms with entropy-guided step selection and stepwise advantages. *arXiv preprint arXiv:2603.12554*, 2026.

Wonseok Lee, Jimyeong Kim, Jungmin Ko, and Wonjong Rhee. Beyond on-policy exploration: Integrating external policy rollouts for reinforcement learning in diffusion language models. *arXiv preprint arXiv:2608.01717*, 2026.

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

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

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

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

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

Haran Raajesh, Kulin Shah, Adam Klivans, and Philipp Krähenbühl. Mask-aware policy gradients for diffusion language models. *arXiv preprint arXiv:2607.15200*, 2026.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Kevin Rojas, Jiahe Lin, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, Molei Tao, and Wei Deng. Improving reasoning for diffusion language models via group diffusion policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 1938–1973, 2026.

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

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

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

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

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

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

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

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

Siyan Zhao, Mengchen Liu, Jing Huang, Miao Liu, Chenyu Wang, Bo Liu, Yuandong Tian, Guan Pang, Sean Bell, Aditya Grover, et al. Inpainting-guided policy optimization for diffusion large language models. In *International Conference on Learning Representations*, volume 2026, pp. 6017–6048, 2026.

Jianyuan Zhong, Kaibo Wang, Ding Ding, Zijin Feng, Haoli Bai, Yang Xiang, Jiacheng Sun, and Qiang Xu. Stabilizing reinforcement learning for diffusion language models. *arXiv preprint arXiv:2603.06743*, 2026.

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

A Proofs of Theoretical Analysis

A.1 Proof of Theorem 4.2

The estimator and its target. For a fixed mask $V \subseteq U$ of size $m > 0$, independent corruption with probability u gives probability $u^m(1-u)^{n-m}$. Its log-score sum therefore has coefficient

$$\int_0^1 u^{m-1}(1-u)^{n-m}\,du = \frac{1}{m\binom{n}{m}} = \frac{1}{n}\binom{n}{m}^{-1}\frac{n}{m} \tag{20}$$

in the conditional ELBO. The last expression is its coefficient under uniform mask-size sampling with weight n/m in Eq. 11, proving $\mathbb{E}Z_{\mathrm{P}} = \mathcal{B}^S_{\pi_\theta}(y)$. The same argument with all L positions eligible proves $\mathbb{E}Z_{\mathrm{F}} = \mathcal{B}_{\pi_\theta}(y)$.

Proof of Theorem 4.2. For each subset $W \subseteq \{1,\dots,L\}$, let z_W be the response y masked at W. The shared score bound is

$$\Lambda_\theta(y) = \max_{1\le i\le L,\ W\subseteq\{1,\dots,L\}} \{-\log p_\theta(y^i \mid z_W)\}. \tag{21}$$

Finite softmax logits give strictly positive target-token probabilities. For fixed y and parameters the maximum is over a finite set, so $\Lambda_\theta(y) < \infty$. Each summand in Eq. 11 lies in $[-\Lambda_\theta(y), 0]$, and there are m terms with common weight n/m. Hence $-n\Lambda_\theta(y) \le Z_{\mathrm{P}} \le 0$, irrespective of token correlations or how the anchor context changes the scores.

For any random variable $X \in [a,b]$, centering at the midpoint gives $\mathrm{Var}(X) \le \mathbb{E}[(X-(a+b)/2)^2] \le (b-a)^2/4$. Applying this to Z_{P} yields $v_{\mathrm{P}} \le n^2\Lambda_\theta(y)^2/4$. The variance of the average of M independent draws is v_{P}/M. Replacing n by L proves the full-estimator bound. Using the same $\Lambda_\theta(y)$ and M, the two upper bounds differ by the factor $(n/L)^2$. □

A.2 MSE Trade-off for Spatial Pruning

Throughout this subsection, fix the response y, the model parameters, and the anchor set S. All expectations and variances are taken over the mask-sampling randomness.

Bias–variance decomposition. Let $Z_{\mathrm{P}}^{(1)},\dots,Z_{\mathrm{P}}^{(M)}$ be independent draws of the conditional-score estimator in Eq. 11, and let $Z_{\mathrm{F}}^{(1)},\dots,Z_{\mathrm{F}}^{(M)}$ be independent draws of its full-score counterpart. The corresponding sample averages are

$$\widehat{\mathcal{B}}^S_{\pi_\theta}(y) = \frac{1}{M}\sum_{j=1}^M Z_{\mathrm{P}}^{(j)}, \qquad \widehat{\mathcal{B}}_{\pi_\theta}(y) = \frac{1}{M}\sum_{j=1}^M Z_{\mathrm{F}}^{(j)}.$$

Appendix A.1 establishes that these estimators are unbiased for their respective targets:

$$\mathbb{E}\widehat{\mathcal{B}}^S_{\pi_\theta}(y) = \mathcal{B}^S_{\pi_\theta}(y), \qquad \mathbb{E}\widehat{\mathcal{B}}_{\pi_\theta}(y) = \mathcal{B}_{\pi_\theta}(y).$$

Consequently, the conditional estimator has bias $b_S(y) = \mathcal{B}^S_{\pi_\theta}(y) - \mathcal{B}_{\pi_\theta}(y)$ relative to the full ELBO, proving Eq. 13. Its estimation error can be decomposed as

$$\widehat{\mathcal{B}}^S_{\pi_\theta}(y) - \mathcal{B}_{\pi_\theta}(y) = \big(\widehat{\mathcal{B}}^S_{\pi_\theta}(y) - \mathcal{B}^S_{\pi_\theta}(y)\big) + b_S(y).$$

Squaring this identity and taking expectations gives

$$\begin{aligned}\mathrm{MSE}(\widehat{\mathcal{B}}^S_{\pi_\theta}(y)) &:= \mathbb{E}\big[(\widehat{\mathcal{B}}^S_{\pi_\theta}(y) - \mathcal{B}_{\pi_\theta}(y))^2\big] \\ &= \mathbb{E}\big[(\widehat{\mathcal{B}}^S_{\pi_\theta}(y) - \mathcal{B}^S_{\pi_\theta}(y))^2\big] + 2b_S(y)\mathbb{E}\big[\widehat{\mathcal{B}}^S_{\pi_\theta}(y) - \mathcal{B}^S_{\pi_\theta}(y)\big] + b_S(y)^2 \\ &= \mathrm{Var}(\widehat{\mathcal{B}}^S_{\pi_\theta}(y)) + b_S(y)^2.\end{aligned}$$

The cross term vanishes because the centered conditional estimator has zero expectation; $b_S(y)$ is deterministic under the fixed response, parameters, and anchors. Writing $v_{\mathrm{P}} = \mathrm{Var}(Z_{\mathrm{P}})$, independence across draws yields

$$\mathrm{Var}(\widehat{\mathcal{B}}^S_{\pi_\theta}(y)) = \frac{1}{M^2}\sum_{j=1}^M \mathrm{Var}(Z_{\mathrm{P}}^{(j)}) = \frac{v_{\mathrm{P}}}{M}.$$

Substituting this identity and the variance bound from Theorem 4.2 gives

$$\mathrm{MSE}(\widehat{\mathcal{B}}^S_{\pi_\theta}(y)) = b_S(y)^2 + \frac{v_{\mathrm{P}}}{M} \le b_S(y)^2 + \frac{n^2 \Lambda_\theta(y)^2}{4M}.$$

Comparison with the full estimator. Since the full estimator is unbiased for the common reference, its MSE equals v_{F}/M, where $v_{\mathrm{F}} = \mathrm{Var}(Z_{\mathrm{F}})$. Subtracting it from the preceding identity yields

$$\mathrm{MSE}(\widehat{\mathcal{B}}^S_{\pi_\theta}(y)) - \mathrm{MSE}(\widehat{\mathcal{B}}_{\pi_\theta}(y)) = b_S(y)^2 - \frac{v_{\mathrm{F}} - v_{\mathrm{P}}}{M},$$

proving Eq. 15.

A.3 Proof of Theorem 4.3

Proof. Fix the cutoff state z_τ and its remaining mask set $\mathcal{M}_\tau$, with $|\mathcal{M}_\tau| = h$. Both continued decoding in Eq. 7 and early completion in Eq. 9 preserve every revealed token. Hence $Y^i_{\mathrm{early}} = Y^i_{\mathrm{full}} = z^i_\tau$ for all $i \notin \mathcal{M}_\tau$, so

$$d_H(Y_{\mathrm{early}}, Y_{\mathrm{full}}) = \frac{1}{L} \sum_{i \in \mathcal{M}_\tau} \mathbb{I}\{Y^i_{\mathrm{early}} \neq Y^i_{\mathrm{full}}\} \le \frac{h}{L} \le \tau,$$

where the last inequality uses $h = \lfloor \tau L \rfloor$. The bound holds for every continuation, establishing the almost-sure claim. □

Adding temporal pruning to spatial pruning. For fixed anchors, couple an unpruned response Y_0 with a spatial-only response Y_S of the same length and prompt. Write $\delta^{\mathrm{out}}_S = \mathbb{E} d_H(Y_0, Y_S)$ for their expected output discrepancy under this coupling. Extend the coupling to a spatio-temporally pruned response $Y_{S,\tau}$ so that $(Y_S, Y_{S,\tau})$ has the shared-prefix coupling in Theorem 4.3. Such an extension is obtained by sampling the latter pair's conditional law given Y_S. Let $\epsilon^{\mathrm{out}}_\tau = \mathbb{E}_{z_\tau} \min\{h/L, T_\tau(z_\tau)\}$ be the averaged temporal bound. The triangle inequality gives

$$\mathbb{E} d_H(Y_0, Y_{S,\tau}) \le \min\{1, \delta^{\mathrm{out}}_S + \epsilon^{\mathrm{out}}_\tau\} \le \min\{1, \delta^{\mathrm{out}}_S + \tau\}. \tag{22}$$

Thus adding the temporal cutoff costs at most τ in normalized output disagreement beyond a separately assessed spatial discrepancy.

A.4 Proof of Theorem 4.5

Lemma A.1 (Bias and Variance of Exponential Transformation)**.** *For a random variable X with $|X| \le K$, let $m = \mathbb{E}X$ and $v = \mathrm{Var}(X)$. Then*

$$|\mathbb{E}e^X - e^m| \le \tfrac{1}{2} e^K v, \qquad \mathrm{Var}(e^X) \le e^{2K} v. \tag{23}$$

Proof. Taylor expansion around m has a zero-mean linear term and a remainder bounded by $e^K(X-m)^2/2$, proving the first inequality. The exponential is e^K-Lipschitz on $[-K, K]$, so $\mathrm{Var}(e^X) \le \mathbb{E}(e^X - e^m)^2 \le e^{2K} v$. □

Lemma A.2 (Lipschitz Continuity of the GRPO Objective)**.** *For fixed A, $g(r; A) = \min\{rA, A\,\mathrm{clip}(r, 1-\varepsilon, 1+\varepsilon)\}$ is $|A|$-Lipschitz in $r \ge 0$.*

Proof. The function is continuous and piecewise linear, with slopes in $\{0, A\}$, for either sign of A. □

Proof of Theorem 4.5. We condition throughout on the fixed response y and advantage A; all expectations and variances below are therefore over the ELBO corruption randomness. Set

$$X = \Delta\widehat{\mathcal{B}}(y), \qquad d = \Delta\mathcal{B}(y), \qquad m = \mathbb{E}X, \qquad \widehat{r} = e^X.$$

By definition, $b_\Delta = m - d$ and $v_\Delta = \mathrm{Var}(X)$. Assumption 4.4 gives $|X| \le K$ almost surely. Since the full estimator is unbiased, $|d| \le K$ as well.

Bias bound. The reference contribution is $L_0 = g(e^d; A)$ and the estimated contribution is $\widehat{L} = g(\widehat{r}; A)$. Insert the intermediate value $g(\mathbb{E}\widehat{r}; A)$ and apply the triangle inequality:

$$|\mathbb{E}\widehat{L} - L_0| \le |\mathbb{E}g(\widehat{r}; A) - g(\mathbb{E}\widehat{r}; A)| \tag{24}$$
$$+ \left|g(\mathbb{E}\widehat{r}; A) - g(e^d; A)\right|. \tag{25}$$

We bound the two terms separately. By Lemma A.2, $g(\cdot; A)$ is $|A|$-Lipschitz. Hence

$$|\mathbb{E}g(\widehat{r}; A) - g(\mathbb{E}\widehat{r}; A)| \le |A|\,\mathbb{E}|\widehat{r} - \mathbb{E}\widehat{r}| \le |A|\sqrt{\mathrm{Var}(\widehat{r})}.$$

Lemma A.1, applied to X, gives

$$\mathrm{Var}(\widehat{r}) = \mathrm{Var}(e^X) \le e^{2K} v_\Delta.$$

Therefore the first term in Eq. 25 is bounded by

$$|A| e^K \sqrt{v_\Delta}.$$

For the second term, use Lemma A.2 once more and then insert e^m:

$$\begin{aligned}\left|g(\mathbb{E}\widehat{r}; A) - g(e^d; A)\right| &\le |A|\left|\mathbb{E}e^X - e^d\right| \\ &\le |A|\left(|\mathbb{E}e^X - e^m| + |e^m - e^d|\right).\end{aligned}$$

The first term is bounded by the Taylor remainder in Lemma A.1:

$$|\mathbb{E}e^X - e^m| \le \tfrac{1}{2} e^K v_\Delta.$$

For the second term, the mean value theorem and $m, d \in [-K, K]$ give

$$|e^m - e^d| \le e^K |m - d| = e^K |b_\Delta|.$$

Thus the second term in Eq. 25 is at most

$$|A| e^K \left(|b_\Delta| + \tfrac{1}{2} v_\Delta\right).$$

Adding the two terms and using $C_A = |A| e^K$ proves

$$|\mathbb{E}\widehat{L} - L_0| \le C_A \left(|b_\Delta| + \sqrt{v_\Delta} + \tfrac{1}{2} v_\Delta\right),$$

which is Eq. 18.

Variance bound. Define $F(x) = g(e^x; A)$. The exponential is e^K-Lipschitz on $[-K, K]$, and $g(\cdot; A)$ is $|A|$-Lipschitz. Consequently,

$$|F(x) - F(x')| \le |A| e^K |x - x'| = C_A |x - x'|$$

for all $x, x' \in [-K, K]$. Let X' be an independent copy of X. The identity

$$\mathrm{Var}(F(X)) = \frac{1}{2}\mathbb{E}\left[(F(X) - F(X'))^2\right]$$

then implies

$$\begin{aligned}\mathrm{Var}(\widehat{L}) &= \mathrm{Var}(F(X)) \\ &\le \frac{1}{2} C_A^2 \mathbb{E}[(X - X')^2] \\ &= C_A^2\,\mathrm{Var}(X) \\ &= C_A^2 v_\Delta.\end{aligned}$$

This proves Eq. 19. □

B Extended Experiments

B.1 Hyperparameter Sensitivity

We investigate the sensitivity of STP to two key hyperparameters: the *Spatial pruning ratio* (γ) and the *Temporal pruning cutoff* (t_{cutoff}). We conduct a two-phase sequential search on the MATH dataset and report accuracy, denoising NFE, and wall-clock training time in Table 5. NFE measures the average number of model forward evaluations per rollout batch during denoising generation.

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 5, the baseline ($t_{\text{cutoff}} = 0.00$) requires 128.0 NFE and takes 126,045 s at 34.2% accuracy. Setting $t_{\text{cutoff}} = 0.05$ reduces NFE to 123.0 and time to 102,190 s, while increasing accuracy to 35.8%. The more aggressive cutoff $t_{\text{cutoff}} = 0.10$ further reduces NFE to 117.0 and is marginally faster at 102,165 s, but lowers accuracy to 35.2%. Thus, temporal pruning produces a clear reduction in denoising computation, whereas $t_{\text{cutoff}} = 0.05$ provides the best accuracy–efficiency trade-off; increasing the cutoff to 0.10 saves a further 6.0 NFE but only 25 s of training time, with a 0.6 percentage-point decrease in accuracy.

In the second phase, we fix $t_{\text{cutoff}} = 0.05$ and sweep the spatial pruning ratio γ from 0.00 to 0.25. NFE decreases monotonically from 123.0 at $\gamma = 0.00$ to 98.4 at $\gamma = 0.25$, a 20.0% reduction, showing a consistent reduction in denoising evaluations as more anchor positions are fixed. Wall-clock time follows the same overall trend through $\gamma = 0.20$, falling from 102,190 s to a minimum of 94,991 s, but increases slightly to 95,290 s at $\gamma = 0.25$. Accuracy remains non-monotonic: it improves from 35.8% at $\gamma = 0.00$ to 36.2% at $\gamma = 0.05$, drops to 32.8% at $\gamma = 0.10$, recovers to the peak of 36.6% at $\gamma = 0.15$, and then declines to 34.2% and 34.0% at $\gamma = 0.20$ and 0.25. The NFE results show that the larger ratios continue to reduce computational work, but the accuracy degradation at $\gamma \geq 0.20$ shows that lower denoising cost does not necessarily translate into better task performance. We therefore use $\gamma = 0.15$ and $t_{\text{cutoff}} = 0.05$ as the default configuration for the MATH benchmark.

Table 5: Hyperparameter sensitivity analysis on the MATH dataset. NFE denotes the Number of Function Evaluations, i.e., the average number of model forward evaluations per rollout batch during denoising generation. Top: Sequential search of the Temporal Pruning Cutoff t_{cutoff} with fixed $\gamma = 0.00$. Bottom: Fine-grained sweep of the Spatial Pruning Parameter γ under the optimized fixed $t_{\text{cutoff}} = 0.05$.

γ	t_{cutoff}	Acc.	NFE	Time (s)
Phase 1: Search of Temporal Cutoff t_{cutoff} (Fixed $\gamma = 0.00$)				
0.00	0.00	34.2%	128.0	126,045
0.00	0.05	**35.8%**	123.0	102,190
0.00	0.10	35.2%	**117.0**	**102,165**
Phase 2: Sweep of Spatial Ratio γ (Fixed $t_{cutoff} = 0.05$)				
0.00	0.05	35.8%	123.0	102,190
0.05	0.05	36.2%	118.8	98,537
0.10	0.05	32.8%	113.9	96,156
0.15	0.05	**36.6%**	108.3	95,361
0.20	0.05	34.2%	103.4	94,991
0.25	0.05	34.0%	**98.4**	**95,290**

B.2 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 different dLLM reinforcement learning objectives. To verify this, we integrate STP with Diffu-GRPO Zhao et al. (2025), SPG Wang et al. (2025), and SDMC Rojas et al. (2026) on the MATH dataset. SDMC is a variance-reduction technique that suppresses the dominant timestep-sampling variance in ELBO estimation via deterministic quadrature over diffusion time. We implement its token-level variant for consistency.

As shown in Table 6, STP consistently improves both accuracy and training efficiency across all three RL algorithms. For Diffu-GRPO, incorporating STP yields a +1.60% accuracy boost alongside a **29.8%** speedup. With SPG, STP delivers a **0.6%** absolute accuracy gain and a **15.4%** speedup. For SDMC, STP yields an accuracy improvement while achieving a **9.0%** speedup. These results demonstrate that STP reduces sampling overhead without conflicting with either policy gradient formulations or variance-reduced ELBO estimators.

Table 6: Compatibility of STP with advanced RL algorithms on MATH. We report absolute accuracy and RL training wall-clock time. Results for Diffu-GRPO and SPG are measured on NVIDIA A800 GPUs, whereas results for the SDMC method are measured on NVIDIA H100 GPUs. The reported speedups are computed within each method using the same GPU type.

Method	Accuracy	Time (s)	Speedup
Diffu-GRPO	32.8%	113,358	-
+ STP	**34.4%**	**79,529**	**+29.8%**
SPG	34.0%	119,664	-
+ STP	**34.6%**	**101,207**	**+15.4%**
SDMC	34.9%	47,171	-
+ STP	**35.0%**	**42,931**	**+9.0%**

B.3 Token Selection Strategies

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 7 shows that Low Confidence Retention significantly outperforms Random Selection (36.6% vs. 33.0%). Although Random Selection exhibits lower wall-clock time due to arbitrary token fixing, it incurs severe performance degradation. The superiority of Low Confidence 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 7: Comparison of Spatial Pruning strategies ($\gamma = 0.15$). Retaining low-confidence tokens for active exploration yields superior accuracy.

Strategy	**Acc.**	**Time (s)**
Random	33.0%	**87,336**
Low Confidence	**36.6%**	101,200

B.4 Ablation Study

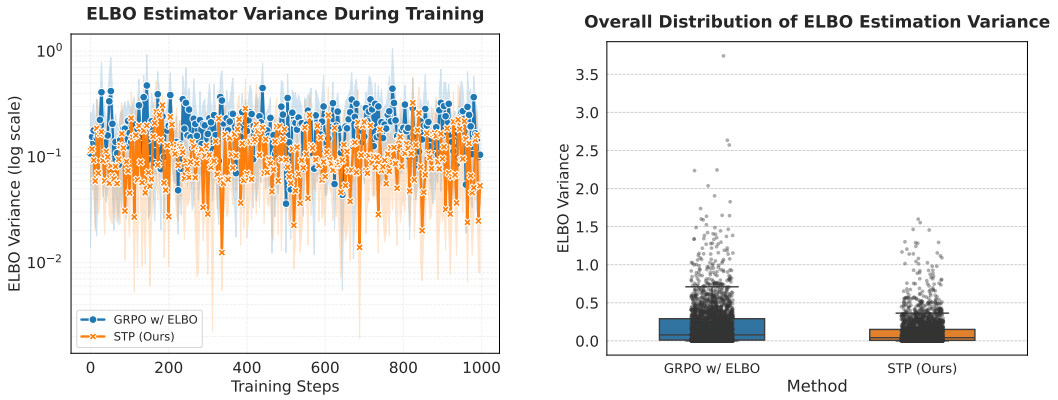
We isolate the effects of Spatial Pruning (SP) and Temporal Pruning (TP) to evaluate their distinct roles in the trade-off between efficiency and performance. Table 8 presents the results. Spatial Pruning alone raises accuracy from 34.2% to 36.6% while reducing training time from 126,045s to 101,200s. Temporal Pruning alone reaches 35.8% accuracy and reduces training time to 102,190s, showing that early completion also improves both measures relative to the baseline. Combining the two components preserves the highest observed accuracy, 36.6%, while lowering training time further to 95,361s. Thus, the ablation indicates that spatial pruning supplies the main accuracy gain, whereas temporal pruning contributes an additional efficiency reduction without sacrificing that gain.

B.5 Empirical diagnostics of ELBO-estimation variance.

We provide an additional diagnostic of ELBO-estimation variance on the MATH benchmark. For each sampled response, we record the ELBO estimates during the first 1,000 training steps and com-

Table 8: Ablation of STP components on MATH. Combining spatial and temporal pruning achieves the best efficiency while preserving the highest observed accuracy.

Method	SP	TP	Acc.	Time (s)
GRPO w/ ELBO	×	×	34.2%	126,045
Spatial Only	✓	×	36.6%	101,200
Temporal Only	×	✓	35.8%	102,190
STP (Ours)	✓	✓	**36.6%**	**95,361**



(a) Variance over Training Steps

(b) Variance Distribution

Figure 2: **Empirical diagnostics of ELBO-estimation variance.** (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, qualitatively consistent with Theorem 4.2.

pute the empirical variance $\mathrm{Var}(\widehat{\mathcal{B}}^S_{\pi_\theta}(y))$ and $\mathrm{Var}(\widehat{\mathcal{B}}_{\pi_\theta}(y))$. As illustrated in Fig. 2a, 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. 2b shows a lower median and fewer outlying variances. These empirical observations corroborate the stability and performance gains discussed in Sec. 5.3 and Sec. 5.4.

C Detailed Hyperparameters

Table 9 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).

D Detailed Related Work

D.1 Diffusion Language Models

Diffusion-based text generation was first explored through continuous representations, where discrete text is mapped to a continuous space before denoising Li et al. (2022); Gong et al. (2022). Discrete diffusion models operate directly on token states Austin et al. (2021); Lou et al. (2024), and masked diffusion provides a particularly convenient formulation in which a model reconstructs masked tokens from bidirectional context Shi et al. (2024); Sahoo et al. (2024). Large masked diffusion language models such as LLaDA Nie et al. (2025) and Dream Ye et al. (2025) demonstrate that this paradigm can reach competitive language-modeling and reasoning quality while allowing flex-

Table 9: 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

ible generation orders and inpainting. Adaptation-based models such as DiffuLLaMA Gong et al. (2025a) and hybrid block-diffusion models Arriola et al. (2025) explore alternative routes to scaling and efficient inference. Commercial systems such as Mercury Khanna et al. (2025) and Gemini Diffusion DeepMind. (2025) further indicate the practical potential of diffusion generation. Our work focuses on masked dLLMs and addresses this cost at the RL rollout level: temporal pruning skips a late refinement segment, while spatial pruning reduces the positions that are actively explored and scored.

D.2 Reinforcement Learning and Likelihood Estimation

PPO Schulman et al. (2017) and GRPO Shao et al. (2024) rely on policy ratios that are readily available for autoregressive models but intractable for dLLMs. The d1 line of work introduced Diffu-GRPO, which uses a mean-field or one-step denoising approximation to make ratio estimation practical Zhao et al. (2025). MMaDA/UniGRPO Yang et al. (2025) and the coupled-GRPO implementation in DiffuCoder Gong et al. (2025b) modify masking or coupled sampling to improve efficiency. LLaDA 1.5 uses preference optimization to reduce dependence on high-variance ELBO estimates Zhu et al. (2025), wd1 removes the explicit policy ratio through a weighted likelihood objective Tang et al. (2025), and SPG uses upper and lower likelihood bounds to control gradient bias Wang et al. (2025). GDPO Rojas et al. (2026) directly studies ELBO-estimation variance and introduces semi-deterministic Monte Carlo (SDMC), combining deterministic integration with stochastic sampling for sequence-level policy optimization. Entropy-Guided Step Selection and Stepwise Advantages Kunde et al. (2026) instead formulates denoising as a finite-horizon decision process, selects steps using entropy, and estimates intermediate advantages with one-step completions. StableDRL Zhong et al. (2026) analyzes reward collapse caused by noisy estimated ratios and proposes unconditional clipping and self-normalization. Mask-Aware Policy Gradients Raajesh et al. (2026) models token prediction and mask-position selection as separate actions, yielding token and masking contributions to the gradient. STP operates at a different level. Spatial pruning changes the conditional corruption and scoring process by fixing selected anchor positions; temporal pruning changes the rollout path by replacing the remaining tail with a parallel completion. This complements policy gradient-level such as SPG and estimator-level methods such as GDPO.

D.3 Guided Exploration, Inpainting, and External Rollouts

Semi-offline RL combines offline information with online generation to improve exploration efficiency Chen et al. (2023). In masked dLLMs, IGPO Zhao et al. (2026) uses inpainting to insert partial ground-truth reasoning traces when ordinary on-policy groups contain only incorrect responses. Its elastic hint injection and correctness-gated updates create reward variation and steer exploration toward promising regions. Prefix-RFT Huang et al. (2025b) provides a related idea for autoregressive models by sampling prefixes from demonstrations, but it does not exploit bidirectional inpainting. ERILS Lee et al. (2026) incorporates higher-reward rollouts from an external policy, controls their lengths, and processes external and on-policy rewards separately. STP shares the general idea of conditioning generation on partial information, but its purpose and source of guidance differ. It uses static model pre-generations as anchors, fixes only selected positions, and scores the remaining positions with a conditional ELBO; it does not inject ground-truth chunks on failure groups as IGPO does. It also does not train on complete external-policy responses as ERILS does. Thus, the contribution of STP is the combination of anchor-conditioned spatial pruning, temporal early completion, and an explicit analysis of the associated estimation and output trade-offs.