OPTIMIZING REMASKING SCHEDULES FOR REASONING IN DISCRETE DIFFUSION MODELS

Radostin Cholakov*, **Zeyneb N. Kaya***, **& Nicole H. Ma***
Stanford University
{radicho, zeynebnk, manicole}@stanford.edu

ABSTRACT

Discrete diffusion language models (DLLMs) have emerged as a new paradigm of language modeling that offers improved inference efficiency and nonlinear generation and reasoning. While standard methods rely on fixed or heuristic schedules (e.g., random or confidence-based), we present LEADS, a framework that enables dynamic inference-time control for DLLMs with a learned remasking scheduler optimized for downstream performance. LEADS chooses what tokens are denoised at each diffusion step based on the internal representations of the model and dynamically allocates compute for token efficiency. On mathematical reasoning tasks, LEADS achieves 19.2% relative improvement (12 pp) over low-confidence based denoising schedules and reduces required diffusion steps by up to 15.3%.

1 INTRODUCTION

Discrete diffusion large language models (DLLMs) have recently shown success as a compelling alternative to autoregressive LMs (Li et al., 2022; Zou et al., 2023; Nie et al., 2025). While autoregressive models generate text sequentially via next-token prediction, DLLMs generate through a multi-step denoising process that repeatedly refines an initially masked sequence with bidirectional attention. This paradigm brings several especially appealing properties for reasoning: (i) improved sampling efficiency and thinking in the added dimension of diffusion steps rather than just chain-of-thought tokens; and (ii) nonlinear reasoning and leveraging global context in planning and self-correction (Fu et al., 2025; Ye et al., 2024; Liu et al., 2025).

Despite this premise, DLLM performance is often bottlenecked by the inference-time scheduling and sampling procedure that decides which positions to commit versus remask at each step (Li et al., 2025; Ma et al., 2025). Current systems typically use linear schedules with heuristic samplers (e.g., random or confidence-based remasking).

In this work, we ask whether models can learn the remasking schedule itself, optimized for task success. We view remasking as a parameterized decision-making problem on internal representations at a diffusion step to select token-level *keep* vs. *remask* actions. The perspective reframes diffusion inference as a controllable process; this enables remasking to increase expected information gain at the next conditioning step and for dynamic schedules to adaptively "think" more or less according to uncertainty.

We propose LEADS (Learned Adaptive Denoising Scheduler), which learns a token-level remasking policy for DLLMs as an inference-time controller optimized via RL. We demonstrate that the learned denoising schedulers improve reasoning accuracy and can reduce the number of diffusion steps with competitive performance, enabling adaptive compute–performance tradeoffs during generation.

2 METHOD

We introduce LEADS, which is composed of a frozen denoiser model and a remasking scheduler head that learns which token positions to keep versus remask at each denoising step to maximize downstream task reward.

*Equal Contribution

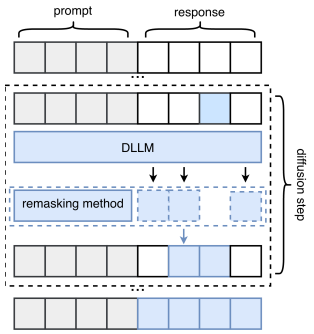

Figure 1: **DLLM Inference.** At each diffusion step, the denoiser receives the input tokens and predicts probability distributions across all generation token positions. The remasking method (i.e. random/low-confidence heuristic or learned remasking scheduler head score outputs from denoiser hidden states) determines positions to remask vs. keep to condition on at the next step. The process repeats for generation.

2.1 Discrete Diffusion Language Models

A discrete diffusion language model (DLLM) generates a completion by iteratively denoising a masked sequence. Given a prompt token sequence $x^{\text{prompt}} \in \mathcal{V}^{L_p}$ and a desired generation length L_g, we form an initial sequence

$$x_0 = [x^{\text{prompt}}; \underbrace{\texttt{[MASK]}, \ldots, \texttt{[MASK]}}_{L_g}] \in (\mathcal{V} \cup \{\texttt{[MASK]}\})^L, \quad L = L_p + L_g.$$

At denoising step $t \in \{0, \ldots, T-1\}$, the denoiser produces token distributions for all positions,$p_\phi(\cdot \mid x_t) \in [0,1]^{L \times |\mathcal{V}|}$, where ϕ denotes frozen model parameters. Let $\hat{x}_{t+1}$ be the model's proposed tokens at each position. A *scheduler/sampler* then selects a subset of positions to *commit* (keep unmasked) while remasking the remaining positions for further refinement. This yields the next sequence x_{t+1}, and the process repeats for T steps. The final completion is the suffix of x_T of length L_g. This process is illustrated in Figure 1.

Heuristic schedulers commonly use a fixed denoising rate and number of diffusion steps (e.g., commit k positions per step, where $k \equiv \text{seq\_len}/\text{steps}$) combined with either random selection or confidence-based selection (commit positions with highest token probability).

2.2 Learning Remasking for Inference-Time Control

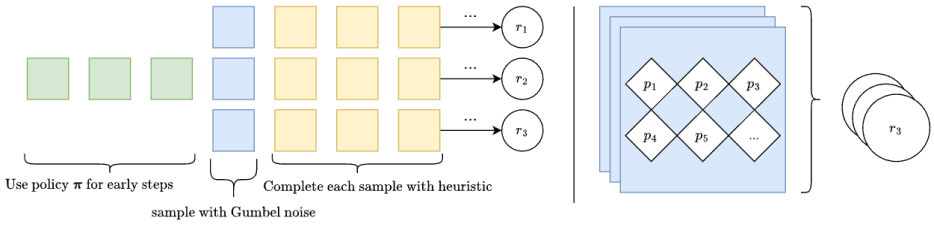

Figure 2: **Training LEADS.** At each diffusion step, the remasking scheduler head outputs token-level keep/remask decisions that determine which positions are committed and which remain masked for further refinement. We use target-step credit assignment by sampling a diffusion step t, running the learned scheduler for steps $< t$ to reach state x_t, then sample G remasking decisions at step t and complete each rollout using a fixed heuristic for steps $> t$. Rewards are computed on final sequence, which update the remasking scheduler head.

We introduce a remasking scheduler policy π_θ that, at each step, decides which positions to keep versus remask. We represent the environment state by the denoiser's per-token internal representa-

tions. Figure 2 illustrates the training procedure for LEADS. Concretely, let $H_t \in \mathbb{R}^{L\times d}$ denote the final-layer hidden states of the denoiser when run on x_t. The policy outputs per-token Bernoulli probabilities $a_{t,j}$ indicating *keep/commit* (1) versus *remask* (0) for each position j. Given sampled actions, we form x_{t+1} by committing tokens at positions with $a_{t,j} = 1$ (setting $x_{t+1,j} = \hat{x}_{t+1,j}$) and remasking positions with $a_{t,j} = 0$ (setting $x_{t+1,j} = \texttt{[MASK]}$), optionally subject to a denoising budget constraint.

Budgeted vs. Adaptive Schedules. We consider two inference regimes:

- **Linear (budgeted) schedule:** commit exactly k generation tokens per step. The policy produces scores/probabilities and we keep the top-k among generation positions to match the standard linear denoising rate.
- **Nonlinear (adaptive) schedule:** commit any token whose confidence exceeds a threshold τ and terminate when all generation positions are committed (or when reaching a maximum step budget). This enables adaptive compute allocation across examples.

2.3 OPTIMIZATION

We train the remasking scheduler to maximize downstream task performance using reinforcement learning. A rollout consists of running the denoising process for up to T steps under a sequence of remasking decisions, producing a final completion. We compute a scalar correctness reward R from the completion. For the adaptive schedule, we optionally add a small efficiency bonus that favors fewer steps: $R \leftarrow R + \alpha(T - \texttt{steps})$.

To update the remasking scheduler, we use Group Relative Policy Optimization (GRPO) style advantage estimation. For a given input at step t, we sample G candidate remasking decisions, execute rollouts to obtain rewards $\{R^{(i)}\}_{i=1}^{G}$, and apply a policy gradient update:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(r^{(i)}\hat{A}^{(i)},\ \text{clip}(r^{(i)}, 1-\epsilon, 1+\epsilon)\hat{A}^{(i)}\right)\right], \text{with } \hat{A}^{(i)} = \frac{R^{(i)} - (\{\bar{R^{(j)}}\})}{\text{std}(\{R^{(j)}\}) + \epsilon_A}.$$

where $r^{(i)}$ is the importance ratio between the new and old policy for the sampled decision.

2.4 TARGET-STEP CREDIT ASSIGNMENT FOR COMPUTE-EFFICIENT RL

Backpropagating through full T-step diffusion rollouts is expensive, so we use *target-step* training: each update optimizes the remasking scheduler at a single denoising step while keeping the rest of the rollout fixed. We sample a target step t (biased toward early steps), generate x_t using the current remasking scheduler without gradients for steps $0, \ldots, t-1$, and at step t sample a group of G remasking decisions with gradients enabled. To isolate the effect of the decision at t, we complete steps $t{+}1, \ldots, T{-}1$ with a fixed low-confidence heuristic, following the intuition that improving information quality at a given denoising level supports subsequent refinement (Sclocchi et al., 2024). This yields a reward for each candidate decision and a low-variance GRPO update at step t.

Because late-step actions often induce little reward variation, we sample $t \in \{0, \ldots, T-1\}$ from an exponentially biased distribution,

$$P(t) \ \propto\ \lambda e^{-\lambda t} + \epsilon,$$

where $\epsilon > 0$ ensures nonzero probability for all steps. Overall, this strategy reduces compute, improves signal-to-noise, and trains the remasking scheduler under the state distribution induced by its own earlier decisions.

2.5 COLD-START TRAINING

To stabilize early training, we cold-start π_θ with strong heuristic remasking policy (low-confidence masking). We treat the heuristic's per-token keep/remask decisions as labels and train π_θ with a weighted binary cross-entropy loss to account for class imbalance. We then switch to GRPO fine-tuning on task reward.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We evaluate whether LEADS can improve reasoning accuracy and reduce inference-time compute in discrete diffusion LLMs. We work with LLaDA (Nie et al., 2025) as our base DLLM. The base denoising model weights are frozen. We add a lightweight parameterized remasking scheduler head. Hidden states from the previous time step of the denoiser are given as an input to the mask predictor head at each diffusion step.

We focus on the task of mathematical reasoning with GSM8K (Cobbe et al., 2021), a compact benchmark of grade-school math word problems designed to probe reasoning. We use binary correctness outcomes for rewards and accuracy.

We evaluate several approaches:

- **Random remasking**: commit k tokens per step chosen uniformly at random among generation positions.
- **Low-confidence remasking**: commit the k tokens with the highest confidence (largest maximum token probability) and remask the rest.
- **LEADS (budgeted)**: commit $k = 2$ tokens per step by selecting the top-k scores among generation positions from the learned remasking scheduler head.
- **LEADS (adaptive)**: commit all positions whose score exceeds threshold τ at each step and terminate once all generation tokens are committed, up to a maximum of $T = 64$ steps.

3.2 RESULTS

Table 1: Results on the GSM8K benchmark with 4-shot examples and using 64 steps (max steps for LEADS_adaptive approach), gen_len 128. Random heuristic and low-confidence heuristic are performed according to the methods of Nie et al. (2025) with LLaDA. 'Mean steps' indicates the average number of diffusion steps taken for generations (constant for all except LEADS (adaptive).

Method	Value	Mean Steps
Random heuristic	0.4924	64
Low confidence heuristic	0.6079	64
LEADS (budgeted)	0.7250	64
LEADS (adaptive)	0.6990	54.2

The results show that LEADS achieves gains above all baselines presented, with a $\sim$12pp improvement over low-confidence heuristics and a $\sim$23pp improvement over random masking. Furthermore, with the adaptive denoising schedule, we observe that we can increase performance while also reducing the number of diffusion steps required; the adaptive approach uses 15.3% fewer diffusion steps while still achieving substantial gains. We demonstrate that the model can thus choose when to think harder and when to save on inference compute according to problem uncertainty. We observe that it tends to predict more tokens briefly at the very earliest timesteps and then primarily towards the mid-late timesteps. More tokens can be predicted at once where tokens carry similar amounts of information for next steps.

4 CONCLUSION

We present LEADS, an approach for learning an optimized remasking scheduler with reinforcement learning for discrete DLLMs that enables adaptive denoising that improves efficiency and reasoning. On mathematical reasoning, the learned remasking scheduler improves reasoning accuracy over heuristic remasking strategies and dynamically allocates inference compute. We demonstrate that diffusion steps can be leveraged to provide a valuable dimension of reasoning and control for DLLMs while being able to flexibly allocate compute budgets based on internal model states and dynamics.

REFERENCES

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Yonggan Fu, Lexington Whalen, Zhifan Ye, Xin Dong, Shizhe Diao, Jingyu Liu, Chengyue Wu, Hao Zhang, Enze Xie, Song Han, Maksim Khadkevich, Jan Kautz, Yingyan Celine Lin, and Pavlo Molchanov. Efficient-dlm: From autoregressive to diffusion language models, and beyond in speed, 2025. URL https://arxiv.org/abs/2512.14067.

Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. Beyond fixed: Training-free variable-length denoising for diffusion large language models, 2025. URL https://arxiv.org/abs/2508.00819.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-lm improves controllable text generation, 2022. URL https://arxiv.org/abs/2205.14217.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising, 2025. URL https://arxiv.org/abs/2410.06264.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps, 2025. URL https://arxiv.org/abs/2501.09732.

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, 2025.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data, 2024. URL https://arxiv.org/abs/2402.16991.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of thoughts: Chain-of-thought reasoning in diffusion language models, 2024. URL https://arxiv.org/abs/2402.07754.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. A survey of diffusion models in natural language processing, 2023. URL https://arxiv.org/abs/2305.14671.