DiFFPO: Training Diffusion LLMs to Reason Fast and Furious via Reinforcement Learning

Hanyang Zhao
Columbia University
hz2684@columbia.edu

Dawen Liang
Netflix
dliang@netflix.com

Wenpin Tang
Columbia University
wt2319@columbia.edu

David D. Yao
Columbia University
yao@columbia.edu

Nathan Kallus
Netflix
nkallus@netflix.com

Abstract

We propose **DiFFPO**, **Di**ffusion **F**ast and **F**urious Policy Optimization, a unified framework for training masked diffusion large language models (dLLMs) to reason not only *better (furious)*, but also *faster* via reinforcement learning (RL). We first generalize the existing baseline approach such as `d1` [23] by proposing to train surrogate policies via off-policy RL, whose likelihood is much more tractable as an approximation to the true dLLM policy. This naturally motivates improved two stage likelihood approximations combined with importance sampling correction, which leads to RL algorithms with better sample efficiency and superior task performance. Second, we propose a new direction of training efficient samplers/controllers of dLLMs policy. Via RL, we incentivize dLLMs' natural multi-token prediction capabilities by letting the model learn to adaptively allocate an inference threshold for each prompt, which yields even better accuracies with lower number of function evaluations (NFEs) compared to the base model. Finally, we consider joint training of the dLLM policy and the sampler together for obtaining the best performance in improving the Pareto frontier of the inference-time compute for dLLMs. We showcase the effectiveness of our pipeline by training open source large diffusion language models over benchmark math and planning tasks.

1 Introduction

Reinforcement Learning from Verifiable Rewards (RLVR, [11]) has achieved remarkable success in enhancing the reasoning capabilities of Large Language Models (LLMs) [5, 9]. The fine-tuned Large Reasoning Models (LRMs) show drastic improvement in solving tasks such as math and coding [12, 16], which require strong reasoning capabilities. These models match, and even surpass the performance of the best human players in math contests [3]. Despite these successes, LRMs are notorious for long inference-time, and overthinking for easy questions [4, 18], which limit their applicability to scenarios that have less tolerance on latencies or inference budgets. A line of recent research on efficient reasoning [19] proposed either training-free early stopping mechanisms, or generation-length-aware RL losses [10, 21]; however, the resulting models are still fundamentally bottle-necked by the left-to-right autoregressive decoding in decoder-only transformers [20], suffering from quadratic inference costs with respect to the length of reasoning traces.

Diffusion LLMs (dLLMs) [13–15, 17, 22], an emerging family of LLMs based on discrete-space diffusion models [1, 6], have the natural premise of going beyond *left-to-right generations* by *any-order generations* and *multi-token predictions* . In contrast with the extensive literature on post-training autoregressive models, the relevant study for dLLMs is quite limited. Notably, there

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

has been little work on post-training frontier dLLMs by RL, raising the question of how RL can be used to enhance the reasoning capabilities of diffusion language models. The purpose of this work is to design a scalable and effective RL approach for improve efficient reasoning of dLLMs. Our contributions can be summarized as follows:

(1) We first propose an efficient RL post-training paradigm for fine-tuning dLLMs from an off-policy RL perspective. Concretely, we generalize `d1` [23] by training (efficient) surrogate policies with performance improvement, as opposed to training base dLLMs policies whose likelihood requires extensive GPU memory, and is inefficient to compute and update. Here we propose novel surrogate policies by conditioning on additional latents at the response-generation level for better approximation, instead of only conditioning on prompts as in `d1`. As an intermediate step, we introduce an importance sampling correction to address the distribution mismatch between the surrogate policies and the true dLLMs behavior policies. We showcase an evident performance gain by our improved techniques over the baseline methods.

(2) Second, we put forward an RL framework to train efficient dLLMs samplers. Motivated by the EB-sampler proposed in [2], we propose to train a prompt-aware inference threshold via RL. This approach naturally leverages the structural property of masked dLLMs for multi-token predictions, and can increase the downstream accuracies with less average compute compared to a prompt-free threshold as in [2]. We also explore the joint training of the policy weights and the sampler to achieve the best performance in enhancing the inference-time computation.

Combining the above two proposals yields a unified pipeline, which we term as **DiFFPO** - **Di**ffusion **F**ast and **F**urious Policy Optimization, for training masked dLLMs to reason not only *better (furious)*, but also *faster* via RL. We demonstrate the effectiveness of DiFFPO in enhancing the inference-time performance of diffusion LLMs on benchmarking math and planning tasks, and push the boundaries of designing scalable RL algorithms towards training efficient LRMs.

2 Preliminaries

In this work, we rely on *masked diffusion models*, which have shown to achieve the state-of-the-art performance among different formulations of dLLMs. Here we follow presentations in MDLM [15].

Masked Diffusion Models. Let `[mask]` be a special masked token, and denote by $\boldsymbol{m}$ the one-hot representation of it. The forward processes of MDLM [1, 15] interpolate between the clean data distribution $\boldsymbol{x} \sim p_{\text{data}}(\cdot)$ and a target non-informative distribution $\text{Cat}(\cdot; \boldsymbol{m})$. The latent variables $\boldsymbol{z}_t$ are defined by $q\left(\boldsymbol{z}_t \mid \boldsymbol{x}\right) = \text{Cat}\left(\mathbf{z}_t; \alpha_t \boldsymbol{x} + (1-\alpha_t)\,\boldsymbol{m}\right)$, where $\alpha_t \in [0,1]$ is a strictly decreasing function in t, with $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$. This process can be viewed as a masking process, because $q\left(\boldsymbol{z}_1 \mid \boldsymbol{x}\right) \approx \boldsymbol{m}$. The posterior of the masking process $(t > s)$ can be explicitly computed as:

$$q\left(\boldsymbol{z}_s \mid \boldsymbol{z}_t, \boldsymbol{x}\right) = \begin{cases} \text{Cat}\left(\boldsymbol{z}_s; \boldsymbol{z}_t\right) & \boldsymbol{z}_t \neq \boldsymbol{m}, \\ \text{Cat}\left(\mathbf{z}_s; \frac{1-\alpha_s}{1-\alpha_t}\boldsymbol{m} + \frac{\alpha_s - \alpha_t}{1-\alpha_t}\boldsymbol{x}\right) & \boldsymbol{z}_t = \boldsymbol{m}. \end{cases} \tag{1}$$

Since $\boldsymbol{x}$ is unknown, MDLMs learn a function approximation $\boldsymbol{x}_\theta(\boldsymbol{z}_t, t)$, and approximate the true posterior with $p_\theta\left(\boldsymbol{z}_s \mid \boldsymbol{z}_t\right)$ by replacing $\boldsymbol{x}$ with approximated $\boldsymbol{x}_\theta$ in (1) when $\boldsymbol{z}_t = \boldsymbol{m}$.

dLLMs Inference. Most existing works focus on a fixed number of multi-token predictions, such as Top-k sampling [8, 10, 14]. In this work, our sampling scheme mainly follows the Entropy-Bounded (EB) sampler proposed in [2], which yields better accuracy and efficiency tradeoff than the Top-k sampler that sweeps over different k's. Jointly unmasking a sequence of variables $X = (\boldsymbol{x}^1, \cdots, \boldsymbol{x}^L)$, the EB-sampler computes the entropy of each unmasked position as the cost of that position $c(l) = H(p^\theta(\boldsymbol{x}^l \mid X))$, and then unmask all positions in U with the maximum cardinality: $\sum_{l \in U} c(\ell) \leq \gamma$, where γ is a predetermined error tolerance hyperparameter.

RL for dLLMs. RL aims at maximizing the expected reward of policy generations, i.e., $\mathbb{E}_{o \sim \pi_\theta(\cdot|c)} r(c, o)$, where $r(c, o)$ is the verifiable reward of generation o with respect to the prompt c, such as correctness or unit test success rates. GRPO [16], as a variant of Reinforce and PPO, first samples a group of outputs $\{o_i\}_{i=1}^G$ from the behaviour (old) policy $\pi_{\theta_{\text{old}}}$ under the same prompt c, and computes the advantage function $A_i = r\left(c, o_i\right) - \frac{1}{G}\sum_{j=1}^G r\left(c, o_j\right)$. Denote $o^{<k}$ as tokens

already generated before time k of completion o, GRPO maximizes the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{o_i \sim \pi_{\theta_{\text{old}}}(\cdot|c)} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{k=1}^{|o_i|} \min\left(\rho_i^k A_i, \text{clip}\left(\rho_i^k, 1-\varepsilon, 1+\varepsilon\right) A_i\right), \tag{2}$$

where $\rho_i^k = \frac{\pi_\theta\left(o_i^k | c, o_i^{<k}\right)}{\pi_{\text{old}}\left(o_i^k | c, o_i^{<k}\right)}$ is the token-level likelihood ratio. Since ρ_i^k is difficult and inefficient to compute in practice, `d1` [23] proposed a *mean-field approximation* $\hat{\rho}_i^k = \frac{\pi_\theta\left(o_i^k | c\right)}{\pi_{\text{old}}\left(o_i^k | c\right)}$ to ρ_i^k in (2) (i.e., the likelihood conditioning on prompts, while omitting the already unmasked tokens.)

3 DiFFPO

3.1 Improved techniques in efficient model post-training

Rethinking via off-policy RL. We first revisit the RL task for dLLMs from an off-policy RL perspective. Since the true likelihood of dLLMs generation is hard to estimate, our proposal is: (1) find an surrogate policy that is close to the actual dLLM policy, while its conditional likelihood is easy to compute and thus much more tractable; (2) optimize the surrogate policy via RL. Given two policies that share the same parameters being $\pi_\theta, \hat{\pi}_\theta$ respectively, and the reward being bounded by M, we have by Pinsker's inequality:

$$\mathbb{E}_{o \sim \pi_\theta(\cdot|c)} r(c, o) \geq \mathbb{E}_{o \sim \hat{\pi}_\theta(\cdot|c)} r(c, o) - M(2 \cdot \text{KL}(\hat{\pi}_\theta \| \pi_\theta))^{\frac{1}{2}}. \tag{3}$$

For instance, we can take a policy $\hat{\pi}_\theta$, which generates token on every position at the first step by conditioning on the prompt, then it is easy to compute the conditional likelihood given any completions under this policy with the same weights as our base dLLM. Yet, such an approximation can be too coarse, because it totally ignores the information of already generated tokens.

Additional conditioning latents and importance sampling. We propose to randomly sample $t \in [0, T]$, and compute $\pi_\theta(o_i^k \mid c, o_i^{<t(k)})$, where $t(k) = t$ if $t < k$, and 0 otherwise. This provides better approximations for tokens when $k > t$ by conditioning on the generations up to a past timestep, while remaining simple and efficient. Note that the generations are still from the base policy, so we can apply importance sampling:

$$\mathbb{E}_{o \sim \hat{\pi}_\theta(\cdot|c)} r(c, o) = \mathbb{E}_{o \sim \pi_\theta(\cdot|c)} \frac{\hat{\pi}_\theta(\cdot \mid c)}{\pi_\theta(\cdot \mid c)} r(c, o). \tag{4}$$

Our ultimate loss objective is then:

$$\mathbb{E}_{o_i \sim \pi_{\theta_{\text{old}}}, t \sim \mathcal{U}[0,T]} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{k=1}^{|o_i|} \underbrace{\min(C, \frac{\pi_{\text{old}}(o_i^k \mid c, o_i^{<t(k)})}{\pi_{\text{old}}(o_i^k \mid c, o_i^{<k})})}_{\text{IS term}} \min\left(\bar{\rho_i}^k A_i, \text{clip}\left(\bar{\rho_i}^k, 1-\varepsilon, 1+\varepsilon\right) A_i\right),$$

where $\bar{\rho}_i^k = \frac{\pi_\theta\left(o_i^k | c, o_i^{<t(k)}\right)}{\pi_{\text{old}}\left(o_i^k | c, o_i^{<t(k)}\right)}$ is the likelihood ratio of the surrogate policy. Compared to `d1` [23], we yield a better likelihood approximation by conditioning on (randomly drawn) latents at the response-generation level, which we refer this as Two Times Mean Field Approximation. Moreover, our loss includes an importance sampling term which builds from off-policy RL.

3.2 Unlocking better adaptive multi-token prediction via sampler post training

Unlike the EB-sampler [2] that uses a fixed threshold to decide which token(s) to unmask for all prompts, we propose to parameterized function to encode prompt features. We obtain a dLLMs embedding of the prompt by averaging over the hidden dimension features $f_\theta^l(c)$ of each position l, and train a linear mapping $\boldsymbol{w}$ on top of the embeddings (we omit the bias term here for clarity). We also introduce an upper threshold $\gamma_{\max}$, and map each feature to the true inference threshold by

$$\gamma_{\boldsymbol{w}}(c) = \gamma_{\max} \text{ sigmoid}\left(\boldsymbol{w}^\top \left(\frac{1}{L} \sum_{l=1}^{L} f_\theta^l(c)\right) + \beta\right), \quad \text{with } \beta = \log \frac{\gamma}{\gamma_{\max} - \gamma}, \tag{5}$$

where $\gamma \in (0, \gamma_{\max})$ is the initial global threshold, since $\gamma_{\boldsymbol{w}_0}(c) = \gamma$ when the initial linear layer weights $\boldsymbol{w}_0$ are set to be all 0. For RL training, we fix a noise level σ for Gaussian exploration $\epsilon \sim \mathcal{N}(0, I)$ before applying sigmoid activation and use the perturbed threshold for inference:

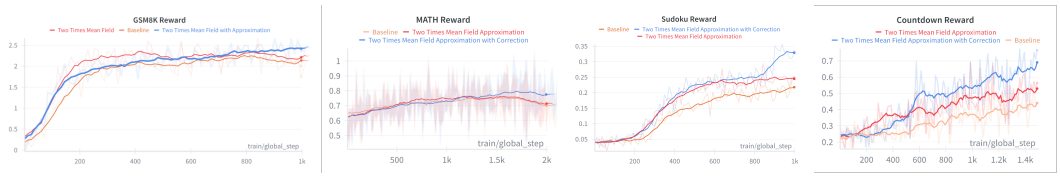
$$\gamma(c) = \gamma_{\max} \operatorname{sigmoid}\left(\boldsymbol{w}^\top \left(\frac{1}{L} \sum_{l=1}^{L} f_\theta^l(c) \right) + \beta + \sigma\epsilon \right). \tag{6}$$

We can then derive the similar GRPO loss for updating $\boldsymbol{w}$ upon each prompt feature.

4 Experimental Results

Experimental Setup. We conduct experiments on training quantized open source diffusion language models LLaDA [14] on benchmark math and planning tasks. We use LoRA [7] ($r = 128$) for parameter-efficient training – the same parameters as in [23]. For a fair comparison, we use the same five random reeds for the baseline `d1` and our proposed algorithm, and report the best performed curve among different seeds.

Improved sample efficiency and task performance. We first showcase the effectiveness of our algorithm on math and planning benchmarks, including GSM8K, MATH, SUDOKU and Countdown.

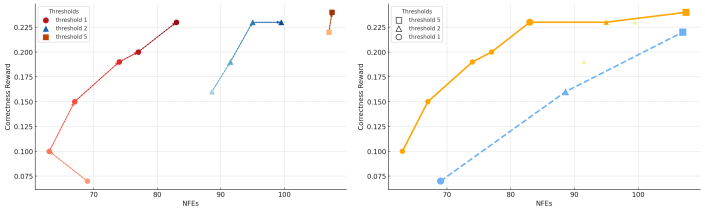


(a) GSM8K (b) Math (c) Sudoku (d) Countdown

Figure 1: Benchmark results (–: DiFFPO; –: DiFFPO without IS; –: `d1` baseline).

As in Figure 1, our algorithm shows a clear margin over `d1` in all tasks. Concretely, using two-times mean field approximation greatly improves the sample efficiency, and applying importance sampling correction further enhances the performance of RL training. Our DiFFPO, combining both higher-order approximation and importance sampling, yields the most evident performance gain.

Improved Pareto frontier via training the sampler only. We study the effect of training the sampler to incentive multi-token predictions for dLLMs via RL. Starting from different thresholds γ and setting $\gamma_{\max} = 2\gamma$, we plot the learning dynamics in Fig 2a, and the efficient frontier evaluated by average correctness rewards in Fig 2b.



(a) Learning Dynamics (b) Accuracy-Efficiency Frontier

Method	Correctness ↑	NFEs ↓
`d1` + Top k	1.65	66.4
`DiFFPO` + Top k	**1.69**	**62.3**
`d1` + EB	1.55	55.7
`DiFFPO` + EB	1.62	42.7
`DiFFPO` + J.T.	**1.67**	**41.3**

(c) Joint Training

Figure 2: Learning Dynamics, Frontiers and Joint training results by training samplers via RL.

Joint training of the model and the sampler. We also consider joint training of the model and sampler by adding up the model and the sampler losses, to improve the inference-time frontier. In Table 2c, we report the optimal checkpoint performance along the training under baseline algorithm. observe that joint training not only improves the (optimal) correctness, but uses less NFEs compared to training the model only.

5 Discussion and Future Works

In this work, we propose DiFFPO – an RL pipeline for post-training dLLMs towards efficient reasoning. We showcase the effectiveness of our proposed algorithms in achieving better sample efficiency, superior performance, and enhancing the accuracy/efficiency tradeoff.

There are several directions to extend our work. It is interesting to understand the generalization behavior of the trained inference threshold; it is also possible to adapt our RL framework to a state (or group)-dependent inference threshold instead of being fixed for each prompt. We leave these problems for future work.

Acknowledgment

The works of Hanyang Zhao and Wenpin Tang are supported by NSF grant DMS-2113779 and DMS-2206038. Wenpin Tang is supported by the Columbia Innovation Hub grant, and the Tang Family Assistant Professorship. The works of Hanyang Zhao, and David Yao are part of a ColumbiaCityU/HK collaborative project that is supported by InnoHK Initiative, The Government of the HKSAR and the AIFT Lab.

References

[1] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Neurips*, volume 34, pages 17981–17993, 2021.

[2] H. Ben-Hamu, I. Gat, D. Severo, N. Nolte, and B. Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. 2025. arXiv:2505.24857.

[3] L. Chen, J. Gu, L. Huang, W. Huang, Z. Jiang, A. Jie, X. Jin, X. Jin, C. Li, and K. Ma. Seed-prover: Deep and broad reasoning for automated theorem proving. 2025. arXiv:2507.23726.

[4] X. Chen, J. Xu, T. Liang, Z. He, J. Pang, D. Yu, L. Song, Q. Liu, M. Zhou, and Z. Zhang. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. 2024. arXiv:2412.21187.

[5] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, and X. Bi. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. 2025. arXiv:2501.12948.

[6] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, and M. Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Neurips*, volume 34, pages 12454–12465, 2021.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[8] Z. Huang, Z. Chen, Z. Wang, T. Li, and G.-J. Qi. Reinforcing the diffusion chain of lateral thought with diffusion language models. 2025. arXiv:2505.10446.

[9] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, and A. Carney. Openai o1 system card. 2024. arXiv:2412.16720.

[10] J. Kim, K. Shah, V. Kontonis, S. Kakade, and S. Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. 2025. arXiv:2502.06768.

[11] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, and S. Lyu. Tulu 3: Pushing frontiers in open language model post-training. 2024. arXiv:2411.15124.

[12] H. Le, Y. Wang, A. D. Gotmare, S. Savarese, and S. C. H. Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. In *Neurips*, volume 35, pages 21314–21328, 2022.

[13] A. Lou, C. Meng, and S. Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. 2023. arXiv:2310.16834.

[14] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. 2025. arXiv:2502.09992.

[15] S. Sahoo, M. Arriola, Y. Schiff, A. Gokaslan, E. Marroquin, J. Chiu, A. Rush, and V. Kuleshov. Simple and effective masked diffusion language models. In *Neurips*, volume 37, pages 130136–130184, 2024.

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

[17] J. Shi, K. Han, Z. Wang, A. Doucet, and M. Titsias. Simplified and generalized masked diffusion for discrete data. In *Neurips*, volume 37, pages 103131–103167, 2024.

[18] J. Su, J. Healey, P. Nakov, and C. Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. 2025. arXiv:2505.00127.

[19] Y. Sui, Y.-N. Chuang, G. Wang, J. Zhang, T. Zhang, J. Yuan, H. Liu, A. Wen, S. Zhong, and H. Chen. Stop overthinking: A survey on efficient reasoning for large language models. 2025. arXiv:2503.16419.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Neurips*, volume 30, 2017.

[21] Y. Xu, H. Dong, L. Wang, D. Sahoo, J. Li, and C. Xiong. Scalable chain of thoughts via elastic reasoning. 2025. arXiv:2505.05315.

[22] J. Ye, Z. Xie, L. Zheng, J. Gao, Z. Wu, X. Jiang, Z. Li, and L. Kong. Dream 7b: Diffusion large language models. 2025. arXiv:2508.15487.

[23] S. Zhao, D. Gupta, Q. Zheng, and A. Grover. d1: Scaling reasoning in diffusion large language models via reinforcement learning. 2025. arXiv:2504.12216.