

# 000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 ENHANCING REASONING FOR DIFFUSION LLMs VIA DISTRIBUTION MATCHING POLICY OPTIMIZATION

Anonymous authors

Paper under double-blind review

## ABSTRACT

Diffusion large language models (dLLMs) are promising alternatives to autoregressive large language models (AR-LLMs), as they potentially allow higher inference throughput. Reinforcement learning (RL) is a crucial component for dLLMs to achieve comparable performance with AR-LLMs on important tasks, such as reasoning. However, RL algorithms that are well-suited for dLLMs' unique characteristics have yet to be developed. This paper proposes **Distribution Matching Policy Optimization (DMPO)**, a principled and theoretically grounded RL fine-tuning method specifically designed to enhance the reasoning capabilities of dLLMs by matching the dLLM policy distribution to the optimal, reward-tilted one through cross-entropy optimization. We identify a key challenge in the implementation with a small training batch size and propose several effective solutions through a novel weight baseline subtraction technique. DMPO exhibits superior performance on multiple reasoning benchmarks without supervised fine-tuning, with an accuracy improvement of up to **54.3%** over previously SOTA baselines and **66.41%** over the base model, underscoring the effectiveness of the distribution matching framework.

## 1 INTRODUCTION

Autoregressive large language models (AR-LLMs) have demonstrated remarkable capabilities in addressing sophisticated reasoning tasks, such as solving challenging math questions and completing coding tasks (Jaech et al., 2024; Anthropic, 2025; Guo et al., 2025a; Novikov et al., 2025; Kimi Team et al., 2025). While these models form the amazing capabilities from pretraining on massive text corpora, the main powerhouse behind the success is scaling the post-training phase with reinforcement learning (RL) techniques, such as Proximal Policy Optimization (PPO, Schulman et al. (2017)) and Group Relative Policy Optimization (GRPO, Shao et al. (2024)), which enhance model abilities through exploration of reward functions and go beyond static datasets. While possessing extraordinary competence, AR-LLMs are known to be expensive for inference due to their sequential, fixed left-to-right generation order, which currently prohibits large-scale deployment.

With the aim of addressing such issues, diffusion large language models (dLLMs) have been investigated as an alternative to the AR models. Unlike their counterparts, dLLMs iteratively refine a sequence from a masked state, allowing for any-order generation, and have shown promising performance in text generation tasks. dLLMs, such as LLaDA (Nie et al., 2025b) and Dream (Ye et al., 2025), have demonstrated competitive performances on many tasks compared to similar-size AR baselines. Recently, commercial models such as Mercury (Inception Labs et al., 2025) and Gemini Diffusion (DeepMind) have demonstrated the capability to achieve a magnitude higher inference throughput without sacrificing generation quality, suggesting dLLM as a promising future direction for language modeling. However, one question that remains largely unanswered is how to transfer the success of RL on LLM to dLLM, thereby further scaling up the model's skills.

Designing RL algorithms for dLLMs faces two major challenges. Due to the bidirectional nature of dLLMs, estimating the log probability of the generated sequences is more expensive than for AR models, making it less favorable to naively adapt LLM post-training algorithms like GRPO to dLLMs, as they heavily rely on such estimation. The GRPO-style algorithms also do not leverage dLLM's unique characteristic of having a *forward* noising process, as they are backward-only algorithms when using generated rollouts. Moreover, existing RL frameworks for enhancing LLM reasoning

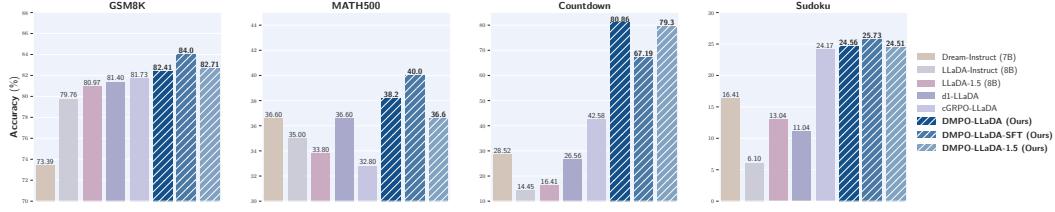


Figure 1: Performance on reasoning benchmarks evaluated with generation length 256. DMPO consistently achieves the best performance across dLLMs, outperforming d1 and cGRPO.

capabilities overly focus on reward maximization (Guo et al., 2025a; Liu et al., 2025c; Zheng et al., 2025a). By targeting only the reward mode, these approaches do not properly utilize dLLM’s potential in generating more diverse responses than LLMs due to the random-order nature (Gong et al., 2025).

To jointly address these challenges, we propose **Distribution Matching Policy Optimization (DMPO)**, a principled and efficient RL fine-tuning method specifically designed for dLLMs. DMPO is designed based on a novel framework theoretically grounded in stochastic optimal control (SOC), which shifts away from the conventional reward maximization paradigm and targets a new goal of matching the entire reward-tilted policy distribution. This enables the model to explore diverse, high-quality reasoning paths and responses during training, addressing concerns about over-focusing on absolute reward values and modes. Moreover, DMPO training leverages importance sampling and a novel weighted denoising cross-entropy (WDCE) loss, which enjoys the key advantage of operating in an *off-policy* manner, allowing the use of replay buffers for improved sample efficiency. More importantly, WDCE is a *forward-only* objective that relies solely on the obtained clean samples and the inexpensive, forward-noising process unique to diffusion LLMs. DMPO largely discards the dependence on rollout trajectories, enabling it to potentially enjoy more speed-up than other dLLM RL algorithms when employed with fast inference techniques.

**Contributions.** The core contributions of this paper are summarized as follows: **(I)** We propose a novel RL learning framework for dLLMs that targets distribution matching rather than reward maximization (Sec. 3.1). **(II)** We propose Distribution Matching Policy Optimization (DMPO), a principled, theoretically-grounded fine-tuning strategy for enhancing dLLM’s reasoning capabilities, supported by importance sampling and weighted denoising cross-entropy (Sec. 3.2). **(III)** We identify a special challenge that occurred for WDCE due to the use of a limited training batch size, and propose two novel techniques to address it: weight baseline subtraction (Sec. 3.3) and weighted direct discriminative optimization (Sec. 3.4). **(IV)** DMPO exhibits superior performances on multiple reasoning benchmarks without supervised fine-tuning (SFT), with an accuracy improvement up to **54.3%** over previously SOTA baselines and **66.41%** over the base model, being top-performing across bi-directional dLLMs (Sec. 4).

## 2 PRELIMINARIES

### 2.1 MASKED DIFFUSION MODELS FOR LANGUAGE MODELING

The **masked (discrete) diffusion models (MDM)** (Lou et al., 2024; Ou et al., 2025; Sahoo et al., 2024; Shi et al., 2024; Zheng et al., 2025f) is a novel method for learning high-dimensional categorical distributions with application to text (Nie et al., 2025b), images (Chang et al., 2022; Bai et al., 2025), DNAs (Hayes et al., 2025), etc. Essentially, it learns the one-dimensional conditional distributions of the data given any subset of observed dimensions. Suppose the data are finite-length sequences with vocabulary  $\mathcal{V} = \{1, 2, \dots, V\}$ . Include the mask token  $M$  into the  $\mathcal{V}$  and let  $\bar{\mathcal{V}} = \{1, 2, \dots, V, M\}$ .

The MDM takes a partially masked sequence  $\mathbf{x} = (x_1, \dots, x_D) \in \bar{\mathcal{V}}^D$  as an input, and outputs  $\pi_\theta(\mathbf{x}) \in \mathbb{R}^{D \times V}$ , whose  $(d, u)$ -th entry  $\pi_\theta(\mathbf{x})_{d,u}$  is set to  $1_{x_d=u}$  if  $x_d \neq M$ , and if  $x_d = M$ , is trained to approximate the conditional probability

$$\Pr_{\mathbf{x} \sim p_{\text{data}}} (X_d = u | \mathbf{X}_{\text{UM}} = \mathbf{x}_{\text{UM}}), \quad \text{where } \mathbf{x}_{\text{UM}} = (x_d : x_d \neq M).$$

By definition, we assume each row of  $\pi_\theta(\mathbf{x})$  is a valid probability vector. The probability of a unmasked sequence  $\mathbf{x} \in \mathcal{V}^D$  under the MDM  $\pi_\theta$  is defined through **random-order autoregressive (AR) generation**: choosing a uniformly random order of the  $D$  positions, and autoregressively

108 sampling each position conditional on the previously sampled ones. Formally,

$$110 \quad p_{\theta}(\mathbf{x}) = \mathbb{E}_{\sigma} p_{\theta}(\mathbf{x}; \sigma), \quad \text{where } \sigma \sim \text{Unif}(S_D) \text{ and } p_{\theta}(\mathbf{x}; \sigma) = \prod_{d=1}^D \pi_{\theta}(x_{\sigma_d} | \mathbf{x}_{\sigma_{<d}}). \quad (1)$$

113 Here,  $S_D$  is the set of all permutations of  $\{1, \dots, D\}$ ;  $\pi_{\theta}(x_{\sigma_d} | \mathbf{x}_{\sigma_{<d}})$  means input  $\mathbf{x}$  with all positions  
114 except  $\sigma_{<d} = \{\sigma_1, \dots, \sigma_{d-1}\}$  masked into the MDM and take the output at position  $(\sigma_d, x_{\sigma_d})$ .  
115

116 The standard way to train a masked discrete diffusion model given i.i.d. samples from  $p_{\text{data}}$  is to  
117 minimize the **denoising cross-entropy (DCE)** loss  $\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathcal{L}_{\theta}(\mathbf{x})$ , which involves the following  
118 definition of the (negative) **evidence lower bound (ELBO)**  $\mathcal{L}_{\theta}$ :

$$119 \quad -\log p_{\theta}(\mathbf{x}) = -\log \mathbb{E}_{\sigma} p_{\theta}(\mathbf{x}; \sigma) \leq -\mathbb{E}_{\sigma} \log p_{\theta}(\mathbf{x}; \sigma) \quad (\text{Jensen's inequality})$$

$$120 \quad = \mathbb{E}_{m \sim \text{Unif}\{1, \dots, |\mathbf{x}|\}} \left[ \frac{|\mathbf{x}|}{m} \mathbb{E}_{\mu_m(\tilde{\mathbf{x}} | \mathbf{x})} \sum_{d: \tilde{x}_d = \mathbf{M}} -\log \pi_{\theta}(\tilde{\mathbf{x}})_{d, x_d} \right] =: \mathcal{L}_{\theta}(\mathbf{x}), \quad (2)$$

123 where the transition distribution  $\mu_m(\cdot | \mathbf{x})$  means to sample a uniformly random subset of  $\{1, \dots, |\mathbf{x}|\}$   
124 of size  $m$  and mask the corresponding entries in  $\mathbf{x}$ , and  $|\mathbf{x}|$  is the length of  $\mathbf{x}$ . The proof of the last  
125 equation can be found in Uria et al. (2016); Ou et al. (2025).

127 When applying to text data, the MDM is also referred to as the **diffusion large language model**  
128 (**dLLM**) (Nie et al., 2025b; Ye et al., 2025; Inception Labs et al., 2025; Song et al., 2025). For the  
129 purpose of reasoning, we typically write  $\mathbf{x} = (\mathbf{q}, \mathbf{o})$ , where  $\mathbf{q}$  is the **prompt** (or query, which is always  
130 assumed to contain no mask state) and  $\mathbf{o}$  is the **response** (or output). We use  $\pi_{\theta}(\mathbf{o} | \mathbf{q}) \in \mathbb{R}^{|\mathbf{o}| \times V}$   
131 to denote the policy model output of the dLLM given a prompt  $\mathbf{q}$  and a partially masked response  
132  $\mathbf{o}$ . The conditional sequence probability of a clean model  $\mathbf{o}$  given a prompt  $\mathbf{q}$ , denoted as  $p_{\theta}(\mathbf{o} | \mathbf{q})$ ,  
133 is similarly defined through (1), where we now use notations  $p_{\theta}(\mathbf{o} | \mathbf{q}; \sigma)$  and  $\pi_{\theta}(o_{\sigma_d} | \mathbf{q}, \mathbf{o}_{\sigma_{<d}})$  to  
134 emphasize the dependence on the prompt  $\mathbf{q}$ . The negative ELBO will be written as  $\mathcal{L}_{\theta}(\mathbf{o} | \mathbf{q})$ .

## 135 2.2 REINFORCEMENT LEARNING FOR ENHANCING REASONING

137 We first present the **Group Relative Policy Optimization (GRPO**, Shao et al. (2024)) method for  
138 LLMs, which is the basis of most of the existing RL methods for dLLMs. Given a pretrained LLM  
139 with policy  $\pi_{\text{ref}}$  that samples from the distribution  $p_{\text{ref}}(\mathbf{o} | \mathbf{q}) = \prod_{d=1}^{|\mathbf{o}|} \pi_{\text{ref}}(o_d | \mathbf{q}, \mathbf{o}_{<d})$ , a reward  
140 function  $r : (\mathbf{q}, \mathbf{o}) \mapsto \mathbb{R}$ , a set of prompts  $\mathcal{D}$ , and a regularization parameter  $\alpha \geq 0$ , each step of the  
141 GRPO aims to solve the following problem: sample  $\mathbf{q} \sim \mathcal{D}$ ,  $\mathbf{o}^{(1:G)} \stackrel{\text{i.i.d.}}{\sim} p_{\theta_{\text{old}}}(\mathbf{o} | \mathbf{q})$ , and maximize

$$144 \quad \mathbb{E} \left\{ \frac{1}{G} \sum_{i=1}^G \frac{1}{|\mathbf{o}^{(i)}|} \sum_{d=1}^{|\mathbf{o}^{(i)}|} \left[ \min \left( \rho_d^{(i)} A_i, \text{clip}(\rho_d^{(i)})_{1 \pm \epsilon} A_i \right) - \alpha \text{KL}(p_{\theta}(\mathbf{o}^{(i)} | \mathbf{q}) \| p_{\text{ref}}(\mathbf{o}^{(i)} | \mathbf{q})) \right] \right\}, \quad (3)$$

147 where the advantages<sup>1</sup> are  $A_i = r(\mathbf{q}, \mathbf{o}^{(i)}) - \text{mean}(r(\mathbf{q}, \mathbf{o}^{(1:G)}))$ , the per-token probability ratios  
148 are  $\rho_d^{(i)} = \frac{\pi_{\theta}(o_d^{(i)} | \mathbf{q}, \mathbf{o}_{<d}^{(i)})}{\pi_{\theta_{\text{old}}}(o_d^{(i)} | \mathbf{q}, \mathbf{o}_{<d}^{(i)})}$ , and the KL regularization term is estimated similarly by the per-token  
149 probability ratios between  $\pi_{\theta}$  and  $\pi_{\text{ref}}$ . The clipping threshold  $\epsilon$  prevents overly large policy updates.  
150

151 While (3) works well for LLMs, it is not directly applicable to dLLMs due to mismatch between  
152 the *dLLM policy (model output)*  $\pi_{\theta}(\mathbf{o} | \mathbf{q})$  and the *sequence likelihood*  $p_{\theta}(\mathbf{o} | \mathbf{q})$ : unlike in LLMs  
153 where these two quantities are easily connected through the chain rule, it is generally non-trivial to  
154 compute the per-token probability given the dLLM model output, and only ELBO (2) is available  
155 as a surrogate. To tackle this issue, diffu-GRPO (Zhao et al., 2025a) proposed to **fully mask all**  
156 **response positions** and partially masks the prompt  $\mathbf{q}$ , and feed this sequence into the model to obtain  
157 the approximate probability  $p_{\theta}(o_d | \mathbf{q})$ . Next, the sequence probability  $p_{\theta}(\mathbf{o} | \mathbf{q})$  is approximated by  
158 mean-field decomposition:  $p_{\theta}(\mathbf{o} | \mathbf{q}) \approx \prod_{d=1}^{|\mathbf{o}|} p_{\theta}(o_d | \mathbf{q})$ . Such approximations do not capture the  
159 correlation between different positions in the response, which produces imprecision. A similar  
160 technique is employed in coupled-GRPO (cGRPO) for code generation tasks in Gong et al. (2025).  
161

<sup>1</sup>As suggested by Liu et al. (2025c), we list here the version without normalization by standard deviation.

### 162 3 DISTRIBUTION MATCHING POLICY OPTIMIZATION

#### 164 3.1 FROM REWARD MAXIMIZATION TO DISTRIBUTION MATCHING

166 To incentivize the reasoning capabilities of large language models,  
 167 reward-maximizing reinforcement learning finetuning algorithms,  
 168 such as TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017),  
 169 and GRPO (Shao et al., 2024), are often employed, with an additional  
 170 entropy regularization term that penalizes the deviation of the  
 171 model from the pretrained one. This process amounts to solving the  
 172 following optimization problem,

$$173 \max_{\theta} \mathbb{E}_{q \sim \mathcal{D}} [\mathbb{E}_{p_{\theta}(o|q)}[r(q, o)] - \alpha \text{KL}(p_{\theta}(\cdot|q) \| p_{\text{ref}}(\cdot|q))] . \quad (4)$$

174 However, existing techniques over-focus on finding and optimizing the **reward mode** and adopt many heuristic techniques to accelerate the mode searching process, neglecting the exploration of the entire distribution landscape, and often result in model mode collapse or reward hacking, causing the model to produce undesirable responses (Weng, 2024). A simple fix to this issue and to encourage diverse model responses is to enforce the optimality of the target policy distribution during the training. It can be shown that the optimal sequence distribution that solves the problem (4) is the following **reward-tilted distribution**:

$$183 p_{*}(o|q) = \frac{1}{Z(q)} p_{\text{ref}}(o|q) e^{r(q, o)/\alpha}, \quad \text{where } Z(q) = \sum_{o} p_{\text{ref}}(o|q) e^{r(q, o)/\alpha}. \quad (5)$$

185 That is to say, we want to use the optimal sequential distribution  $p_{*}(o|q)$  as the **supervision signal** throughout the learning process, so that we can learn a dLLM policy  $\pi_{\theta}$  which produces a sequence distribution  $p_{\theta}$  matching  $p_{*}$ . We can thus obtain a policy that not only explores the dominant reward mode, but is guaranteed to sample other high-reward trajectories with a likelihood proportional to the reward value. This motivates us to consider the following task of **policy distribution matching**,

191 **Policy Distribution Matching Learning:** Given a pretrained dLLM policy  $\pi_{\text{ref}}(o|q)$  that samples  
 192 from a distribution  $p_{\text{ref}}(o|q)$ , a reward function  $r : (q, o) \mapsto \mathbb{R}$ , a set of prompts  $\mathcal{D}$ , and  
 193 temperature  $\alpha > 0$ , learn a dLLM policy  $\pi_{\theta}(o|q)$  to produce the desired optimal sequence  
 194 distribution  $p_{*}(o|q)$  in (5) by optimizing the following objective:

$$195 \min_{\pi_{\theta}} \mathbb{E}_{q \sim \mathcal{D}} \mathcal{F}(p_{\theta}(\cdot|q), p_{*}(\cdot|q)). \quad (6)$$

197 Here,  $\mathcal{F}$  is a class of functionals such that  $\text{argmin}_p \mathcal{F}(p, p_{*}) = p_{*}$ . Note that the original entropy-  
 198 regularized entropy optimization problem is equivalent to choosing  $\mathcal{F}$  to be the reverse KL between  
 199  $p$  and  $p_{*}$ , i.e.,  $\mathcal{F}(p_{\theta}, p_{*}) = \text{KL}(p_{\theta} \| p_{*}) = \mathbb{E}_{p_{\theta}} [\log \frac{p_{\theta}}{p_{*}}]$ . While this objective in theory can also lead  
 200 to the same optimal distribution with the desired property, it is widely known that reverse KL is  
 201 *mode-seeking*, i.e., it tends to match the most dominant mode in  $p_{*}$  while potentially neglecting other  
 202 modes, which may lead to reward hacking.

204 To address this issue, we consider a series of new objectives  $\mathcal{F}$  with more desirable convergence  
 205 guarantees that steadily lead to optimization towards the desired sequence distribution, and propose  
 206 **Distribution Matching Policy Optimization (DMPO)** (Alg. 1), which targets matching the entire  
 207 reward-tilted policy distribution. In Sec. 3.2, we introduce **weighted denoising cross-entropy**  
 208 (WDCE), a **scalable** implementation of the forward KL using importance sampling. In Secs. 3.3  
 209 and 3.4, we discuss an important failure case of forward KL with **small training batch size**, and  
 210 propose a series of novel techniques such as **weight baseline subtraction** (Sec. 3.3) and **weighted**  
 211 **direct discriminative optimization** (Sec. 3.4) to address it.

#### 212 3.2 WEIGHTED DENOISING CROSS-ENTROPY

214 Unlike the reverse KL objective considered by many existing works, which are known to be prone  
 215 to mode seeking and collapse, one alternative choice is to use the forward KL divergence (or **cross-  
 216 entropy**, CE) for the functional, i.e.,  $\mathcal{F}(p_{\theta}, p_{*}) = \text{KL}(p_{*} \| p_{\theta})$ , which tends to cover all the modes

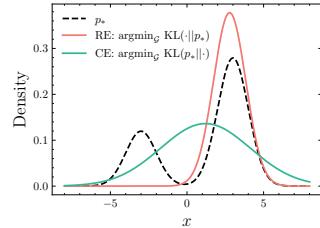


Figure 2: Illustration of relative entropy (mode-seeking) and cross-entropy (mass-covering) for fitting a target  $p_{*}$  ( $\mathcal{G}$  is the set of Gaussian distributions).

---

216 **Algorithm 1** Distribution Matching Policy Optimization (DMPO)

---

217 **Require:** Training dataset  $\mathcal{D}$ , number of prompts per batch  $B$ , number of rollouts per prompt  $N$ ,  
218 frequency for sampling buffer  $F$ , model policy  $\pi_\theta$ .

219 1: **for**  $\text{step} = 0, 1, 2, \dots$  **do**

220 2:   **if**  $\text{step} \bmod F = 0$  **then**           ▷ Prepare the buffer using the current policy, denoted  $\pi_v$ .

221 3:     Sample  $B$  prompts  $\{\mathbf{q}^{(i)}\}_{1 \leq i \leq B}$  from the dataset  $\mathcal{D}$ .

222 4:     **for**  $1 \leq i \leq B$  (in parallel, with gradient computation disabled) **do**

223 5:       Sample  $N$  orders and generate  $N$  rollouts  $\{\mathbf{o}^{(i,n)}\}_{1 \leq n \leq N}$  conditional on prompt  $\mathbf{q}^{(i)}$ .

224 6:       Evaluate reward and compute weights  $w(\mathbf{o}^{(i,n)} | \mathbf{q}^{(i)}; \sigma^{(i,n)})$  according to (10).

225 7:       Compute the weight baseline according to (13), (14), or (15), and obtain the real  
226 weights  $w_{\text{real}}(\mathbf{o}^{(i,n)} | \mathbf{q}^{(i)}; \sigma^{(i,n)})$  according to (12).

227 8:     For each  $\mathbf{o}^{(i,n)}$ , sample a mask assignment and obtain  $\tilde{\mathbf{o}}^{(i,n)}$ .

228 9:     Feed all pairs of  $(\mathbf{q}^{(i)}, \tilde{\mathbf{o}}^{(i,n)})$  into  $\pi_\theta$  and compute the WDCE loss (11), then update  $\theta$ .

229 **return**  $\pi_\theta$

---

231  
232 of the optimal distribution and can retain the response diversity. The CE loss is also widely used in  
233 another domain known as stochastic optimal control (SOC) (Domingo-Enrich et al., 2024; 2025),  
234 which is also closely connected with our work. This amounts to solving the following task,

235 
$$\min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}} \mathbb{E}_{p_*(\mathbf{o}|\mathbf{q})} \left[ \log \frac{p_*(\mathbf{o}|\mathbf{q})}{p_\theta(\mathbf{o}|\mathbf{q})} \right]. \quad (7)$$

236 However, objective (7) is not directly amenable to practical implementation, as we do not have  
237 access to real samples from the  $p_*$ , nor can we exactly compute  $\log p_*$  due to the presence of the  
238 unknown partition function  $Z(\mathbf{q})$ . To bypass this issue, we draw inspiration from the recent work  
239 masked diffusion neural sampler (MDNS, Zhu et al. (2025g)), which proposes a training framework  
240 for learning a masked diffusion neural sampler with stochastic optimal control and cross-entropy  
241 minimization. While targeting a different task, the core of MDNS resides in solving the same  
242 distribution matching problem with cross-entropy loss, and it proposes a practically implementable  
243 and scalable variant of (7), named **weighted denoising cross-entropy (WDCE)** loss. The central  
244 idea is to introduce a reference policy and leverage *importance sampling*, so that we can treat i.i.d.  
245 samples as importance-weighted samples from  $p_*$ . Taking advantage of this approach, we now derive  
246 WDCE for the purpose of dLLM policy learning.

247 First, given the relationship between the policy output and sequence distribution of the masked dLLM  
248 (1), it is clear that we can match the correct target sequence distribution  $p_*(\mathbf{o}|\mathbf{q})$  as long as we train  
249  $p_\theta(\mathbf{o}|\mathbf{q}; \sigma)$  to match the *order-specific* ones, i.e.,  $p_*(\mathbf{o}|\mathbf{q}; \sigma)$ , given by

250 
$$p_*(\mathbf{o}|\mathbf{q}; \sigma) = \frac{1}{Z(\mathbf{q})} p_{\text{ref}}(\mathbf{o}|\mathbf{q}; \sigma) e^{r(\mathbf{q}, \mathbf{o})/\alpha}. \quad (8)$$

251 Leveraging this fact, given any prompt  $\mathbf{q}$ , we can express the cross-entropy loss as follows:

252 
$$\begin{aligned} \text{KL}(p_*(\cdot|\mathbf{q}) \| p_\theta(\cdot|\mathbf{q})) &= \mathbb{E}_{p_*(\mathbf{o}|\mathbf{q})} [-\log p_\theta(\mathbf{o}|\mathbf{q})] + \text{const} \\ &= \mathbb{E}_{\sigma} \mathbb{E}_{p_*(\mathbf{o}|\mathbf{q}; \sigma)} [-\log p_\theta(\mathbf{o}|\mathbf{q})] + \text{const} \\ &= \mathbb{E}_{\sigma} \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q}; \sigma)} \frac{p_*(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)} [-\log p_\theta(\mathbf{o}|\mathbf{q})] + \text{const}, \end{aligned} \quad (9)$$

253 where  $p_v$  is the sequence probability under a reference policy model  $v$  that does not involve gradient  
254 computation, and in practice, one often chooses  $v \leftarrow \theta := \text{stopgrad}(\theta)$  to be a copy of the policy  
255 model detached from the computation graph, and periodically synchronizes with the current model  
256 policy  $p_\theta$ , which is also commonly referred to as  $p_{\theta_{\text{old}}}$  in the literature. The importance weight  
257  $w(\mathbf{o}|\mathbf{q}; \sigma) := \frac{p_*(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)}$  captures the mismatch between  $p_v$  and  $p_*$  and ensures the mathematical  
258 correctness of the objective, and  $\log p_\theta(\mathbf{o}|\mathbf{q})$  is an intractable sequence log probability under the  
259 current dLLM policy. We discuss the computation of these two components in parallel below.

260 **Importance weight  $w(\mathbf{o}|\mathbf{q}; \sigma)$ .** We simplify it with the pretrained model and the reward:

261 
$$w(\mathbf{o}|\mathbf{q}; \sigma) = \frac{1}{Z(\mathbf{q})} \frac{p_{\text{ref}}(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)} e^{\frac{r(\mathbf{q}, \mathbf{o})}{\alpha}} \propto \exp \left( \frac{r(\mathbf{q}, \mathbf{o})}{\alpha} + \log \frac{p_{\text{ref}}(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)} \right) =: e^{\ell(\mathbf{o}|\mathbf{q}; \sigma)}. \quad (10)$$

270 Recall that the order-specific probability of a sequence is computed via (1). To ensure that the sample  
 271 distribution after importance sampling is valid and normalized, we keep track of the **log weights**  
 272  $\ell(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})$ , and taking softmax among those corresponding to the same prompt  $\mathbf{q}$  to compute the real  
 273 weight  $w(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})$ . This is equivalent to estimating the unknown partition function  $Z(\mathbf{q})$  using an  
 274 empirical estimator of the following expectation:

$$276 \quad Z(\mathbf{q}) = \mathbb{E}_{\boldsymbol{\sigma}} \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})} \left[ \frac{p_{\text{ref}}(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})}{p_v(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})} e^{r(\mathbf{q}, \mathbf{o})/\alpha} \right],$$

279 The need to estimate partition functions is common in RL algorithms for LLM, such as in GflowNet  
 280 (Bengio et al., 2021; Kimi Team et al., 2025). In contrast to these approaches that learn such functions  
 281 independently, our estimation approach is training-free and more efficient.

282 **Sequence log probability**  $\log p_v(\mathbf{o}|\mathbf{q})$ . Unlike the case of LLM, the exact sequence log probability  
 283 is intractable due to the presence of expectation over the random order  $\boldsymbol{\sigma}$ . However, similar to  
 284 the training of dLLM, we can leverage the negative ELBO (2) as a surrogate. Combined with the  
 285 importance weight  $w(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})$ , we introduce the **weighted denoising cross-entropy (WDCE)** loss  
 286 for dLLM policy distribution matching:

$$288 \quad \min_{\theta} \mathbb{E}_{\mathbf{q} \sim \mathcal{D}} \mathbb{E}_{\boldsymbol{\sigma}} \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})} \left\{ w(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma}) \mathbb{E}_{m \sim \text{Unif}\{1, \dots, |\mathbf{o}|\}} \left[ \frac{|\mathbf{o}|}{m} \mathbb{E}_{\mu_m(\tilde{\mathbf{o}}|\mathbf{o})} \sum_{d: \tilde{o}_d = M} -\log \pi_{\theta}(\tilde{\mathbf{o}}|\mathbf{q})_{d, o_d} \right] \right\}. \quad (11)$$

291 Notably, this loss highly resembles the DCE loss used in pre-training and the supervised fine-tuning  
 292 (SFT) phase of dLLM. One major difference is that instead of using i.i.d. samples from  $p_*$ , we  
 293 use importance sampling to weight samples from  $p_v$  and obtain a valid training objective with  
 294 theoretical guarantees. WDCE differs significantly from other popular RL training techniques such  
 295 as PPO/GRPO in two key aspects.

296 **WDCE is an off-policy loss.** The WDCE loss remains valid as the model parameter  $\theta$  gets updated,  
 297 since both the sampling policy  $p_v$  and the important sampling target policy  $p_*$  are independent of the  
 298 current model policy  $p_{\theta}$ . This allows us to save generated rollouts in a replay buffer and reuse them  
 299 for multiple training updates, without worrying excessively about numerical instability, leading to  
 300 improved sample efficiency. On the other hand, for on-policy methods, to use a replay buffer, one  
 301 would need to estimate importance weights with respect to the current model policy  $p_{\theta}(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})$ , i.e.,  
 302  $\frac{p_{\theta}(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})}{p_v(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})}$ . Different from the case of LLM, where such estimation can be done in one model forward  
 303 pass, an accurate estimation in dLLM **per training update** is expensive, rendering the on-policy  
 304 method less efficient. Moreover, for large models, where rollout generation and sequence likelihood  
 305 estimation are typically handled by different implementations (such as vLLM and FSDP), this could  
 306 lead to more nuanced, hard-to-detect biases that secretly undermine the algorithm’s performance  
 307 (Yao et al., 2025). With WDCE, we are largely free of such concerns.

308 **WDCE is a forward loss.** Unlike the GRPO-style of algorithms that typically require keeping track  
 309 of the entire rollout trajectories, WDCE leverages the forward noising process in training, which is  
 310 a characteristic unique to **diffusion** LLMs. Once we obtain the final samples and their associated  
 311 weights, we can discard the trajectories and perform training using the cheap forward process by  
 312 randomly masking the data. This implies that the training speed when using WDCE largely depends  
 313 on the model inference speed. With the advances of dLLM efficient inference techniques such as  
 314 fast decoding algorithms and KV-cache techniques (Ma et al., 2025; Hu et al., 2025; Wu et al., 2025;  
 315 Liu et al., 2025b), WDCE could also enjoy a great boost in efficiency. This method also effectively  
 316 utilizes dLLM’s potential in surpassing LLMs in inference throughput, distinguishing it from other  
 317 RL baselines that merely adapt LLM algorithms to dLLM. We defer a more detailed discussion of  
 318 such properties to App. B.2.

319 **Finally, we remark that while we developed the WDCE loss through the lens of policy distribution**  
 320 **matching learning (6), it can also be derived through the perspective of stochastic optimal control**  
 321 **(SOC)** as in Wang et al. (2025a); Zhu et al. (2025g), which we detail in App. B.1. In particular,  
 322 we emphasize that the ELBO approximation of *sequence-level* log probabilities in matching KL  
 323 divergence in (9) is equivalent to precisely matching *path-level* probabilities in the SOC framework,  
 justifying the validity of such a heuristic approximation.

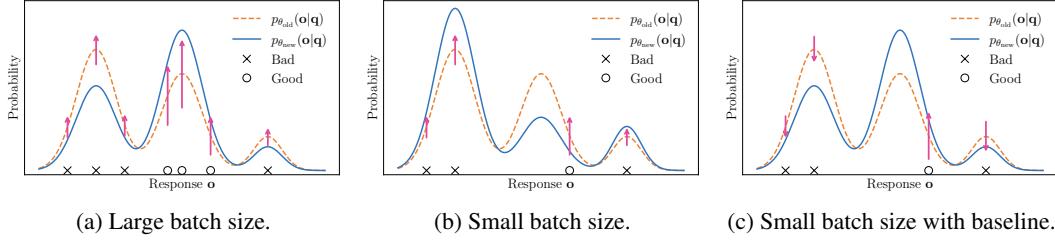


Figure 3: Demonstration of the effect of weight baseline. The orange and blue curves represent the probability  $p_\theta(o|q)$  before and after update, and the magenta arrows represent the weights. (a) When batch size is large, distribution mode coverage is good. Though bad responses have positive weights, the correct ones will have larger weights to force the distribution updates towards the right direction. (b) When batch size is small, some modes (e.g., the good one in the middle) may not be sampled. Without **weight baseline subtraction**, the dominant positive weights of the bad responses lead to wrong update directions. (c) With **weight baseline subtraction**, the bad responses will appropriately be penalized, leading to the desired update direction.

### 3.3 EFFECTIVE TRAINING WITH NEGATIVE GRADIENT INSERTION

While theoretically, minimizing the WDCE loss (11) probably leads to convergence of the model sequence distribution to  $p_*(o|q)$ , this could face practicality issues in real implementation due to the often limited number of rollouts generated per prompt. Ideally, we would want to promote the likelihood of “good” responses while decreasing those of “bad” responses. However, with WDCE, any response  $o$  will be associated with a positive weight  $w(o|q; \sigma)$  due to the softmax operation, which may lead to ineffective learning in the low-batch-size scenario.

We note that this issue does not arise when the batch size is sufficiently large for the following reason. When having a large batch of diverse responses that make up a good coverage of the sample space, despite having all positive weights, since the model cannot increase likelihood on all responses (as it is a probability distribution that sums up to 1), the “bad” responses will be automatically and implicitly penalized due to not having larger weights than the “good” responses.

When the batch size is small, the scenario is different as is illustrated in Fig. 3. In such a case, the model will tend to promote **both “good” and “bad” responses** due to the positive weights, and potentially penalize the likelihood of other unseen responses to maintain a valid probability distribution. This could be detrimental to achieving distribution matching, as these unseen responses may have high reward values and correspond to an undiscovered distribution mode.

To address this issue, we inject negative gradient (Ren & Sutherland, 2025; Deng et al., 2025) by designing a **weight baseline** and subtract it from the obtained weights to facilitate an effective reinforcement on the good samples, i.e.,

$$w_{\text{real}}(o|q; \sigma) = w(o|q; \sigma) - w_{\text{base}}(o|q; \sigma). \quad (12)$$

This approach resembles that adopted by PPO/GRPO. However, distinct from these methods, we rate responses based on the log weights  $\ell(o|q; \sigma)$ , whose larger values indicate a better alignment with the target optimal distribution. As a consequence, we promote responses that are more likely to be sampled from  $p_*$  and penalize those that are less likely. Based on this perspective, we consider the following three methods for choosing  $w_{\text{base}}(o|q; \sigma)$ .

**Group weight baseline.** When the dLLM policy is close to optimal, the original log weight  $\ell(o|q; \sigma)$  should behave approximately like constants for a group of different responses  $\{o^{(n)}\}_{1 \leq n \leq N}$ , leading to nearly uniform weight value for  $\{w(o^{(n)}|q; \sigma^{(n)})\}_{1 \leq n \leq N}$  after group softmax. We can thus choose the baseline as 1 to encourage convergence to this optimal situation:

$$w_{\text{base}}(o^{(n)}|q; \sigma^{(n)}) = 1, \forall n. \quad (13)$$

**Individual weight baseline.** We can also consider the individual weight value of each response. For samples with smaller log weights, a stronger penalization is more desirable. A natural, adaptive way of designing penalization strength is to use softmax over the log weights with *negative* reward: let  $\ell_-(o|q; \sigma) := -\frac{r(q, o)}{\alpha} + \log \frac{p_{\text{ref}}(o|q; \sigma)}{p_v(o|q; \sigma)}$ , and define

$$w_{\text{base}}(o^{(n)}|q; \sigma^{(n)}) = \frac{N \exp(\ell_-(o^{(n)}|q; \sigma^{(n)}))}{\sum_k \exp(\ell_-(o^{(k)}|q; \sigma^{(k)}))}, \forall n. \quad (14)$$

Table 1: Model performances on reasoning benchmarks. **best** and second best results are highlighted. DMPO consistently outperforms other baselines across different generation length.

Task Sequence Length	GSM8K			MATH500			Countdown			Sudoku		
	128	256	512	128	256	512	128	256	512	128	256	512
Dream-Instruct (7B)	56.63	73.39	76.65	31.00	36.60	36.40	22.66	28.52	27.34	14.45	16.41	11.77
LLaDA-Instruct (8B)	71.87	79.76	83.62	28.20	35.00	38.80	23.44	14.45	14.84	12.94	6.10	7.37
LLaDA-1.5 (8B)	73.09	80.97	84.38	26.80	33.80	40.00	26.17	16.41	23.83	15.19	13.04	8.98
d1-LLaDA	75.28	81.40	84.38	30.00	36.60	40.80	34.38	26.56	30.47	21.97	11.04	8.69
cGRPO-LLaDA	67.40	81.73	84.23	21.40	32.80	38.40	30.08	42.58	37.11	24.17	24.17	21.97
<b>DMPO-LLaDA (Ours)</b>	74.83	82.41	<b>85.22</b>	30.00	<b>38.20</b>	<b>42.80</b>	<b>67.19</b>	<b>80.86</b>	82.81	<b>32.76</b>	24.56	19.97
<b>DMPO-LLaDA-SFT (Ours)</b>	<b>80.06</b>	<b>84.00</b>	84.09	<b>31.80</b>	<b>40.00</b>	41.20	54.69	67.19	77.34	25.20	<b>25.73</b>	<b>23.78</b>
<b>DMPO-LLaDA-1.5 (Ours)</b>	77.56	82.71	84.61	30.20	36.60	41.00	59.77	79.30	<b>83.20</b>	25.34	24.51	23.34

Note that this  $w_{\text{base}}(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})$  now corresponds to a bad target distribution given by  $p_{*-}(\mathbf{o}|\mathbf{q}) \propto p_{\text{ref}}(\mathbf{o}|\mathbf{q})e^{-r(\mathbf{q}, \mathbf{o})/\alpha}$  which is tilted by the negative reward. The minus sign in the loss before  $w_{\text{base}}(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma})$  means we want to drive the dLLM policy away from this bad distribution.

**Model weight baseline.** Finally, we can determine whether to promote or penalize specific responses by comparing  $w(o|q; \sigma)$  with the importance weight under the current model policy  $p_\theta(o|q)$ . This pushes the model further towards the optimal one  $p_*(o|q)$ . Note that this does not incur additional computation overhead as we can estimate  $\log p_{\bar{\theta}}(o|q)$  using negative ELBO (2), which is already computed in the WDCE loss. Let  $\ell_\theta(o|q; \sigma) := \log \frac{p_{\bar{\theta}}(o|q; \sigma)}{p_v(o|q; \sigma)}$ , and define

$$w_{\text{base}}(\mathbf{o}^{(n)} | \mathbf{q}; \boldsymbol{\sigma}^{(n)}) = \frac{N \exp(\ell_{\theta}(\mathbf{o}^{(n)} | \mathbf{q}; \boldsymbol{\sigma}^{(n)}))}{\sum_k \exp(\ell_{\theta}(\mathbf{o}^{(k)} | \mathbf{q}; \boldsymbol{\sigma}^{(k)}))}, \quad \forall n. \quad (15)$$

We remark that the group weight and model weight baselines (13) and (15) can also be interpreted as an *approximate variance reduction*. See App. B.3 for discussion.

### 3.4 WEIGHTED DIRECT DISCRIMINATIVE OPTIMIZATION

To explore the full potential of the distribution matching framework in (6), we also investigate other choices for the potential  $\mathcal{F}$  other than the cross-entropy. One particularly interesting objective is the following **direct discriminative optimization (DDO)** loss.

$$\mathcal{F}(p_\theta(\cdot|\mathbf{q}), p_*(\cdot|\mathbf{q})) = -\mathbb{E}_{p_*(\mathbf{o}|\mathbf{q})} \log \sigma \left( \log \frac{p_\theta(\mathbf{o}|\mathbf{q})}{p_*(\mathbf{o}|\mathbf{q})} \right) - \mathbb{E}_{p_*(\mathbf{o}|\mathbf{q})} \log \sigma \left( -\log \frac{p_\theta(\mathbf{o}|\mathbf{q})}{p_*(\mathbf{o}|\mathbf{q})} \right), \quad (16)$$

where  $\sigma(t) = 1/(1 + e^{-t})$ . The global optimum of (16) is also  $p_*(\cdot | \mathbf{q})$ , thus being a valid functional for distribution matching learning. For a more detailed justification, see App. B.4.

This is inspired by Zheng et al. (2025e), which proposes a GAN-like (Goodfellow et al., 2014) training loss for the SFT of vision models. One interesting trait of this objective is its natural incorporation of negative gradients for bad samples due to the GAN nature, as is shown in the analysis therein:

$$\nabla_{\theta} \mathcal{F}(p_{\theta}(\cdot|\mathbf{q}), p_{*}(\cdot|\mathbf{q})) = \sum \sigma \left( -\log \frac{p_{\theta}(\mathbf{o}|\mathbf{q})}{p_{*}(\mathbf{o}|\mathbf{q})} \right) (p_{\theta}(\mathbf{o}|\mathbf{q}) - p_{*}(\mathbf{o}|\mathbf{q})) \nabla_{\theta} \log p_{\theta}(\mathbf{o}|\mathbf{q}).$$

From the expression, as the first term is always non-negative, and the middle term  $p_\theta(o|q) - p_*(o|q)$  applies a penalty for bad response  $o$ , thus providing a gradient direction for increasing  $p_\theta(o|q)$ . Leveraging this property, we adapt it for RL finetuning of dLLM and introduce the **weighted direct discriminative optimization (WDDO)** loss, again through importance sampling to represent  $p_*(o|q)$ .

$$\mathcal{F}(p_\theta(\cdot|\mathbf{q}), p_*(\cdot|\mathbf{q})) = -\mathbb{E}_{\boldsymbol{\sigma} \sim p_{\boldsymbol{\theta}}(\mathbf{o}|\mathbf{a}, \boldsymbol{\sigma})} \left[ w(\mathbf{o}|\mathbf{q}; \boldsymbol{\sigma}) \log \sigma \left( \log \frac{p_\theta(\mathbf{o}|\mathbf{q})}{p_*(\mathbf{o}|\mathbf{q})} \right) + \log \sigma \left( -\log \frac{p_\theta(\mathbf{o}|\mathbf{q})}{p_*(\mathbf{o}|\mathbf{q})} \right) \right],$$

where  $w(\mathbf{q}|\mathbf{q}; \boldsymbol{\sigma})$  is the importance weight defined in (10).

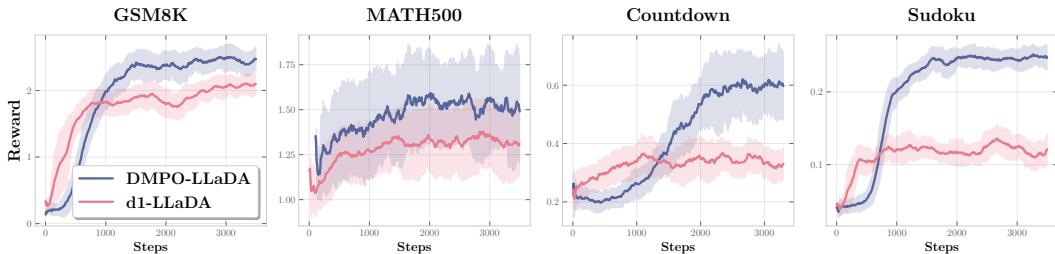


Figure 4: Reward dynamics during training. DMPO consistently produces higher rewards than d1.

## 4 EXPERIMENTS

**Model and baselines.** We apply DMPO to LLaDA-8B-Instruct (Nie et al., 2025b), a state-of-the-art open-sourced, native masked dLLM that has not been post-trained with RL techniques. To clearly demonstrate the potential of DMPO, we follow an R1-Zero-like training recipe (Guo et al., 2025a; Liu et al., 2025c) and directly apply DMPO to the LLaDA model without first performing SFT on curated datasets. We refer to the model obtained via this pipeline as **DMPO-LLaDA**. We benchmark our method against a series of top-performing dLLM base models of comparable model size, such as Dream-Instruct (7B, Ye et al. (2025)), LLaDA-Instruct (8B, Nie et al. (2025b)), and LLaDA-1.5 (8B, Zhu et al. (2025a)). Our main RL baseline is d1 (Zhao et al., 2025a), a state-of-the-art RL finetuning approach developed for dLLMs that combines both SFT and diffu-GRPO (an adapted version of GRPO). We also compare with cGRPO, which was used to fine-tune a Dream-based coding dLLM in Gong et al. (2025). In the main result table (Tab. 1), **DMPO-LLaDA** uses the group weight baseline (13) on **GSM8K**, **MATH500**, and **Sudoku**, and the individual weight baseline (14) on **Countdown**. **DMPO-LLaDA-SFT** and **DMPO-LLaDA-1.5** adopt the individual weight baseline (14) in all cases.

**Experimental setups.** We perform experiments on 4 different reasoning benchmarks: **GSM8k** (Cobbe et al., 2021), **MATH500** (Lightman et al., 2023; Hendrycks et al., 2021), **Sudoku** (Arel, 2025), and **Countdown** (Pan et al., 2025). For all pretrained dLLM models, we evaluate the latest available checkpoints for each task. For d1 and cGRPO, we reproduce their results exactly following the provided guidelines. To ensure a fair comparison, we train DMPO-LLaDA on the same datasets as d1 for each task with rollouts generated using a fixed sequence length of 256. All evaluations are conducted with zero-shot prompting using three different generation lengths: 128, 256, and 512, following a similar practice as in Zhao et al. (2025a). See App. C for more details of experiments.

**DMPO incentivizes superior reasoning capabilities.** We report in Tab. 1 the performance of DMPO together with that of the base model LLaDA-Instruct (8B), **LLaDA-1.5 (8B)**, the models obtained by d1 and **cGRPO** post-training strategies, and other pretrained dLLM models. DMPO consistently outperforms both the LLaDA-Instruct baseline and the d1/**cGRPO** models, achieving the best performance among the listed state-of-the-art dLLMs. Notably, DMPO achieves excellent gains over the LLaDA-Instruct baseline, with an accuracy improvement of an average of **+2.40%** on **GSM8K**, **+3.00%** on **MATH500**, **+59.38%** on **Countdown**, **+16.96%** on **Sudoku**. DMPO also demonstrates superior performance over d1, the current SOTA RL baseline for dLLM, especially on planning tasks, with an increase of **+46.48%** on **Countdown** and **+11.86%** on **Sudoku**. This underscores the overall effectiveness of DMPO for enhancing model reasoning capabilities.

**DMPO consistently achieves higher rewards.** In Fig. 4, we present the reward dynamics of DMPO across training steps and compare with that of d1. DMPO consistently achieves higher reward values after an initial warm-up phase and ultimately discovers responses with higher rewards than d1, possibly because it continuously explores the reward distribution landscape throughout training. In the first 1,000 steps, DMPO often produces lower reward values than d1, potentially due to the lack of an SFT phase before RL scaling. Moreover, we observe that the performance of DMPO does not saturate after 4,000 gradient steps, suggesting its greater potential than GRPO-type algorithms.

**Weight baseline subtraction is crucial for small batch size training.** We test the different choices presented for negative gradient insertion in Secs. 3.3 and 3.4 when training on the **Sudoku** dataset with a small batch size, and the result is visualized in Fig. 5. As shown by the curves, without weight baseline subtraction, the model does not improve as training progresses. All the proposed weight baselines in (13), (14), and (15) effectively increase the reward value during training. Weighted DDO achieves the fastest reward increase during the initial 1k steps but suffers from instability afterwards.

**DMPO benefits from other means of post-training techniques.** To showcase the robustness and general efficacy of DMPO, we apply it to LLaDA-SFT and LLaDA-1.5. LLaDA-SFT is obtained

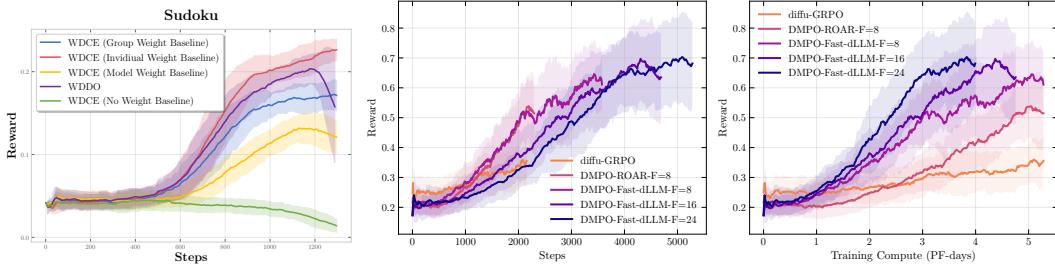


Figure 5: Effects of negative gradient insertion on Sudoku. Figure 6: Comparison of training dynamics on Countdown.  $F$  is the frequency of sampling the buffer.

by performing SFT of LLaDA-Instruct (8B) on s1k (Muennighoff et al., 2025), a dataset of 1k examples of high-quality reasoning questions with distilled reasoning traces from Gemini Thinking. LLaDA-1.5 is obtained by performing DPO on 350K preference pairs covering a wide range of topics such as writing and reasoning. We then apply DMPO to these base models to obtain DMPO-LLaDA-SFT and DMPO-LLaDA-1.5, with performance reported in Tab. 1. DMPO continues to deliver performance gains for post-trained models, with consistent and significant accuracy improvements over base models, especially at generation lengths of 128 and 256 for the math reasoning datasets, with **+4.78%** and **+2.60%** on GSM8K, **+1.80%** and **+3.40%** on MATH500 compared with d1. This underscores that DMPO is a powerful method that integrates smoothly with existing solutions.

**DMPO enables efficient and fast training.** Due to its *off-policy* and *forward* nature, DMPO achieves considerable training acceleration compared with GRPO-type methods. In Fig. 6, we compare head-to-head the training dynamics of diffu-GRPO, DMPO with random-order autoregressive (ROAR) sampler, and DMPO with Fast-dLLM (an approximate KV-cache mechanism enabled, confidence-based heuristic sampler for dLLMs, from Wu et al. (2025)) on Countdown under the same amount of training compute (measured in PF-days, where 1 PF-day =  $8.64 \times 10^{19}$  floating point operations). Due to its off-policy nature, DMPO enables heavy reuse of each sampled buffer of rollouts and achieves a sample efficiency  $2 \sim 3 \times$  that of diffu-GRPO. Regarding training-compute efficiency, as a forward-loss-based algorithm, DMPO enjoys a flexible choice of rollout sampler. With fast-dLLM, DMPO gains an acceleration of up to  $8 \times$  per rollout sampling, and achieves the same level of reward as d1 with only 31% of the training budget (1.8 PF-days v.s. 5.78 PF-days). This empirical evidence emphasizes that DMPO is not only an effective algorithm but also highly sample- and compute-efficient.

**DMPO exhibits stable training despite highly stale data.** As is evident from Figs. 4 and 6, DMPO enjoys a largely stable dynamics despite using up to  $24 \times$  stale data (which means 24 parameter updates on the same batch of rollouts), without suffering from high variance of importance sampling. While this seems to contradict the general belief that on-policy learning beats off-policy learning for LLM RL, we argue that this is not the case because the off-policy in DMPO is inherently different from that used in diffu-GRPO or GRPO. Note that the latter considers the importance weight of the form  $\frac{\pi_\theta}{\pi_{\text{old}}}$ , which **inevitably diverges** as the number of parameter updates on  $\theta$  increases. However, DMPO uses importance weight of the form  $\frac{p_s}{p_{\text{old}}}$ , which is independent of the current policy model  $\pi_\theta$  and remains stable over a long horizon of training, enabling the use of a low buffer sampling frequency and highly stale rollouts without sacrificing performances. Moreover, DMPO adopts *sequence-level* importance sampling, in contrast to the *token-level* importance sampling used in diffu-GRPO or cGRPO, thereby providing an additional layer of stability. This advantage is also discussed in depth in Group Sequence Policy Optimization (GSPO, Zheng et al. (2025a)), which similarly considers *sequence-level* importance sampling.

Additional experimental results and discussion can be found in App. C.3.

## 5 CONCLUSION

This paper proposed Distribution Matching Policy Optimization (DMPO), a novel RL fine-tuning framework for dLLMs. DMPO leverages the unique characteristics of dLLMs via importance sampling and a WDCE loss, enabling off-policy training and forward-only computation that naturally exploit dLLM inference capabilities. The main limitation of this work is that we focus on a single pretrained dLLM and four elementary reasoning benchmarks, and DMPO’s performance on other pretrained dLLMs and tasks in different domains remains unknown. Our work opens several promising directions for future research, such as investigating the distribution matching framework for other sequence models and studying the design of more effective weight baseline techniques.

540 REFERENCES  
541

542 Anthropic. Introducing claude 4, May 2025. URL <https://www.anthropic.com/news/claude-4>. Accessed: 2025-09-01.

544 Arel. Arel's sudoku generator. <https://www.ocf.berkeley.edu/~arel/sudoku/main.html>, 2025. Accessed: 2025-07-01.

546 Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=tyEyYT267x>.

551 Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL <https://openreview.net/forum?id=h7-XixPCAL>.

555 Jinbin Bai, Tian Ye, Wei Chow, Enxin Song, Qing-Guo Chen, Xiangtai Li, Zhen Dong, Lei Zhu, and Shuicheng YAN. Meissonic: Revitalizing masked generative transformers for efficient high-resolution text-to-image synthesis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=GJsuYHhAga>.

560 Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=WBcBhT1NKO>.

564 Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27381–27394. Curran Associates, Inc., 2021. URL [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/e614f646836aaed9f89ce58e837e2310-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/e614f646836aaed9f89ce58e837e2310-Paper.pdf).

570 Victor Besnier, Mickael Chen, David Hurých, Eduardo Valle, and Matthieu Cord. Halton scheduler for masked generative image transformer. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=RDVrlWAb7K>.

574 Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 28266–28279. Curran Associates, Inc., 2022. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b5b528767aa35f5b1a60fe0aaeca0563-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b5b528767aa35f5b1a60fe0aaeca0563-Paper-Conference.pdf).

580 Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, 2024. URL <https://openreview.net/forum?id=kQwSbv0BR4>.

584 Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11315–11325, 2022. doi: 10.1109/CVPR52688.2022.01103.

587 Chen-Hao Chao, Wei-Fang Sun, Hanwen Liang, Chun-Yi Lee, and Rahul Krishnan. Beyond masked and unmasked: Discrete diffusion models via partial masking. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=nqpbbvEZwF>.

592 Haoxuan Chen, Yinuo Ren, Martin Renqiang Min, Lexing Ying, and Zachary Izzo. Solving inverse problems via diffusion-based priors: An approximation-free ensemble sampling approach. *arXiv preprint arXiv:2506.03979*, 2025a.

594 Tong Chen, Yinuo Zhang, Sophia Tang, and Pranam Chatterjee. Multi-objective-guided discrete flow  
 595 matching for controllable biological sequence design. *arXiv preprint arXiv:2505.07086*, 2025b.  
 596

597 Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong  
 598 reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.

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

602

603 Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In  
 604 *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=mZn2Xyh9Ec>.

605

606 DeepMind. Gemini diffusion. <https://deepmind.google/models/gemini-diffusion/>. Accessed: 2025-09-24.

607

608 Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis.  
 609 On the effect of negative gradient in group relative deep reinforcement optimization. In *The*  
 610 *Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=2K9QsDaqkM>.

611

612 Justin Deschenaux and Caglar Gulcehre. Beyond autoregression: Fast LLMs via self-distillation  
 613 through time. In *The Thirteenth International Conference on Learning Representations*, 2025.  
 614 URL <https://openreview.net/forum?id=uZ5K4HeNwd>.

615

616 Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky T. Q.  
 617 Chen. Stochastic optimal control matching. In A. Globerson, L. Mackey, D. Bel-  
 618 grave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural In-  
 619 formation Processing Systems*, volume 37, pp. 112459–112504. Curran Associates, Inc.,  
 620 2024. URL [https://proceedings.neurips.cc/paper\\_files/paper/2024/file/cc32ec39a5073f61d38c338d963df30d-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2024/file/cc32ec39a5073f61d38c338d963df30d-Paper-Conference.pdf).

621

622 Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching:  
 623 Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control.  
 624 In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=xQBRrtQM8u>.

625

626 Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam  
 627 Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English,  
 628 and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In  
 629 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett,  
 630 and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine  
 631 Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR,  
 632 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/esser24a.html>.

633

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

637

638 Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sher-  
 639 jil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In  
 640 Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Ad-  
 641 vances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.,  
 642 2014. URL [https://proceedings.neurips.cc/paper\\_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf).

643

644 Daya Guo, Dejian Yang, Huawei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,  
 645 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-R1: Incentivizing reasoning capability in LLMs  
 646 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

647

Wei Guo, Jaemoo Choi, Yuchen Zhu, Molei Tao, and Yongxin Chen. Proximal diffusion neural  
 648 sampler. *arXiv preprint arXiv:2510.03824*, 2025b.

648 Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert  
 649 Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wigert, et al. Simulating 500 million years of  
 650 evolution with a language model. *Science*, pp. 850–858, 2025. doi: 10.1126/science.ads0018.  
 651

652 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn  
 653 Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH  
 654 dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information  
 655 Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL [https://datasets-benchmarks-proceedings.neurips.cc/paper\\_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf](https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf).  
 656

658 Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and  
 659 Jiangchao Yao. Wide-in, narrow-out: Revokable decoding for efficient and effective DLLMs.  
 660 *arXiv preprint arXiv:2507.18578*, 2025.

661 Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,  
 662 and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International  
 663 Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeFYf9>.  
 664

666 Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S Abdelfattah, Jae-sun Seo, Zhiru Zhang, and  
 667 Udit Gupta. Accelerating diffusion language model inference via efficient KV caching and guided  
 668 diffusion. *arXiv preprint arXiv:2505.21467*, 2025.

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

672 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec  
 673 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. OpenAI o1 system card. *arXiv preprint  
 674 arXiv:2412.16720*, 2024.

676 Amin Karimi Monsefi, Nikhil Bhendawade, Manuel Rafael Ciosici, Dominic Culver, Yizhe Zhang,  
 677 and Irina Belousova. FS-DFM: Fast and accurate long text generation with few-step diffusion  
 678 language models. *arXiv preprint arXiv:2509.20624*, 2025.

680 Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham M. Kakade, and Sitan Chen. Train for the worst,  
 681 plan for the best: Understanding token ordering in masked diffusions. In *Forty-second International  
 682 Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=DjJmre5IkP>.  
 683

685 Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun  
 686 Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with  
 687 LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

688 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan  
 689 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth  
 690 International Conference on Learning Representations*, 2023.

692 Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan,  
 693 Di ZHANG, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. In  
 694 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL  
 695 <https://openreview.net/forum?id=oCBKGw5HNF>.

696 Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang,  
 697 and Linfeng Zhang. dLLM-Cache: Accelerating diffusion large language models with adaptive  
 698 caching. *arXiv preprint arXiv:2506.06295*, 2025b.

699 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min  
 700 Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*,  
 701 2025c.

702 Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.  
703

704

705 Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios  
706 of the data distribution. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller,  
707 Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International  
708 Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp.  
709 32819–32848. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/lou24a.html>.  
710

711

712 Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dKV-Cache: The cache for diffusion  
713 language models. In *The Thirty-ninth Annual Conference on Neural Information Processing  
714 Systems*, 2025. URL <https://openreview.net/forum?id=Gppo2JImHs>.  
715

716 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke  
717 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time  
718 scaling. *arXiv preprint arXiv:2501.19393*, 2025.

719 Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan  
720 Li. Scaling up masked diffusion models on text. 2025a. URL <https://openreview.net/forum?id=WNvvwK0tut>.  
721

722

723 Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin,  
724 Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *The Thirty-ninth Annual  
725 Conference on Neural Information Processing Systems*, 2025b. URL <https://openreview.net/forum?id=Knqic0znVF>.  
726

727 Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance  
728 for discrete state-space diffusion and flow models. In *The Thirteenth International Conference  
729 on Learning Representations*, 2025. URL <https://openreview.net/forum?id=XsgH154y07>.  
730

731

732 Alexander Novikov, Ngan Vu, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wag-  
733 ner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. AlphaE-  
734 volv: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*,  
735 2025.

736

737 Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan  
738 Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean  
739 data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL  
740 <https://openreview.net/forum?id=sMyXP8Tanm>.

741 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong  
742 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser  
743 Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan  
744 Leike, and Ryan Lowe. Training language models to follow instructions with human feed-  
745 back. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Ad-  
746 vances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Asso-  
747 ciates, Inc., 2022. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf).  
748

749

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

752

753 Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Cheng-Hao Liu, Sarthak Mittal, Nouha  
754 Dziri, Michael M. Bronstein, Pranam Chatterjee, Alexander Tong, and Joey Bose. Steering  
755 masked discrete diffusion models via discrete denoising posterior prediction. In *The Thirteenth  
756 International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=Ombm8S40zN>.

756 Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *The Thirteenth*  
 757 *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=tPNHOoZF19>.

758

759 Yinuo Ren, Haoxuan Chen, Grant M. Rotskoff, and Lexing Ying. How discrete and continuous  
 760 diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral  
 761 framework. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL  
 762 <https://openreview.net/forum?id=6awxwQE182>.

763

764 Yinuo Ren, Haoxuan Chen, Yuchen Zhu, Wei Guo, Yongxin Chen, Grant M. Rotskoff, Molei Tao,  
 765 and Lexing Ying. Fast solvers for discrete diffusion models: Theory and applications of high-order  
 766 algorithms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*,  
 767 2025b. URL <https://openreview.net/forum?id=Ouk1L6Q3s0>.

768

769 Yinuo Ren, Wenhao Gao, Lexing Ying, Grant M Rotskoff, and Jiequn Han. DriftLite: Lightweight  
 770 drift control for inference-time scaling of diffusion models. *arXiv preprint arXiv:2509.21655*,  
 771 2025c.

772

773 Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuta Takida, Yuki Mitsufuji, and Molei Tao. Theory-  
 774 informed improvements to classifier-free guidance for discrete diffusion models. *arXiv preprint  
 775 arXiv:2507.08965*, 2025a.

776

777 Kevin Rojas, Yuchen Zhu, Sichen Zhu, Felix X-F. Ye, and Molei Tao. Diffuse everything: Multimodal  
 778 diffusion models on arbitrary state spaces. In *Forty-second International Conference on Machine  
 779 Learning*, 2025b. URL <https://openreview.net/forum?id=AjbiICRt6q>.

780

781 Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M  
 782 Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion  
 783 language models. In *The Thirty-eighth Annual Conference on Neural Information Processing  
 784 Systems*, 2024. URL <https://openreview.net/forum?id=L4uaAR4ArM>.

785

786 Subham Sekhar Sahoo, Zhihan Yang, Yash Akhauri, Johnna Liu, Deepansha Singh, Zhoujun Cheng,  
 787 Zhengzhong Liu, Eric Xing, John Thickstun, and Arash Vahdat. Esoteric language models. *arXiv  
 788 preprint arXiv:2506.01928*, 2025.

789

790 John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region  
 791 policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International  
 792 Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*,  
 793 pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL <https://proceedings.mlr.press/v37/schulman15.html>.

794

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

797

798 Shiv Shankar. PADRE: Pseudo-likelihood based alignment of diffusion language models. In *2nd AI  
 799 for Math Workshop @ ICML 2025*, 2025. URL <https://openreview.net/forum?id=gzdqCqN095>.

800

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

804

805 Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized  
 806 masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information  
 807 Processing Systems*, 2024. URL <https://openreview.net/forum?id=xcqSOfHt4g>.

808

809 Qingyu Shi, Jinbin Bai, Zhuoran Zhao, Wenhao Chai, Kaidong Yu, Jianzong Wu, Shuangyong Song,  
 810 Yunhai Tong, Xiangtai Li, Xuelong Li, et al. Mudit: Liberating generation beyond text-to-image  
 811 with a unified discrete diffusion model. *arXiv preprint arXiv:2505.23606*, 2025.

812

813 Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang  
 814 Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with  
 815 high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

810 Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. PepTune: De novo generation of therapeutic pep-  
 811 tides with multi-objective-guided discrete diffusion. In *Forty-second International Conference on*  
 812 *Machine Learning*, 2025a. URL <https://openreview.net/forum?id=FQoy1Y1Hd8>.

813 Sophia Tang, Yuchen Zhu, Molei Tao, and Pranam Chatterjee. TR2-D2: Tree search guided trajectory-  
 814 aware fine-tuning for discrete diffusion. *arXiv preprint arXiv:2509.25171*, 2025b.

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

817 Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural  
 818 autoregressive distribution estimation. *Journal of Machine Learning Research*, 17(205):1–37,  
 819 2016. URL <http://jmlr.org/papers/v17/16-272.html>.

820 Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan  
 821 Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformer reinforcement  
 822 learning. <https://github.com/huggingface/trl>, 2020.

823 Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey  
 824 Levine, Aviv Regev, Hanchen, and Tommaso Biancalani. Fine-tuning discrete diffusion models via  
 825 reward optimization with applications to DNA and protein design. In *The Thirteenth International*  
 826 *Conference on Learning Representations*, 2025a. URL <https://openreview.net/forum?id=G328D1xt4W>.

827 Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforce-  
 828 ment learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*,  
 829 2025b.

830 Lilian Weng. Reward hacking in reinforcement learning. *lilianweng.github.io*, Nov 2024. URL  
 831 <https://lilianweng.github.io/posts/2024-11-28-reward-hacking/>.

832 Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song  
 833 Han, and Enze Xie. Fast-dLLM: Training-free acceleration of diffusion LLM by enabling KV  
 834 cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

835 Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu,  
 836 Qiushan Guo, Weilin Huang, et al. DanceGRPO: Unleashing GRPO on visual generation. *arXiv*  
 837 *preprint arXiv:2505.07818*, 2025.

838 Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. MMaDA:  
 839 Multimodal large diffusion language models. In *The Thirty-ninth Annual Conference on Neural*  
 840 *Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=wczmXLuLGd>.

841 Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your  
 842 efficient RL framework secretly brings you off-policy RL training, August 2025. URL <https://fengyao.notion.site/off-policy-rl>.

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

845 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian  
 846 Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming  
 847 Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua  
 848 Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao  
 849 Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang.  
 850 DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual*  
 851 *Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=2a36EMSSTp>.

852 Oussama Zekri and Nicolas Boullé. Fine-tuning discrete diffusion models with policy gradient  
 853 methods. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.  
 854 URL <https://openreview.net/forum?id=rXFzVRZsbt>.

864 Ruixiang Zhang, Shuangfei Zhai, Jiatao Gu, Yizhe Zhang, Huangjie Zheng, Tianrong Chen,  
 865 Miguel Ángel Bautista, Joshua M. Susskind, and Navdeep Jaitly. Flexible language model-  
 866 ing in continuous space with transformer-based autoregressive flows. In *The Thirty-ninth Annual*  
 867 *Conference on Neural Information Processing Systems*, 2025a. URL <https://openreview.net/forum?id=MR7Fn23hSE>.

868

870 Ruixiang Zhang, Shuangfei Zhai, Yizhe Zhang, James Thornton, Zijing Ou, Joshua M. Susskind,  
 871 and Navdeep Jaitly. Target concrete score matching: A holistic framework for discrete diffu-  
 872 sion. In *Forty-second International Conference on Machine Learning*, 2025b. URL <https://openreview.net/forum?id=ZMrdvSm7xi>.

873

874 Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. d1: Scaling reasoning in diffusion  
 875 large language models via reinforcement learning. In *The Thirty-ninth Annual Conference on*  
 876 *Neural Information Processing Systems*, 2025a. URL <https://openreview.net/forum?id=7ZVRlBFuEv>.

877

878 Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shao-  
 879 han Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint*  
 880 *arXiv:2507.20673*, 2025b.

881

882 Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong  
 883 Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization.  
 884 *arXiv preprint arXiv:2507.18071*, 2025a.

885

886 Haoyang Zheng, Xinyang Liu, Cindy Xiangrui Kong, Nan Jiang, Zheyuan Hu, Weijian Luo, Wei  
 887 Deng, and Guang Lin. Ultra-fast language generation via discrete diffusion divergence instruct.  
 888 *arXiv preprint arXiv:2509.25035*, 2025b.

889

890 Huangjie Zheng, Shansan Gong, Ruixiang Zhang, Tianrong Chen, Jiatao Gu, Mingyuan Zhou,  
 891 Navdeep Jaitly, and Yizhe Zhang. Continuously augmented discrete diffusion model for categorical  
 892 generative modeling. *arXiv preprint arXiv:2510.01329*, 2025c.

893

894 Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su,  
 895 Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with  
 896 forward process. *arXiv preprint arXiv:2509.16117*, 2025d.

897

898 Kaiwen Zheng, Yongxin Chen, Huayu Chen, Guande He, Ming-Yu Liu, Jun Zhu, and Qinsheng  
 899 Zhang. Direct discriminative optimization: Your likelihood-based visual generative model is  
 900 secretly a GAN discriminator. In *Forty-second International Conference on Machine Learning*,  
 901 2025e. URL <https://openreview.net/forum?id=OJ6WE7F8tK>.

902

903 Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked  
 904 diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical  
 905 sampling. In *The Thirteenth International Conference on Learning Representations*, 2025f. URL  
 906 <https://openreview.net/forum?id=CTC7CmirNr>.

907

908 Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang, Muhan Zhang, Lester Mackey,  
 909 Tommi Jaakkola, Stephen Bates, and Dinghuai Zhang. Coevolutionary continuous discrete diffu-  
 910 sion: Make your diffusion language model a latent reasoner. *arXiv preprint arXiv:2510.03206*,  
 911 2025.

912

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

916

917 Sichen Zhu, Yuchen Zhu, Molei Tao, and Peng Qiu. Diffusion generative modeling for spatially  
 918 resolved gene expression inference from histology images. In *The Thirteenth International*  
 919 *Conference on Learning Representations*, 2025b. URL <https://openreview.net/forum?id=FtjLUHyZAO>.

920

921 Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang  
 922 Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. FlowRL: Matching reward distributions for LLM  
 923 reasoning. *arXiv preprint arXiv:2509.15207*, 2025c.

918 Yuzhu Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Di[M]O: Distilling masked  
919 diffusion models into one-step generator. In *Proceedings of the IEEE/CVF International Conference*  
920 *on Computer Vision (ICCV)*, pp. 18606–18618, October 2025d.  
921

922 Yuzhu Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Soft-Di[M]O: Improving  
923 one-step discrete image generation with soft embeddings. *arXiv preprint arXiv:2509.22925*,  
924 2025e.

925 Yuchen Zhu, Tianrong Chen, Lingkai Kong, Evangelos Theodorou, and Molei Tao. Trivialized  
926 momentum facilitates diffusion generative modeling on Lie groups. In *The Thirteenth International*  
927 *Conference on Learning Representations*, 2025f. URL <https://openreview.net/forum?id=DTatjJTD11>.

928

929 Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. MDNS:  
930 Masked diffusion neural sampler via stochastic optimal control. In *The Thirty-ninth Annual*  
931 *Conference on Neural Information Processing Systems*, 2025g. URL <https://openreview.net/forum?id=xIH95kXNR2>.

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972 A RELATED WORK  
973974 Here, we focus on the literature for discrete diffusion models, as well as the methods for fine-tuning  
975 MDMs, dLLMs, and LLMs. We also briefly review several GRPO-style algorithms for domains  
976 outside of LLMs.  
977978 **Discrete Diffusion Models.** Diffusion models have been top-performing approaches for generating  
979 various data modalities (Zhu et al., 2025f; Esser et al., 2024; Zhu et al., 2025b; Rojas et al., 2025b;  
980 Zheng et al., 2025e; Chen et al., 2025a; Ren et al., 2025c). Discrete diffusion models (Austin et al.,  
981 2021; Campbell et al., 2022; Lou et al., 2024; Zhang et al., 2025b), a natural extension of diffusion  
982 models to finite state spaces, have emerged as powerful approaches for generating categorical,  
983 sequence data, with applications to text (Nie et al., 2025a;b; Ye et al., 2025), images (Chang et al.,  
984 2022; Bai et al., 2025; Shi et al., 2025), and biological sequences (Tang et al., 2025a; Chen et al.,  
985 2025b). One of the most effective variants of discrete diffusion models is masked diffusion models  
986 (MDM) (Sahoo et al., 2024; Ou et al., 2025; Shi et al., 2024) and its variants (Arriola et al., 2025;  
987 Sahoo et al., 2025; Chao et al., 2025). Recently, continuous latents have also been introduced into the  
988 modeling of discrete data (Zhang et al., 2025a; Zhou et al., 2025; Zheng et al., 2025c), resulting in  
989 improved and more appealing performance.  
990991 One particularly important line of development for discrete diffusion models centers on their inference  
992 techniques, with the aim of improving generation quality (Nisonoff et al., 2025; Rojas et al., 2025a;  
993 Kim et al., 2025; Besnier et al., 2025) and accelerating sampling speed (Ren et al., 2025b; Ben-Hamu  
994 et al., 2025; Wu et al., 2025; Hong et al., 2025). Besides these training-free approaches, learning-based  
995 approaches, such as few-step distillation, have also achieved decent success for discrete diffusion  
996 models (Deschenaux & Gulcehre, 2025; Karimi Monsefi et al., 2025; Zheng et al., 2025b; Zhu et al.,  
997 2025d;e). DMPO is closely tied to the literature on fast inference, as it can benefit from it by enjoying  
998 a similar training speed acceleration due to its forward nature.  
9991000 **Fine-tuning general discrete diffusion models.** Earlier works on fine-tuning discrete diffusion  
1001 models primarily focus on applications in biological and chemical domains, e.g., SVDD (?), DDPP  
1002 (Rector-Brooks et al., 2025), DRAKES (Wang et al., 2025a), SEPO (Zekri & Boullé, 2025), and  
1003 TR2-D2 (Tang et al., 2025b). Although these methods work well for their respective tasks, they are  
1004 not directly applicable to dLLMs due to the unique challenges posed by the language domain, such  
1005 as large model size, high dimensionality, and the need to maintain linguistic coherence and diversity.  
10061007 **Fine-tuning diffusion LLMs.** Recently, numerous works have proposed RL algorithms for fine-  
1008 tuning dLLMs, with most existing works being adaptations of the GRPO algorithm (Shao et al.,  
1009 2024) for AR LLMs. For example, Zhao et al. (2025a) proposed Diffu-GRPO that estimates the  
1010 per-token response log probabilities via masking all except the required response positions, and  
1011 partially masking the prompt to get the model output, while their sequence log probability is estimated  
1012 by mean-field approximation. Gong et al. (2025) introduced Coupled GRPO that modified the Diffu-  
1013 GRPO method by not partially masking the prompt, and using complementary pairs of masks to  
1014 mask the same response that fully uses the model output, which we also adopt in our experiments.  
1015 Yang et al. (2025) proposed UniGRPO, which involves a structured noise strategy and a modified  
1016 log-likelihood approximation (both per-token and sequence). [Shankar \(2025\) proposed an alignment  
1017 method of dLLMs based on pseudo-likelihood.](#) Concurrent with our work, TraceRL (Wang et al.,  
1018 2025b) improves dLLM RL training by minimizing a training-inference gap. wd1 (Tang et al., 2025c)  
1019 introduces additional regularization to the old policy, alongside the regularization applied to the  
1020 reference model policy, which resembles the case discussed in App. B.2. We highlight that all these  
1021 methods are GRPO-style algorithms that require estimating per-token response log probabilities,  
1022 which are typically intractable and challenging for dLLMs. In contrast, our method offers the  
1023 advantage of being a forward one, with greater efficiency and accuracy.  
10241025 **Fine-tuning LLMs.** For fine-tuning LLMs, pre-LLM era works such as Trust Region Policy  
1026 Optimization (TRPO, Schulman et al. (2015)) and Proximal Policy Optimization (PPO, Schulman  
1027 et al. (2017)) have been widely used for RLHF (Ouyang et al., 2022). Since the huge success of  
1028 GRPO (Shao et al., 2024) on DeepSeek-R1 (Guo et al., 2025a), there have been many follow-up  
1029 works that improve GRPO in various ways, for instance: GRPO Done Right (Dr-GRPO, Liu et al.  
1030 (2025c)), Decoupled clip and Dynamic sAmpling Policy Optimization (DAPO, Yu et al. (2025)),

1026 Group Policy Gradient (GPG, Chu et al. (2025)), Group Sequence Policy Optimization (GSPO, Zheng  
 1027 et al. (2025a)), Geometric-Mean Policy Optimization (GMPO, Zhao et al. (2025b)), etc.

1028 Apart from the aforementioned policy gradient-based methods, GFlowNet (Bengio et al., 2021) has  
 1029 also been applied to finetuning LLMs, with successful applications seen in Kimi 1.5 (Kimi Team  
 1030 et al., 2025) and FlowRL (Zhu et al., 2025c). Notably, concurrent with our work, FlowRL shares the  
 1031 same high-level goal as our DMPO, targeting also policy distribution matching rather than merely  
 1032 reward maximization for AR-LLMs. However, distinct from DMPO, FlowRL derives its objectives  
 1033 from reverse KL and utilizes GFlowNet objectives. In contrast, our approach considers forward KL,  
 1034 which is known to be mass-covering, and implements it using importance sampling and weighted  
 1035 denoising cross-entropy.

1036 **GRPO-style algorithms for fine-tuning diffusion and flow-based models.** GRPO-type algorithms  
 1037 have also been adapted to diffusion and flow-based models, such as flow-GRPO (Liu et al., 2025a)  
 1038 and DanceGRPO (Xue et al., 2025). Aside from that, there are also SOC-based fine-tuning algorithms  
 1039 for diffusion models, such as adjoint matching (Domingo-Enrich et al., 2025), with which our work  
 1040 shares similarity at a high level. Concurrent with our work, DiffusionNFT (Zheng et al., 2025d) has  
 1041 been proposed to finetune continuous diffusion models for text-to-image generation tasks. While  
 1042 formulated in drastically different ways, DiffusionNFT shares a similarity with our DMPO in that it is  
 1043 also an algorithm that primarily depends on model forward passes rather than backward trajectories.

## B THEORY OF DISTRIBUTION MATCHING POLICY OPTIMIZATION

### B.1 DISTRIBUTION MATCHING POLICY OPTIMIZATION FROM THE STOCHASTIC OPTIMAL CONTROL PERSPECTIVE

1050 This section aims at providing an alternative derivation of DMPO from the perspective of stochastic  
 1051 optimal control (SOC), which is inspired by DRAKES (Wang et al., 2025a) and MDNS (Zhu  
 1052 et al., 2025g). We will first introduce the necessary background on continuous-time Markov chains  
 1053 (CTMCs), then show how MDM sampling can be viewed as a CTMC. Finally, we derive the DMPO  
 1054 framework from the SOC perspective.

1055 **Introduction to Continuous-time Markov Chains.** To derive the SOC framework for fine-tuning,  
 1056 we view the sampling of an MDM as a time-indexed stochastic process, and the proper mathematical  
 1057 tool is the **continuous-time Markov chain (CTMC)**. A CTMC  $X = (X_t)_{t \in [0,1]}$  is a stochastic  
 1058 process taking value in a discrete state space  $\mathcal{X}$ . Its law is characterized by the **rate matrix**  
 1059  $Q = (Q_t)_{t \in [0,1]}$ , defined as

$$Q_t(x, y) = \lim_{\Delta t \rightarrow 0} \frac{\Pr(X_{t+\Delta t} = y | X_t = x) - 1_{x=y}}{\Delta t}, \quad \forall x, y \in \mathcal{X}. \quad (17)$$

1060 By definition, the off-diagonal entries of  $Q_t$  are non-negative, and each row sums to zero.

1061 The **path** of  $X$ , i.e.,  $t \mapsto X_t(\omega)$ , is piecewise constant with discontinuous jumps, and one typically  
 1062 assumes that the path is right continuous with left limits. The **path measure** a CTMC  $X$  is a  
 1063 probability measure on the space of paths defined as  $\mathbb{P}^X(A) := \Pr(X \in A)$ , which is the distribution  
 1064 of  $X$ . The following lemma shows how to compute the **Radon-Nikodým (RN) derivative** between  
 1065 two path measures driven by CTMCs with different rate matrices and initial distributions:

1066 **Lemma 1.** *Given two CTMCs with rate matrices  $Q^1, Q^2$  and initial distributions  $\mu_1, \mu_2$  on  $\mathcal{X}$ , let  
 1067  $\mathbb{P}^1, \mathbb{P}^2$  be the associated path measures. Then, for any path  $\xi = (\xi_t)_{t \in [0,1]}$ ,*

$$\log \frac{d\mathbb{P}^1}{d\mathbb{P}^2}(\xi) = \log \frac{d\mu_1}{d\mu_2}(\xi_0) + \sum_{t: \xi_{t-} \neq \xi_t} \log \frac{Q_t^1(\xi_{t-}, \xi_t)}{Q_t^2(\xi_{t-}, \xi_t)} + \int_0^1 (Q_t^1(\xi_t, \xi_t) - Q_t^2(\xi_t, \xi_t)) dt. \quad (18)$$

1068 For the proof, see Campbell et al. (2024, App. C.1), Ren et al. (2025a, Thm. 3.3), or Zhu et al. (2025g,  
 1069 Lem. 1). An intuitive interpretation of (18) is to view the RN derivative as the limit of density ratios  
 1070 between finite-dimensional joint distributions, and approximate the transition probability by (17).

1080  
1081 **Masked Diffusion Models as Continuous-Time Markov Chains.** We will now delve into the  
1082 CTMC formulation of sampling from an MDM. To avoid notational clutter, we use superscript to  
1083 denote the position index, and subscript to denote the time index (e.g.,  $\xi_t = (\xi_t^1, \dots, \xi_t^D)$ ). We present  
1084 the theory only in the case of *unconditional generation* with sequence length  $D$  for simplicity of  
1085 notations, but it can be easily generalized to the case of conditional generation of  $\mathbf{o}$  given a  
1086 prompt  $\mathbf{q}$ .

1087 As shown in Ou et al. (2025), by introducing a noise schedule  $\gamma(t) = \frac{1}{1-t}$ ,<sup>2</sup> the random order  
1088 autoregressive sampling of an MDM  $\pi_\theta$  can be viewed as a CTMC with the rate matrix  $Q^\theta =$   
1089  $(Q_t^\theta)_{t \in [0,1]}$  such that for  $\mathbf{x} \neq \mathbf{y} \in \bar{\mathcal{V}}^D$ ,

$$1090 Q_t^\theta(\mathbf{x}, \mathbf{y}) = \gamma(t) \pi_\theta(\mathbf{x})_{d,n}, \text{ if } \mathbf{x}^d = \mathbf{M} \text{ and } \mathbf{y} = \mathbf{x}^{d \leftarrow n},$$

1091 and 0 if otherwise, where  $\mathbf{x}^{d \leftarrow n}$  means the sequence obtained by replacing the  $d$ -th position of  $\mathbf{x}$  by  
1092  $n$ . The diagonal terms of  $Q_t^\theta$  can be computed as

$$1093 Q_t^\theta(\mathbf{x}, \mathbf{x}) = - \sum_{\mathbf{y} \neq \mathbf{x}} Q_t^\theta(\mathbf{x}, \mathbf{y}) = - \sum_{d: \mathbf{x}^d = \mathbf{M}} \sum_n Q_t^\theta(\mathbf{x}, \mathbf{x}^{d \leftarrow n}) \\ 1094 = - \sum_{d: \mathbf{x}^d = \mathbf{M}} \sum_n \gamma(t) \pi_\theta(\mathbf{x})_{d,n} = -\gamma(t) \cdot |\{d : \mathbf{x}^d = \mathbf{M}\}|. \quad (19)$$

1095 Therefore, if  $\mathbb{P}^\theta, \mathbb{P}^{\theta'}$  are the path measures of the sampling processes of two MDMs parameterized by  
1096  $\theta$  and  $\theta'$ , respectively, then by (18), assuming that the jump from  $\xi_{t-}$  to  $\xi_t$  is at the  $d(t)$ -th position,  
1097 we have

$$1102 \log \frac{d\mathbb{P}^{\theta'}}{d\mathbb{P}^\theta}(\xi) = \sum_{t: \xi_{t-} \neq \xi_t} \log \frac{\pi_{\theta'}(\xi_{t-})_{d(t), \xi_t^{d(t)}}}{\pi_\theta(\xi_{t-})_{d(t), \xi_t^{d(t)}}}, \forall \xi = (\xi_t)_{t \in [0,1]}, \quad (20)$$

1103 as the first term in (18) is always zero (both initial distributions are the point mass on the fully masked  
1104 sequence), and the diagonal terms in the third term cancel out due to (19).

1105 Moreover, as proved in Ou et al. (2025), the training of an MDM  $\pi_\theta$  given i.i.d. samples from the  
1106 target distribution  $p_{\text{data}}$  can be interpreted as minimizing the KL divergence between the target path  
1107 measure  $\mathbb{P}^*$  and the parameterized path measure  $\mathbb{P}^\theta$ , where  $\mathbb{P}^*$  is defined as the path measure of the  
1108 CTMC with rate matrix  $Q_t^*(\mathbf{x}, \mathbf{x}^{d \leftarrow n}) = \gamma(t) \Pr_{\mathbf{x} \sim p_{\text{data}}}(\mathbf{X}^d = n | \mathbf{X}^{\text{UM}} = \mathbf{x}^{\text{UM}}) \mathbf{1}_{\mathbf{x}^d = \mathbf{M}}$ , i.e., with  
1109 the ground-truth conditional distribution. Moreover, one can derive

$$1112 \text{KL}(\mathbb{P}^* \parallel \mathbb{P}^\theta) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{m \sim \text{Unif}\{1, \dots, D\}} \left[ \frac{D}{m} \mathbb{E}_{\mu_m(\tilde{\mathbf{x}} | \mathbf{x})} \sum_{d: \tilde{\mathbf{x}}^d = \mathbf{M}} -\log \pi_\theta(\tilde{\mathbf{x}})_{d, \mathbf{x}^d} \right] + \text{const},$$

1113 where const does not depend on  $\theta$ , and  $\mu_m(\cdot | \mathbf{x})$  means to sample a uniformly random subset of  
1114  $\{1, \dots, D\}$  of size  $m$  and mask the corresponding positions in  $\mathbf{x}$ . Note that this is exactly the denoising  
1115 cross-entropy loss  $\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathcal{L}_\theta(\mathbf{x})$  as presented in (2). In other words, minimizing the KL divergence  
1116 between sequence-level probabilities  $p_*(\xi_1)$  and  $p_\theta(\xi_1) \approx e^{-\mathcal{L}_\theta(\xi_1)}$  in (11) can be interpreted as  
1117 precisely minimizing the KL divergence between path-level probabilities  $\mathbb{P}^*(\xi)$  and  $\mathbb{P}^\theta(\xi)$ .

1118 **Fine-tuning MDMs as a Stochastic Optimal Control Problem on Path Measures.** The task of  
1119 fine-tuning a pretrained MDM can be viewed as a stochastic optimal control (SOC) problem on the  
1120 space of path measures: given a pretrained MDM  $\pi_{\text{ref}}$  which generates a distribution  $p_{\text{ref}}$ , we define its  
1121 induced **reference path measure** as  $\mathbb{P}_1^{\text{ref}}$ , with rate matrix  $Q_t^{\text{ref}}(\mathbf{x}, \mathbf{x}^{d \leftarrow n}) = \gamma(t) \pi_{\text{ref}}(\mathbf{x})_{d,n} \mathbf{1}_{\mathbf{x}^d = \mathbf{M}}$ ,  
1122 and has terminal distribution  $\mathbb{P}_1^{\text{ref}} = p_{\text{ref}}$ . We aim at finding a target rate matrix  $Q^*$  such that the  
1123 associated target path measure  $\mathbb{P}^*$  has a terminal distribution  $p_*$  defined in the following way of  
1124 tilting by reward:

$$1125 p_*(\mathbf{x}) = \frac{1}{Z} p_{\text{ref}}(\mathbf{x}) e^{r(\mathbf{x})/\alpha}, \quad \mathbf{x} \in \mathcal{V}^D, \quad \text{where } Z = \sum_{\mathbf{x}} p_{\text{ref}}(\mathbf{x}) e^{r(\mathbf{x})/\alpha}.$$

1126  
1127 <sup>2</sup>The choice of noise schedule is essentially not important for MDM. In fact,  $\gamma$  can be any positive function  
1128 with  $\int_0^1 \gamma(t) dt = \infty$ . Here, we follow the convention in most of the literature on MDM and choose this specific  
1129  $\gamma$  such that the conditional distribution of  $\xi_t \in \bar{\mathcal{V}}^D$  given  $\xi_1 \in \mathcal{V}^D$  is obtained by independently masking each  
1130 position in  $\xi_1$  with probability  $1 - t$ .

1134 This can be achieved by defining the target path measure  $\mathbb{P}^*$  as  
 1135  
 1136 
$$\mathbb{P}^*(\xi) = \mathbb{P}^{\text{ref}}(\xi_{[0,1]} | \xi_1) p_*(\xi_1) = \mathbb{P}^{\text{ref}}(\xi) \frac{p_*(\xi_1)}{p_{\text{ref}}(\xi_1)} = \frac{1}{Z} \mathbb{P}^{\text{ref}}(\xi) e^{r(\xi_1)/\alpha}, \forall \xi = (\xi_t)_{t \in [0,1]}. \quad (21)$$
  
 1137  
 1138

1139 We use a network  $\pi_\theta$  to parameterize the new rate matrix, initialized at  $\pi_{\text{ref}}$ . Given a current path  
 1140 measure  $\mathbb{P}^\theta$  induced by a CTMC with rate matrix  $Q_t^\theta(\mathbf{x}, \mathbf{x}^{d \leftarrow n}) = \gamma(t) \pi_\theta(\mathbf{x})_{d,n} 1_{x^d=M}$ , we can first  
 1141 derive the RN derivative between the path measures by (20):  
 1142

$$\begin{aligned} \log \frac{d\mathbb{P}^*}{d\mathbb{P}^\theta}(\xi) &= \log \frac{d\mathbb{P}^*}{d\mathbb{P}^{\text{ref}}}(\xi) + \log \frac{d\mathbb{P}^0}{d\mathbb{P}^\theta}(\xi) \\ &= \frac{r(\xi_1)}{\alpha} - \log Z + \sum_{t: \xi_{t-} \neq \xi_t} \log \frac{Q_t^0(\xi_{t-}, \xi_t)}{Q_t^\theta(\xi_{t-}, \xi_t)} + \int_0^1 \sum_{y \neq \xi_t} (Q_t^\theta - Q_t^0)(\xi_t, y) dt \\ &= \frac{r(\xi_1)}{\alpha} + \sum_{t: \xi_{t-} \neq \xi_t} \log \frac{\pi_{\text{ref}}(\xi_{t-})_{d(t), \xi_t^{d(t)}}}{\pi_\theta(\xi_{t-})_{d(t), \xi_t^{d(t)}}} - \log Z =: W^\theta(\xi) - \log Z, \end{aligned} \quad (22)$$

1152 where we assume that the jump from  $\xi_{t-}$  to  $\xi_t$  is at the  $d(t)$ -th position. The idea of the weighted  
 1153 denoising cross-entropy (WDCE) loss is essentially to treat i.i.d. samples from the current policy  $\mathbb{P}^\theta$   
 1154 as weighted samples from  $\mathbb{P}^*$ , and minimizing the following loss:  
 1155

$$\begin{aligned} \text{KL}(\mathbb{P}^* \parallel \mathbb{P}^\theta) + \text{const} &= \mathbb{E}_{p_*(\mathbf{x})} \mathcal{L}_\theta(\mathbf{x}) = \mathbb{E}_{\mathbb{P}^*(\xi)} \mathcal{L}_\theta(\xi_1) \\ &= \mathbb{E}_{\mathbb{P}^v(\xi)} \frac{d\mathbb{P}^*}{d\mathbb{P}^v}(\xi) \mathcal{L}_\theta(\xi_1) = \mathbb{E}_{\mathbb{P}^v(\xi)} \frac{1}{Z} e^{W^v(\xi)} \mathcal{L}_\theta(\xi_1), \end{aligned}$$

1160 where  $\mathbb{P}^v$  is the path measure induced by a CTMC with rate matrix  $Q^v$  where the network is par-  
 1161 meterized by  $v$  (e.g., the old parameters  $\theta_{\text{old}}$ ), whose parameters do not involve gradient calculation.  
 1162 For instance, we can set  $v = \theta_{\text{old}}$ . Note that  $Z = \mathbb{E}_{\mathbb{P}^v(\xi)} e^{W^v(\xi)}$ , which, if estimated via samples, is  
 1163 equivalent to doing softmax normalization on the logits  $W^v(\xi)$  in the batch. Comparing with the  
 1164 WDCE loss (11) presented in Sec. 3.2, we conclude that they are essentially the same.  
 1165

## 1166 B.2 GENERALIZING WDCE TO ZERO TEMPERATURE WITH PROXIMAL DESCENT

1168 Recall that our target distribution is (5), which is under a temperature  $\alpha > 0$ . We propose to generalize  
 1169 the WDCE loss (11) to incorporate the limiting case  $\alpha \rightarrow 0$  from the viewpoint of **proximal descent**  
 1170 (Guo et al., 2025b).

1171 The reward maximization problem (4) provides a variational characterization of the target distribution  
 1172  $p_*(\mathbf{o}|\mathbf{q})$ . Suppose now we have a dLLM policy  $\pi_{\theta_{\text{old}}}(\mathbf{o}|\mathbf{q})$  that outputs a distribution  $p_{\theta_{\text{old}}}(\mathbf{o}|\mathbf{q})$ . We  
 1173 define the next target distribution  $p_{\text{tar}}(\mathbf{o}|\mathbf{q})$  as  
 1174

$$p_{\text{tar}}(\mathbf{o}|\mathbf{q}) = \underset{p_\theta(\mathbf{o}|\mathbf{q})}{\text{argmax}} \left\{ \mathbb{E}_{p_\theta(\mathbf{o}|\mathbf{q})} [r(\mathbf{q}, \mathbf{o})] - \alpha \text{KL}(p_\theta(\cdot|\mathbf{q}) \| p_{\text{ref}}(\cdot|\mathbf{q})) - \frac{1}{\eta'} \text{KL}(p_\theta(\cdot|\mathbf{q}) \| p_{\theta_{\text{old}}}(\cdot|\mathbf{q})) \right\}, \quad (23)$$

1178 where  $\eta' > 0$  is the step size. Let  $\eta = \frac{\eta'}{1 + \eta' \alpha} \in (0, \frac{1}{\alpha})$ . It is easy to see that the solution is given by  
 1179

$$\begin{aligned} p_{\text{tar}}(\mathbf{o}|\mathbf{q}) &\propto_{\mathbf{o}} p_{\theta_{\text{old}}}(\mathbf{o}|\mathbf{q})^{1-\eta\alpha} p_{\text{ref}}(\mathbf{o}|\mathbf{q})^{\eta\alpha} e^{\eta r(\mathbf{q}, \mathbf{o})}, \\ &\propto_{\mathbf{o}} p_{\theta_{\text{old}}}(\mathbf{o}|\mathbf{q})^{1-\eta\alpha} p_*(\mathbf{o}|\mathbf{q})^{\eta\alpha}. \end{aligned} \quad (24)$$

1184 In fact, the term inside the brackets in (23) is  $-\frac{1}{\eta} \text{KL}(p_\theta(\cdot|\mathbf{q}) \| p_{\text{tar}}(\cdot|\mathbf{q})) + \text{const}$ . This means the  
 1185 next target distribution is a geometric interpolation between the current model distribution  $p_{\theta_{\text{old}}}$  and  
 1186 the optimal distribution  $p_*$ , with  $\eta > 0$  being a step size parameter. (24) is well-defined even when  
 1187  $\alpha = 0$ , although in this case, the target distribution concentrates on the set of maximizers of  $r(\mathbf{q}, \mathbf{o})$   
 1188 (e.g., all correct question-response pairs) without regularization from the base model  $p_{\text{ref}}(\mathbf{o}|\mathbf{q})$ .  
 1189

1188 For  $\alpha = 0$ ,  $p_{\text{tar}}(\mathbf{o}|\mathbf{q}) \propto_{\mathbf{o}} p_{\theta_{\text{old}}}(\mathbf{o}|\mathbf{q}) e^{\eta r(\mathbf{q}, \mathbf{o})}$ . We can similarly solve the distribution matching  
 1189 problem via the WDCE loss:

$$\begin{aligned} \text{KL}(p_{\text{tar}}(\cdot|\mathbf{q})\|p_{\theta}(\cdot|\mathbf{q})) &= \mathbb{E}_{p_{\text{tar}}(\mathbf{o}|\mathbf{q})}[-\log p_{\theta}(\mathbf{o}|\mathbf{q})] + \text{const} \\ &= \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q})} \underbrace{\frac{p_{\text{tar}}(\mathbf{o}|\mathbf{q})}{p_v(\mathbf{o}|\mathbf{q})}}_{=:w(\mathbf{o}|\mathbf{q})} [-\log p_{\theta}(\mathbf{o}|\mathbf{q})] + \text{const} \\ &\leq \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q})} w(\mathbf{o}|\mathbf{q}) \mathcal{L}_{\theta}(\mathbf{o}|\mathbf{q}) + \text{const}, \end{aligned}$$

1197 where the importance weight  $w(\mathbf{o}|\mathbf{q}) \propto_{\mathbf{o}} \exp\left(\eta r(\mathbf{q}, \mathbf{o}) + \log \frac{p_{\theta_{\text{old}}}(\mathbf{o}|\mathbf{q})}{p_{\theta_v}(\mathbf{o}|\mathbf{q})}\right)$ . For  $v \leftarrow \theta_{\text{old}}$ , the weight  
 1198 simplifies to the softmax of  $\eta r(\mathbf{q}, \mathbf{o})$  over all responses for the same prompt  $\mathbf{q}$ . The weight baseline  
 1199 subtraction tricks also apply here.

1200 We remark that when picking  $\alpha = 0$ , through the proximal gradient descent formulation, DMPO  
 1201 becomes completely *forward-only*, as it eliminates the need for estimating the sequence log probability  
 1202 ratio of the form  $\log \frac{p_{\text{ref}}(\mathbf{o}|\mathbf{q})}{p_v(\mathbf{o}|\mathbf{q})}$ , making it the best option to incorporate fast dLLM inference techniques  
 1203 for RL training speed-up. However, in this case, we can no longer guarantee the diversity in the target  
 1204 optimal distribution, and thus, we save this direction for future investigation.

### 1206 B.3 INSIGHTS FOR WEIGHT BASELINES: APPROXIMATE VARIANCE REDUCTION

1207 We first recall a classical equality in statistics regarding the **score function**: if  $p_{\theta}(x)$  is a probability  
 1208 density or probability mass function parameterized by a continuous parameter  $\theta$ , then under certain  
 1209 weak regularity conditions, we have  $\mathbb{E}_{p_{\theta}(x)} \nabla_{\theta} \log p_{\theta}(x) = 0$ .

1210 Therefore,

$$\begin{aligned} 0 &= \mathbb{E}_{p_{\theta}(\mathbf{o}|\mathbf{q})} \nabla_{\theta} \log p_{\theta}(\mathbf{o}|\mathbf{q}) = \nabla_{\theta} \mathbb{E}_{p_{\bar{\theta}}(\mathbf{o}|\mathbf{q})} \log p_{\theta}(\mathbf{o}|\mathbf{q}) \\ &= \nabla_{\theta} \mathbb{E}_{\sigma} \mathbb{E}_{p_{\bar{\theta}}(\mathbf{o}|\mathbf{q}; \sigma)} \log p_{\theta}(\mathbf{o}|\mathbf{q}) \\ &= \nabla_{\theta} \mathbb{E}_{\sigma} \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q}; \sigma)} \frac{p_{\bar{\theta}}(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)} \log p_{\theta}(\mathbf{o}|\mathbf{q}). \end{aligned}$$

1211 Combined with (9), we can see that subtracting  $\frac{p_{\bar{\theta}}(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)}$  from the weight does not change the  
 1212 gradient of the CE loss, i.e.,

$$\nabla_{\theta} \text{KL}(p_{*}(\cdot|\mathbf{q})\|p_{\theta}(\cdot|\mathbf{q})) = \nabla_{\theta} \mathbb{E}_{\sigma} \mathbb{E}_{p_v(\mathbf{o}|\mathbf{q}; \sigma)} \left( \frac{p_{*}(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)} - \lambda \frac{p_{\bar{\theta}}(\mathbf{o}|\mathbf{q}; \sigma)}{p_v(\mathbf{o}|\mathbf{q}; \sigma)} \right) [-\log p_{\theta}(\mathbf{o}|\mathbf{q})], \forall \lambda \in \mathbb{R}.$$

1213 Theoretically, there is an optimal choice of  $\lambda$  that minimizes the variance. The natural choice of  
 1214  $\lambda = 1$  means implicitly matching the probability  $p_{\theta}(\mathbf{o}|\mathbf{q}; \sigma)$  to fit  $p_{*}(\mathbf{o}|\mathbf{q}; \sigma)$ , which corresponds  
 1215 to our model weight baseline (15). When the frequency for sampling buffer  $F$  is small, we can  
 1216 assume  $p_{\theta}(\mathbf{o}|\mathbf{q}; \sigma)$  does not deviate too much from  $p_v(\mathbf{o}|\mathbf{q}; \sigma)$ , thus this ratio should be close to  
 1217 1, which corresponds to our group weight baseline (13). Finally, as we actually use the negative  
 1218 ELBO  $\mathcal{L}_{\theta}(\mathbf{o}|\mathbf{q})$  instead of  $-\log p_{\theta}(\mathbf{o}|\mathbf{q})$  in computing the loss, the variance reduction only holds  
 1219 *approximately*.

### 1220 B.4 PROOFS FOR THE WEIGHTED DIRECT DISCRIMINATIVE OPTIMIZATION OBJECTIVE

1221 For notational simplicity, we ignore the conditional dependence on  $\mathbf{q}$ . Write

$$\mathcal{F}(p_{\theta}) = -\mathbb{E}_{p_{*}} \log \frac{p_{\theta}}{p_{\theta} + p_v} - \mathbb{E}_{p_v} \log \frac{p_v}{p_{\theta} + p_v}.$$

1222 For any fixed  $\mathbf{o}$ , consider the function

$$p_{\theta}(\mathbf{o}) \mapsto -p_{*}(\mathbf{o}) \log \frac{p_{\theta}(\mathbf{o})}{p_{\theta}(\mathbf{o}) + p_v(\mathbf{o})} - p_v(\mathbf{o}) \log \frac{p_v(\mathbf{o})}{p_{\theta}(\mathbf{o}) + p_v(\mathbf{o})}.$$

1223 The derivative with respect to  $p_{\theta}(\mathbf{o})$  is  $-\frac{p_{*}(\mathbf{o})}{p_{\theta}(\mathbf{o})} + \frac{p_{*}(\mathbf{o}) + p_v(\mathbf{o})}{p_{\theta}(\mathbf{o}) + p_v(\mathbf{o})}$ , which is  $> 0$  if  $p_{\theta}(\mathbf{o}) > p_{*}(\mathbf{o})$  and  
 1224  $< 0$  if  $p_{\theta}(\mathbf{o}) < p_{*}(\mathbf{o})$ . Therefore, this function is minimized at  $p_{\theta}(\mathbf{o}) \leftarrow p_{*}(\mathbf{o})$ , which completes the  
 1225 proof.

---

## 1242 C DETAILS OF EXPERIMENTS AND FURTHER RESULTS

### 1244 C.1 INTRODUCTION OF DATASETS AND REWARDS USED

1246 To ensure a fair comparison, we use the same datasets and training rewards as d1 (Zhao et al., 2025a).  
 1247 For a self-contained presentation, we list the datasets and the rewards below.

1249 **GSM8K.** GSM8k (Cobbe et al., 2021) is a mathematical reasoning dataset featuring multi-step  
 1250 grade school math problems. We conduct fine-tuning on the train split and evaluate on the test split.<sup>3</sup>

1251 The reward is decomposed as follows:

- 1253 *XML Structure Reward*: +0.125 for each correctly placed opening and closing tag  
     (<reasoning>, </reasoning>, <answer>, </answer>) and -0.001 for each extra token after the closing tag </answer>.
- 1256 *Soft Format Reward*: +0.5 for responses matching the pattern  
     <reasoning>...</reasoning><answer>...</answer>.
- 1258 *Strict Format Reward*: +0.5 for matching the specified format precisely with correct line  
     breaks.
- 1260 *Integer Answer Reward*: +0.5 if the retrieved answer parses as an integer.
- 1262 *Correctness Reward*: +2 if the returned answer equals the ground truth exactly.

1264 **MATH500.** MATH500 (Lightman et al., 2023) is a mathematical reasoning dataset, as well as a  
 1265 curated collection of 500 high-school-level problems sampled from the MATH (Hendrycks et al.,  
 1266 2021) dataset. We conduct fine-tuning on the train split and evaluate on the test split.<sup>4</sup>

1267 The reward comprises

- 1269 *Format Reward*: 1 when answer tags are present and \boxed appears inside them; 0.75  
     when the tags are present but \boxed is absent; 0.50 when the tags are missing but  
     \boxed is present; 0.25 when neither the tags nor \boxed appear.
- 1272 *Correctness Reward*: +2 when the correct answer is enclosed in \boxed{ }.

1274 **Countdown.** Countdown (Pan et al., 2025) is a planning task that requires solving a combinatorial  
 1275 arithmetic challenge, which is to form a target number using basic arithmetic operations with a  
 1276 provided set of 3 numbers, where each number can only be used once. We train on the training split  
 1277 of the dataset from the TinyZero project (Pan et al., 2025), restricting to instances that use only three  
 1278 numbers, and evaluate on 256 synthetically generated countdown questions with three numbers.

1279 The reward checks if an arithmetic expression constructed from given numbers reaches a target value.  
 1280 More specifically, it is 1 when the equation equals the target and uses exactly the available numbers,  
 1281 0.1 when the equation uses the right numbers but does not reach the target, and 0 if otherwise.

1282 **Sudoku.** Sudoku is a planning task that requires solving  $4 \times 4$  Sudoku puzzles, which demand  
 1283 constraint satisfaction and logical elimination to correctly fill the grid. We use the training dataset  
 1284 from <https://github.com/Black-Phoenix/4x4-Sudoku-Dataset>, in particular, the  
 1285 subset containing one million unique puzzles, which was synthetically generated using code from  
 1286 Arel (2025). For evaluation purposes, we randomly generate 256 Sudoku puzzles using this generator.  
 1287 The reward equals the fraction of originally blank cells that the model fills correctly.

### 1289 C.2 TRAINING HYPERPARAMETERS AND EVALUATION

1291 We choose the training hyperparameters following Zhao et al. (2025a) for a fair comparison. We also  
 1292 use the Transformer Reinforcement Learning library (TRL, von Werra et al. (2020) to implement  
 1293 DMPO. During training, we also employed the same Low-Rank Adaptation (LoRA, Hu et al. (2022))

---

1294 <sup>3</sup><https://huggingface.co/datasets/openai/gsm8k>

1295 <sup>4</sup><https://huggingface.co/datasets/ankner/math-500>

1296 with a rank of  $r = 128$  and scaling factor  $\alpha = 64$ . For all tasks, the training was conducted on 8  
 1297 NVIDIA H100 or H200 GPUs with the hyperparameters described below.  
 1298

1299 We use a maximum generation length 256 tokens, a batch size of 8 per GPU, and gradient accumulation  
 1300 steps of 2, and 16 generated rollouts per prompt. We optimized the model using the AdamW  
 1301 optimizer (Loshchilov & Hutter, 2019) with parameters  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ , weight decay of 0.1,  
 1302 learning rate of  $3 \times 10^{-6}$ , and gradient clipping at 0.2. For each clean sequence, we sampled 4  
 1303 partially masked tokens to compute the WDCE/WDDO loss. For rollouts generation during training,  
 1304 we use a semi-autoregressive random order sampler (with temperature 0) and fast-dLLM (with  
 1305 temperature 0.2) with a block size of 32 to generate diverse responses, which is the recommended  
 1306 practice for using LLaDA series models as is described in Nie et al. (2025b). We train 4,000 steps  
 1307 (number of gradient updates) for GSM8K and MATH500, Countdown, and Sudoku, respectively.

1308 For the reproduction of the d1 results, we follow the guidelines listed in Zhao et al. (2025a) and  
 1309 first perform SFT on s1k (Muennighoff et al., 2025) before applying diffu-GRPO. We use the  
 1310 recommended hyperparameter setups and train for up to 13,000 iterations on each dataset before  
 1311 evaluating the results.

1312 For computational efficiency, we use Flash Attention 2 (Dao, 2024) and 4-bit quantization. All  
 1313 experiments on DMPO share these hyperparameters. The main result reported in Tab. 1 used the  
 1314 group weight baseline defined in (13). The ablation study in Fig. 5 also follows the same set of  
 1315 hyperparameters above, except for using different choices of weight baselines.

1316 For the evaluation of all model checkpoints, we consider three different generation lengths: 128,  
 1317 256, and 512. We correspondingly use 128, 256, and 512 steps for generation. For the LLaDA  
 1318 series of models, such as LLaDA-Instruct, LLaDA-1.5, d1-LLaDA, and our own DMPO-LLaDA,  
 1319 we employ the semi-autoregressive sampler with a block size of 32, a greedy decoding scheme with  
 1320 a temperature of 0, and the top- $k$  remasking scheme to achieve the best inference results. For the  
 1321 Dream model, we also employ the recommended practice and perform inference with temperature  
 0.95 and the top- $k$  remasking scheme.

### 1323 C.3 FURTHER EXPERIMENTAL RESULTS

1325 **Ablation studies on the hyperparameter dependence.** We provide an ablation study on two of  
 1326 the main hyperparameters in Alg. 1, namely the number of rollouts  $N$  and the frequency for sampling  
 1327 buffer  $F$ , in Figs. 7 and 8, respectively. For each run shown in Fig. 7, we train for 6 hours using  
 1328 8 NVIDIA H200 GPUs. For each run shown in Fig. 8, we train for 8 hours with 8 NVIDIA H200  
 1329 GPUs. We only vary the resampling buffer frequency  $F$  and the number of rollouts sampled per  
 1330 prompt  $N$ , while fixing other hyperparameters, such as the total effective batch size, to maintain a  
 1331 fair comparison.

1332 For the number of rollouts per question  $N$ , we observe that a larger number of  $N$  does not necessarily  
 1333 lead to longer training time, even with the same number of steps, due to the parallelism of the  
 1334 generation process, since we kept the total batch size fixed while varying the hyperparameter  $N$ . The  
 1335 algorithm is robust across various values of  $N$  ranging from 4 to 32 thanks to the mechanism for  
 1336 inserting negative gradients.

1337 For the buffer sampling frequency  $F$ , we observe that it significantly affects training speed. The  
 1338 figure clearly demonstrates the advantage of DMPO due to its *off-policy* nature, whereas a purely  
 1339 on-policy realization of WDCE loss (with  $F = 1$ ) is not only extremely slow but also does not show  
 1340 a significant boost in per-step reward gains. The figure also underscores the unique benefit of WDCE  
 1341 being a *forward* loss: given the generated rollouts and their weights, one can train using the simple  
 1342 forward process via random masking. Our algorithm is robust to choices of  $F$  up to 24, whereas an  
 1343 even larger  $F$  may cause slight instability later in training when the reward is high.

1344  
 1345 **Visualizing rollout entropy of DMPO** In Fig. 9, we compare the reward and entropy of the  
 1346 generated rollouts during training for both the relative-entropy-based (diffu-GRPO) RL algorithm  
 1347 and the cross-entropy-based (DMPO) RL algorithm. Here, in both experiments, we fix  $N = 16$   
 1348 and  $F = 8$  and evaluate the entropy of generated samples every 10 generations. The evaluation of  
 1349 entropy is as follows: we use random-order autoregressive generation with block length 32, and at  
 the  $d$ -th step of unmasking (where  $d$  ranges from 1 to  $D = |\mathcal{O}|$ ), we compute the entropy of the

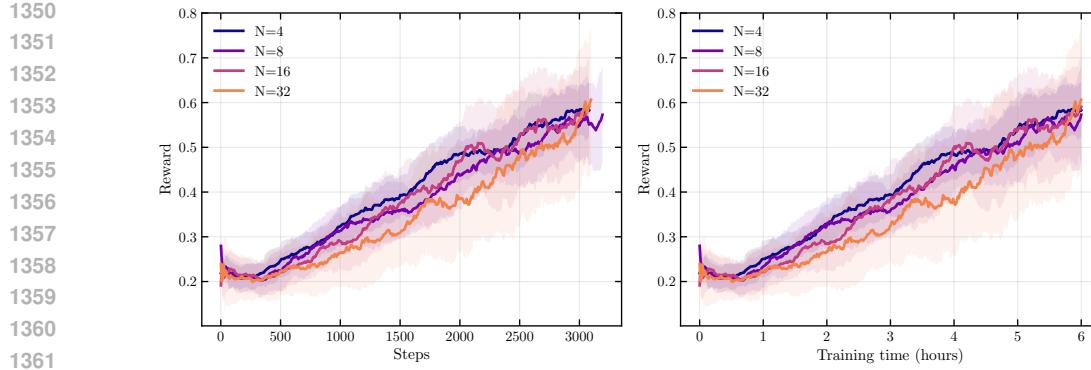


Figure 7: **Ablation study of the number of rollouts per prompt  $N$  on Countdown dataset under the same training time and compute.** The performance is robust to this hyperparameter.

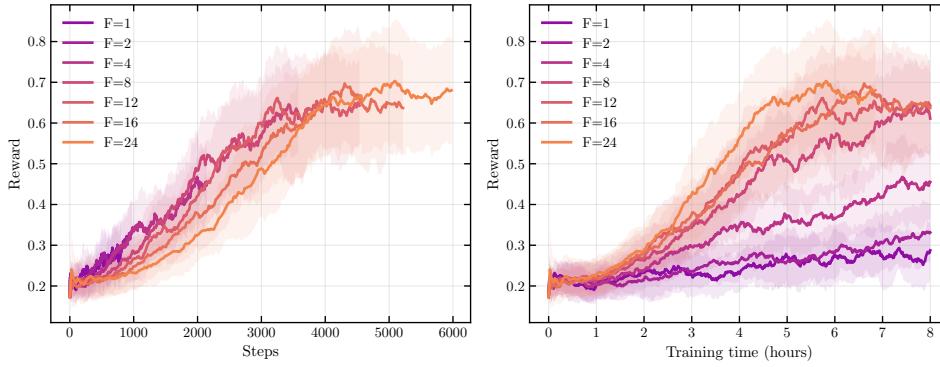


Figure 8: **Ablation study of the resampling frequency  $F$  on Countdown dataset.** A larger  $F$  is generally more time-efficient though may cause instability when the reward is high.

predicted logits at the  $d$ -th position, and take average of all the  $D$  entropy values as the final value of sequential entropy. From the figure, the trend of consistently higher sample entropy for WDCE loss than for diffu-GRPO agrees with our expectation that cross-entropy-based methods are less prone to mode-seeking and maintain a higher level of diversity throughout training.

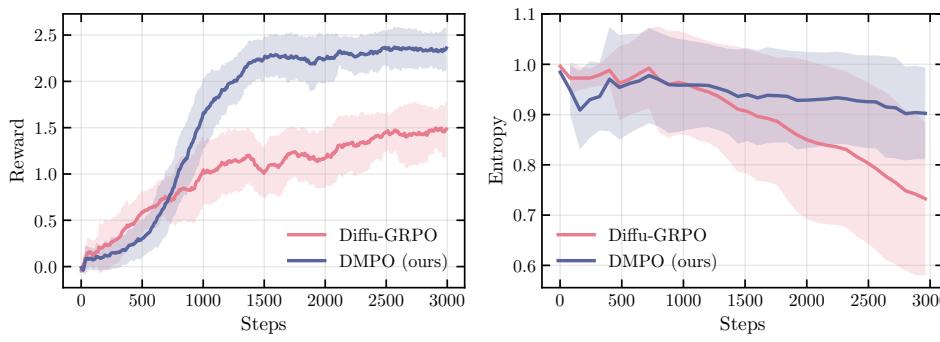


Figure 9: **Reward (left) and entropy (right) of generated rollouts during training for both diffu-GRPO (Zhao et al., 2025a) and DMPO.**

#### C.4 EXAMPLE OUTPUTS OF THE MODEL AFTER FINE-TUNING

We present two example outputs of the DMPO-LLaDA model in the following.

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

**Model:** DMPO-LLaDA**Dataset:** MATH500**Generation Sequence Length:** 512**Question:** The product of  $3t^2 + 5t + a$  and  $4t^2 + bt - 2$  is  $12t^4 + 26t^3 - 8t^2 - 16t + 6$ . What is  $a + b$ ?

1421

To find the values of  $a$  and  $b$  such that the product of  $3t^2 + 5t + a$  and  $4t^2 + bt - 2$  equals  $12t^4 + 26t^3 - 8t^2 - 16t + 6$ , we need to expand the left-hand side and compare coefficients with the right-hand side.

First, expand the product:

$$(3t^2 + 5t + a)(4t^2 + bt - 2)$$

Using the distributive property (FOIL method for binomials):

$$3t^2 \cdot 4t^2 + 3t^2 \cdot bt + 3t^2 \cdot (-2) + 5t \cdot 4t^2 + 5t \cdot bt + 5t \cdot (-2) + a \cdot 4t^2 + a \cdot bt + a \cdot (-2)$$

Simplify each term:

$$12t^4 + 3bt^3 - 6t^2 + 20t^3 + 5bt^2 - 10t + 4at^2 + abt - 2a$$

Combine like terms:

$$12t^4 + (3b + 20)t^3 + (-6 + 5b + 4a)t^2 + (-10 + ab)t - 2a$$

We need this to equal:

$$12t^4 + 26t^3 - 8t^2 - 16t + 6$$

By comparing coefficients, we get the following system of equations:

$$3b + 20 = 26 \implies 3b = 6 \implies b = 2$$

$$-6 + 5(2) + 4a = -8 \implies 4 + 4a = -8 \implies 4a = -12 \implies a = -3$$

Thus,  $a = -3$  and  $b = 2$ .

$$a + b = -3 + 2 = -1$$

</reasoning><answer> -1 </answer>

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461 **Model:** DMPO-LLaDA1462 **Dataset:** GSM8K1463 **Generation Sequence Length:** 5121464 **Question:** Katarina has 5 less cookies than Max has. Max has 12 more cookies than the  
1465 Cookie Monster, and Summer has 23 more cookies than Max. If Katarina has 68 cookies,  
1466 how many cookies do they have in total?1467 To determine the total number of cookies Katarina, Max, and Summer have, we need to find  
1468 out how many cookies each person has and then sum them up.

1469 1. We know that Katarina has 68 cookies.

1470 2. Katarina has 5 less cookies than Max.

1471 3. Max has 12 more cookies than the Cookie Monster.

1472 4. Summer has 23 more cookies than Max.

1473 Let's denote the number of cookies each person has as follows:

1474 -  $K$  for Katarina's cookies1475 -  $M$  for Max's cookies1476 -  $C$  for the Cookie Monster's cookies1477 -  $S$  for Summer's cookies

1478 From the problem, we have:

1479 -  $K = 68$ 1480 -  $K = M - 5$ 1481 -  $M = C + 12$ 1482 -  $S = M + 23$ 1483 First, we substitute the known value of  $K$  into the equations:1484 -  $68 = M - 5$ 1485 -  $M = C + 12$ 1486 -  $S = M + 23$ 1487 First, solve for  $M$ :

1488 
$$68 = M - 5$$

1489 
$$M = 68 + 5$$

1490 
$$M = 73$$

1491 Next, solve for  $C$ :

1492 
$$73 = C + 12$$

1493 
$$C = 73 - 12$$

1494 
$$C = 61$$

1495 Finally, solve for  $S$ :

1496 
$$73 = S - 23$$

1497 
$$S = 73 + 23$$

1498 
$$S = 96$$

1499 Now, we have the number of cookies each person has:

1500 - Katarina: 68 cookies

1501 - Max: 73 cookies

1502 - The Cookie Monster: 61 cookies

1503 - Summer: 96 cookies

1504 To find the total number of cookies, we sum these amounts:

1505 
$$68 + 73 + 61 + 96 = 298$$

1506 Thus, the total number of cookies is 298.1507 </reasoning><answer>298</answer>

1508

1509

1510

1511