dFlowGRPO: Rate-Aware Policy Optimization for Discrete Flow Models

Anonymous authors
Paper under double-blind review

Abstract

Discrete flow models (DFMs) are a class of flexible generative models for generating discrete data, and diffusion large language models (dLLMs) can be viewed as a special case with a specific choice of a mixture path and a masked source distribution. While several recent works have explored reinforcement learning for dLLMs, its application to more general discrete flow models remains underexplored. In this work, we present discrete Flow-GRPO (dFlowGRPO), a unified reinforcement learning framework for discrete flow models that supports a broad family of probability paths and non-masked source distributions. We derive the full trajectory probability for DFMs and formulate the denoising process as a Markov decision process, enabling dFlowGRPO to incorporate information from both the associated conditional transition rates and the posterior model during reinforcement learning. We apply dFlowGRPO to FUDOKI, a recent multimodal discrete flow model, and evaluate it on both image generation and multimodal understanding tasks. Empirical results show that dFlowGRPO outperforms existing GRPO-type methods for dLLMs on text-to-image generation tasks and achieves performance competitive with continuous flow-based models trained using Flow-GRPO, while also demonstrating strong capabilities on understanding tasks.

1 Introduction

Diffusion large language models (dLLMs) (Nie et al., 2025a;b; Ye et al., 2025) are a popular and powerful paradigm in text generation, speeding up the inference process by unmasking tokens in parallel, as an alternative to autoregressive models. Discrete flow models (DFMs) (Campbell et al., 2024; Gat et al., 2024; Shaul et al., 2025) provide a flexible matching-based framework to learn the denoising process from a source distribution to the data distribution, by marginalizing the conditional transition rate similar to continuous flow matching (Albergo & Vanden-Eijnden, 2023; Liu et al., 2023; Lipman et al., 2023), offering a larger design space than dLLMs. Specifically, dLLMs are a special case of DFMs under a specific choice of a mixture probability path and a conditional transition rate.

To improve the performance of dLLMs, several works on reinforcement learning for dLLMs have recently emerged. Since the log-likelihood of dLLMs cannot be decomposed naturally like autoregressive models, the existing works on RL for dLLMs mainly focus on the estimation of the log-likelihood of dLLMs, such as mean-field approximation (Zhao et al., 2025a) and ELBO-based approximation (Zhu et al., 2025; Ou et al., 2025a). Despite the success of these RL methods for dLLMs, no existing work focuses on RL for DFMs and takes general probability paths and conditional rates into account. In this work, we propose dFlowGRPO, a unified rate-aware policy optimization framework for DFMs by formulating the denoising process as a Markov decision process, parallel to Flow-GRPO (Liu et al., 2025a). By deriving the probability of full trajectory, we find that the transition probability ratio of the denoising process of DFMs can be written as a product of the token-wise expected posterior ratio weighted by a rate-dependent term, which can be estimated efficiently using MC samples without additional forward passes. Notably, the proposed GRPO objective incorporates both the information of conditional transition rate and posterior model, providing an effective signal for policy optimization. See figure 1 for an overview of dFlowGRPO.

Empirically, we evaluate the unified dFlowGRPO framework on both image generation and multimodal understanding tasks, using FUDOKI (a multimodal DFM, Wang et al., 2025b) as our base

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

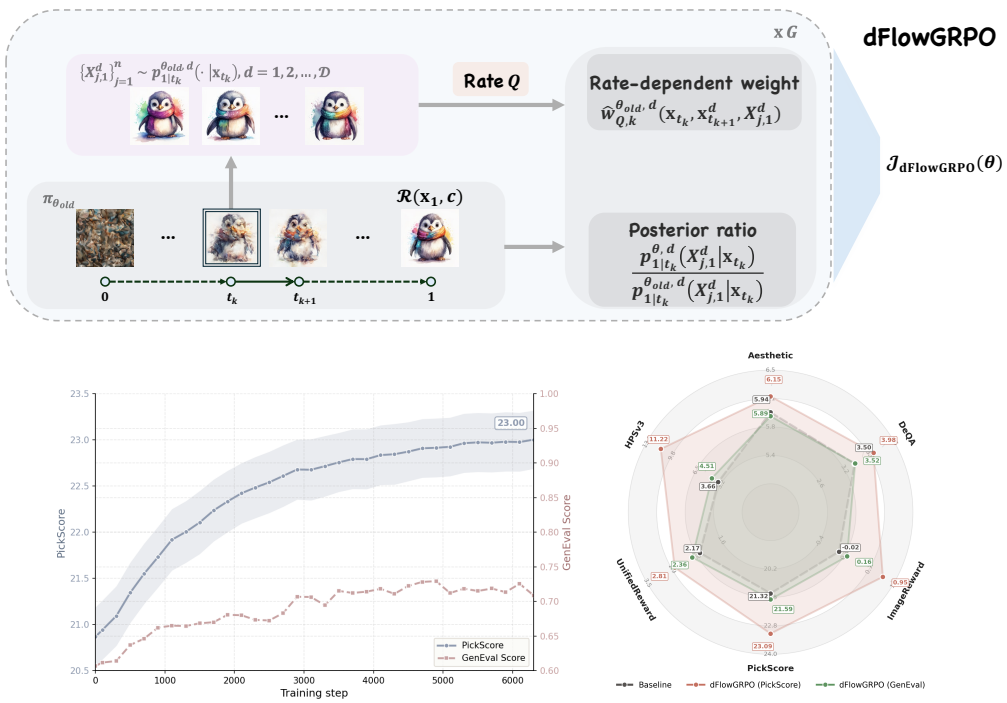

Figure 1: (Top) Overview of the proposed dFlowGRPO framework ((2)). At each denoising step, we sample MC samples from posterior model for computing rate-dependent weight and posterior ratio, and then we calculate the rate-aware transition probability ratio by (4). (Bottom, Left) PickScore and GenEval Score steadily increase during the training process of dFlowGRPO trained with PickScore reward without KL regularization. (Bottom, Right) The performance of dFlowGRPO trained with PickScore and GenEval reward without KL regularization, evaluated by other metrics on DrawBench prompts (Saharia et al., 2022). (These results use 8 NFEs for training and 20 NFEs for evaluation.)

model. For text-to-image generation, we use human preference PickScore (model-based reward, Kirstain et al., 2023) and GenEval (verifiable reward, Ghosh et al., 2023) as our reward models for online RL training. Without classifier-free guidance, dFlowGRPO improves FUDOKI from 20.87 to 23.00 on PickScore and from 61% to 93% on GenEval without evident reward hacking across the reported auxiliary metrics, demonstrating the effectiveness and superior performance of dFlowGRPO on image generation tasks (see figure 1 and table 1). For multimodal understanding, we train the base model on a multimodal dataset, ScienceQA (Lu et al., 2022), with verifiable correctness reward (derived by exact matching and LLM extraction). dFlowGRPO boosts the understanding accuracy from 75.1% to 81.1% on ScienceQA (with LLM extraction), with limited evidence of reward hacking across the reported auxiliary benchmarks (see figure 2). Additionally, we compare dFlowGRPO against other RL methods on FUDOKI, including online DPO (Guo et al., 2024), diffu-GRPO and its variant (Zhao et al., 2025a). Experimental results show that dFlowGRPO consistently outperforms these approaches.

Contribution We summarize our main contributions as follows.

1. We propose dFlowGRPO, a unified rate-aware policy optimization framework for DFMs with general probability path and conditional rate by utilizing the information of both conditional rate and posterior model.
2. We derive the transition probability ratio for the denoising trajectory, which can be written as a product of token-wise expected posterior ratio reweighted by a rate-dependent term. This term can be estimated using MC samples without additional forward passes.
3. We empirically show the effectiveness of dFlowGRPO by applying it to a multimodal DFM. dFlowGRPO improves the reward of base model significantly and consistently on text-to-image generation and multimodal understanding tasks.

2 Related Work

Reinforcement learning for continuous diffusion and flow-based models Reinforcement learning methods, such as RLHF (Ouyang et al., 2022) and RLVR (Lambert et al., 2024), have achieved significant successes in improving the capabilities of AR-based LLMs. Several works have boosted the performance of continuous diffusion and flow-based models via reinforcement learning by developing variants of existing policy gradient methods, such as PPO Schulman et al. (2017) and GRPO Shao et al. (2024); Guo et al. (2025). For example, DPOK (Fan et al., 2023) and DDPO (Black et al., 2023) proposed policy gradient method by formulating the denoising process of diffusion models as a Markov decision process, achieving better performance than reward-weighted methods (Lee et al., 2023). Flow-GRPO Liu et al. (2025a) and DanceGRPO Xue et al. (2025) generalize GRPO to continuous Flow-based models by converting ODE to the equivalent SDE to derive transition probability. For DPO variants, Wallace et al. (2024) proposed Diffusion-DPO to improve text-to-image performance of diffusion models with a tractable DPO objective. Liu et al. (2025b) generalize DPO to continuous flow-based models for video generation.

Discrete diffusion and flow-based models As an alternative to AR-based models, discrete diffusion models (Austin et al., 2021; Hoogeboom et al., 2022; Campbell et al., 2022; Sun et al., 2023; Lou et al., 2024) aim to learn a denoising process, usually modeled as a continuous-time Markov chain (CTMC), which can transport a source distribution to the data distribution, parallel to the continuous counterparts (Ho et al., 2020; Song et al., 2021b;a). Diffusion large language models (Nie et al., 2025a;b) have shown promising results in language modeling, achieving performance comparable to that of autoregressive models. Instead of learning the time-reversal of the forward process in diffusion-based models, discrete flow-based models (Gat et al., 2024; Shaul et al., 2025; Campbell et al., 2024) offer a flexible framework for learning a transition rate by marginalizing a pre-specified conditional transition rate (velocity), providing a larger design space than discrete diffusion models, similar to continuous flow matching (Liu et al., 2023; Albergo & Vanden-Eijnden, 2023; Lipman et al., 2023).

Reinforcement learning for dLLMs The main challenge of RL for dLLMs is the log-likelihood estimation, which cannot be naturally decomposed as in autoregressive models. To tackle this issue, existing works have proposed various methods for estimating the log-likelihood, including mean-field approximation (Zhao et al., 2025a) and ELBO approximation (Zhu et al., 2025; Bie et al., 2025; 2026; Ma et al., 2026; Ou et al., 2025a). Additionally, several variants and efficient strategies for ELBO-based methods have been developed, such as token-level ELBO-based methods (Yang et al., 2025; Gong et al., 2026), complementary masking (Li et al., 2025; 2026), and semi-deterministic MC (Rojas et al., 2026). Tang et al. (2026) proposed an advantage-weighted log-likelihood maximization approach for RL. Huang et al. (2025) introduced diffusion chain-of-lateral-thought for dLLMs by optimizing the entire trajectory with a GRPO-based method. By formulating the denoising process of dLLMs as a Markov decision process, Wang et al. (2025a) and Zhang et al. (2026) derived the exact transition probability ratio for dLLMs and developed GRPO- and DPO-based methods, respectively. Wang et al. (2026) accelerated RL training by aggregating neighboring steps and incorporated a value model to improve training stability.

3 Preliminary

In this section, we briefly introduce discrete flow-based models and formulate the denoising process as a Markov decision process.

3.1 Discrete Flow Models

Consider a data distribution q_1 in a discrete space $\mathcal{S}^{\mathcal{D}}$, where $\mathcal{S}$ is the vocabulary and $\mathcal{D}$ is the sequence length. We call a stochastic process $(\mathbf{x}_t)_{t\in[0,1]}$ a continuous-time Markov chain (CTMC) on $\mathcal{S}^{\mathcal{D}}$ if it satisfies the Markov property and for any $x, z \in \mathcal{S}^{\mathcal{D}}$

$$\mathbb{P}(\mathbf{x}_{t+h} = z|\mathbf{x}_t = x) = \delta_x(z) + Q_t(x, z)h + o(h), \tag{1}$$

where $(Q_t(x, z))_{x,z\in\mathcal{S}^{\mathcal{D}}}$ is the transition rate matrix satisfying the rate properties: $\sum_{z\in\mathcal{S}^{\mathcal{D}}} Q_t(x, z) = 0$ and $Q_t(x, z)\mathbf{1}(x \neq z) \geq 0$ for any $x, z \in \mathcal{S}^{\mathcal{D}}$. Here, $\mathbf{1}(\cdot)$ is the

indicator function. Denote $(q_t)_{t\in[0,1]}$ as the marginal probability distribution (path) corresponding to $(\mathbf{x}_t)_{t\in[0,1]}$. We say Q_t can generate q_t if it satisfies the Kolmogorov forward equation $\partial_t q_t(z) = \sum_{x\in\mathcal{S}^\mathcal{D}} q_t(x)Q_t(x,z)$ for $t \in [0,1]$.

Discrete flow models aim to learn a transition rate Q_t that can transport a source distribution q_0 (e.g., a uniform distribution or a point mass at the mask token) to the data distribution q_1 by marginalizing a conditional transition rate $Q_t(x,z|x_1)$, parallel to continuous flow matching Liu et al. (2023); Albergo & Vanden-Eijnden (2023). Specifically, given a pre-defined conditional probability path $q_{t|1}(x|x_1)$ and a conditional transition rate $Q_t(x,z|x_1)$ that can generate $q_{t|1}$ conditional on x_1, the marginal transition rate $Q_t(x,z) = \sum_{x_1\in\mathcal{S}^\mathcal{D}} Q_t(x,z|x_1)q_{1|t}(x_1|x)$ can generate the marginal probability path q_t (see Proposition 3.1 in Campbell et al., 2024), where $q_{1|t}$ is the posterior corresponding to $q_{t|1}$. For computational tractability, the conditional probability path and the associated transition rate are usually restricted to factorize across dimensions conditional on the terminal state $\mathbf{x}_1$ (that is, $q_{t|1}(x|x_1) = \prod_{d=1}^{\mathcal{D}} q^d_{t|1}(x^d|x^d_1)$ and $Q_t(x,z|x_1) = \sum_{d=1}^{\mathcal{D}} \delta_{x^{\backslash d}}(z^{\backslash d})Q^d_t(x^d,z^d|x^d_1)$), which implies that the marginal transition rate matrix is sparse:

$$Q_t(x,z) = \sum_{d\in[\mathcal{D}]} \delta_{x^{\backslash d}}(z^{\backslash d})Q^d_t(x,z^d),$$

where $Q^d_t(x,z^d) = \sum_{x^d_1\in\mathcal{S}} Q^d_t(x^d,z^d|x^d_1)q^d_{1|t}(x^d_1|x)$. In particular, choosing the mixture path $q^d_{t|1}(x^d|x^d_1) = \kappa_t\delta_{x^d_1}(x^d) + (1-\kappa_t)\delta_{\mathbf{m}}(x^d)$ with the conditional rate $Q^d_t(x^d,z^d|x^d_1) = \frac{\dot\kappa_t}{1-\kappa_t}(\delta_{x^d_1}(z^d) - \delta_{x^d}(z^d))$, we can recover the masked diffusion models (Shi et al., 2024; Ou et al., 2025b; Sahoo et al., 2024; Nie et al., 2025b), where κ_t is the time scheduler.

Training In practice, we can parametrize the posterior $q_{1|t}$ using a neural network $p^{\theta,d}_{1|t}$. For general conditional probability paths, a generally applicable training objective is the following cross-entropy loss (Campbell et al., 2024; Shaul et al., 2025; Gat et al., 2024; Wang et al., 2025b)

$$\mathcal{L}(\theta) = -\mathbb{E}_{\mathbf{x}_1\sim q_1(\cdot),\mathbf{x}_t\sim q_{t|1}(\cdot|\mathbf{x}_1)}[\sum_{d\in[\mathcal{D}]} \log p^{\theta,d}_{1|t}(\mathbf{x}^d_1|\mathbf{x}_t)].$$

Sampling To improve sampling efficiency for general probability paths and to tackle the ill-defined issue of (1) when the step size h is not sufficiently small, one can use the following always-valid Euler sampler[1] at each denoising step Shaul et al. (2025); Wang et al. (2025b):

1. Given $\mathbf{x}_t$, for each $d \in [\mathcal{D}]$, sample $X^d_1 \sim p^{\theta,d}_{1|t}(\cdot|\mathbf{x}_t)$ in parallel, where $[\mathcal{D}] = \{1,2,\dots,\mathcal{D}\}$;
2. For each $d \in [\mathcal{D}]$, set $\mathbf{x}^d_{t+h} = \mathbf{x}^d_t$ with probability $\exp(-h\lambda^d_t(\mathbf{x}^d_t, X^d_1))$ or set $\mathbf{x}^d_{t+h} = z^d \neq \mathbf{x}^d_t$ with probability $\left\{1-\exp(-h\lambda^d_t(\mathbf{x}^d_t,X^d_1))\right\}\frac{Q^d_t(\mathbf{x}^d_t,z^d|X^d_1)}{\lambda^d_t(\mathbf{x}^d_t,X^d_1)}$, where $\lambda^d_t(\mathbf{x}^d_t,X^d_1) = \sum_{z^d\neq\mathbf{x}^d_t} Q^d_t(\mathbf{x}^d_t,z^d|X^d_1)$.

The details of the Euler solver can be found in algorithm 1.

3.2 Denoising as a Markov Decision Process

Consider a time discretization scheme $0 = t_0 < t_1 \cdots < t_K = 1$. Let $p(\mathbf{x}_{t_{k+1}}|\mathbf{x}_{t_k},\mathbf{c})$ be the transition probability at the k-th timestep given a prompt (or context) $\mathbf{c}$. Following DDPO (Black et al., 2023) and Flow-GRPO (Liu et al., 2025a), we can formulate the denoising process as a Markov Decision Process defined by the tuple $(\mathbf{S},\mathcal{A},P,r,\rho_0)$, where $\mathbf{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $P : \mathbf{S}\times\mathcal{A}\times\mathbf{S}\to\mathbb{R}$ is the transition probability, $r : \mathbf{S}\times\mathcal{A}\to\mathbb{R}$ is the reward function and $\rho_0 : \mathbf{S}\to\mathbb{R}$ is the distribution of the initial state s_0 (considering a constant discount factor for simplicity). At the k-th step with a state $\mathbf{s}_k = (\mathbf{c},k,\mathbf{x}_{t_k})$, an agent takes a stochastic action $\mathbf{a}_k = \mathbf{x}_{t_{k+1}}$ with probability $\pi(\mathbf{a}_k|\mathbf{s}_k) = p(\mathbf{x}_{t_{k+1}}|\mathbf{x}_{t_k},\mathbf{c})$ and receives a reward $R(\mathbf{s}_k,\mathbf{a}_k)$, moving to the next state

[1]Refer to the official flow matching PyTorch library: `https://github.com/facebookresearch/flow_matching`

$\mathbf{s}_{k+1}$ with deterministic transition $P(\mathbf{s}_{k+1}|\mathbf{s}_k, \mathbf{a}_k) = (\delta_{\mathbf{c}}, \delta_{k+1}, \delta_{\mathbf{x}_{t_{k+1}}})$. In continuous diffusion or flow-based models, it is common to take $\rho_0(\mathbf{s}_0) = (p_c(\mathbf{c}), \delta_0, \mathcal{N}(\mathbf{0}, \mathbf{I}))$ and assign the reward of the entire denoising trajectory to the terminal state; that is, $R(\mathbf{s}_k, \mathbf{a}_k) = \mathcal{R}(\mathbf{x}_1, \mathbf{c})$ if $k = K-1$, and 0 otherwise. The goal of RL is to maximize the expected return $\mathbb{E}_{\mathbf{c}\sim p_c, \pi_\theta(\cdot|\mathbf{c})}[\sum_{k=0}^{K-1} R(\mathbf{s}_k, \mathbf{a}_k)]$, which has the gradient $\mathbb{E}_{\mathbf{c}\sim p_c, \pi_\theta(\cdot|\mathbf{c})}[\sum_{k=0}^{K-1} \nabla_\theta \log p_\theta(\mathbf{x}_{t_{k+1}}|\mathbf{x}_{t_k}, \mathbf{c})\mathcal{R}(\mathbf{x}_1, \mathbf{c})]$.

4 DISCRETE FLOW-GRPO

In this section, we introduce Discrete Flow-GRPO (dFlowGRPO), a unified reinforcement learning framework for discrete flow models based on the GRPO framework Shao et al. (2024); Guo et al. (2025).

4.1 GRPO FOR DISCRETE FLOW MODELS

Given a prompt $\mathbf{c} \sim p_c$ and the number of denoising steps K, we can generate G final samples $\{\mathbf{x}_1^{(i)}\}_{i\in[G]}$ using the Euler solver discussed in the previous section. Let $\{(\mathbf{x}^{(i)})\}_{i\in[G]} \triangleq \{(\mathbf{x}_{t_0}^{(i)}, \mathbf{x}_{t_1}^{(i)}, \ldots, \mathbf{x}_{t_K}^{(i)})\}_{i\in[G]}$ be the corresponding denoising trajectory. Since the denoising process can be formulated as a Markov decision process and the full trajectory probability can be decoupled $(p(\mathbf{x}^{(i)}) = p(\mathbf{x}_0^{(i)}) \prod_{k=0}^{K-1} p(\mathbf{x}_{t_{k+1}}^{(i)}|\mathbf{x}_{t_k}^{(i)}, \mathbf{c}))$, following Flow-GRPO (Liu et al., 2025a), we can use the following training objective to optimize the policy model:

$$\mathcal{J}_{\text{dFlowGRPO}}(\theta) = \mathbb{E}_{\mathbf{c}\sim p_c, \{\mathbf{x}^{(i)}\}_{i=1}^{G}\sim\pi_{\theta_{old}}(\cdot|\mathbf{c})} \Big\{ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{k=0}^{K-1}\Big(\min([r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}}\hat{A}_k^{(i)}, \text{clip}([r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}}, 1-\epsilon, 1+\epsilon)\hat{A}_k^{(i)}) - \beta D_{KL}(\pi_\theta\|\pi_{ref})\Big)\Big\}, \tag{2}$$

where ϵ is the clipping parameter, $r_k^{(i)}(\theta)$ is the step-level transition probability ratio between the current policy and the old policy and the estimated advantage $\hat{A}_k^{(i)}$ can be computed using the normalized reward in the group, which is independent of k:

$$r_k^{(i)}(\theta) = \frac{p_\theta(\mathbf{x}_{t_{k+1}}^{(i)}|\mathbf{x}_{t_k}^{(i)}, \mathbf{c})}{p_{\theta_{old}}(\mathbf{x}_{t_{k+1}}^{(i)}|\mathbf{x}_{t_k}^{(i)}, \mathbf{c})}; \quad \hat{A}_k^{(i)} = \frac{\mathcal{R}(\mathbf{x}_1^{(i)}, \mathbf{c}) - \text{mean}(\{\mathcal{R}(\mathbf{x}_1^{(i)}, \mathbf{c})\}_{i\in[G]})}{\text{std}(\{\mathcal{R}(\mathbf{x}_1^{(i)}, \mathbf{c})\}_{i\in[G]})}.$$

Remark 1. Since the transition probability ratio can be written as a product of the token-level probability ratio (see (4)), we use the geometric mean of the token-level probability ratio for training stability in (2), which is similar to GSPO (Zheng et al., 2025), GMPO (Zhao et al., 2025b) and ESPO (Ou et al., 2025a). We summarize dFlowGRPO in algorithm 2 and derive its gradient in section C. Similarly, we use the following estimator for KL regularization to improve training stability:

$$\begin{aligned} D_{KL}(\pi_\theta\|\pi_{ref}) &\approx [r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} - \log[r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} - 1 \\ &\le \frac{1}{\mathcal{D}}\sum_{d\in[\mathcal{D}]}\Big\{\frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} - \log\frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} - 1\Big\}, \end{aligned}$$

where the right-hand side is the token-level KL regularization similar to standard GRPO (Shao et al., 2024).

4.2 RATE-AWARE TRANSITION PROBABILITY RATIO

To compute the GRPO objective, we have to calculate the transition probability ratio for the Euler solver (algorithm 1). Following the discussion in Wan et al. (2026) and Liang et al. (2025), in the time interval $[t_k, t_{k+1}]$, the Euler sampler can be viewed as a CTMC with a time-homogeneous transition rate $Q_t^{\text{Euler}}(x, z) = \sum_{d=1}^{\mathcal{D}} \delta_{x^{\backslash d}}(z^{\backslash d})\delta_{\mathbf{x}_{t_k}^d}(x^d)\hat{Q}_{t_k}^d(\mathbf{x}_{t_k}, z^d)$, where $\hat{Q}_{t_k}^d(\mathbf{x}_{t_k}, z^d) = Q_{t_k}^d(\mathbf{x}_{t_k}, z^d|X_1^d)$ is the estimator of the oracle rate using a single MC sample

$X_1^d \sim p_{1|t}^d(\cdot|\mathbf{x}_{t_k})$. We first define the following unnormalized rate-dependent weight:

$$\tilde{w}_{Q,k}^d(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d) = \mathbf{1}(\mathbf{x}_{t_k}^d = \mathbf{x}_{t_{k+1}}^d)\Big(\exp\Big(-(t_{k+1}-t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big) + \mathbf{1}(\mathbf{x}_{t_k}^d \neq \mathbf{x}_{t_{k+1}}^d)\Big(\frac{Q_{t_k}^d(\mathbf{x}_{t_k}^d, \mathbf{x}_{t_{k+1}}^d|X_1^d)}{\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)}\Big\{1-\exp\Big(-(t_{k+1}-t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big\}\Big), \tag{3}$$

where $\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d) = \sum_{z^d \neq \mathbf{x}_{t_k}^d} Q_{t_k}^d(\mathbf{x}_{t_k}^d, z^d|X_1^d)$. We derive the transition probability ratio for the denoising trajectory of DFMs in the following theorem.

Theorem 1. *For sampling with the Euler solver (algorithm 1) and posterior model $p_{1|t}^\theta$, the transition probability is $p_\theta(\mathbf{x}_{t_{k+1}}|\mathbf{x}_{t_k}) = \prod_{d\in[\mathcal{D}]} p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})$ (here, we omit the prompt $\mathbf{c}$ for simplicity), where*

$$p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k}) = \mathbb{E}_{X_1^d \sim p_{1|t_k}^{\theta,d}(\cdot|\mathbf{x}_{t_k})}[\tilde{w}_{Q,k}^d(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d)].$$

Consequently, the transition probability ratio between the current policy and the old policy is

$$r_k(\theta) = \prod_{d\in[\mathcal{D}]} \frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} = \prod_{d\in[\mathcal{D}]} \mathbb{E}_{X_1^d \sim p_{1|t_k}^{\theta_{old}}(\cdot|\mathbf{x}_{t_k})}\Big\{\underbrace{w_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d)}_{\textit{rate-dependent weight}} \underbrace{\frac{p_{1|t_k}^\theta(X_1^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old}}(X_1^d|\mathbf{x}_{t_k})}}_{\textit{posterior ratio}}\Big\}, \tag{4}$$

where the rate-dependent weight (satisfying $\mathbb{E}_{X_1^d \sim p_{1|t_k}^\theta(\cdot|\mathbf{x}_{t_k})}[w_{Q,k}^{\theta,d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d)] = 1$) is defined by

$$w_{Q,k}^{\theta,d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d) = \frac{\tilde{w}_{Q,k}^d(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d)}{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}. \tag{5}$$

Theorem 1 indicates that the transition probability ratio in the denoising trajectory is a product of dimension-wise expected posterior ratios, each reweighted by a rate-dependent term. In particular, for the mixture path with masked source distribution $q_0^d(x) = \delta_{\mathbf{m}}$ and conditional rate $Q_t^d(x^d, z^d|x_1^d) = \frac{\dot{\kappa}_t}{1-\kappa_t}(\delta_{x_1^d}(z^d) - \delta_{x^d}(z^d))$, the transition probability ratio is

$$r_k^{(i)}(\theta) = \prod_{d:\mathbf{x}_{t_k}^d = \mathbf{m} \text{ and } \mathbf{x}_{t_{k+1}}^d \neq \mathbf{m}} \frac{p_{1|t}^{\theta,d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{1|t}^{\theta_{old},d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})},$$

which recovers the transition probability ratio derived in Theorem 3.2 of Zhang et al. (2026). We defer the proof of Theorem 1 to section B.

Estimating rate-dependent weight For a general probability path (e.g., metric-induced probability path introduced in Shaul et al., 2025; Wang et al., 2025b), we can use the empirical version of the transition probability ratio to estimate $r_k(\theta)$ at time t_k without additional forward passes; that is,

$$\hat{r}_k(\theta) = \prod_{d\in[\mathcal{D}]} \frac{\hat{p}_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{\hat{p}_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} = \prod_{d\in[\mathcal{D}]} \Big(\frac{1}{n}\sum_{j=1}^n \Big\{\hat{w}_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_{j,1}^d)\frac{p_{1|t_k}^\theta(X_{j,1}^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old}}(X_{j,1}^d|\mathbf{x}_{t_k})}\Big\}\Big), \tag{6}$$

where $\{X_{j,1}^d\}_{j\in[n]}$ are i.i.d. samples from $p_{1|t_k}^{\theta_{old},d}(\cdot|\mathbf{x}_{t_k})$ and

$$\hat{w}_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_{j,1}^d) = \frac{\tilde{w}_{Q,k}^d(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_{j,1}^d)}{\frac{1}{n}\sum_{l=1}^n \tilde{w}_{Q,k}^d(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_{l,1}^d)}. \tag{7}$$

When we only use one single Monte Carlo sample (i.e., $n = 1$), the weight estimator is equal to one, which implies that the transition probability ratio estimator reduces to the product of the token-wise posterior ratios. In this scenario, the estimation of the transition probability ratio is not only independent of the conditional rate but also unrelated to the next state $\mathbf{x}_{t_{k+1}}$, which has a non-negligible bias. Additionally, the ratio estimation with a single MC sample also leads to slow convergence during training; we will discuss this in the ablation study.

5 EXPERIMENTS

In this section, we conduct comprehensive experiments on image generation and multimodal understanding tasks to evaluate the performance of the proposed dFlowGRPO framework. Results on a second DFM are provided in section F.

5.1 EXPERIMENTAL SETUP

We choose FUDOKI (Wang et al., 2025b), a multimodal DFM, as our base model. The metric-induced probability path and the corresponding kinetic-optimal conditional rate (Shaul et al., 2025) for FUDOKI are defined by

$$q^d_{t|1}(x^d|x^d_1) = \text{softmax}(-\beta_t \cdot \tilde{d}(x^d, x^d_1)),$$
$$Q^d_t(x^d, z^d|x^d_1) = q^d_{t|1}(z^d|x^d_1)\dot{\beta}_t[\tilde{d}(x^d, x^d_1) - \tilde{d}(z^d, x^d_1)],$$

where $\beta_t = 3 \cdot (\frac{t}{1-t})^{0.9}$ and $\tilde{d}(\cdot, \cdot)$ is a metric calculated based on the embedding layers adopted by FUDOKI. For image generation, the sequence length for the response is set to 576 and the resolution of the generated images is 384×384 in FUDOKI. For multimodal understanding, the sequence length for the text part is set to 500. To accelerate training, we do not use classifier-free guidance, and the number of denoising steps is set as 8 by default for all experiments. We adopt LoRA with rank 48 and $\alpha = 96$ to fine-tune FUDOKI. The implementation details can be found in section E.

Table 1: Evaluation of text-to-image generation ability on GenEval benchmark. †The results of FUDOKI (with CFG) are presented in their original paper.

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Attribute	Overall ↑
Generation-Only							
SDXL (Podell et al., 2024)	0.98	0.74	0.39	0.85	0.15	0.23	0.55
DALL-E 3 (Betker et al.)	0.96	0.87	0.47	0.83	0.43	0.45	0.67
SD3.5-L (Esser et al., 2024)	0.98	0.89	0.73	0.83	0.34	0.47	0.71
FLUX.1-dev (Labs, 2024)	0.98	0.93	0.75	0.93	0.68	0.65	0.82
SD3.5-M (Esser et al., 2024)	0.98	0.78	0.50	0.81	0.24	0.52	0.63
w/ Flow-GRPO (Liu et al., 2025a)	1.00	0.99	0.95	0.92	0.99	0.86	0.95
Unified Understanding & Generation							
Show-o (Xie et al., 2025)	0.95	0.52	0.49	0.82	0.11	0.28	0.53
w/ Mask-GRPO (Luo et al., 2025)	0.99	0.90	0.69	0.85	0.35	0.59	0.73
Janus-Pro-7B (Chen et al., 2025)	0.99	0.89	0.59	0.90	0.79	0.66	0.80
MMaDA (Yang et al., 2025)	0.96	0.60	0.45	0.81	0.14	0.25	0.56
w/ UniGRPO(Yang et al., 2025)	0.99	0.76	0.61	0.84	0.20	0.37	0.63
w/ MaskGRPO (Ma et al., 2026)	0.99	0.85	0.66	0.89	0.73	0.69	0.80
Lumina-Dimoo (Xin et al., 2025)	1.00	0.94	0.85	0.89	0.85	0.76	0.88
w/ selfGRPO (Xin et al., 2025)	1.00	0.96	0.87	0.95	0.85	0.82	0.91
FUDOKI† (Wang et al., 2025b)	0.96	0.85	0.56	0.88	0.68	0.67	0.77
w/ dFlowGRPO	0.97	0.93	0.92	0.91	0.88	0.97	0.93

5.2 TEXT-TO-IMAGE GENERATION

For text-to-image generation, we evaluate dFlowGRPO's generation capabilities on the GenEval (Ghosh et al., 2023), PickScore (Kirstain et al., 2023) and DrawBench (Saharia et al., 2022) benchmarks. We train dFlowGRPO using GenEval and PickScore rewards on their corresponding training splits without KL regularization, respectively. We evaluate dFlowGRPO with 20 NFEs for each reward. table 1 presents the performance of dFlowGRPO on the GenEval benchmark, where the proposed framework improves FUDOKI significantly, achieving the best overall performance (0.93) among the unified multimodal models with fewer denoising steps without CFG. The experimental results of dFlowGRPO trained with PickScore reward are reported in figure 1. We can see that the PickScore and GenEval Score increase consistently during the training of dFlowGRPO, improving the PickScore of FUDOKI from 20.87 to 23.00 without evident reward hacking across the reported auxiliary metrics. Additionally, for evaluation results on DrawBench, we observe that dFlowGRPO trained with PickScore reward consistently improves other metrics, including HPSv3 (Ma et al., 2025), Aesthetic Score (Schuhmann), DeQA (You et al., 2025), ImageReward (Xu et al., 2023) and UnifiedReward (Wang et al., 2025c). For dFlowGRPO trained with GenEval reward, the fine-tuned

model also improves five rewards evaluated on DrawBench, with only a slight decrease in Aesthetic Score.

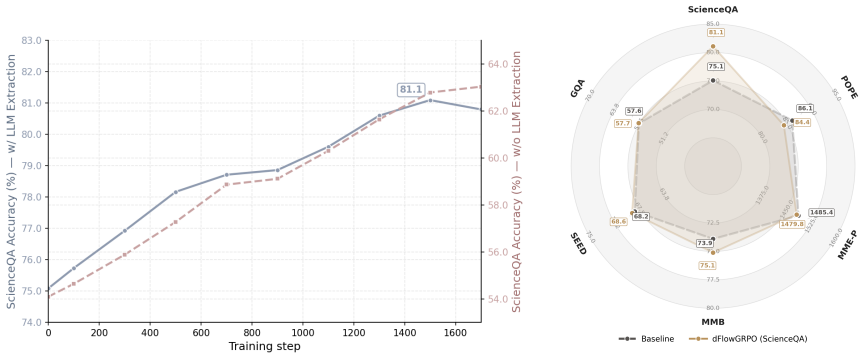

Figure 2: (Left) Training dynamics of dFlowGRPO trained on ScienceQA training split; the evaluation results are obtained based on test split. (Right) Evaluation results of dFlowGRPO on several benchmarks.

5.3 Multimodal Understanding

We evaluate the understanding capability of dFlowGRPO on six benchmarks, including POPE (Li et al., 2023b), MME-P (Fu et al.), SEED (Li et al., 2023a), MMB (Liu et al., 2024), GQA (Hudson & Manning, 2019), and ScienceQA (Lu et al., 2022). We fine-tune FUDOKI with dFlowGRPO on ScienceQA training split with KL regularization $\beta = 0.01$, and the 0-1 correctness reward for each answer is obtained by exact matching between the generated answer and the ground truth; otherwise, a judge LLM is used (see section E.2 for its prompt and error analysis). figure 2 shows the training dynamics of dFlowGRPO and the evaluation results on multimodal understanding benchmarks. We can observe that the proposed dFlowGRPO improves the accuracy of FUDOKI on ScienceQA from 75.1% to 81.1% with LLM extraction and from 54.1% to 59.1% without LLM extraction. Additionally, dFlowGRPO achieves better performance than the base model on GQA, SEED and MMB after training, showing limited evidence of reward hacking across the reported auxiliary benchmarks.

5.4 Ablation Study and Discussion

In this subsection, we conduct ablation studies on the effect of KL regularization and the effect of the number of MC samples, and compare dFlowGRPO with the mean-field approximation method for discrete flow models. Comparison with online DPO, further ablations, additional benchmarks, unseen-object generalization, the effect of NFEs, and qualitative results are provided in sections D.2 to D.7 and G.

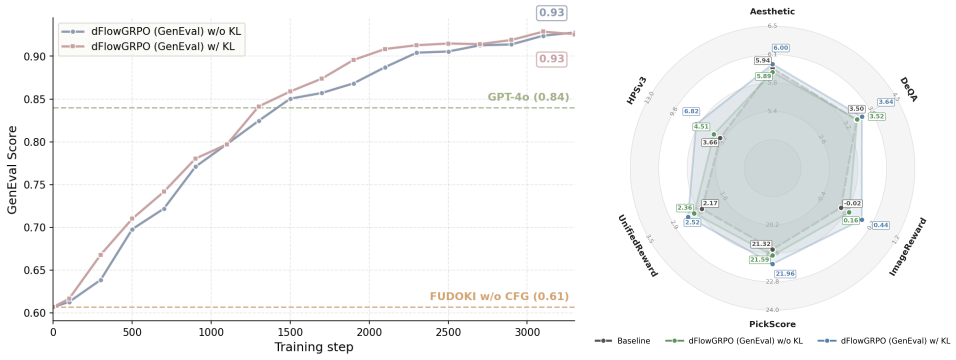

Figure 3: (Left) Training dynamics of dFlowGRPO on GenEval training split (w/ and w/o KL regularization); the evaluation results are obtained on test split. (Right) Performance of dFlowGRPO evaluated by other metric on DrawBench.

Effect of KL regularization. We train dFlowGRPO with GenEval reward without and with KL regularization $\beta = 0.01$. figure 3 presents the training dynamics of dFlowGRPO in both settings, where the training dynamics are similar and both curves achieve 0.93 after 3300 training steps. We also evaluate both models on DrawBench with different reward functions. We can see that KL regularization can mitigate reward hacking and significantly improve most of the other rewards.

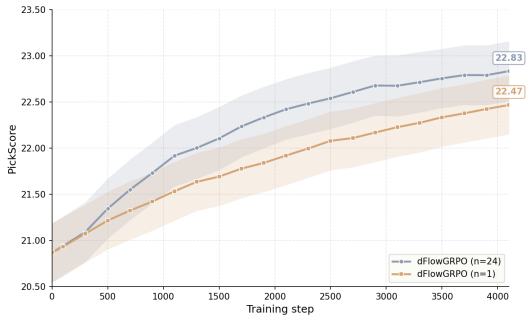

Figure 4: Comparison of dFlowGRPO with $n = 24$ and $n = 1$.

Effect of MC samples. To investigate the effect of the number of MC samples, we train dFlowGRPO with $n = 1$ on the PickScore dataset. figure 4 shows the training dynamics of dFlowGRPO with $n = 1$ and $n = 24$ for the first 4100 gradient updates. We can observe that the PickScore reward of dFlowGRPO with $n = 24$ achieves 22.50 after 2300 training steps (which requires 29.44 ($\times 8$) H100 GPU hours), but the reward of dFlowGRPO with $n = 1$ increases slowly during training and achieves 22.47 after 4100 training steps (which requires 38.13 ($\times 8$) H100 GPU hours), showing that more MC samples can accelerate training and convergence.

Comparison with mean-field approximation method. To verify the effectiveness of dFlowGRPO, we compare it with mean-field approximation methods (Zhao et al., 2025a) on FUDOKI, a non-masked model. The diffu-GRPO objective and its geometric-mean variant, the diffu-GSPO objective, are defined in (8) and (9), respectively. figure 5 presents the training dynamics of dFlowGRPO and two mean-field approximation methods trained with the PickScore reward without KL regularization. For a fair comparison, we use the same number of denoising steps and the same clipping parameters for all three methods. Results show that both mean-field baselines collapse during training, whereas dFlowGRPO improves steadily throughout and achieves a higher reward.

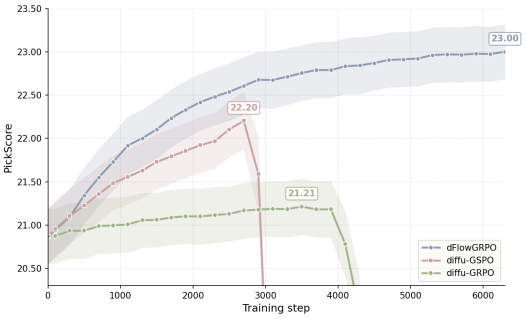

Figure 5: Comparison with mean-field approximation methods on PickScore.

6 Conclusion

In this paper, we introduce dFlowGRPO, a unified RL training framework for DFMs. By deriving the transition probability ratio of the denoising process, dFlowGRPO leverages information from both the conditional rate and the posterior model. We evaluate the proposed method on both text-to-image generation and multimodal understanding tasks, demonstrating the effectiveness and strong performance across various benchmarks.

Building on our framework, one can also apply different rate-aware transition probability ratios to dLLMs by considering alternative conditional rates constructed according to the detailed balance condition (see Appendix F in Campbell et al., 2024). We leave this interesting direction for future work.

AI USE STATEMENT

In this work, LLM judges were used to score model outputs during evaluation. Generative AI tools were used to provide feedback on experiments, edit code, and polish the language and readability of the manuscript. All AI-assisted text and code were reviewed, and the AI-edited code and judge-based evaluation were verified by the authors. We take responsibility for the final content of this work.

Generative AI tools were **not** used to generate synthetic datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for proving mathematical claims; assist in writing proofs; propose or refine hypotheses; translate text; clean and reformat dataset; support qualitative or thematic data analysis; or interpret results.

REPRODUCIBILITY STATEMENT

The supplementary material includes the training and evaluation code used in this work. The appendix provides complete proofs, gradient derivations, implementation details, training configurations, computational costs, evaluation protocols, and additional experimental results.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, et al. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, et al. LLaDA2. 1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*, 2026.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28266–28279, 2022.

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 *International Conference on Machine Learning*, 2024.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. MME: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, volume 37, pp. 133345–133385, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Shansan Gong, Ruixiang ZHANG, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. Reinforcing the diffusion chain of lateral thought with diffusion language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. LaViDa: A large diffusion language model for multimodal understanding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Shufan Li, Yuchen Zhu, Jiuxiang Gu, Kangning Liu, Zhe Lin, Yongxin Chen, Molei Tao, Aditya Grover, and Jason Kuen. LaViDa-R1: Advancing reasoning for unified multimodal diffusion language models. *arXiv preprint arXiv:2602.14147*, 2026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023b.

Yuchen Liang, Yingbin Liang, Lifeng Lai, and Ness Shroff. Discrete diffusion models: Novel analysis and new sampler guarantees. *arXiv preprint arXiv:2509.16756*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di ZHANG, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Yifu Luo, Xinhao Hu, Keyu Fan, Haoyuan Sun, Zeyu Chen, Bo Xia, Tiantian Zhang, Yongzhe Chang, and Xueqian Wang. Reinforcement learning meets masked generative models: Mask-GRPO for text-to-image generation. *arXiv preprint arXiv:2510.13418*, 2025.

Tianren Ma, Mu Zhang, Yibing Wang, and Qixiang Ye. Consolidating reinforcement learning for multimodal discrete diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15086–15095, 2025.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011.

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

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *International Conference on Learning Representations*, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *International Conference on Learning Representations*, 2025b.

Jingyang Ou, Jiaqi Han, Minkai Xu, Shaoxuan Xu, Jianwen Xie, Stefano Ermon, Yi Wu, and Chongxuan Li. Principled RL for diffusion llms emerges from a sequence-level perspective. *arXiv preprint arXiv:2512.03759*, 2025a.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

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

Chrisoph Schuhmann. Laion aesthetics, aug 2022. *URL https://laion. ai/blog/laion-aesthetics*.

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

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

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: a kinetic-optimal perspective. In *International Conference on Learning Representations*, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *International Conference on Learning Representations*, 2023.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. wd1: Weighted policy optimization for reasoning in diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238, June 2024.

Zhengyan Wan, Yidong Ouyang, , Liyan Xie, Fang Fang, Hongyuan Zha, and Guang Cheng. Corrected samplers for discrete flow models. In *Forty-third International Conference on Machine Learning*, 2026.

Guanghan Wang, Gilad Turok, Yair Schiff, Marianne Arriola, and Volodymyr Kuleshov. d2: Improved techniques for training reasoning diffusion language models. *arXiv preprint arXiv:2509.21474*, 2025a.

Jin Wang, Yao Lai, Aoxue Li, Shifeng Zhang, Jiacheng Sun, Ning Kang, Chengyue Wu, Zhenguo Li, and Ping Luo. FUDOKI: Discrete flow-based unified understanding and generation via kinetic-optimal velocities. *arXiv preprint arXiv:2505.20147*, 2025b.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025c.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. Revolutionizing reinforcement learning framework for diffusion large language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: one single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, 2025.

Yi Xin, Qi Qin, Siqi Luo, Kaiwen Zhu, Juncheng Yan, Yan Tai, Jiayi Lei, Yuewen Cao, Keqi Wang, Yibin Wang, et al. Lumina-Dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

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

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

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

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching large language models to regress accurate image quality scores using score distribution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 14483–14494, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Wenxuan Zhang, Lemeng Wu, Changsheng Zhao, Ernie Chang, Mingchen Zhuge, Zechun Liu, Andy Su, Hanxian Huang, Jun Chen, Chong Zhou, et al. dTRPO: Trajectory reduction in policy optimization of diffusion large language models. *arXiv preprint arXiv:2603.18806*, 2026.

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

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

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

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

A ALGORITHMS

In this section, we present the always-valid Euler solver (see algorithm 1 and Shaul et al., 2025; Wang et al., 2025b) and the proposed dFlowGRPO algorithm (see algorithm 2).

Algorithm 1 Euler solver (Algorithm 1 in Shaul et al., 2025)

Require: A conditional rate $Q_t^d(x^d, z^d|x_1^d)$, a posterior $p_{1|t}$, time partition $0 = t_0 < t_1 < \cdots < t_K = 1$.

1: Draw $\mathbf{x}_0 \sim p_0$.
2: **for** $k = 0$ to $K - 1$ **do**
3: Set $Z = \mathbf{x}_{t_k}$.
4: **for** $d = 1$ to $\mathcal{D}$ **do**
5: Sample $X_1^d \sim p_{1|t_k}^d(\cdot|\mathbf{x}_{t_k})$. (We can sample n i.i.d. copies $\{X_{j,1}^d\}_{j\in[n]}$ of X_1^d and use the empirical mean $\frac{1}{n}\sum_{i=1}^n Q_{t_k}^d(\mathbf{x}_{t_k}^d, z^d|X_{j,1}^d)$ to estimate $Q_{t_k}^d(\mathbf{x}_{t_k}, z^d)$ for variance reduction.)
6: Set $\lambda_{k+1}^d = \sum_{z^d \neq \mathbf{x}_{t_k}^d} Q_{t_k}^d(\mathbf{x}_{t_k}^d, z^d|X_1^d)$.
7: Set $Z^d = \begin{cases} z^d, & \text{with probability } \Big\{1 - \exp\Big(-(t_{k+1} - t_k)\lambda_{k+1}^d\Big)\Big\}\frac{Q_{t_k}^d(\mathbf{x}_{t_k}^d, z^d|X_1^d)}{\lambda_{k+1}^d} \\ Z^d, & \text{with probability } \exp\Big(-(t_{k+1} - t_k)\lambda_{k+1}^d\Big) \end{cases}$,
where $z^d \neq Z^d$.
8: **end for**
9: Set $\mathbf{x}_{t_{k+1}} = Z$.
10: **end for**
11: **return** $\mathbf{x}_1 \sim p_1$

B PROOF OF THEOREM 1

Proof. Note that for each denoising step, the tokens of $\mathbf{x}_{t_{k+1}}$ are independent conditioning on the last state $\mathbf{x}_{t_k}$; that is, $p_\theta(\mathbf{x}_{t_{k+1}}|\mathbf{x}_{t_k}) = \prod_{d\in[\mathcal{D}]} p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})$. According to the sampling algorithm (algorithm 1), the transition probability for the d-th token is:

$$
\begin{aligned}
&p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k}) \\
&= \mathbb{E}_{X_1^d \sim p_{1|t_k}^{\theta,d}(\cdot|\mathbf{x}_{t_k})}\Big\{\mathbf{1}(\mathbf{x}_{t_k}^d = \mathbf{x}_{t_{k+1}}^d)\Big(\exp\Big(-(t_{k+1} - t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big) \\
&\quad + \mathbf{1}(\mathbf{x}_{t_k}^d \neq \mathbf{x}_{t_{k+1}}^d)\Big(\frac{Q_{t_k}^d(\mathbf{x}_{t_k}^d, \mathbf{x}_{t_{k+1}}^d|X_1^d)}{\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)}\Big\{1 - \exp\Big(-(t_{k+1} - t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big\}\Big)\Big\} \\
&= \mathbb{E}_{X_1^d \sim p_{1|t_k}^{\theta_{old},d}(\cdot|\mathbf{x}_{t_k})}\Big\{\mathbf{1}(\mathbf{x}_{t_k}^d = \mathbf{x}_{t_{k+1}}^d)\exp\Big(-(t_{k+1} - t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\frac{p_{1|t_k}^{\theta,d}(X_1^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old},d}(X_1^d|\mathbf{x}_{t_k})} \\
&\quad + \mathbf{1}(\mathbf{x}_{t_k}^d \neq \mathbf{x}_{t_{k+1}}^d)\frac{Q_{t_k}^d(\mathbf{x}_{t_k}^d, \mathbf{x}_{t_{k+1}}^d|X_1^d)}{\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)}\Big\{1 - \exp\Big(-(t_{k+1} - t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big\}\frac{p_{1|t_k}^{\theta,d}(X_1^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old},d}(X_1^d|\mathbf{x}_{t_k})}\Big\},
\end{aligned}
$$

where $\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d) = \sum_{z^d \neq \mathbf{x}_{t_k}^d} Q_{t_k}^d(\mathbf{x}_{t_k}^d, z^d|X_1^d)$.

Thus, the transition probability ratio between the current policy and the old policy is

$$
r_k(\theta) = \prod_{d\in[\mathcal{D}]} \frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} = \prod_{d\in[\mathcal{D}]} \mathbb{E}_{X_1^d \sim p_{1|t_k}^{\theta_{old}}(\cdot|\mathbf{x}_{t_k})}\Big\{w_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d)\frac{p_{1|t_k}^{\theta,d}(X_1^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old},d}(X_1^d|\mathbf{x}_{t_k})}\Big\},
$$

where the rate-dependent weight $w_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d)$ is defined by

$$
\begin{aligned}
w_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}, \mathbf{x}_{t_{k+1}}^d, X_1^d) = \frac{1}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}&\Big\{\mathbf{1}(\mathbf{x}_{t_k}^d = \mathbf{x}_{t_{k+1}}^d)\Big(\exp\Big(-(t_{k+1} - t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big) \\
&+ \mathbf{1}(\mathbf{x}_{t_k}^d \neq \mathbf{x}_{t_{k+1}}^d)\Big(\frac{Q_{t_k}^d(\mathbf{x}_{t_k}^d, \mathbf{x}_{t_{k+1}}^d|X_1^d)}{\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)}\Big\{1 - \exp\Big(-(t_{k+1} - t_k)\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d)\Big)\Big\}\Big)\Big\}.
\end{aligned}
$$

Algorithm 2 dFlowGRPO

Require: A conditional rate $Q_t^d(x^d, z^d|x_1^d)$, a posterior model $p_{1|t}^\theta$, time partition $0 = t_0 < t_1 < \cdots < t_K = 1$, the number of MC samples n.

while not converged **do**
  Sample $\mathbf{c} \sim p_c$.
  for $i = 1$ to G **do**
    Draw $\mathbf{x}_0^{(i)} \sim p_0$.
    for $k = 0$ to $K - 1$ **do**
      Set $Z^{(i)} = \mathbf{x}_{t_k}^{(i)}$.
      for $d = 1$ to $\mathcal{D}$ **do**
        Sample $\{X_{j,1}^{(i),d}\}_{j\in[n]} \overset{i.i.d.}{\sim} p_{1|t_k}^{\theta_{old},d}(\cdot|\mathbf{x}_{t_k}^{(i)}, \mathbf{c})$.
        Set $Q_{t_k}^{(i),d}(\mathbf{x}_{t_k}^{(i),d}, z^d) = \frac{1}{n}\sum_{j\in[n]} Q_{t_k}^{(i),d}(\mathbf{x}_{t_k}^{(i),d}, z^d|X_{j,1}^{(i),d})$
        Set $\lambda_{k+1}^{(i),d} = \sum_{z^d \neq \mathbf{x}_{t_k}^{(i),d}} Q_{t_k}^{(i),d}(\mathbf{x}_{t_k}^{(i),d}, z^d)$.
        Set $Z^{(i),d} = \begin{cases} z^d, & \text{w.p. } \left\{1 - \exp\Big(-(t_{k+1} - t_k)\lambda_{k+1}^{(i),d}\Big)\right\} \frac{Q_{t_k}^{(i),d}(\mathbf{x}_{t_k}^{(i),d}, z^d)}{\lambda_{k+1}^{(i),d}} \\ Z^{(i),d}, & \text{w.p. } \exp\Big(-(t_{k+1} - t_k)\lambda_{k+1}^{(i),d}\Big) \end{cases}$,
        where $z^d \neq Z^{(i),d}$.
      end for
      Set $\mathbf{x}_{t_{k+1}}^{(i)} = Z^{(i)}$.
      Compute the rate-dependent weight $\hat{w}_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}^{(i)}, \mathbf{x}_{t_{k+1}}^{(i),d}, X_{j,1}^{(i),d})$ by (7) for each $j \in [n], d \in [\mathcal{D}]$.
    end for
  end for
  Set $\hat{A}^{(i)} = \frac{\mathcal{R}(\mathbf{x}_1^{(i)},\mathbf{c}) - \text{mean}(\{\mathcal{R}(\mathbf{x}_1^{(i)},\mathbf{c})\}_{i\in[G]})}{\text{std}(\{\mathcal{R}(\mathbf{x}_1^{(i)},\mathbf{c})\}_{i\in[G]})}$.
  Compute $\hat{r}_k^{(i)}(\theta) = \prod_{d\in[\mathcal{D}]} \frac{\hat{p}_\theta(\mathbf{x}_{t_{k+1}}^{(i),d}|\mathbf{x}_{t_k}^{(i)},\mathbf{c})}{\hat{p}_{\theta_{old}}(\mathbf{x}_{t_{k+1}}^{(i),d}|\mathbf{x}_{t_k}^{(i)},\mathbf{c})}$ for each $k \in \{0, 1, \ldots, K-1\}$ by (6) and update θ with dFlowGRPO objective $\mathcal{J}_{\text{dFlowGRPO}}(\theta)$ ((2)).
end while

The transition probability ratio derived above can be estimated by the empirical mean for a general probability path and the associated conditional rate. In the following, we give an example for the specific choice of the mixture path with conditional rate $Q_t^d(x^d, z^d|x_1^d) = \frac{\dot{\kappa}_t}{1-\kappa_t}(\delta_{x_1^d}(z^d) - \delta_{x^d}(z^d))$.

Example (mixture path). For the mixture path with conditional rate $Q_t^d(x^d, z^d|x_1^d) = \frac{\dot{\kappa}_t}{1-\kappa_t}(\delta_{x_1^d}(z^d) - \delta_{x^d}(z^d))$, we have

$$\lambda_{t_k}^d(\mathbf{x}_{t_k}^d, X_1^d) = \begin{cases} 0, & \text{if } \mathbf{x}_{t_k}^d = X_1^d \\ \frac{\dot{\kappa}_{t_k}}{1-\kappa_{t_k}}, & \text{if } \mathbf{x}_{t_k}^d \neq X_1^d \end{cases},$$

which implies that

$$\begin{aligned} r_k(\theta) &= \prod_{d\in[\mathcal{D}]} \frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} \\ &= \prod_{d\in[\mathcal{D}]} \Big\{ \mathbf{1}(\mathbf{x}_{t_k}^d \neq \mathbf{x}_{t_{k+1}}^d) \frac{p_{1|t_k}^{\theta,d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old},d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} \\ &\qquad + \mathbf{1}(\mathbf{x}_{t_k}^d = \mathbf{x}_{t_{k+1}}^d) \frac{p_{1|t_k}^{\theta,d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k}) + (1 - p_{1|t_k}^{\theta,d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k}))g_Q(k)}{p_{1|t_k}^{\theta_{old},d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k}) + (1 - p_{1|t_k}^{\theta_{old},d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k}))g_Q(k)} \Big\}, \end{aligned}$$

where

$$g_Q(k) = \exp(-(t_{k+1} - t_k)\frac{\dot{\kappa}_{t_k}}{1-\kappa_{t_k}}).$$

In particular, if the source distribution is $\delta_{\mathbf{m}}$, then we have

$$r_k(\theta) = \prod_{d\in[\mathcal{D}]} \frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})} = \prod_{d:\mathbf{x}_{t_k}^d=\mathbf{m} \text{ and } \mathbf{x}_{t_{k+1}}^d\neq\mathbf{m}} \frac{p_{1|t_k}^{\theta,d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})}{p_{1|t_k}^{\theta_{old},d}(\mathbf{x}_{t_{k+1}}^d|\mathbf{x}_{t_k})},$$

which is independent of the time scheduler.

□

C Derivation of the Gradient

In this section, we derive the gradient of our proposed dFlowGRPO objective. For simplicity, we omit the clipping operation and take $\beta = 0$.

By Theorem 1, we have

$$r_k^{(i)}(\theta) = \prod_{d\in[\mathcal{D}]} \frac{p_\theta^d(\mathbf{x}_{t_{k+1}}^{(i),d}|\mathbf{x}_{t_k}^{(i)})}{p_{\theta_{old}}^d(\mathbf{x}_{t_{k+1}}^{(i),d}|\mathbf{x}_{t_k}^{(i)})} = \prod_{d\in[\mathcal{D}]} \mathbb{E}_{X_1^d\sim p_{1|t_k}^{\theta_{old}}(\cdot|\mathbf{x}_{t_k})} \Big\{ w_{Q,k}^{\theta_{old},d}(\mathbf{x}_{t_k}^{(i)}, \mathbf{x}_{t_{k+1}}^{(i),d}, X_1^d) \frac{p_{1|t_k}^{\theta,d}(X_1^d|\mathbf{x}_{t_k}^{(i)})}{p_{1|t_k}^{\theta_{old},d}(X_1^d|\mathbf{x}_{t_k}^{(i)})} \Big\}.$$

Thus, we can obtain that

$$\begin{aligned} &\nabla_\theta \mathcal{J}_{\text{dFlowGRPO}}(\theta) \\ &= \mathbb{E}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{k=0}^{K-1} \hat{A}_k^{(i)} \nabla_\theta [r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} \\ &= \mathbb{E}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{k=0}^{K-1} \hat{A}_k^{(i)} [r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} \Big(\frac{1}{\mathcal{D}}\sum_{d=1}^{\mathcal{D}} \nabla_\theta \log p_\theta^d(\mathbf{x}_{t_{k+1}}^{(i),d}|\mathbf{x}_{t_k}^{(i)})\Big) \\ &= \mathbb{E}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{k=0}^{K-1} \hat{A}_k^{(i)} [r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} \Big(\frac{1}{\mathcal{D}}\sum_{d=1}^{\mathcal{D}} \frac{\mathbb{E}_{X_1^d\sim p_{1|t_k}^{\theta}(\cdot|\mathbf{x}_{t_k})}\Big\{\tilde{w}_{Q,k}^d(\mathbf{x}_{t_k}^{(i)}, \mathbf{x}_{t_{k+1}}^{(i),d}, X_1^d)\nabla_\theta \log p_{1|t_k}^{\theta,d}(X_1^d|\mathbf{x}_{t_k}^{(i)})\Big\}}{p_\theta^d(\mathbf{x}_{t_{k+1}}^{(i),d}|\mathbf{x}_{t_k}^{(i)})}\Big) \\ &= \mathbb{E}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{K}\sum_{k=0}^{K-1} \hat{A}_k^{(i)} [r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} \Big(\frac{1}{\mathcal{D}}\sum_{d=1}^{\mathcal{D}} \mathbb{E}_{X_1^d\sim p_{1|t_k}^{\theta}(\cdot|\mathbf{x}_{t_k})}\Big\{w_{Q,k}^{\theta,d}(\mathbf{x}_{t_k}^{(i)}, \mathbf{x}_{t_{k+1}}^{(i),d}, X_1^d)\nabla_\theta \log p_{1|t_k}^{\theta,d}(X_1^d|\mathbf{x}_{t_k}^{(i)})\Big\}\Big), \end{aligned}$$

where we use the definition of the rate-dependent weight (see (3) and (5)) and the expectation is taken over $\mathbf{c} \sim p_c, \{\mathbf{x}^{(i)}\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|\mathbf{c})$. Consequently, the gradient of the dFlowGRPO objective has a form similar to the gradient of standard Flow-GRPO ($[r_k^{(i)}(\theta)]^{\frac{1}{\mathcal{D}}} \approx 1$), with the policy score replaced by the posterior score weighted by rate-dependent weight.

D Some Discussions and Additional Experiments

In this section, we discuss other RL methods for DFMs such as mean-field approximation methods (Zhao et al., 2025a) and online DPO (Guo et al., 2024), and conduct additional experiments for these methods. We also provide additional ablation studies and evaluate dFlowGRPO on additional benchmarks, unseen-object generalization, and with different NFEs.

D.1 diffu-GRPO and diffu-GSPO

Zhao et al. (2025a) proposed to use mean-field approximation to estimate the log-likelihood for dLLMs in RL training. The corresponding diffu-GRPO is

$$\mathcal{J}_{\text{diffu-GRPO}}(\theta) = \mathbb{E}_{\mathbf{c}\sim p_c, \{\mathbf{x}^{(i)}\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|\mathbf{c})} \Big\{ \frac{1}{G}\sum_{i=1}^{G} \frac{1}{\mathcal{D}} \sum_{d=1}^{\mathcal{D}} \Big(\min(r_d^{(i)}(\theta)\hat{A}^{(i)}, \text{clip}(r_d^{(i)}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}^{(i)}) - \beta D_{KL}(\pi_\theta \| \pi_{ref}) \Big) \Big\}, \tag{8}$$

where $r_d^{(i)}(\theta) = \frac{p_\theta(\mathbf{x}_1^{(i),d}|\mathbf{x}_0^{(i)}, \mathbf{c})}{p_{\theta_{old}}(\mathbf{x}_1^{(i),d}|\mathbf{x}_0^{(i)}, \mathbf{c})}$ is the token-level posterior ratio at $\mathbf{x}_0$. If the source distribution is a masked point mass, then the ratio only depends on $\mathbf{c}$ and $\mathbf{x}_1^{(i)}$.

Following GSPO (Zheng et al., 2025), we can propose diffu-GSPO to improve the performance of diffu-GRPO by simply replacing the arithmetic mean with geometric mean of token-level posterior ratios (see figure 5 for comparison):

$$\mathcal{J}_{\text{diffu-GSPO}}(\theta) = \mathbb{E}_{\mathbf{c}\sim p_c, \{\mathbf{x}^{(i)}\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|\mathbf{c})} \Big\{ \frac{1}{G}\sum_{i=1}^{G} \Big(\min\Big(\Big[\prod_{d=1}^{\mathcal{D}} r_d^{(i)}(\theta)\Big]^{\frac{1}{\mathcal{D}}} \hat{A}^{(i)}, \text{clip}\Big(\Big[\prod_{d=1}^{\mathcal{D}} r_d^{(i)}(\theta)\Big]^{\frac{1}{\mathcal{D}}}, 1-\epsilon, 1+\epsilon\Big)\hat{A}^{(i)} \Big) - \beta D_{KL}(\pi_\theta \| \pi_{ref}) \Big) \Big\}. \tag{9}$$

Compared with our proposed dFlowGRPO, the mean-field approximation in diffu-GRPO and diffu-GSPO only uses the information of the samples at the terminal time, leading to inaccurate log-likelihood or trajectory probability estimation. Additionally, it can make RL training unstable or even collapse for a general probability path (see figure 5).

D.2 dFlowDPO: Generalization to Online DPO

In this section, we generalize our approach to online DPO (Guo et al., 2024). Consider the following regularized RLHF problem:

$$\pi^* = \arg\max_{\pi_\theta} \mathbb{E}_{\mathbf{c}\sim p_c, \mathbf{x}\sim\pi_\theta(\cdot|\mathbf{c})} \Big\{ [r(\mathbf{x}_1, \mathbf{c})] - \beta D_{KL}\Big(\pi_\theta(\mathbf{x}_{t_0:t_K}|\mathbf{c}) \Big\| \pi_{\text{old}}(\mathbf{x}_{t_0:t_K}|\mathbf{c}) \Big) \Big\}.$$

The optimal policy π^* has the closed-form $\pi^*(\mathbf{x}_{t_0:t_K}|\mathbf{c}) \propto \exp(r(\mathbf{x}_1, \mathbf{c})/\beta)\pi_{old}(\mathbf{x}_{t_0:t_K}|\mathbf{c})$. Plugging $r(\mathbf{x}_1, \mathbf{c}) = \beta \log \frac{\pi^*(\mathbf{x}_{t_0:t_K}|\mathbf{c})}{\pi_{old}(\mathbf{x}_{t_0:t_K}|\mathbf{c})} + \log C(\mathbf{c})$ into the log-likelihood of BT model, then we have the following online DPO training objective for discrete flow models:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim p_c, \{\mathbf{x}^{(i)}\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|\mathbf{c})} \Big\{ -\log\sigma\Big(\beta \log \frac{\pi_\theta(\mathbf{x}^+_{t_0:t_K}|\mathbf{c})}{\pi_{old}(\mathbf{x}^+_{t_0:t_K}|\mathbf{c})} - \beta \log \frac{\pi_\theta(\mathbf{x}^-_{t_0:t_K}|\mathbf{c})}{\pi_{old}(\mathbf{x}^-_{t_0:t_K}|\mathbf{c})} \Big) \Big\} \\
\le\ &\mathbb{E}_{\mathbf{c}\sim p_c, \{\mathbf{x}^{(i)}\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot|\mathbf{c})} \Big\{ -\frac{1}{K}\sum_{k=0}^{K-1} \log\sigma\Big(\beta K \Big\{ \log \frac{p_\theta(\mathbf{x}^+_{t_{k+1}}|\mathbf{x}^+_{t_k}, \mathbf{c})}{p_{\theta_{old}}(\mathbf{x}^+_{t_{k+1}}|\mathbf{x}^+_{t_k}, \mathbf{c})} - \log \frac{p_\theta(\mathbf{x}^-_{t_{k+1}}|\mathbf{x}^-_{t_k}, \mathbf{c})}{p_{\theta_{old}}(\mathbf{x}^-_{t_{k+1}}|\mathbf{x}^-_{t_k}, \mathbf{c})} \Big\} \Big) \Big\} \\
\triangleq\ &\mathcal{J}_{\text{dFlowDPO}}(\theta),
\end{aligned}$$

where we use Jensen's inequality (for training efficiency, similar to Wallace et al. (2024)), and $\mathbf{x}^+_{t_0:t_K}$ and $\mathbf{x}^-_{t_0:t_K}$ are the trajectories with highest reward and lowest reward in the group, respectively.

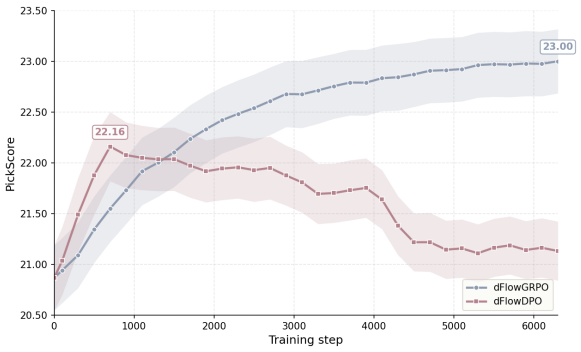

Figure 6: Comparison of dFlowGRPO and dFlowDPO on PickScore.

We investigate the performance of dFlowDPO on PickScore. For a fair comparison, we set the same number of denoising steps for both dFlowDPO and dFlowGRPO without KL regularization. We choose $\beta = 100/K$ for dFlowDPO, as suggested by Liu et al. (2025a). figure 6 presents the training dynamics of dFlowDPO and dFlowGRPO with PickScore reward, where dFlowDPO increases faster than dFlowGRPO initially and then decreases steadily after 700 training steps.

D.3 Additional Ablation Studies

We provide the additional experimental results on the number of MC samples and the rate-dependent weight below. For the ablation results for the rate-dependent weight, we set the weight in the computation of the transition probability ratio estimation to 1, and keep the number of MC samples to 24. All variants below are trained under *an identical compute budget of 13 hours* on 8 H100 GPUs on PickScore. Due to limited computing resources, this budget is only intended to verify the effectiveness of the rate-dependent weight rather than to train the models to convergence.

Table 2: Additional ablation studies on the number of MC samples and the rate-dependent weight.

(a) Number of MC samples

n	1	4	8	16	24
PickScore	21.6634	21.8340	**21.8556**	21.7828	21.8243

(b) Rate-dependent weight ($n = 24$)

Weighting	PickScore
Rate-aware	**21.8243**
$w = 1$	21.5541

We can see that the performance achieves its best result with $n = 8$, and the rate-dependent weight improves the performance by 0.27 points on PickScore.

D.4 Results on Additional Benchmarks

We have further evaluated the trained models on two additional text-to-image benchmarks that are not used as training rewards, DPG-Bench (a benchmark used in the FUDOKI paper) and T2I-CompBench++ (a benchmark used in Flow-GRPO). All evaluation results below are obtained with 20 NFEs and without CFG, and the results of the model trained on GenEval with KL regularization are reported.

We can see that the trained models perform comparably to the base model on DPG-Bench, where the model trained with the PickScore reward slightly improves the overall score and the model

Table 3: Evaluation results on additional text-to-image benchmarks. The best result in each column is highlighted in bold.

(a) DPG-Bench (4 images per prompt).

Model	Entity	Attribute	Relation	Other	Global	Overall
FUDOKI	**85.82**	84.18	**90.95**	**72.40**	**82.37**	77.95
w/ dFlowGRPO (PickScore)	84.59	83.74	89.44	68.40	80.24	**78.77**
w/ dFlowGRPO (GenEval)	85.40	**84.28**	89.48	68.00	79.64	77.86

(b) T2I-CompBench++ (10 images per prompt).

Model	Color	Shape	Texture	Spatial	3D-Spatial	Non-Spatial	Numeracy
FUDOKI	0.4643	0.3488	0.4301	0.1027	0.2164	0.2848	0.3777
w/ dFlowGRPO (GenEval)	**0.5532**	**0.3944**	**0.5065**	**0.2793**	**0.3066**	**0.2924**	**0.4370**

trained with the GenEval reward is slightly worse. One possible main reason is that the prompts in DPG-Bench are longer and more complex than those in the GenEval and PickScore training splits. Additionally, similar to the results in Flow-GRPO, the model trained on GenEval can improve the performance on T2I-CompBench++ since the prompts in some categories of T2I-CompBench++, such as color, spatial, and numeracy, are similar to the GenEval training prompts.

For understanding tasks, we evaluate the model trained on ScienceQA on two further benchmarks, MMMU and MM-Vet, with 32 NFEs for MMMU and 100 NFEs for MM-Vet.[2]

Table 4: Evaluation results on additional multimodal understanding benchmarks. The best result in each column is highlighted in bold.

Model	MMMU (validation)	MM-Vet
FUDOKI	0.3378	34.17
w/ dFlowGRPO (ScienceQA)	**0.3533**	**34.77**

These results show no evident reward hacking or catastrophic forgetting on the two additional understanding benchmarks.

D.5 Generalization to Unseen Objects

We additionally train a model on a subset of the GenEval object categories and evaluate it on prompts built from the held-out categories. We partition the 80 GenEval object categories into 60 training categories and 20 test categories with no overlap. Each split is built only from its own categories, so no object appearing in the test split is seen in any training prompt. We train dFlowGRPO for 3000 steps with $\beta_{\mathrm{KL}} = 0.01$ and evaluate at 20 NFEs, both without CFG.

Table 5: Generalization results on GenEval prompts constructed from held-out object categories. The best result in each column is highlighted in bold.

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Attribute	Overall
FUDOKI	0.7375	0.4583	0.6627	0.7234	0.3249	0.5585	0.5602
w/ dFlowGRPO (GenEval)	**0.7625**	**0.7760**	**0.8200**	**0.7926**	**0.7145**	**0.7606**	**0.7720**

dFlowGRPO improves every category and raises the overall score from 0.5602 to 0.7720. Since no test object appears in any training prompt, these improvements generalize to objects that are never seen during training.

[2]The judge models used by FUDOKI for these two benchmarks have been retired by OpenAI, so we reevaluate both FUDOKI and our model with the same judge, `gpt-4-turbo-2024-04-09`.

D.6 Evaluation Across Random Seeds

The main results are reported at a single evaluation seed. To check that they are not an artifact of that choice, we reevaluate the base model and our models at two additional seeds on the full test sets (2048 PickScore prompts, 2212 GenEval prompts and 2015 ScienceQA prompts), at 20 NFEs and without CFG. The results are reported in table 6.

Table 6: Evaluation results across three random seeds. The first seed column is the one reported in the main text (seed 42 for generation, seed 99 for understanding).

Benchmark	Model	Seed (reported)	Seed 43	Seed 44
PickScore	FUDOKI	20.8672	20.8432	20.8554
PickScore	w/ dFlowGRPO	**23.0016**	**22.9858**	**22.9814**
GenEval	FUDOKI	0.6069	0.5959	0.6326
GenEval	w/ dFlowGRPO	**0.9278**	**0.9254**	**0.9294**
ScienceQA	FUDOKI	0.7509	0.7573	0.7524
ScienceQA	w/ dFlowGRPO	**0.8109**	**0.8025**	**0.8089**

The results are consistent across seeds, and the improvements of dFlowGRPO over the base model are much larger than the seed-to-seed variation. Multi-seed training runs are not available due to the computational cost of training large DFM models.

D.7 Evaluation with Different NFEs

To investigate the effect of NFEs, we report the evaluation results on PickScore and GenEval with different NFEs, which is presented in table 7. The results show that dFlowGRPO has a strong performance with few-step generation (e.g., 8 NFEs), achieving the best rewards around 16–32 NFEs.

Table 7: Evaluation with different NFEs on image generation tasks. The models are trained with corresponding rewards.

NFEs	4	8	16	32	48	64
GenEval	0.46	0.91	0.93	0.93	0.93	0.92
PickScore	21.55	22.88	22.99	22.98	22.97	22.94

E Implementation Detail

In this section, we provide additional implementation and evaluation details omitted from the main text. We first describe the training setup and computational cost, and then present the ScienceQA answer extraction prompt together with an analysis of extraction errors.

E.1 Training Setup and Computational Cost

We use 8 NVIDIA H100 GPUs for all experiments. We fix group size $G = 24$ and the number of MC samples $n = 24$ across tasks. We train the model using batches of 16 prompts per gradient update, with gradient clipping at 1.0 and the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set the learning rate 1×10^{-5} for text-to-image generation and 1×10^{-6} for multimodal understanding. We use asymmetric clipping parameters $\epsilon_{\text{high}} = 1.5 \cdot \epsilon_{\text{low}} = 1.5 \times 10^{-3}$ for GRPO objective by searching from $\{1.5 \times 10^{-2}, 1.5 \times 10^{-3}, 1.5 \times 10^{-4}\}$, following the clip-higher strategy proposed in DAPO (Yu et al., 2026). For training stability, we update the old policy once per 48 gradient updates. We train dFlowGRPO for 3300 steps on GenEval, 6300 steps on PickScore, and 1500 steps on ScienceQA. We report the computational cost of our experiments in table 8.

Table 8: The wall-clock time of different settings for the first 100 gradient updates on 8 H100 GPUs.

Method	Time (hours)
dFlowGRPO (ScienceQA)	2.56
dFlowGRPO (PickScore)	1.28
dFlowGRPO (GenEval)	1.91
dFlowGRPO (GenEval, w/ KL)	2.02
dFlowGRPO (PickScore, $n = 1$)	0.93
diffu-GRPO	0.87
diffu-GSPO	0.83
dFlowDPO	0.84

E.2 ScienceQA Answer Extraction Prompt

We follow the VLMEvalKit protocol. The answer letter is first extracted by exact rule-based matching; the LLM judge is called only when that fails. The prompt is copied from VLMEvalKit:

```
You are an AI assistant who will help me to match an answer with
several options of a single-choice question.  You are provided with
a question, several options, and an answer, and you need to find
which option is most similar to the answer.  If the meaning of all
options are significantly different from the answer, output Z. Your
should output a single uppercase character in A, B, C, D (if they
are valid options), and Z.
Example 1:
Question:  What is the main object in image?
Options:  A. teddy bear B. rabbit C. cat D. dog
Answer:  a cute teddy bear
Your output:  A
Example 2:
Question:  What is the main object in image?
Options:  A. teddy bear B. rabbit C. cat D. dog
Answer:  Spider
Your output:  Z
Example 3:
Question:  {question}?
Options:  {options}
Answer:  {model output}
Your output:
```

On the ScienceQA test split (2015 prompts), we checked every question on which the judge model was called and counted how often it read the wrong letter. The results are reported in table 9. We can also see that our model can generate more answers that can be extracted by rule-based matching, which is consistent with the higher accuracy of our model.

Table 9: Answer-extraction statistics on the ScienceQA test split.

Model	LLM judge calls	Extraction errors
FUDOKI	52 (2.58%)	9
w/ dFlowGRPO	25 (1.24%)	2

F Additional Experiments on a Second DFM Backbone

To further demonstrate the effectiveness and robustness of our approach, we apply dFlowGRPO to a second base model, FS-DFM 1.3B[3] (Monsefi et al., 2025), a text-only DFM built on a mixture path

[3]Hugging Face: `https://huggingface.co/aminr8/FS-DFM`.

Table 10: Evaluation results of dFlowGRPO on FS-DFM 1.3B for sentiment-controlled review completion on the IMDB test set.

Model	Success rate	Success (pos. label)	Success (neg. label)	$\mathrm{distinct}_1$	Reward
FS-DFM 1.3B	0.6007	0.4479	0.7535	0.9223	1.5214
w/ dFlowGRPO	0.9915	0.9960	0.9870	0.9961	1.9870

with a uniform source and the conditional rate

$$q^d_{t|1}(x^d \mid x^d_1) = \kappa_t \delta_{x^d_1}(x^d) + \frac{1-\kappa_t}{|\mathcal{S}|},$$
$$Q^d_t(x^d, z^d \mid x^d_1) = \frac{\dot{\kappa}_t}{1-\kappa_t}\left[\delta_{x^d_1}(z^d) - \delta_{x^d}(z^d)\right],$$

with $\kappa_t = t$.

The task is review completion on the IMDB dataset (Maas et al., 2011), a standard corpus of 50k movie reviews annotated with binary (positive/negative) sentiment; we draw prompts from its labelled train/test splits (10k / 2k). The prompt is a three-word phrase ("Positive movie review:" or "Negative movie review:") followed by the first 32 tokens of a real review, and the model generates the next 32 tokens. Note that only the generated 32 tokens are scored, and the goal is for their sentiment to match the review's label. The reward is

$$\mathcal{R} = P(\text{target sentiment} \mid \text{generated text}) + \mathrm{distinct}_1(\text{generated text}),$$

where $P(\cdot)$ is given by a sentiment classifier[4] and $\mathrm{distinct}_1$ is the ratio of unique unigrams to total unigrams in the generated text, which penalizes degenerate repetition.

In our implementation, we set the number of MC samples to 24 and the learning rate to $5\mathrm{e}{-6}$ with $\beta_{\mathrm{KL}} = 0.01$, and keep the optimizer settings the same as in the paper's experiments. We train the model on the training set for 1000 training steps and report the evaluation results on the test set below. The number of sampling steps is set to 20. dFlowGRPO significantly improves the performance, further demonstrating the effectiveness and superiority of the proposed method.

We can see that dFlowGRPO raises the reward from 1.5214 to 1.9870 and the success rate from 0.6007 to 0.9915.

G Qualitative Results

In this section, we present qualitative results, including visualizations of the sampling trajectories during training and comparisons between the base model and the model after dFlowGRPO training.

G.1 Trajectory Results

figures 7 and 8 show how the denoising trajectories change over the course of training. We can see that dFlowGRPO training enables the model to form more details in the early denoising steps in the course of training.

G.2 Results at Terminal Time

We present the qualitative results for dFlowGRPO and the base model, FUDOKI, on both image generation and multimodal understanding tasks in figures 9 to 13. We can see that the generation quality and multimodal understanding capability of dFlowGRPO significantly outperform those of the base model.

[4]Hugging Face: `https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english`.

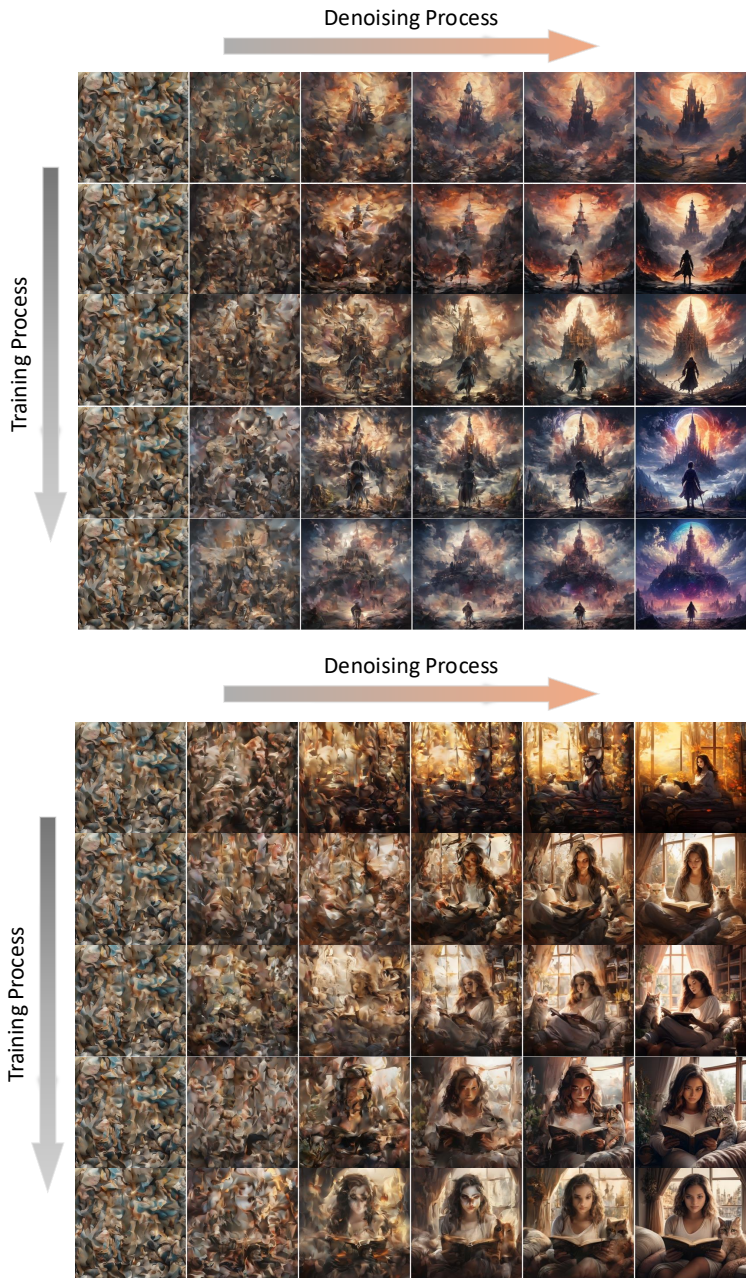

Figure 7: Visualization of the trajectory of dFlowGRPO during training with PickScore reward.

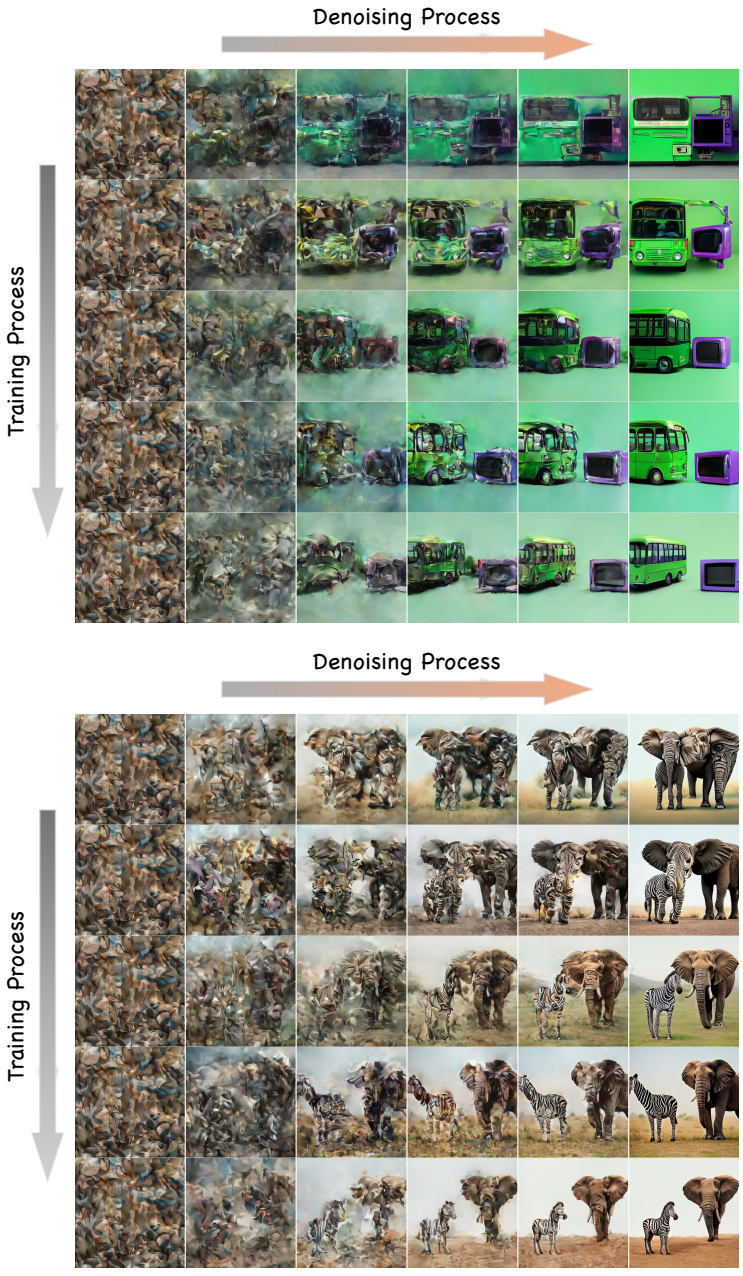

Figure 8: Visualization of the trajectory of dFlowGRPO during training with GenEval reward.

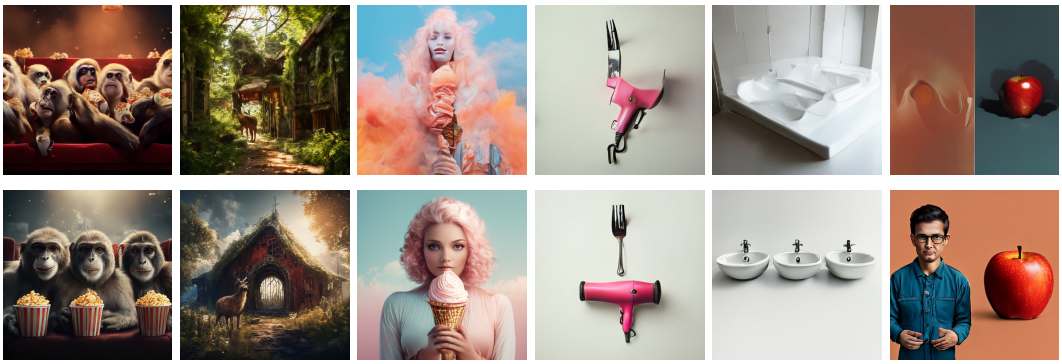

Figure 9: Qualitative comparison of generation quality between the base model FUDOKI (top) and dFlow-GRPO (bottom).

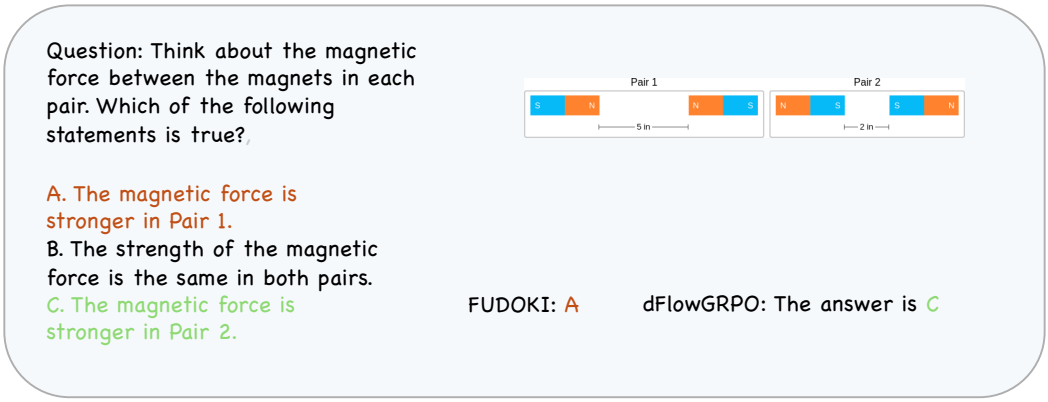

Figure 10: Qualitative comparison of understanding capability between dFlowGRPO and the base model FUDOKI. The correct answer is highlighted in green and the wrong answer is highlighted in red.



Figure 11: Qualitative comparison of understanding capability between dFlowGRPO and the base model FUDOKI. The correct answer is highlighted in green and the wrong answer is highlighted in red.

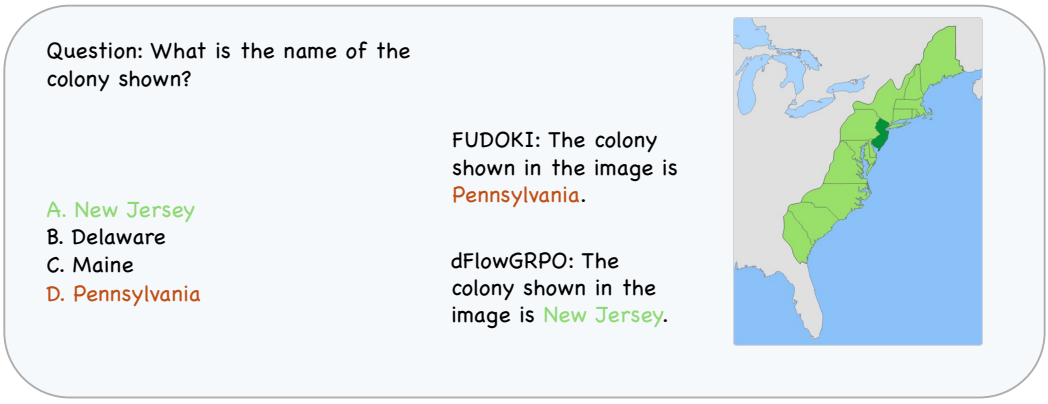

Figure 12: Qualitative comparison of understanding capability between dFlowGRPO and the base model FUDOKI. The correct answer is highlighted in green and the wrong answer is highlighted in red.



Figure 13: Qualitative comparison of understanding capability between dFlowGRPO and the base model FUDOKI. The correct answer is highlighted in green and the wrong answer is highlighted in red.