MixGRPO: Unlocking Flow-based GRPO Efficiency with Mixed ODE-SDE

Junzhe Li[1,2,3⋆], Yutao Cui[1*], Tao Huang[1*], Weijie Kong[1], Chuxuan Zeng[4], Yinping Ma[3], Chun Fan[3], Miles Yang[1], Zhao Zhong[1✉], and Liefeng Bo[1]

[1] Hunyuan, Tencent, China
[2] School of Computer Science, Peking University, China
[3] Computer Center, Peking University, China
[4] China Mobile Communications Group, China

Abstract. Although GRPO substantially improves flow-matching models for human preference alignment in vision generation, mainstream methods such as DanceGRPO rely on Global-Stochastic Differential Equations (SDE) sampling across full timesteps in the Markov Decision Process (MDP), which remains computationally inefficient. In this paper, we propose ***MixGRPO***, a novel framework that leverages the flexibility of mixed sampling strategies through the integration of SDE and Ordinary Differential Equations (ODE). This redesign streamlines optimization within the MDP, **delivering gains in both generation performance and training efficiency**. Specifically, MixGRPO introduces a sliding window mechanism, using SDE sampling and GRPO-guided optimization only within the window, while applying ODE sampling outside. This design confines sampling randomness to the time-steps within the window, thereby reducing the optimization overhead, and allowing for more focused gradient updates to accelerate convergence. Additionally, as time-steps beyond the sliding window are not involved in optimization, higher-order solvers are supported for faster sampling. So we present a faster variant, termed ***MixGRPO-Flash***, which further improves training efficiency while achieving comparable performance. MixGRPO exhibits substantial gains across multiple dimensions of human preference alignment, outperforming DanceGRPO in both effectiveness and efficiency, **with nearly 50% lower training time**. Notably, MixGRPO-Flash further **reduces training time by 71%**.[5]

Keywords: Vision Generation · Flow-matching · Online RL · GRPO

1 Introduction

Recent advances [20,21,48,49] in vision generation have demonstrated that probability flow models can achieve improved performance by incorporating Reinforcement Learning from Human Feedback (RLHF) [30] strategies during the

⋆ Equal contribution. (lijunzhe1028@gmail.com)
✉ Corresponding authors.
[5] Code is available in https://github.com/Tencent-Hunyuan/MixGRPO

Fig. 1: Comparison of MixGRPO and DanceGRPO with **varying denoising timesteps optimized**. MixGRPO achieves higher performance with lower overhead.

post-training stage to maximize rewards. Specifically, methods [20, 49] based on Group Relative Policy Optimization (GRPO) [36], have recently been studied, achieving optimal alignment with human preferences.

Current GRPO methods in probability flow models, such as Flow-GRPO and DanceGRPO, combine a rollout phase followed by an online optimization phase. During rollouts, they employ Stochastic Differential Equation (SDE) sampling at each denoising step to introduce randomness, thereby enabling the stochastic exploration required by RLHF. This allows the entire denoising process to be framed as a Markov Decision Process (MDP) in a stochastic environment, where GRPO is then applied to optimize the complete state-action sequence in the online optimization phase. However, the requirement to optimize the entire denoising steps introduces two challenges, *prohibitive computational overhead* and *unfocused and inefficient optimization*. i) The overhead arises because computing the policy ratio requires separate full-trajectory sampling from both the old ($\pi_{\theta_{\text{old}}}$) and new policies ($\pi_\theta$). Although DanceGRPO proposes a workaround by optimizing a random subset of steps, our findings in Figure 1 reveal a critical trade-off: reducing the subset size to save computation severely compromises model performance. ii) Early-step gradients prioritize global structure while late-step gradients focus on fine details, leading to conflicting update signals. This may result in an inefficient training process.

To address these issues, we propose MixGRPO, a method that optimizes a strategic subset of denoising steps to drastically reduce computational overhead while ensuring a more focused and efficient optimization. The policy optimization is confined only to the SDE sub-interval. This interval operates as a sliding window, systematically advancing from low-SNR (signal-to-noise ratio) steps to high-SNR denoising steps as training progresses. This curriculum-based approach is conceptually analogous to temporal discounting in RL [2, 12, 31], as it prioritizes the optimization of initial, high-impact steps that establish global structure from a vast exploration space (Figure 2 Right), before progressively shifting focus to the refinement of local details. This selective optimization not only reduces the computational burden but also fosters a more dedicated exploration. Furthermore, this decoupling allows us to accelerate the deterministic ODE portions with fast solvers (*e.g.,* DPMSolver++ [25]), since their posterior distributions are irrelevant to the optimization, thereby reducing rollout time without compromising final image quality. Finally, we adopt Coefficients-Preserving Sampling (CPS) [43] as a stable stochastic sampler in our framework, which reduces sampling artifacts and yields more reliable reward feedback.

To evaluate the performance and efficiency of MixGRPO, we first conduct comprehensive experiments on FLUX [16] by directly comparing against DanceGRPO [49], a representative Global-SDE baseline. Results show that our method consistently surpasses DanceGRPO in both effectiveness and efficiency. In particular, MixGRPO improves ImageReward [48] from 1.088 to 1.629, exceeding DanceGRPO's 1.436, while generating images with better semantic fidelity, aesthetics, and fewer distortions. Moreover, compared with DanceGRPO under its official setting, MixGRPO reduces training time by nearly 50%, and MixGRPO-Flash further achieves a 71% training-time reduction. These results highlight the advantage of our Mixed ODE-SDE with sliding-window formulation. We further validate MixGRPO on both in-domain and out-of-domain reward models, and extend evaluation to cross-dataset settings, where MixGRPO demonstrates strong generalization. Beyond this, we verify robustness across different backbone models and scenarios, including SD3.5-M [5] and HunyuanImage-3.0 [4], and further extend the method to video generation on HunyuanVideo-1.5 [41], where consistent gains are observed. Finally, to demonstrate the competitiveness of MixGRPO, we compare with other RL-based alignment methods, including Flow-DPO [21], Flow-GRPO [20], and DiffusionNFT [55]. Across these comparisons, MixGRPO achieves advanced alignment performance while delivering substantial training acceleration.

To summarize, the key contributions are outlined below:

- We propose a mixed ODE-SDE framework for GRPO training with flow-based models. This approach alleviates computational overhead by confining the stochastic optimization to a specific sub-interval of the MDP.
- We introduce an SDE sliding window to dynamically schedule the optimized timesteps. By guiding the optimization from broad exploration to fine-grained refinement, this strategy significantly enhances performance.
- Our hybrid framework enables the use of high-order ODE solvers to accelerate the deterministic sampling of $\pi_{\theta_{\text{old}}}$ during GRPO training. This yields substantial speedups with negligible performance degradation.
- Comprehensive experiments across multiple frameworks demonstrate the versatility of MixGRPO. Our method consistently achieves substantial performance gains while significantly reducing training overhead, highlighting its broad applicability and effectiveness.

2 Related Work

RLHF for vision generation has advanced from differentiable reward-based fine tuning [48] to PPO-style [27] and DPO-style [21], and more recently to GRPO-based optimization in flow-based models, where SDE-induced stochasticity enables broader exploration. Methods such as Flow-GRPO and DanceGRPO improve alignment quality, but still suffer from a key bottleneck: optimizing many imbalanced timesteps, which leads to redundant effective MDP horizons under Global-SDE sampling. In parallel, sampling research has progressed from ODE

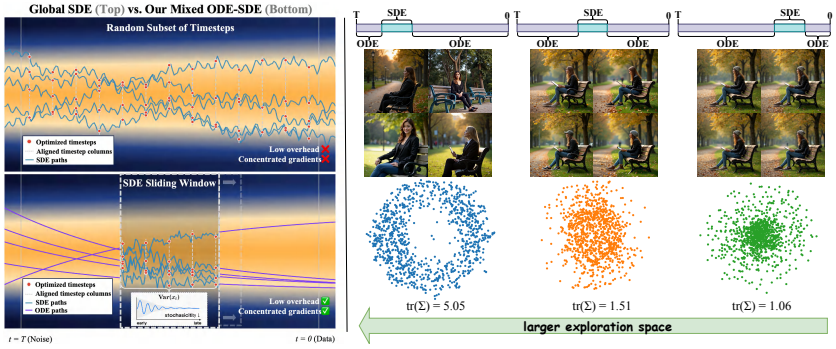

Fig. 2: (Left) **Methods Comparison.** By employing the mixed ODE-SDE and sliding window strategy, MixGRPO reduces the number of optimized timesteps and achieves more efficient training than DanceGRPO [49]. (Right) **SDE Exploration Analysis.** As training progresses, MixGRPO gradually slides the SDE window from a high-noise to a low-noise regime, causing the exploration space to shrink and the distribution of sampled images to transition from dispersed to concentrated.

solvers to high-order variants. Motivated by the SDE-ODE equivalence in probability flow, we establish a mixed SDE-ODE MDP formulation and systematically explore different scheduling strategies for optimizing selected SDE timesteps, establishing a more efficient paradigm, MixGRPO. A more comprehensive review of RL for Vision Generation and Sampling Methods is provided in **Appendix 7**.

3 Method

3.1 Mixed ODE-SDE Sampling in GRPO

According to Flow-GRPO [20], the SDE sampling in flow matching can be framed as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \rho_0, P, \mathcal{R})$. The agent produces a trajectory during the discrete sampling process, defined as $\Gamma = \{(\mathbf{s}_t, \mathbf{a}_t)\}_{t=0}^{T}$, where the reward is provided only at the final step by the reward model, specifically $\mathcal{R}(\mathbf{s}_i, \mathbf{a}_i) \triangleq R(\mathbf{x}_T, c)$ if $i = T$, and 0 otherwise.

In MixGRPO, we propose a sampling method that combines SDE and ODE. MixGRPO defines a time interval $S = [t_l, t_r) \subseteq [0, 1)$, which corresponds to a subset of denoising timesteps, such that $0 \leq l < r \leq T$ and $t_i = \frac{i}{T}$. We use SDE sampling within the interval S and ODE sampling outside, while S shifts along the denoising direction throughout the training process (See Figure 2 Left). MixGRPO restricts the stochastic exploration to the interval S, shortening the sequence length of the MDP to a subset $\Gamma_{\text{MixGRPO}} = \{(\mathbf{s}_t, \mathbf{a}_t)\}_{t=l}^{r}$ and requires policy optimization only within S:

$$\max_{\theta} \mathbb{E}_{\Gamma_{\text{MixGRPO}} \sim \pi_\theta} \left[\sum_{t \in S} \left(\mathcal{R}(\mathbf{s}_t, \mathbf{a}_t) - \beta D_{\text{KL}} \big(\pi(\cdot \mid \mathbf{s}_t) \| \pi_{\text{ref}}(\cdot \mid \mathbf{s}_t) \big) \right) \right], \tag{1}$$

MixGRPO reduces training overhead while also enabling more concentrated gradient updates. Next, we derive the specific sampling form. For a deterministic

reverse *probability flow ODE* [39], it takes the following form:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = f(\mathbf{x}_t, t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t), \tag{2}$$

where $q_t(\mathbf{x}_t)$ represents the evolution process of the reverse probability distribution from 0 to T. $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ is the *score function* at time t. According to the Fokker-Planck equation [29, 32], [39] has demonstrated that Eq. (2) has the following equivalent *probability flow SDE*, which maintains the same marginal distribution at each time t:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = f(\mathbf{x}_t, t) - g^2(t) \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) + g(t) \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t}. \tag{3}$$

In MixGRPO, we mix ODE and SDE for sampling. Under standard regularity assumptions (continuous $g(t)$, locally Lipschitz drift/score fields, finite second moments, and sufficiently small Δt), the mixed process preserves the same time-marginal dynamics as the corresponding probability-flow ODE up to discretization/score-approximation error; a full proof is provided in Appendix 8. The specific form is as follows:

$$\mathrm{d}\mathbf{x}_t = \begin{cases} \left[f(\mathbf{x}_t, t) - g^2(t) \mathbf{s}_t(\mathbf{x}_t) \right] \mathrm{d}t + g(t) \mathrm{d}\mathbf{w}, & \text{if } t \in S, \\ \left[f(\mathbf{x}_t, t) - \frac{1}{2} g^2(t) \mathbf{s}_t(\mathbf{x}_t) \right] \mathrm{d}t, & \text{otherwise,} \end{cases} \tag{4}$$

where we define the score function as $\mathbf{s}_t \triangleq \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$. In particular, for Flow Matching (FM) [19], especially the Rectified Flow (RF) [22], the sampling process can be viewed as a deterministic ODE:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \mathbf{v}_t. \tag{5}$$

Eq. (5) is actually a special case of the Eq. (2) with $\mathbf{v}_t = f(\mathbf{x}_t, t) - \frac{1}{2} g^2(t) \mathbf{s}_t(\mathbf{x}_t)$. So we can derive mixed ODE-SDE sampling form for RF as follows:

$$\mathrm{d}\mathbf{x}_t = \begin{cases} \left[\mathbf{v}_t - \frac{1}{2} g^2(t) \mathbf{s}_t(\mathbf{x}_t) \right] \mathrm{d}t + g(t) \mathrm{d}\mathbf{w}, & \text{if } t \in S, \\ \mathbf{v}_t \mathrm{d}t, & \text{otherwise.} \end{cases} \tag{6}$$

In the RF framework, the model is used to predict the velocity field as $\mathbf{v}_\theta(\mathbf{x}_t, t) = \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}$. Following [20], the *score function* is represented as $\mathbf{s}_t(\mathbf{x}_t) = -\frac{\mathbf{x}_t}{t} - \frac{1-t}{t} \mathbf{v}_\theta(\mathbf{x}_t, t)$. The $g(t)$ is represented as the standard deviation of the noise $g(t) = \sigma_t$. According to the definition of the standard Wiener process, we use $\mathrm{d}\mathbf{w} = \sqrt{\mathrm{d}t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Applying Euler-Maruyama discretization for SDE and Euler discretization for ODE, we build the final denoising process in MixGRPO:

$$\mathbf{x}_{t+\Delta t} = \begin{cases} \mathbf{x}_t + \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \Delta t + \sigma_t \sqrt{\Delta t} \epsilon, & \text{if } t \in S, \\ \mathbf{x}_t + \mathbf{v}_\theta(\mathbf{x}_t, t) \Delta t, & \text{otherwise,} \end{cases} \tag{7}$$

where the SDE drift term for the sampling process is defined as $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \triangleq \mathbf{v}_\theta(\mathbf{x}_t, t) + \frac{\sigma_t^2 (\mathbf{x}_t + (1-t)\mathbf{v}_\theta(\mathbf{x}_t, t))}{2t}$. For timesteps $t \in S$, Eq. (7) induces (via Euler–Maruyama) the conditional policy density

$$q_\theta(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t, c) = \mathcal{N}\left(\mathbf{x}_t + \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \Delta t, \sigma_t^2 \Delta t \, \mathbf{I} \right), \quad t \in S, \tag{8}$$

with the same form for $q_{\theta_{\text{old}}}$. For $t \notin S$, the update is deterministic (Dirac transition), and these steps are excluded from GRPO ratio/KL evaluation. According to Eq. (1), GRPO optimization is only performed on timesteps within the interval S for each group of N samples. This design not only shortens the effective MDP horizon but also concentrates gradient updates, leading to the following training objective:

$$\mathcal{J}_{\text{MixGRPO}}(\theta) = \mathbb{E}_{c\sim\mathcal{C},\ \{\mathbf{x}_T^i\}_{i=1}^N\sim\pi_{\theta_{\text{old}}}(\cdot|c)}\Bigg[\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|S|}\sum_{t\in S}\Big(\min\big(r_t^i(\theta)A^i, \text{clip}(r_t^i(\theta),1-\varepsilon,1+\varepsilon)A^i\big) - \beta\mathcal{J}_{\text{KL}}\Big)\Bigg], \tag{9}$$

where ε is the clipping threshold, $r_t^i(\theta)$ is the policy ratio, and A^i is the advantage score. For optimized timesteps ($t \in S$), we parameterize the policy by the mean of the Gaussian transition above, while the covariance is fixed to $\sigma_t^2 \Delta t\, \mathbf{I}$ for both θ and θ_{old}. Therefore, the ratio and KL are defined as follows according to [20]:

$$\begin{aligned} r_t^i(\theta) &= \frac{q_\theta(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t,c)}{q_{\theta_{\text{old}}}(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t,c)}, \quad t\in S, \quad A^i = \frac{R\left(\mathbf{x}_T^i,c\right) - \text{mean}\left(\{R\left(\mathbf{x}_T^i,c\right)\}_{i=1}^N\right)}{\text{std}\left(\{R\left(\mathbf{x}_T^i,c\right)\}_{i=1}^N\right)}, \\ \mathcal{J}_{\text{KL}} &= D_{\text{KL}}(q_\theta(\cdot|\mathbf{x}_t,c)\,||\,q_{\theta_{\text{old}}}(\cdot|\mathbf{x}_t,c)) = \frac{||\mathbf{x}_{t+\Delta t}(\theta) - \mathbf{x}_{t+\Delta t}(\theta_{\text{old}})||^2}{2\sigma_t^2\Delta t}, \quad t\in S. \end{aligned} \tag{10}$$

MixGRPO reduces the NFE of π_θ and optimized timesteps compared to Global-SDE sampling. However, the NFE of $\pi_{\theta_{\text{old}}}$ is not reduced, as complete inference is required to obtain the final image for reward calculation. In Section 3.3, we will introduce the use of higher-order ODE solvers, which also reduce the NFE of $\pi_{\theta_{\text{old}}}$ leading to further speedup. In summary, the mixed ODE-SDE sampling significantly streamlines the MDP, enhancing training efficiency by lowering optimization overhead and enabling more focused gradient updates.

3.2 Sliding Window as Optimized Timestep Scheduler

In this section, we will introduce the sliding window to describe the movement of S, which leads to a significant improvement in the quality of the generated images. Along the denosing time-steps $\{0, 1, \ldots, T-1\}$, MixGRPO defines a SDE sliding window $W(l)$ and optimization is only employed at the timesteps within $W(l)$.

$$W(l) = \{t_l, t_{l+1}, \ldots, t_{l+w-1}\}, \quad l \le T - w, \tag{11}$$

where l is the *left boundary* of the sliding window. The movement strategy during training is governed by three hyperparameters: **(1) window size** w, which determines the number of timesteps included in each window; **(2) shift interval** τ, which specifies the number of training steps between two consecutive window shifts; and **(3) window stride** s, which indicates how many denoising timesteps the left boundary l advances at each shift. In this work, we use a fixed progressive-constant scheduler as a simple and stable default across experiments.

Algorithm 1 MixGRPO Training Process

Require: initial policy model π_θ; reward models $\{R_k\}_{k=1}^{K}$; prompt dataset $\mathcal{C}$; total sampling steps T; number of samples per prompt N;
Require: sliding window $W(l)$, window size w, shift interval τ, window stride s
Init left boundary of $W(l)$: $l \leftarrow 0$
for training iteration $m = 1$ **to** M **do**
  Sample batch prompts $\mathcal{C}_b \sim \mathcal{C}$
  Update old policy model: $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  for each prompt $\mathbf{c} \in \mathcal{C}_b$ **do**
    Init the same noise $\mathbf{x_0} \sim \mathcal{N}(0, \mathbf{I})$ ▷ according to DanceGRPO [49]
    for generate i-th image from $i = 1$ **to** N **do**
      for sampling timestep $t = 0$ **to** $T - 1$ **do**
        if $t \in W(l)$ **then**
          Use SDE Sampling to get $\mathbf{x}_{t+1}^i$
        else
          Use ODE Sampling to get $\mathbf{x}_{t+1}^i$
        end if
      end for
    end for
    for i-th image from $i = 1$ **to** N **do**
      Calculate advantages: $A_i \leftarrow \sum_{k=1}^{K} \frac{R(\mathbf{x}_T^i, \mathbf{c})_k^i - \mu_k}{\sigma_k}$
    end for
    for optimization timestep $t \in W(l)$ **do**
      Update policy model: $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{J}$
    end for
  end for
  if m mod τ is 0 **then** ▷ move sliding window
    $l \leftarrow \min(l + s,\ T - w)$
  end if
end for

The detailed sliding-window strategy and the MixGRPO algorithm are provided in Algorithm 1.

In MixGRPO, denoising follows a natural stochastic-to-deterministic trajectory along probability flow. We exploit this property with a sliding-window curriculum: optimization starts in low-SNR (high-stochasticity) regions and progressively shifts to high-SNR (more deterministic) regions for refinement, as illustrated in Figure 2 (right). This coarse-to-fine schedule stabilizes early optimization and improves final alignment quality. Table 4 shows that progressive movement consistently outperforms random selection, while even a frozen window on early timesteps remains competitive, indicating that early denoising steps contribute disproportionately to optimization payoff [5, 14].

3.3 Trade-off Between Overhead and Performance

Unlike DanceGRPO [49], which relies on Global-SDE sampling, MixGRPO employs a mixed ODE-SDE method, allowing the use of higher-order ODE solvers

Fig. 3: Qualitative comparison. MixGRPO achieve superior performance in semantics, aesthetics and text-image alignment.

to accelerate GRPO training-time sampling. However, accelerating the ODE phase before the SDE window amplifies the solver's numerical errors due to the window's stochasticity, severely degrading image quality and corrupting the reward signal (See Appendix 13 for details). In contrast, accelerating the ODE phase after the window provides a significant speed-up without compromising the image fidelity required for reliable reward evaluation. Therefore, MixGRPO exclusively accelerates the post-window ODE timesteps.

[9] has demonstrated the equivalence between the ODE sampling of flow matching models (FM) and DDIM, and Section 3.1 has also shown that diffusion probabilistic models (DPM) and FM share the same ODE form during the denoising process. Therefore, the higher-order ODE solvers *e.g.,* DPM-Solver Series [24, 25, 56], UniPC [54] designed for DPM sampling acceleration are also applicable to FM. We have reformulated DPM-Solver++ [25] to apply it in the FM framework for ODE sampling acceleration and released detailed derivations in Appendix 9.

By applying higher-order solvers, we achieve acceleration in the sampling of $\pi_{\theta_{\text{old}}}$ during GRPO training, which is essentially a balance between overhead and performance. Excessive acceleration leads to fewer timesteps, which inevitably results in a decline in image generation quality, thereby accumulating errors in the computation of rewards. We have found in practice that the 2nd-order DPM-Solver++ is sufficient to provide significant acceleration while ensuring that the generated images align well with human preferences in Table 8.

Ultimately, we integrate DPM-Solver++ with both progressive and frozen sliding-window strategies, leading to MixGRPO-Flash and MixGRPO-Flash*. Notably, MixGRPO-Flash* enables acceleration over a larger number of ODE timesteps. A detailed description of the algorithm is provided in Appendix 10. These faster versions offer more substantial acceleration relative to base version of our MixGRPO, while simultaneously surpassing DanceGRPO in their alignment with human preferences.

Table 1: Comparison between Mixed ODE-SDE and Global-SDE. MixGRPO achieves the best performance across multiple metrics. MixGRPO-Flash significantly reduces training time while outperforming DanceGRPO. **Bold**: rank 1. Underline: rank 2. *The Frozen strategy means that optimization is only employed at the initial denoising steps.

Model	$\text{NFE}_{\pi_{\theta_{\text{old}}}}\downarrow$	$\text{NFE}_{\pi_\theta}\downarrow$	Iteration Time (s)↓	Human Preference Alignment			
				HPS-v2.1↑	Pick Score↑	ImageReward↑	Unified Reward↑
FLUX	/	/	/	0.313	0.227	1.088	3.370
DanceGRPO	25	14	291.284	0.356	0.233	1.436	3.397
	25	4	149.978	0.334	0.225	1.335	3.374
	25	4*	150.059	0.333	0.229	1.235	3.325
MixGRPO	25	4	149.326	**0.369**	**0.238**	**1.645**	**3.419**
MixGRPO-Flash	16 (Avg)	4	<u>112.372</u>	<u>0.358</u>	<u>0.236</u>	1.528	<u>3.407</u>
MixGRPO-Flash*	8	4*	**83.278**	0.357	0.232	<u>1.624</u>	3.402

3.4 Coefficients-Preserving Sampling for Stable Exploration

In addition to the mixed ODE-SDE design, we adopt Coefficients-Preserving Sampling (CPS) [43] to improve sampling stability in the stochastic segment. Compared with standard SDE discretization, CPS better preserves the interpolation structure of flow matching while injecting controlled stochasticity, reducing high-frequency artifacts that can lead to reward hacking. In our implementation, CPS is used as an alternative stochastic discretization while keeping the same per-step Gaussian covariance scale $\sigma_t^2 \Delta t\, \mathbf{I}$ for GRPO density-ratio/KL evaluation in Eq. (9); therefore, the policy-ratio and KL parameterization remain unchanged. This yields cleaner rollouts and more reliable reward signals for GRPO optimization. Detailed derivations are provided in Appendix 19.

4 Experiments

4.1 Experiment Setup

Dataset We conduct T2I experiments using the prompts provided by the HPDv2 dataset, which is the official dataset for the HPS-v2 benchmark [47]. The training set contains 103,700 prompts; in fact, MixGRPO already achieves strong human preference alignment before completing one full epoch, using only 9,600 prompts. The test set consists of 400 prompts. The prompts are diverse, encompassing four styles: "Animation", "Concept Art", "Painting", and "Photo".
Base Model Following DanceGRPO [49], we use FLUX.1-dev [16] as the primary backbone, a strong flow-matching-based text-to-image model. To further verify the robustness of MixGRPO, we also extend our method to additional backbones, including LoRA fine-tuning on SD3.5-M [5]. More implementation details are in Appendix 11.
Overhead Evaluation For the evaluation of overhead, we use two metrics: the number of function evaluations (NFE) [24] and the time consumption per iteration during training. The NFE is decomposed into $\text{NFE}_{\pi_{\theta_{\text{old}}}}$ and NFE_{π_θ}. $\text{NFE}_{\pi_{\theta_{\text{old}}}}$ denotes the number of forward passes of the reference model used for rollout sampling. NFE_{π_θ} is the number of forward passes of the policy model solely for the policy ratio. Additionally, the average training time per GRPO iteration provides a more accurate reflection of the acceleration effect.

Fig. 4: Qualitative results of MixGRPO-Flash. As the training overhead is reduced, generated images maintain high quality.

Reward Model We employ four reward models: HPS-v2.1 [47], Pick Score [15], ImageReward [48], and Unified Reward [44]. Although all four are preference-aligned metrics, they capture complementary aspects of quality. In particular, ImageReward emphasizes image-text alignment and fidelity, while Unified Reward focuses more on semantic consistency. Following the findings of DanceGRPO [49], we adopt multi-reward guidance to obtain stronger and more stable alignment performance. To validate the generalization capability of MixGRPO, we conduct cross-domain evaluations across both reward models and datasets.

4.2 Main Results

Mixed ODE-SDE GRPO vs. Global-SDE GRPO We first evaluated the overhead and performance of MixGRPO against Global-SDE GRPO methods (e.g., DanceGRPO), with results shown in Table 1. In the official DanceGRPO setting, 14 timesteps are randomly selected from 25 for optimization. Our results show that MixGRPO achieves better performance while optimizing only 4 timesteps, reducing per-iteration GRPO training time from 291.284s to 150.839s. Qualitative results in Figure 3 further show that MixGRPO improves both aesthetic quality and semantic alignment. For fairness, we also modify DanceGRPO to use 4 timesteps (either randomly selected or the initial ones), matching the overhead of MixGRPO; however, DanceGRPO remains inferior to MixGRPO in alignment performance under both settings. For MixGRPO-Flash, we evaluate both progressive and frozen strategies. Although MixGRPO-Flash is slightly weaker than MixGRPO, all evaluation metrics are still higher than

those of DanceGRPO (official setting), and the fastest setting reduces time from 291.284s to 83.278s. As shown in Figure 4, MixGRPO-Flash maintains strong image quality even as overhead changes. More visualized results are provided in Appendix 20.

Cross-domain Reward Models To evaluate the generalization ability of MixGRPO, we follow DanceGRPO and use HPS-v2.1 (aesthetics) and CLIP Score [10] (semantic-consistency) as in-domain reward models, under both single-reward and two-reward training settings. We further assess out-of-domain performance using Pick Score, ImageReward, and Unified Reward. As shown in Table 2, MixGRPO consistently outperforms both DanceGRPO and the FLUX baseline on in-domain and out-of-domain metrics. These results indicate that MixGRPO improves overall image-generation alignment rather than relying on reward hacking (i.e., overfitting to in-domain reward models while failing to match human preference).

MixGRPO vs. Other Alignment Methods In addition to DanceGRPO, we compare MixGRPO with other representative alignment methods on SD3.5-M, including DiffusionNFT [55], Offline/Online Flow-DPO, and Flow-GRPO, with results shown in Table 3. For fairness, the settings of the other alignment methods follow the best configurations reported in their original papers. We use HPS-v2.1, Pick Score, and ImageReward as multi-guidance and evaluation metrics. The results show that MixGRPO can converge to the best performance during training while optimizing only 4 timesteps. More experimental details are provided in Appendix 11. We also compare with ReNO [6], an alignment method for single-step generators; as shown in Appendix 12, MixGRPO achieves better performance without incurring the overhead of step distillation.

Extension to Larger-Scale Model and Video Generation We further validate the scalability of MixGRPO beyond standard T2I settings. On the industrial-scale HunyuanImage-3.0 [4] backbone (80B parameters) trained on 512 GPUs, MixGRPO maintains stable optimization and substantial efficiency gains, demonstrating that its advantages persist at very large scale. Appendix 14 provides additional experimental details and visualizations of human blind-test preferences. We also extend MixGRPO to HunyuanVideo-1.5 [41] for text-to-video alignment, where it consistently outperforms Flow-GRPO in training stability and reward improvement. These results confirm that our mixed ODE-SDE design generalizes well across both larger image models and video generation tasks (see Appendix 15 for training stability curves and more details).

4.3 Ablation Experiments

Sliding Window Hyperparamters As introduced in Section 3.2, the *moving strategy*, *shift interval* τ, *window size* w and *window stride* s are parameters

Table 2: Comparison under cross-rewards. The results demonstrate that MixGRPO achieves the best performance on both in-domain and out-of-domain rewards.

Reward Model	Method	In Domain		Out-of-Domain		
		HPS-v2.1	CLIP Score	Pick Score	ImageReward	Unified Reward
/	FLUX	0.313	0.388	0.227	1.088	3.370
HPS-v2.1	DanceGRPO	0.367	0.349	0.227	1.141	3.270
	MixGRPO	**0.373**	**0.372**	**0.228**	**1.396**	**3.370**
HPS-v2.1 & CLIP Score	DanceGRPO	0.346	0.400	0.228	1.314	3.377
	MixGRPO	**0.349**	**0.415**	**0.229**	**1.416**	**3.430**

Table 3: Comparison with other alignment methods. The results demonstrate that MixGRPO outperforms others in training efficiency and performance.

Model	$\text{NFE}_{\pi_{\theta_{\text{old}}}}$	NFE_{π_θ}	HPS-v2.1	Pick Score	ImageReward
SD3.5-M	/	/	0.307	0.227	1.163
Offline Flow-DPO	40	40	0.304	0.222	1.452
Online Flow-DPO	40	40	0.313	0.221	**1.500**
DiffusionNFT	10	10	0.313	0.235	1.494
Flow-GRPO	10	10	0.331	0.232	1.457
DanceGRPO	10	10	0.309	0.226	1.433
MixGRPO	10	4	**0.342**	**0.236**	1.485

introduced by the sliding window. We conducted ablation experiments on each of them. For moving strategy, we compared three approaches: *frozen*, where the window remains stationary; *random*, where a random window position is selected at each iteration; and *progressive*, where the sliding window moves incrementally with denoising steps. For *progressive* strategy, we further evaluated different scheduling schemes in which the interval τ starts from 25 and evolves over training. As shown in Table 4, a constant schedule under the *progressive* strategy yields the strongest overall trade-off. For *shift interval* τ, we use $\tau = 25$ as a robust default setting (see Table 5). The number of forward passes for π_θ increases with the growth of the window size w, leading to greater time overhead. We compared different settings of w, and results are shown in Table 6. Ultimately, we use $w = 4$ as a balanced default setting between overhead and performance. For *window stride* s, we use $s = 1$ as the default choice, as shown in Table 7.

High Order ODE Solver MixGRPO-Flash, which incorporates a high-order ODE solver to accelerate the sampling process of the reference model, achieves an effective trade-off between speed and performance. For MixGRPO-Flash, we first conducted ablation experiments on the order of the solver, using DPM-Solver++ [25] as the high-order solver with the *progressive* strategy. The results, as shown in Table 8, indicate that the second-order mid-point method is the optimal setting, optimizing the most human preference alignment metrics while simultaneously accelerating the process.

Then we compared two acceleration approaches. One is MixGRPO-Flash, and the other is MixGRPO-Flash*. Both utilize a second-order ODE solver for acceleration, but they differ in their sliding-window moving strategies. The

Table 4: Ablation for moving strategies.

Strategy	Interval Schedule	HPS-v2.1	Pick Score	ImageReward	Unified Reward
Frozen	/	0.354	0.234	1.580	3.403
Random	Constant	0.365	0.237	1.513	3.388
Progressive	Decay (Linear)	0.365	0.235	1.566	3.382
	Decay (Exp)	0.360	**0.239**	**1.632**	3.416
	Constant	**0.367**	0.237	1.629	**3.418**

Table 5: Ablation for shift interval τ.

τ	HPS-v2.1	Pick Score	ImageReward	Unified Reward
15	0.366	0.237	1.509	3.403
20	0.366	**0.238**	1.610	3.411
25	**0.367**	0.237	**1.629**	**3.418**
30	0.350	0.229	1.589	3.385

Table 6: Ablation for window size w.

w	NFE_{π_θ}	HPS-v2.1	Pick Score	ImageReward	Unified Reward
1	1	0.359	0.234	1.629	3.235
2	2	0.362	0.235	1.588	**3.419**
4	4	0.367	0.237	**1.629**	3.418
6	6	**0.370**	**0.238**	1.547	3.398

Table 7: Ablation for window stride s.

s	HPS-v2.1	Pick Score	ImageReward	Unified Reward
1	0.367	0.237	**1.629**	**3.418**
2	0.357	0.236	1.575	3.391
3	**0.370**	0.236	1.578	3.404
4	0.368	**0.238**	1.575	3.407

quantitative results are presented in Table 9. MixGRPO-Flash requires the window to move throughout the training process, resulting in a smaller portion of the ODE being accelerated compared to MixGRPO-Flash*. Consequently, MixGRPO-Flash* not only achieves a higher degree of acceleration for the reference model but also yields superior results in the ImageReward [48] and Unified Reward [44] metrics.

Cross-Dataset Experiments and Sensitivity Analysis Following the cross-dataset protocol in Appendix 16, we train on HPD-v2 and Pick-a-Pic-v1 datasets reciprocally and evaluate on both ID and OOD splits. The same hyperparameter setting (progressive-constant, $\tau = 25$, $w = 4$, $s = 1$) remains consistently strong across datasets, while performance stays stable within a reasonable range of (τ, w, s) (Figure 5). These results indicate that MixGRPO is robust to dataset shifts and insensitive to delicate hyperparameter tuning, achieving reliable gains without additional tuning cost.

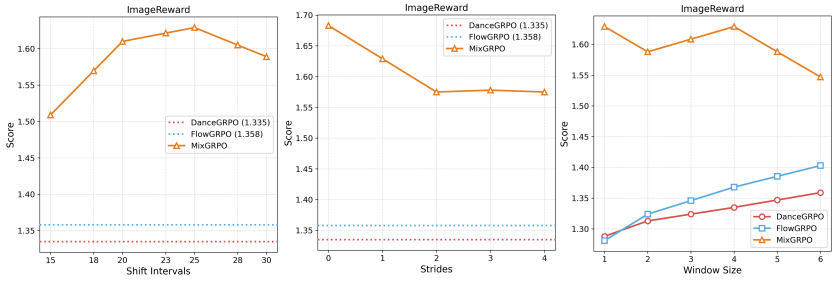

Fig. 5: Sensitivity analysis of hyperparameters (w, τ, s). The results show that MixGRPO is not sensitive to hyperparameters and consistently outperforms Global-SDE.

Coefficients-Preserving Sampling We further ablate the stochastic sampler by comparing standard SDE sampling and CPS [43] under the same training

Table 8: Comparison for different ODE solvers. The second-order Midpoint method achieves the best performance.

Order	Solver	Type	HPS-v2.1	Pick Score	ImageReward	Unified Reward
1	/		**0.367**	0.236	1.570	3.403
2	Midpoint		0.358	**0.237**	**1.578**	**3.407**
	Heun		0.362	0.233	1.488	3.399
3	/		0.359	0.234	1.512	3.387

Table 9: Comparison for sampling steps of the reference model. MixGRPO-Flash preserves performance while providing acceleration.

Method	Sampling Overhead		Human Preference Alignment			
	$NFE_{\pi_{old}}$	Time per Image (s)	HPS-v2.1	Pick Score	ImageReward	Unified Reward
DanceGRPO	25	9.301	0.334	0.225	1.335	3.374
MixGRPO-Flash	19 (Avg)	7.343	0.357	0.236	1.564	3.394
	16 (Avg)	6.426	**0.362**	**0.237**	1.578	3.407
	13 (Avg)	5.453	0.344	0.229	1.447	3.363
MixGRPO-Flash*	12	4.859	0.353	0.230	1.588	3.396
	10	4.214	0.359	0.234	1.548	**3.430**
	8	**3.789**	0.357	0.232	**1.624**	3.402

Table 10: Comparison between SDE and CPS sampling, CPS effectively improves alignment performance.

Model	HPS-v2.1	Pick Score	ImageReward	Unified Reward
FLUX	0.313	0.227	1.088	3.370
MixGRPO-SDE	0.367	0.237	1.629	3.418
MixGRPO-CPS	**0.369**	**0.238**	**1.645**	**3.419**

setup. As shown in Table 10, CPS consistently improves all reward metrics, indicating that reducing sampling artifacts provides more reliable optimization signals for preference alignment.

5 Limitation

A key limitation of GRPO is that its performance ceiling is inherently tied to the capability of the reward model. Specifically, when the reward model is not sufficiently powerful to provide a comprehensive assessment of image quality, GRPO is prone to reward hacking, particularly in the later stages of training (see Appendix 18 for concrete examples). As confirmed by several studies [28, 45, 46], reward hacking is fundamentally driven by limitations of the reward model, rather than the RL algorithm itself. In this context, the primary goal of MixGRPO is not to eliminate these reward-model limitations, but to achieve faster convergence and stronger performance under the guidance of existing, imperfect reward models. In addition, our current sliding-window scheduler still relies on predefined hyperparameters (w, τ, s). Although we observe strong robustness across datasets and backbones, an adaptive scheduler based on online signals (e.g., reward convergence or gradient variance) remains an important direction for future work. Looking ahead, while ultimate performance remains contingent on reward-model quality, we plan to develop and train stronger reward models in future work.

6 Conclusion

In this work, we have presented MixGRPO, a novel hybrid ODE-SDE framework for improving GRPO training efficiency and performance. We have proposed a strategy to confine optimization to a dynamic stochastic interval managed by a sliding window. This guides the optimization process from broad exploration to

fine-grained refinement, thus enhancing performance. Experiments demonstrate that MixGRPO has achieved superior performance in both single-reward and multi-reward settings while substantially reducing overhead. Furthermore, we have presented MixGRPO-Flash, a variant offering a flexible trade-off between performance and computational cost.

Acknowledgements

We thank Yiming Cheng from the Department of Engineering Physics, Tsinghua University (cheng-ym25@mails.tsinghua.edu.cn) for his helpful assistance with figure preparation and writing. This research is supported by the Science and Technology Development Special Fund Project of Xinjiang Production and Construction Corps (Grant No. 2025AB027).

References

1. Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. arXiv preprint arXiv:2303.08797 (2023)
2. Amit, R., Meir, R., Ciosek, K.: Discount factor as a regularizer in reinforcement learning. In: International conference on machine learning. pp. 269–278. PMLR (2020)
3. Black, K., Janner, M., Du, Y., Kostrikov, I., Levine, S.: Training diffusion models with reinforcement learning. arXiv preprint arXiv:2305.13301 (2023)
4. Cao, S., Chen, H., Chen, P., Cheng, Y., Cui, Y., Deng, X., Dong, Y., Gong, K., Gu, T., Gu, X., et al.: Hunyuanimage 3.0 technical report. arXiv preprint arXiv:2509.23951 (2025)
5. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
6. Eyring, L., Karthik, S., Roth, K., Dosovitskiy, A., Akata, Z.: Reno: Enhancing one-step text-to-image models through reward-based noise optimization. Neural Information Processing Systems (NeurIPS) (2024)
7. Fan, Y., Lee, K.: Optimizing ddpm sampling with shortcut fine-tuning. arXiv preprint arXiv:2301.13362 (2023)
8. Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Lee, K., Lee, K.: Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. Advances in Neural Information Processing Systems **36**, 79858–79885 (2023)
9. Gao, R., Hoogeboom, E., Heek, J., Bortoli, V.D., Murphy, K.P., Salimans, T.: Diffusion meets flow matching: Two sides of the same coin (2024), https://diffusionflow.github.io/, accessed: 2026-03-04
10. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 7514–7528 (2021)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)

12. Hu, H., Yang, Y., Zhao, Q., Zhang, C.: On the role of discount factor in offline reinforcement learning. In: International conference on machine learning. pp. 9072–9098. PMLR (2022)
13. Jordan, K., Jin, Y., Boza, V., Jiacheng, Y., Cesista, F., Newhouse, L., Bernstein, J.: Muon: An optimizer for hidden layers in neural networks (2024), https://kellerjordan.github.io/posts/muon/, accessed: 2026-03-04
14. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems **35**, 26565–26577 (2022)
15. Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., Levy, O.: Pick-a-pic: An open dataset of user preferences for text-to-image generation. Advances in Neural Information Processing Systems **36**, 36652–36663 (2023)
16. Labs, B.F.: Flux. https://github.com/black-forest-labs/flux (2024), accessed: 2026-03-04
17. Lee, K., Liu, H., Ryu, M., Watkins, O., Du, Y., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Gu, S.S.: Aligning text-to-image models using human feedback. arXiv preprint arXiv:2302.12192 (2023)
18. Liang, Z., Yuan, Y., Gu, S., Chen, B., Hang, T., Cheng, M., Li, J., Zheng, L.: Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 13199–13208 (2025)
19. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
20. Liu, J., Liu, G., Liang, J., Li, Y., Liu, J., Wang, X., Wan, P., Zhang, D., Ouyang, W.: Flow-grpo: Training flow matching models via online rl. arXiv preprint arXiv:2505.05470 (2025)
21. Liu, J., Liu, G., Liang, J., Yuan, Z., Liu, X., Zheng, M., Wu, X., Wang, Q., Qin, W., Xia, M., et al.: Improving video generation with human feedback. arXiv preprint arXiv:2501.13918 (2025)
22. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
24. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in Neural Information Processing Systems **35**, 5775–5787 (2022)
25. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. arXiv preprint arXiv:2211.01095 (2022)
26. Ma, Y., Wu, X., Sun, K., Li, H.: Hpsv3: Towards wide-spectrum human preference score (2025), https://arxiv.org/abs/2508.03789, accessed: 2026-03-04
27. Miao, Z., Wang, J., Wang, Z., Yang, Z., Wang, L., Qiu, Q., Liu, Z.: Training diffusion models towards diverse image generation with reinforcement learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10844–10853 (2024)
28. Nishimura-Gasparian, K., Dunn, I., Sleight, H., Turpin, M., Hubinger, E., Denison, C., Perez, E.: Reward hacking behavior can generalize across tasks—ai alignment forum. In: AI Alignment Forum (2024)
29. Øksendal, B.: Stochastic differential equations. In: Stochastic differential equations: an introduction with applications, pp. 38–50. Springer (2003)

30. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
31. Pitis, S.: Rethinking the discount factor in reinforcement learning: A decision theoretic approach. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 7949–7956 (2019)
32. Risken, H., Risken, H.: Fokker-planck equation. Springer (1996)
33. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
34. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)
35. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
36. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
37. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
38. Song, J., Zhang, Q., Yin, H., Mardani, M., Liu, M.Y., Kautz, J., Chen, Y., Vahdat, A.: Loss-guided diffusion models for plug-and-play controllable generation. In: International Conference on Machine Learning. pp. 32483–32498. PMLR (2023)
39. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
40. Tang, Z., Peng, J., Tang, J., Hong, M., Wang, F., Chang, T.H.: Inference-time alignment of diffusion models with direct noise optimization. arXiv preprint arXiv:2405.18881 (2024)
41. Team, T.H.F.M.: Hunyuanvideo 1.5 technical report (2025), https://arxiv.org/abs/2511.18870, accessed: 2026-03-04
42. Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., Naik, N.: Diffusion model alignment using direct preference optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8228–8238 (2024)
43. Wang, F., Yu, Z.: Coefficients-preserving sampling for reinforcement learning with flow matching. arXiv preprint arXiv:2509.05952 (2025)
44. Wang, Y., Zang, Y., Li, H., Jin, C., Wang, J.: Unified reward model for multimodal understanding and generation. arXiv preprint arXiv:2503.05236 (2025)
45. Weng, L.: Reward hacking in reinforcement learning. lilianweng.github.io (Nov 2024), https://lilianweng.github.io/posts/2024-11-28-reward-hacking/, accessed: 2026-03-04
46. Wu, J., Gao, Y., Ye, Z., Li, M., Li, L., Guo, H., Liu, J., Xue, Z., Hou, X., Liu, W., et al.: Rewarddance: Reward scaling in visual generation. arXiv preprint arXiv:2509.08826 (2025)
47. Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., Li, H.: Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. arXiv preprint arXiv:2306.09341 (2023)
48. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: Learning and evaluating human preferences for text-to-image generation. Advances in Neural Information Processing Systems **36**, 15903–15935 (2023)

49. Xue, Z., Wu, J., Gao, Y., Kong, F., Zhu, L., Chen, M., Liu, Z., Liu, W., Guo, Q., Huang, W., et al.: Dancegrpo: Unleashing grpo on visual generation. arXiv preprint arXiv:2505.07818 (2025)
50. Yeh, P.H., Lee, K.H., Chen, J.C.: Training-free diffusion model alignment with sampling demons. arXiv preprint arXiv:2410.05760 (2024)
51. yifan123, G.U.: Discussion on flow-grpo issue 7. https://github.com/yifan123/flow_grpo/issues/7#issuecomment-2870678379 (2025), accessed: 2025-05-12
52. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6613–6623 (2024)
53. Yuan, H., Chen, Z., Ji, K., Gu, Q.: Self-play fine-tuning of diffusion models for text-to-image generation. Advances in Neural Information Processing Systems **37**, 73366–73398 (2024)
54. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. Advances in Neural Information Processing Systems **36**, 49842–49869 (2023)
55. Zheng, K., Chen, H., Ye, H., Wang, H., Zhang, Q., Jiang, K., Su, H., Ermon, S., Zhu, J., Liu, M.Y.: Diffusionnft: Online diffusion reinforcement with forward process. arXiv preprint arXiv:2509.16117 (2025)
56. Zheng, K., Lu, C., Chen, J., Zhu, J.: Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)