OP-GRPO: EFFICIENT OFF-POLICY GRPO FOR FLOW-MATCHING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Post training via GRPO has demonstrated remarkable effectiveness in improving the generation quality of flow-matching models. However, GRPO suffers from inherently low sample efficiency due to its on-policy training paradigm. To address this limitation, we present OP-GRPO, the first **O**ff-**P**olicy **GRPO** framework tailored for flow-matching models. First, we maintain a replay buffer that retains high-quality trajectories and reuses them by mixing a small fraction of off-policy samples into each rollout batch. Second, to mitigate the distribution shift introduced by off-policy samples, we propose a sequence-level importance sampling correction that keeps GRPO's clipping anchored to the current policy, reducing the clipped off-policy samples while ensuring stable policy updates. Third, we theoretically and empirically show that late denoising steps yield ill-conditioned off-policy ratios, and mitigate this by truncating trajectories at late steps. Across image and video generation benchmarks, OP-GRPO achieves comparable or superior performance to Flow-GRPO with only 34.2% of the training steps on average, yielding substantial gains in training efficiency while maintaining quality.

1 INTRODUCTION

Flow-matching models (Lipman et al., 2022; Liu et al., 2022; Yang et al., 2025) have become a standard backbone for modern visual generation (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020). By formulating generation as continuous transport, they improve both sampling efficiency and visual fidelity. They now underpin state-of-the-art systems across modalities, from image generators such as Stable Diffusion 3.5 (SD3.5) (Esser et al., 2024), Flux (Labs et al., 2025), and Qwen-Image (Wu et al., 2025) to video generators such as Wan (Wan et al., 2025), Open-Sora (Peng et al., 2025), and HunyuanVideo (Kong et al., 2025).

To align these models with human preferences, reinforcement learning (RL) has been widely adopted (Black et al., 2023; Xu et al., 2023; Wallace et al., 2024; Liu et al., 2025b; Xue et al., 2025). Among RL methods, Group Relative Policy Optimization (GRPO) (Liu et al., 2025b; Xue et al., 2025) is particularly effective: by optimizing group-normalized advantages over sampled trajectories, it improves aesthetic quality, semantic consistency, and text rendering (Kirstain et al., 2023; Ghosh et al., 2023; Wu et al., 2023).

However, these benefits come at a high computational cost. GRPO-based fine-tuning of large flow-matching models is highly inefficient (Li et al., 2025c): methods such as Flow-GRPO typically require thousands of GPU hours to converge, which limits their scalability. The inefficiency arises from two factors. First, GRPO is strictly on-policy: it samples fresh trajectories under the current policy and discards them after a single update. As a result, even high-quality trajectories cannot be reused, and trajectory sampling accounts for a large fraction of the total cost. Second, on difficult prompts the model rarely produces a successful trajectory, so the rewards within a group degenerate to all zeros; the group-normalized advantages then vanish and the gradient carries no signal, stalling policy improvement.

Both problems point to the same solution: relaxing the on-policy constraint. If high-quality trajectories were retained and reused rather than discarded, the sampling cost would be amortized over multiple updates, and the rare successful trajectories could continue to inform the policy. This is the central idea of off-policy learning, which has been studied in classical RL for its sample efficiency and stability (Watkins & Dayan, 1992; Mnih et al., 2013; Haarnoja et al., 2018).

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.

However, applying off-policy learning to flow-matching GRPO is not straightforward. Reusing trajectories from earlier policies introduces a distributional shift relative to the current rollout policy, which distorts GRPO's clipped objective and destabilizes training. Moreover, this off-policy mismatch is not uniform along the trajectory: as denoising approaches the low-noise regime, the reverse transition becomes nearly deterministic, so the per-step importance weights become ill-conditioned and the updates unstable. Naively transferring off-policy RL therefore fails at the low-noise steps, where flow-matching differs fundamentally from large language model (LLM).

To address these challenges, we propose OP-GRPO: the first **O**ff-**P**olicy **GRPO** framework tailored to flow-matching models. Our main contributions are as follows:

(1) OP-GRPO maintains a replay buffer that retains high-quality off-policy trajectories, and reuses them by mixing a small fraction of off-policy samples into each rollout batch. We further analyze why this hybrid rather than a fully off-policy design is essential for stable training.

(2) OP-GRPO adopts an off-policy objective that makes such reuse stable and unbiased. A sequence-level importance sampling correction keeps GRPO's clipping anchored to the current policy, reducing clipped samples from over 40% to 11.8%. A trajectory truncation then removes the off-policy samples' ill-conditioned late denoising steps.

(3) OP-GRPO is validated extensively on both image and video generation, reaching Flow-GRPO's final performance with only 34.2% of its training steps while matching or surpassing its generalization metrics and quality.

2 Related Work

2.1 GRPO in Flow-matching Models

Recent GRPO-based alignment for flow-matching models explores several directions. Flow-GRPO (Liu et al., 2025c) and DanceGRPO (Xue et al., 2025) establish the paradigm through an ODE-to-SDE formulation, and MixGRPO (Li et al., 2025a) generalizes it with mixed ODE-SDE schedules. Another line targets the stability of the policy update: GRPO-Guard (Wang et al., 2025a) and TempFlow-GRPO (He et al., 2026b) analyze timestep-wise shifts in the importance ratio and reweight it across denoising steps, while BranchGRPO (Li et al., 2025c), Fine-Grained GRPO (Zhou et al., 2025) and Neighbor GRPO (He et al., 2025) refine trajectory structure or credit assignment. Closer to our efficiency goal, Flash-GRPO (He et al., 2026a) accelerates video alignment through one-step policy optimization. Despite these advances, these methods remain on-policy: trajectories are discarded after each update, incurring high computational cost. Our work instead extends flow-matching GRPO to an off-policy setting through trajectory replay, improving sample efficiency and reducing training cost.

2.2 Off-policy-based GRPO in LLM

Recent research has focused on extending GRPO beyond its original on-policy formulation to leverage historical experience and improve sample efficiency in LLM post-training. RePO (Li et al., 2025b) augments GRPO with a replay buffer that retrieves diverse off-policy samples for advantage estimation, increasing effective optimization steps without increasing computation and empirically improving reasoning performance over vanilla GRPO on multiple benchmarks. ReMix (Liang et al., 2025) generalizes this idea by enabling GRPO to leverage much larger volumes of off-policy data through a mix-policy proximal optimization framework that incorporates off-policy trajectories under KL-convex constraints, enabling stable reuse of historical data while preserving policy improvement guarantees. BAPO (Wan et al., 2026) further proposes batch-level adaptation for selective reuse of high-value and difficult samples, providing theoretical insights into off-policy stability and empirical gains over standard GRPO. Liu et.al. (Liu et al., 2025a) investigates the off-policy effects under verl (Sheng et al., 2024) and Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023) settings, and further explores several rollout correction methods.

3 PRELIMINARIES

Flow-matching models. Flow-matching models (Lipman et al., 2022; Liu et al., 2022) learn a continuous transport from a base distribution $p_0 = \mathcal{N}(0, I)$ to the data distribution p_1. The transport is driven by a time-dependent velocity field $v_\theta(x, t)$ that defines an ODE,

$$\frac{dx_t}{dt} = v_\theta(x, t),\ t \in [0, 1] \tag{1}$$

At inference, samples are generated by integrating this ODE with an explicit Euler solver,

$$x_{k-1} = x_k + \Delta t v_\theta(x_k, t_k) \tag{2}$$

On-policy GRPO. The generation process forms a trajectory whose transitions follow the rollout policy, so the trajectory log-probability decomposes as $\log p_{\text{off}}(\mathbf{z}_{0:T} \mid c) = \sum_{t=1}^{T} \log p_{\text{off}}(\mathbf{z}_{t-1} \mid \mathbf{z}_t, c)$. GRPO (Schulman et al., 2017) estimates advantages from a group of G samples $\{x_0^i\}_{i=1}^G$ generated for a prompt c,

$$\hat{A}_t^i = \frac{R(x_0^i, c) - \text{mean}(\{R(x_0^j, c)_{j=1}^G\})}{\text{std}(\{R(x_0^j, c)\}_{j=1}^G)}. \tag{3}$$

where the terminal reward is shared across all denoising steps, i.e. $\hat{A}_t^i = \hat{A}^i$. The GRPO objective is

$$\mathcal{J}(\theta) = \mathbb{E}_{c\sim\mathcal{C},\{x_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|c)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{T}\sum_{t=1}^{T}\left(\min\left(r_t^i\hat{A}_t^i, \text{clip}\left(r_t^i(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t^i\right)\right), \tag{4}$$

with importance sampling ratio $r_t^i(\theta) = \frac{p_\theta(x_{t-1}^i|x_t^i,c)}{p_{\theta_{\text{old}}}(x_{t-1}^i|x_t^i,c)}$.

4 METHOD

In this section, we present OP-GRPO, an off-policy framework of GRPO designed to improve sample efficiency and accelerate training. We first introduce a high-quality replay buffer for storing informative off-policy trajectories with stepwise decay (Section 4.1). We then describe the hybrid rollout that injects off-policy samples into on-policy groups, and explain the necessity and rationality of the hybrid design (Section 4.2). Finally, we formulate the off-policy optimization objective (Section 4.3.2), consisting of a sequence-level importance sampling correction and a trajectory truncation for numerical stability. The overall framework of OP-GRPO is illustrated in Fig. 1 and refer to Algorithms 1 and 2 for the pseudocode.

4.1 HIGH-QUALITY REPLAY BUFFER

To preserve high-quality samples collected during the GRPO rollout phase, we maintain a replay buffer $\mathcal{B}_{\text{off}}$ to store these trajectories. During training iterations, a subset of stored samples is retrieved for reuse, thereby improving sample efficiency. Specifically, for each sampled group conditioned on c, we denote the set as $\{x_i\}_{i=1}^G$, where sample x is formally defined as a triplet,

$$x = \big(c, \mathbf{z}_{0:T}, \log p_{\text{off}}(\mathbf{z}_{0:T} \mid c), R\big), \tag{5}$$

where p_{off} denotes the distribution of diffusion policy, $\mathbf{z}_{0:T} = (\mathbf{z}_0, \mathbf{z}_1, \cdots, \mathbf{z}_T)$ denotes the complete latent trajectory generated by π_{off}, $\log p_{\text{off}}(\mathbf{z}_{0:T} \mid c)$ denotes the log-probability of the generated trajectory and R denotes the reward. We store the full latent trajectory together with its log-probability under p_{off}, rather than only the final sample, so that a replayed trajectory can be re-scored under the current policy and its distributional shift corrected at reuse time (Section 4.3.1).

To ensure that $\mathcal{B}_{\text{off}}$ continuously retains the most informative and recent high-quality trajectories, we adopt a reward-based replacement strategy. Specifically, within each group, we select the highest-reward trajectory as the most informative sample for retention. We compare the newly selected trajectory with the lowest-reward entry currently stored in the buffer, and replace it if the new trajectory achieves a higher reward.

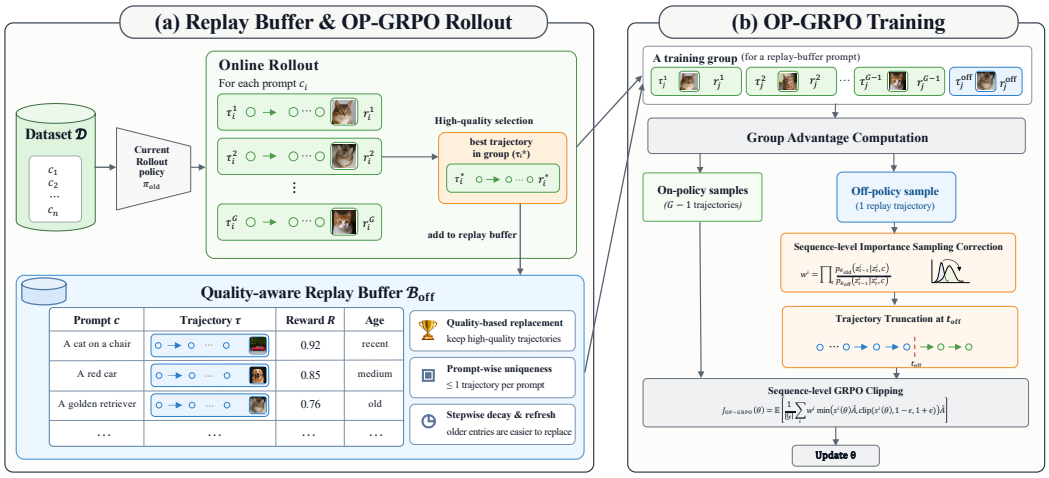


Figure 1: Framework of OP-GRPO: (a) OP-GRPO rollout and (b) OP-GRPO training. Blue regions represent samples from the replay buffer, green ones represent samples from the dataset.

Meanwhile, $\mathcal{B}_{\text{off}}$ enforces a uniqueness constraint to maintain the effectiveness and diversity of the stored trajectories. For each conditioning input c, at most one trajectory is retained. This design prevents the buffer from collapsing to a small set of trivial or repetitive samples and ensures that the stored data remains informative across the prompt distribution. Without this constraint, a few easy prompts that repeatedly yield high rewards would crowd out the buffer, so that off-policy replay would be dominated by a narrow set of conditions and lose coverage. When a newly selected trajectory corresponds to a conditioning input cthat already exists in the buffer, it is directly compared with the entry under the same condition, rather than with the lowest-reward entry.

Furthermore, to ensure that samples in the buffer are effectively refreshed and to prevent excessively off-policy samples from being repeatedly introduced into training, the buffer $\mathcal{B}_{\text{off}}$ is designed to favor recently collected trajectories. This keeps the stored samples close to the current policy and and mitigates the risk of the training process becoming overly off-policy. To this end, we apply a stepwise decay to the rewards of trajectories in the buffer, so that older samples naturally become easier to replace, allowing newer, higher-quality trajectories to be incorporated more readily. The decay is calibrated so that even the highest-reward entry becomes replaceable after roughly ten update steps, which empirically keeps the buffer close to the current policy without discarding useful samples prematurely (see Section C for the schedule per task).

4.2 Hybrid Off-policy Rollout

Given the replay buffer, a straightforward stategy for resuing its trajectories is to train purely on off-policy samples within the buffer. A fully off-policy design, however, is not a feasible alternative. First, training solely on a buffer of high-quality samples prevents the model from exploring new solutions, causing the policy to overfit to the stored trajectories and lose generalization. Second, a fully off-policy scheme requires a larger buffer, which amplifies the distributional gap between p_{off} and p_{old}, making the importance ratios and the correction term of Section 4.3.1 unstable and the resulting gradients untrustworthy.

We therefore resort to the hybrid RL paradigm (Song et al., 2022; Nakamoto et al., 2023) and mix only a fraction of off-policy samples into each group: on-policy rollouts maintain exploration and algorithmic stability, while off-policy replay makes better use of the rare, high-quality samples. This also mitigates the reward-degeneration problem raised in Section 1, since a high-reward buffer trajectory injected into an otherwise all-failing group restores intra-group contrast and recovers the vanishing gradient. Such a hybrid setting is also common practice in LLM off-policy RL, where purely off-policy PPO-style variants are rare (Yan et al., 2025; Wan et al., 2026).

Concretely, a majority of prompts are drawn from the original dataset and a minority are sampled from the buffer $\mathcal{B}_{\text{off}}$ in each batch. or a dataset prompt, G trajectories are generated on-policy per prompt. For a buffer prompt, only $(G-1)$ new trajectories are generated, and the remaining slot is filled by the stored buffer trajectory. Formally, for a given input c the group is

$$\mathcal{G}(c) = \left\{\mathbf{z}_i^{\text{on}} \sim p_{\text{old}}(\cdot|c)\right\}_{i=1}^{G-1} \cup \left\{\mathbf{z}^{\text{off}} \sim \mathcal{B}_{\text{off}}\right\}, \tag{6}$$

where G is the group size. The resulting trajectories are thoroughly shuffled before training to avoid biases induced by a fixed sample ordering. The off-policy proportion serves as a single hyperparameter that governs the trade-off between convergence speed and training stability.

4.3 Off-policy Optimization

Building on the standard GRPO objective, OP-GRPO additionally handles the off-policy trajectories, which cannot be optimized in the same way as on-policy ones. Once a fraction of each group comes from the buffer, the objective can no longer be applied as is: the reused trajectories no longer follow the current rollout policy p_{old}, and their late, low-noise denoising steps are numerically fragile. We therefore introduce two complementary components, a sequence-level importance sampling correction that realigns the off-policy samples while keeping GRPO's clipping intact (Section 4.3.1), and a trajectory truncation that removes the ill-conditioned steps (Section 4.3.2), which we detail next.

4.3.1 Sequence-level Importance Sampling Correction

Directly applying Eq. (4) to a group containing an off-policy sample is invalid, since that sample is drawn from an earlier policy p_{off} rather than the current rollout policy p_{old}. The resulting distributional mismatch biases the gradient and can destabilize training. A directly remedy is to absorb the shift between p_{off} and p_{old} into the per-step importance sampling ratio in Eq. (4), but this distorts the clipping mechanism: clipping is meant to bound the update of p_θ relative to its starting point p_{old}, yet once the shift is absorbed into the ratio, the p_{old} terms cancel, as shown in Eq. (7):

$$r_t^i(\theta) = \frac{p_{\theta_{\text{old}}}(z_{t-1}^i|z_t^i,c)}{p_{\theta_{\text{off}}}(z_{t-1}^i|z_t^i,c)} \times \frac{p_\theta(z_{t-1}^i|z_t^i,c)}{p_{\theta_{\text{old}}}(z_{t-1}^i|z_t^i,c)} = \frac{p_\theta(z_{t-1}^i|z_t^i,c)}{p_{\theta_{\text{off}}}(z_{t-1}^i|z_t^i,c)}. \tag{7}$$

As a consequence, clipping now measures the discrepancy between p_θ and p_{off} rather than p_{old}, so the ratios become extreme and over **40%** of off-policy samples are clipped, undermining the benefit of reuse. The issue is that a single per-step ratio cannot both correct the shift and bound the update, which motivates separating the two roles and lifting both to the sequence level.

Since a trajectory represents a single sample with one trajectory-level advantage $\hat{A}^i$, we bound its update at the same granularity: we use the length-normalized geometric mean of the per-step ratios between p_θ and p_{old}, following the group sequence policy optimization of (Zheng et al., 2025),

$$s^i(\theta) = \left(\prod_t \frac{p_\theta(z_{t-1}^i \mid z_t^i, c)}{p_{\theta_{\text{old}}}(z_{t-1}^i \mid z_t^i, c)}\right)^{1/L}, \tag{8}$$

where L is the number of steps in the trajectory. The length normalization makes s^i invariant to the trajectory length, so a single clipping threshold ϵ applies consistently across trajectories regardless of how many denoising steps they contain.

To correct the distributional shift, we further multiply each trajectory by the importance sampling weight $w^i = \frac{P_{\pi_{\text{old}}}(\tau^i)}{P_{\pi_{\text{off}}}(\tau^i)} = \prod_t \frac{p_{\theta_{\text{old}}}(z_{t-1}^i|z_t^i,c)}{p_{\theta_{\text{off}}}(z_{t-1}^i|z_t^i,c)}$, which maps p_{off} back to p_{old}. The resulting objective is,

$$\mathcal{J}_{\text{OP-GRPO}}(\theta) = \mathbb{E}_{c\sim\mathcal{C},\,\{x_i\}_{i=1}^{\mathcal{G}}} \frac{1}{|\mathcal{G}|} \sum_i w^i \cdot \min\Big(s^i(\theta)\,\hat{A}^i,\ \text{clip}\big(s^i(\theta), 1-\epsilon, 1+\epsilon\big)\,\hat{A}^i\Big). \tag{9}$$

The two factors s^i and w^i play distinct roles and are normalized accordingly. The clipping ratio s^i bounds how far p_θ moves from p_{old}, for which the geometric mean gives a stable, length-invariant scale. The weight w^i keeps the raw likelihood ratio, which makes the off-policy objective an unbiased estimator (proof in Section B.1). Crucially, p_{old} no longer cancels, so clipping still bounds the update against the correct reference, and $w^i = 1$ recovers a sequence-level GRPO objective for on-policy samples. Empirically, OP-GRPO clips only 11.8% of off-policy samples, far below the 40% of the naive substitution, while keeping training stable.

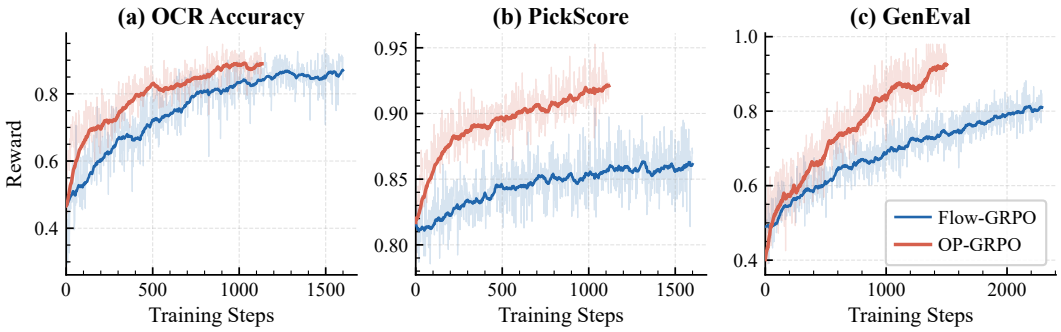


Figure 3: **Training Curves of OP-GRPO and Flow GRPO.**

4.3.2 Trajectory Truncation for Numerical Stability

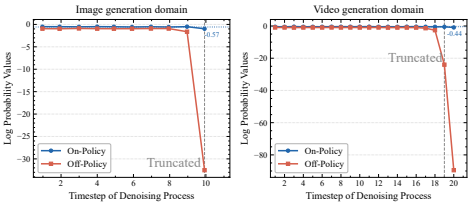


Figure 2: Log-probability of on-policy and off-policy samples across denoising steps.

Reusing off-policy latents introduces a numerical difficulty that is intrinsic to off-policy reuse itself and concentrated at the late, low-noise denoising steps. As shown in Fig. 2, the per-step log-probability of an off-policy sample stays stable across most steps but drops sharply in the final few. Here the reverse transition is nearly deterministic ($\sigma_t \to 0$), so the per-step log-probability is highly sensitive to any policy mismatch. Since an off-policy sample comes from a stale policy, its deviation is amplified into ill-conditioned log-probabilities that destabilize the update. This sensitivity can be made precise from the Gaussian form of the reverse transition. Its log-probability contains a term $|z_{t-1} - \mu_\theta(z_t, t)|^2/(2\sigma_t^2)$, whose numerator reflects the policy mismatch $|\Delta\mu|^2$. As $\sigma_t^2 \to 0$ at the late steps, this term diverges (full derivation in Section B.3).

The ill-conditioning is confined to the off-policy latents at the low-noise steps. Since these steps are nearly deterministic and contribute little to the perceptual content, we discard exactly this segment rather than reuse it. Concretely, we truncate the off-policy trajectory at a step t_{off} and regenerate the remaining low-noise steps with the current policy,

$$\tau_m = \left\{ z_i^{\text{off}} \right\}_{i=T}^{t_{\text{off}}} \cup \left\{ z_j \right\}_{j=t_{\text{off}}}^{0}, \tag{10}$$

where τ_m denotes the mixed-policy trajectory, z^{off} denotes the reused off-policy latents, and z_j denotes the latents re-generated by the current policy, as in Eq. (2). Both s^i and w^i are then aggregated only over the reused segment $t \in [t_{\text{off}}, T]$, so the ill-conditioned low-noise steps no longer enter the objective and training is stabilized.

5 Experiments

5.1 Experimental Setup

We evaluate OP-GRPO across two generation models and three tasks.

Models. We use **SD3.5-M** (Esser et al., 2024) for text-to-image and **Wan2.1-1.4B** (Wan et al., 2025) for text-to-video generation, covering both image and video domains.

Tasks. Following Flow-GRPO, we consider three tasks: (1) **Compositional Image Generation**, evaluated with GenEval (Ghosh et al., 2023); (2) **Visual Text Generation** (Chen et al., 2023), evaluated via a rule-based recognition pipeline; and (3) **Human Preference Alignment**, using PickScore (Kirstain et al., 2023) as the reward.

Table 1: Performance on **task metrics** (Compositional Image Generation, Visual Text Rendering, Human Preference) and **unseen metrics**. ImgRwd: ImageReward; UniRwd: UnifiedReward. Best score in blue.

Model	Step	Task Metric	Image Quality		Preference Score		
			Aesthetic	DeQA	ImgRwd	PickScore	UniRwd
SD3.5-M	—	0.63 / 0.59 / 21.72	5.31	4.07	0.92	22.18	3.29
Compositional Image Generation							
Flow-GRPO	1020	0.68	5.27	4.05	1.37	22.24	3.35
OP-GRPO	1020	0.88	5.22	4.02	1.21	22.35	3.49
Flow-GRPO	Best	0.95	5.18	4.00	1.12	22.37	3.51
OP-GRPO	Best	0.96	5.17	3.98	1.13	22.42	3.57
Visual Text Rendering							
Flow-GRPO	720	0.70	5.28	3.92	0.93	22.31	3.30
OP-GRPO	720	0.81	5.30	3.89	0.95	22.39	3.38
Flow-GRPO	Best	0.92	5.32	3.83	0.98	22.54	3.46
OP-GRPO	Best	0.93	5.32	3.85	0.98	22.56	3.45
Human Preference Alignment							
Flow-GRPO	720	22.19	5.44	4.13	1.09	22.19	3.43
OP-GRPO	720	23.27	5.92	4.17	1.22	23.27	3.59
Flow-GRPO	Best	23.32	6.01	4.18	1.26	23.32	3.66
OP-GRPO	Best	23.64	6.09	4.23	1.31	23.44	3.72

Generalization Evaluation. To assess generalization beyond the training objectives, we report several unseen metrics covering two dimensions: image quality, via Aesthetic Predictor (Schuhmann, 2022) and DeQA (You et al., 2025), and human preference, via ImageReward (Xu et al., 2023), PickScore (Kirstain et al., 2023), and UnifiedReward (Wang et al., 2025b).

5.2 Training Efficiency of OP-GRPO

OP-GRPO converges substantially faster than Flow-GRPO while reaching higher final performance, as shown by the training curves on the three tasks in Fig. 3. Trained for 1,150 versus 1,600 steps, it reaches Flow-GRPO's final performance with only **34.2%** of the steps on average, while attaining **1.17%** higher final performance. This confirms the efficiency gain from off-policy learning. The speed-up is most pronounced in the early phase of training, where reusing high-reward trajectories restores the intra-group contrast that on-policy rollouts often lack, yielding stronger learning signals per step.

5.3 OP-GRPO on Image Generation

Quantitative Results. We evaluate OP-GRPO on the SD3.5-M backbone against Flow-GRPO on both task specific metrics (GenEval, OCR accuracy and PickScore) and cross-task metrics, at an intermediate step and the best step (Table 1. See Section C for details. At the intermediate step, OP-GRPO outperforms Flow-GRPO across all three tasks on both metric types, improving efficiency without sacrificing quality. At the best step, it matches or slightly surpasses Flow-GRPO.

Visualization. We compare intermediate-step generations of the two methods in Fig. 4. OP-GRPO gives more precise object count and attribute control in compositional generation, more legible text in visual text rendering, and richer detail in human preference alignment, consistent with Table 1.

Generalization. OP-GRPO does not overfit the task-specific reward, as reflected by its cross-task metrics in Table 1. We further validate it on the more comprehensive T2I-CompBench++ (Huang et al., 2023; 2025), where OP-GRPO retains clear advantages over Flow-GRPO (Table 2).

Scalability to larger backbones. To test whether these gains persist on a much larger model, we additionally apply OP-GRPO to FLUX.1-dev (Labs, 2024) on the Human Preference Alignment task, using the same recipe as SD3.5-M without any model-specific tuning. OP-GRPO again converges faster and reaches a higher training reward than Flow-GRPO, indicating that its off-policy replay

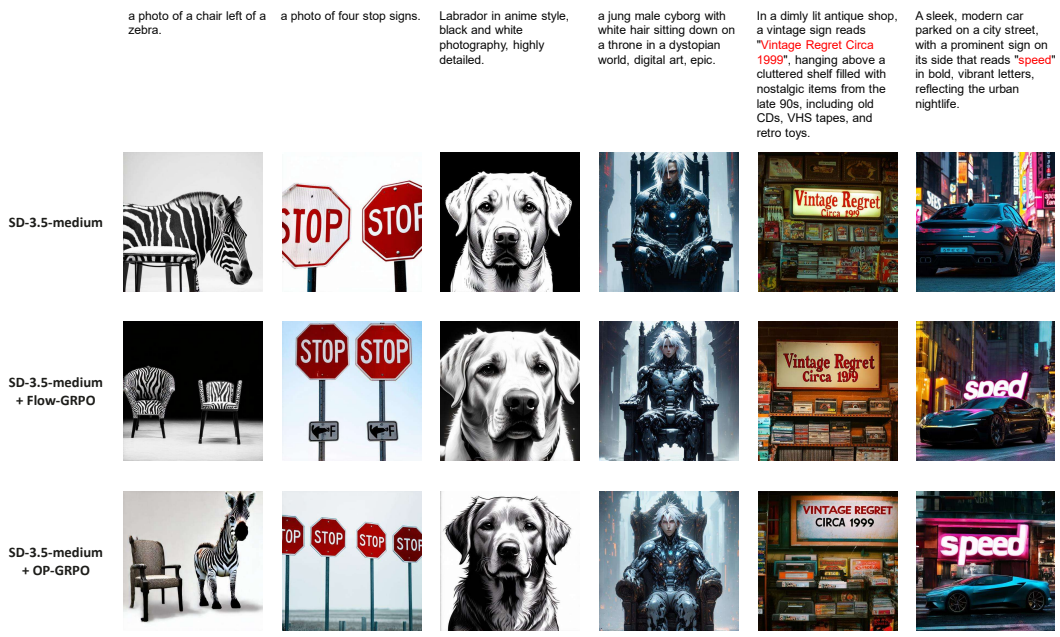


Figure 4: Visual Results of OP-GRPO and Flow GRPO using SD3.5-M.

Table 2: **T2I-CompBench++ Result.** We report results using the Best model trained on the GenEval. 2D, 3D, and Non-Sp. denote the 2D-Spatial, 3D-Spatial, and Non-Spatial dimensions.

Model	Color	Shape	2D-Sp	3D-Sp	Numeracy	Non-Sp	Average
FLUX.1 Dev	0.738	0.574	0.289	0.391	0.621	0.315	0.488
SD3.5-M	0.796	0.569	0.283	0.376	0.595	0.312	0.489
Intermediate checkpoints							
Flow-GRPO	0.769	0.578	0.412	0.405	0.644	0.312	0.520
OP-GRPO	0.824	0.603	0.498	0.437	0.660	0.318	0.567
Best checkpoints							
Flow-GRPO	0.838	0.613	0.545	0.447	0.675	0.320	0.573
OP-GRPO	0.845	0.614	0.550	0.461	0.682	0.319	0.579

and trajectory truncation remain effective as the backbone scales up. We report the full setup and training curves in Section D.1.

5.4 OP-GRPO on Video Generation

We further evaluate OP-GRPO on the more challenging video generation setting, using the Visual Text Generation task on Wan2.1-1.4B (Fig. 5 for qualitative comparisons, Fig. 7 for convergence). OP-GRPO reaches Flow-GRPO's final performance with only 30.1% of the training steps, and its final generation quality substantially surpasses Flow-GRPO. This gap stems from the purely on-policy nature of Flow-GRPO: on difficult tasks, the rare high-quality samples it discovers are immediately discarded, yielding slow and unstable improvement. OP-GRPO instead retains and repeatedly reuses these samples, enabling more effective exploitation of high-quality trajectories.

5.5 Ablation Study

We ablate the two components that render OP-GRPO's off-policy reuse effective: the sequence-level correction with its truncation, and the off-policy proportion. All ablations use the SD3.5-M backbone on the GenEval task, following the setup of Fig. 3.

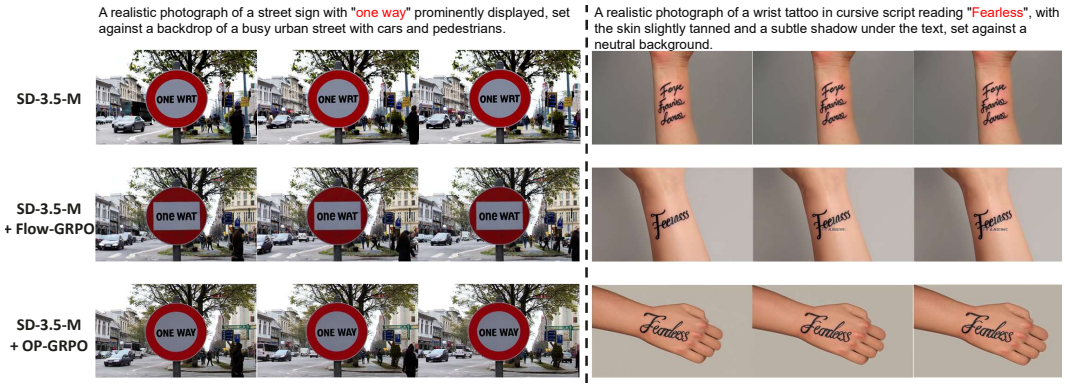


Figure 5: Visual Results of OP-GRPO and Flow GRPO on OCR task on video generation model Wan2.1-1.4B.

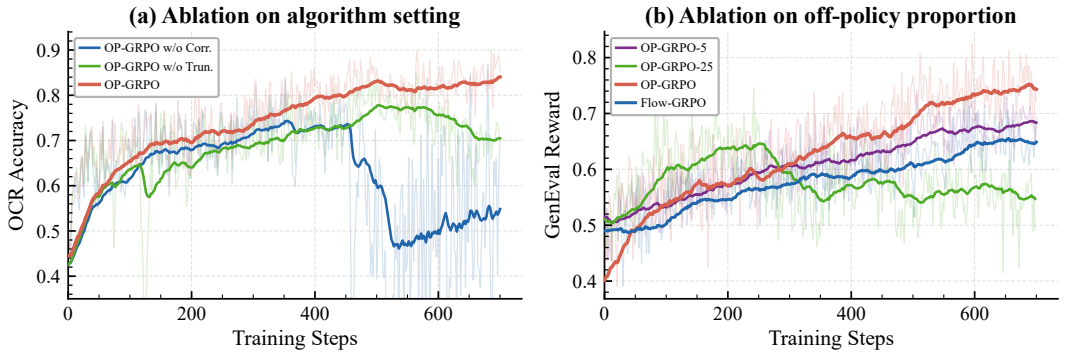


Figure 6: Ablation study of OP-GRPO.

Sequence correction and truncation. Removing the correction (*OP-GRPO w/o corr*) leaves the mismatch between p_{off} and p_{old} uncorrected, so its bias accumulates and training diverges. Removing the truncation (*OP-GRPO w/o trun*) retains the ill-conditioned low-noise steps (Section 4.3.2), where the vanishing variance ($\sigma_t^2 \to 0$) amplifies the per-step log-ratios and destabilizes the importance weights. Only with both does OP-GRPO stay stable while preserving the speed-up (Fig. 6(a)). The correction also keeps clipping bounded against p_{old} rather than the stale p_{off}, reducing the clipped fraction of off-policy samples from over **40%** to **11.8%** (Section 4.3.1).

Off-policy proportion. We vary the off-policy proportion in Fig. 6(b), with OP-GRPO-5 and OP-GRPO-25 denoting 5% and 25%. Off-policy samples accelerate convergence at every proportion, but the value governs a trade-off: too low restores the intra-group contrast too rarely and the gain is limited (23.1% for OP-GRPO-5), while too high makes each batch stale, dispersing the weights w^i and destabilizing updates. We adopt 25% as the best trade-off throughout.

6 Conclusion

In this work, we investigate the efficiency and stability bottlenecks of GRPO-based RL training for flow-matching models. We identify that the on-policy training paradigm and degenerate reward signals hinder learning efficiency, while naive off-policy reuse introduces distributional shift and training instability. To address these issues, we propose OP-GRPO, an off-policy GRPO framework that integrates replay-based sampling, sequence-level distribution correction, and truncated denoising. Experiments on image and video generation benchmarks show that OP-GRPO accelerates convergence while preserving generation quality competitive with fully on-policy baselines, reaching Flow-GRPO's performance with only 34.2% of the training steps on average. One limitation is that our evaluation focuses on image and video generation; extending OP-GRPO to other domains, such as audio or 3D generation, remains an open direction for future work.

AI USE STATEMENT

We used generative AI tools (LLMs) only for tasks in the recommended-disclosure category: editing the manuscript to improve readability and phrasing, drafting and refining portions of the text, and creating and editing scientific figures (including plotting code). We did not use generative AI tools for any task in the required-disclosure category, including generating synthetic data, developing theoretical models, formulating or proving mathematical claims, proposing hypotheses, designing methodology or experiments, implementing methods, cleaning datasets, or interpreting results; all of these were carried out by the authors. We have reviewed all AI-assisted work: all AI-edited text was checked by the authors for accuracy, and all figures and plotting code were verified against the underlying experimental results. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work studies reinforcement learning for aligning flow-matching generative models and does not involve human subjects, personally identifiable information, or the release of new datasets. All models and datasets used in our experiments are publicly available and are used in accordance with their intended use and licenses. We do not foresee ethical concerns arising from our method beyond those generally associated with visual generative models. We adhere to the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

We have made substantial efforts to ensure the reproducibility of our results. The full method is described in Sections 4.1 to 4.3, and the complete training and rollout procedures are provided as pseudocode in Algorithms 1 and 2. All theoretical claims, including the unbiasedness of the off-policy correction and the ill-conditioning analysis, are stated with their assumptions and complete proofs in Sections B.1 to B.4. The experimental setup, including backbones, tasks, evaluation metrics, and datasets, is detailed in Section 5.1, and the key hyperparameters for each task are summarized in Section C. All datasets and pretrained models used in our experiments are publicly available. To further facilitate reproduction, we provide a runnable subset of our source code as **supplementary material**, covering the FLUX.1-dev PickScore experiment (the OP-GRPO training entry point together with the replay buffer, sequence-level correction, and trajectory truncation, and the dependencies required to run it). We will release the complete code upon acceptance.

REFERENCES

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

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36:9353–9387, 2023.

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 *Forty-first international conference on machine learning*, 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.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Dailan He, Guanlin Feng, Xingtong Ge, Yazhe Niu, Yi Zhang, Bingqi Ma, Guanglu Song, Yu Liu, and Hongsheng Li. Neighbor grpo: Contrastive ode policy optimization aligns flow models. *arXiv preprint arXiv:2511.16955*, 2025.

Xiaoxuan He, Siming Fu, Zeyue Xue, Weijie Wang, Ruizhe He, Yuming Li, Dacheng Yin, Shuai Dong, Haoyang Huang, Hongfa Wang, Nan Duan, and Bohan Zhuang. Flash-grpo: Efficient alignment for video diffusion via one-step policy optimization, 2026a. URL https://arxiv.org/abs/2605.15980.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 132869–132895, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d75f561eaaf2cb754bc8d7e36d8af362-Paper-Conference.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025.

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.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025. URL https://arxiv.org/abs/2412.03603.

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

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025a.

Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. Repo: Replay-enhanced policy optimization. *arXiv preprint arXiv:2506.09340*, 2025b.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. *arXiv preprint arXiv:2509.06040*, 2025c.

Jing Liang, Hongyao Tang, Yi Ma, Jinyi Liu, Yan Zheng, Shuyue Hu, Lei Bai, and Jianye Hao. Squeeze the soaked sponge: Efficient off-policy reinforcement finetuning for large language model. *arXiv preprint arXiv:2507.06892*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying RL collapse from the training-inference mismatch. https://richardli.xyz/rl-collapse, September 2025a.

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*, 2025b.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025c.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36:62244–62269, 2023.

Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, Yuhui Wang, Anbang Ye, Gang Ren, Qianran Ma, Wanying Liang, Xiang Lian, Xiwen Wu, Yuting Zhong, Zhuangyan Li, Chaoyu Gong, Guojun Lei, Leijun Cheng, Limin Zhang, Minghao Li, Ruijie Zhang, Silan Hu, Shijie Huang, Xiaokang Wang, Yuanheng Zhao, Yuqi Wang, Ziang Wei, and Yang You. Open-sora 2.0: Training a commercial-level video generation model in $200k, 2025. URL https://arxiv.org/abs/2503.09642.

Christoph Schuhmann. Laion aesthetics, August 2022.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

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*, pp. 8228–8238, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning. *arXiv preprint arXiv:2602.20722*, 2026.

Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint arXiv:2510.22319*, 2025a.

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

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL https://arxiv.org/abs/2508.02324.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

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.

Jianhao Yan, Yafu Li, Zican Chen, Chao Fang, Fandong Yin, Zhenghao Wang, Chao Wang, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

Xiaofeng Yang, Cheng Chen, xulei yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 13896–13920, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2460396f2d0d421885997dd1612ac56b-Paper-Conference.pdf.

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.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-grained grpo for precise preference alignment in flow models. *arXiv preprint arXiv:2510.01982*, 2025.

A PSEUDOCODE

The rollout procedure of OP-GRPO is illustrated in Algorithm 1.

Algorithm 1 OP-GRPO Rollout

Require: Prompt dataset $\mathcal{B}$, replay buffer $\mathcal{B}_{\text{off}}$, batch size M, off-policy ratio α, group size G, truncation step t_{off}
Ensure: Generated batch samples

for rollout iteration **do**
  Sample αM off-policy samples from $\mathcal{B}_{\text{off}}$
  for each sampled off-policy sample **do**
    Let $x = (c, \mathbf{z}_{0:T}, \log p_{\text{off}}(\mathbf{z}_{0:T}|c), R)$
  end for
  Sample $(1-\alpha)M$ prompts from $\mathcal{B}$
  Repeat each prompt G times
  Randomly assign off-policy samples into batch positions
  Shuffle all prompts
  for each prompt c in batch **do**
    if c is not associated with off-policy sample **then**
      Rollout trajectory $\mathbf{z}_{0:T}$ following current policy
    else
      $\mathbf{z}_{0:t_{\text{off}}} \leftarrow \mathbf{z}^{\text{off}}_{T:t_{\text{off}}}$
      Sample $\mathbf{z}_{t_{\text{off}}-1:0}$ using current policy
      $\mathbf{z}_{0:T} \leftarrow \text{concatenate}(\mathbf{z}_{T:t_{\text{off}}}, \mathbf{z}_{t_{\text{off}}-1:0})$
    end if
  end for
  Evaluate reward $\mathbf{R}$ for all generated samples
  for each prompt **do**
    Form group $\mathcal{G}$ containing its G samples
    Select $x^* = \arg\max_{x \in \mathcal{G}} R(x)$
    if x^* exists in $\mathcal{B}_{\text{off}}$ **then**
      Compare $R(x^*)$ with the matched buffer sample
      Replace buffer sample if $R(x^*)$ is larger
    else
      Let $x_{\min}$ be the sample with minimum reward in $\mathcal{B}_{\text{off}}$
      Insert x^* if $R(x^*) > R(x_{\min})$
    end if
  end for
end for

The training procedure of OP-GRPO is illustrated in Algorithm 2.

B THEORETICAL ANALYSIS

B.1 UNBIASEDNESS OF THE OFF-POLICY CORRECTION

We show that the importance sampling weight w^i in Eq. (9) makes the off-policy objective an unbiased estimator of the corresponding on-policy objective, for an *arbitrary* per-trajectory surrogate. This isolates the role of w^i: it removes the distributional shift introduced by reusing samples from p_{off}, independently of how the surrogate itself is defined (e.g. the sequence-level clipped objective we adopt, or any other choice).

Let τ denote a trajectory generated for a conditioning input c, let $P_{\pi_{\text{old}}}(\tau)$ and $P_{\pi_{\text{off}}}(\tau)$ be its likelihoods under the current rollout policy p_{old} and the off-policy p_{off}, and let $w(\tau) = P_{\pi_{\text{old}}}(\tau)/P_{\pi_{\text{off}}}(\tau)$. Let $F(\tau, \theta)$ be any function of the trajectory and the parameters (the per-trajectory surrogate; in our case $F(\tau,\theta) = \min(s(\theta)\hat{A},\ \text{clip}(s(\theta), 1-\epsilon, 1+\epsilon)\hat{A})$).

Algorithm 2 OP-GRPO Training

Require: Batch $\mathcal{D} = \{(\mathbf{z}_{0:T}^{(i)}, A^{(i)}, \log p_{\text{off}}^{(i)})\}$, current policy π_θ, rollout policy p_{old}, reference policy π_{ref}, training steps K, clipping range ϵ, KL weight β
Ensure: Updated parameters $\bar{\theta}$

for $k = 1$ to K **do**
  Clamp advantages: $\tilde{A}^{(i)} = \text{clamp}(A^{(i)})$
  for each sample i in $\mathcal{D}$ **do**
    Sequence ratio (geometric mean over retained steps): $s^{(i)} = \big(\prod_{t=t_{\text{off}}}^{T} p_\theta / p_{\text{old}}\big)^{1/L}$
    if sample i is off-policy **then**
      $w^{(i)} = \exp\big(\sum_{t=t_{\text{off}}}^{T} \big[\log p_{\text{old}}(z_{t-1}^{(i)} | z_t^{(i)}, c^{(i)}) - \log p_{\text{off}}(z_{t-1}^{(i)} | z_t^{(i)}, c^{(i)})\big]\big)$
    else
      $w^{(i)} = 1$
    end if
  end for
  Policy loss: $\mathcal{L}_{\text{policy}} = -\frac{1}{|\mathcal{D}|} \sum_i w^{(i)} \min\big(s^{(i)} \tilde{A}^{(i)},\ \text{clip}(s^{(i)}, 1-\epsilon, 1+\epsilon)\, \tilde{A}^{(i)}\big)$
  KL loss: $\mathcal{L}_{\text{KL}} = \mathbb{E}_{\mathbf{z} \sim \pi_\theta}\big[\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})\big]$
  Total loss: $\mathcal{L} = \mathcal{L}_{\text{policy}} + \beta \mathcal{L}_{\text{KL}}$
  $\bar{\theta} \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for
return $\pi_{\bar{\theta}}$

Proposition 1. *Assume* $\text{supp}(P_{\pi_{old}}(\cdot)) \subseteq \text{supp}(P_{\pi_{off}}(\cdot))$*, so that* $w(\tau)$ *is well defined wherever* p_{old} *has mass. Then for every* θ*,*

$$\mathbb{E}_{\tau \sim P_{\pi_{off}}}\big[\, w(\tau)\, F(\tau, \theta)\,\big] \;=\; \mathbb{E}_{\tau \sim P_{\pi_{old}}}\big[\, F(\tau, \theta)\,\big]. \tag{11}$$

Consequently, evaluating the surrogate on off-policy trajectories reweighted by w *yields, in expectation, exactly the on-policy objective; averaging over a group as in Eq.* (9) *preserves this identity by linearity.*

Proof. The identity is the trajectory-level form of importance sampling. Writing the off-policy expectation as an integral over trajectories and substituting $w(\tau) = P_{\pi_{\text{old}}}(\tau)/P_{\pi_{\text{off}}}(\tau)$,

$$\begin{aligned} \mathbb{E}_{\tau \sim P_{\pi_{\text{off}}}}\big[w(\tau) F(\tau, \theta)\big] &= \int P_{\pi_{\text{off}}}(\tau)\, \frac{P_{\pi_{\text{old}}}(\tau)}{P_{\pi_{\text{off}}}(\tau)}\, F(\tau, \theta)\, d\tau \\ &= \int P_{\pi_{\text{old}}}(\tau)\, F(\tau, \theta)\, d\tau = \mathbb{E}_{\tau \sim P_{\pi_{\text{old}}}}\big[F(\tau, \theta)\big]. \end{aligned} \tag{12}$$

The support assumption $\text{supp}(P_{\pi_{\text{old}}}) \subseteq \text{supp}(P_{\pi_{\text{off}}})$ ensures the integrand is well defined wherever $P_{\pi_{\text{old}}}(\tau) > 0$: on the set where $P_{\pi_{\text{off}}}(\tau) = 0$ we also have $P_{\pi_{\text{old}}}(\tau) = 0$, so the factor $P_{\pi_{\text{off}}}(\tau)$ cancels the ratio and no probability mass is lost. The identity therefore holds for the single-trajectory surrogate. Since the group objective in Eq. (9) averages F over the trajectories in a group, taking expectations term by term and applying linearity extends the identity to the full group average, which establishes the claim. □

Proposition 1 clarifies what w^i does and does not guarantee. It guarantees that off-policy reuse introduces *no additional bias* relative to the on-policy objective: the reweighted off-policy estimate targets the same quantity as if the trajectories had been drawn on-policy. It does *not* claim that the surrogate F is itself an unbiased estimator of the true policy gradient; as in PPO, GRPO, and GSPO, the clipping inside F trades exactness for stability, and this bias is shared by the on-policy objective rather than caused by off-policy reuse. This is precisely why we keep w^i as the raw likelihood ratio in Eq. (9): any geometric-mean normalization of w^i would break Eq. (11) and reintroduce distributional bias, whereas the geometric mean is applied only to the clipping ratio s^i, which plays no role in Proposition 1.

B.2 Reverse Process and Per-step Log-Probability

We follow the SDE formulation of Flow-GRPO (Liu et al., 2025c), which stochasticizes the deterministic flow ODE so that the reverse process becomes a sequence of Gaussian transitions amenable to policy optimization. Let t index the (discretized) denoising steps and σ_t denote the corresponding noise level. Each reverse transition takes the form

$$z_{t-1} = \mu_\theta(z_t, t) + \sigma_t \sqrt{\Delta t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{13}$$

where $\mu_\theta(z_t, t)$ is the drift determined by the learned velocity field v_θ and Δt is the step size. Following Flow-GRPO, the noise level is scheduled as

$$\sigma_t = \eta \sqrt{\tfrac{t}{1-t}}, \tag{14}$$

with a hyperparameter η controlling the injected stochasticity. Crucially, $\sigma_t \to 0$ as $t \to 0$, i.e. the transitions become nearly deterministic in the low-noise regime near the data manifold. The transition is Gaussian, $p_\theta(z_{t-1} \mid z_t, c) = \mathcal{N}\big(z_{t-1}; \mu_\theta(z_t, t), \sigma_t^2\big)$, with log-density

$$\log p_\theta(z_{t-1} \mid z_t, c) = -\frac{\|z_{t-1} - \mu_\theta(z_t, t)\|^2}{2\sigma_t^2 \Delta t} - \log\big(\sigma_t \sqrt{\Delta t}\big) - \tfrac{1}{2}\log(2\pi). \tag{15}$$

B.3 Ill-conditioning of Off-policy Samples at Low Noise

We now derive why the per-step log-probability of an off-policy sample becomes ill-conditioned as $\sigma_t \to 0$. Consider a single reused step. Since the sample $z_{t-1} = \mu_{\theta_{\text{off}}}(z_t, t) + \sigma_t \sqrt{\Delta t}\, \epsilon$ was generated by the off-policy p_{off}, its per-step log-probability under the current policy differs from that under p_{off} by

$$\begin{aligned} \log \frac{p_{\theta_{\text{old}}}(z_{t-1} \mid z_t, c)}{p_{\theta_{\text{off}}}(z_{t-1} \mid z_t, c)} &= -\frac{\|\Delta\mu_t + \sigma_t \sqrt{\Delta t}\, \epsilon\|^2}{2\sigma_t^2 \Delta t} + \frac{\|\sigma_t \sqrt{\Delta t}\, \epsilon\|^2}{2\sigma_t^2 \Delta t} \\ &= -\frac{\|\Delta\mu_t\|^2}{2\sigma_t^2 \Delta t} - \frac{\Delta\mu_t \cdot \epsilon}{\sigma_t \sqrt{\Delta t}}, \end{aligned} \tag{16}$$

where $\Delta\mu_t = \mu_{\theta_{\text{old}}}(z_t, t) - \mu_{\theta_{\text{off}}}(z_t, t)$ is the drift gap between the current rollout policy and the off-policy that generated the sample. Taking expectation over $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and using $\mathbb{E}[\epsilon] = 0$,

$$\mathbb{E}_\epsilon\left[\log \frac{p_{\theta_{\text{old}}}}{p_{\theta_{\text{off}}}}\right] = -\frac{\|\Delta\mu_t\|^2}{2\sigma_t^2 \Delta t}, \qquad \mathrm{Var}_\epsilon\left[\log \frac{p_{\theta_{\text{old}}}}{p_{\theta_{\text{off}}}}\right] = \frac{\|\Delta\mu_t\|^2}{\sigma_t^2 \Delta t}. \tag{17}$$

Both the mean and the variance of this per-step quantity scale as $\|\Delta\mu_t\|^2/\sigma_t^2$, where the drift gap $\|\Delta\mu_t\|$ arises because the sample was produced by a stale off-policy p_{off}. As $\sigma_t \to 0$ at the late, low-noise steps, this quantity diverges, so the per-step log-probabilities of off-policy samples become ill-conditioned exactly at those steps. The same $1/\sigma_t^2$ divergence has been observed for the single-step ratio in on-policy flow-matching GRPO (Wang et al., 2025a; He et al., 2026b); here it is driven by the off-policy drift gap $\|\Delta\mu_t\|$, which is larger than in the on-policy case because p_{off} is stale relative to the current policy.

These ill-conditioned per-step terms are the log-probabilities that the objective sums over the trajectory, $\log \frac{P_{\pi_{\text{old}}}(\tau)}{P_{\pi_{\text{off}}}(\tau)} = \sum_{t=1}^{T} \log \frac{p_{\theta_{\text{old}}}(z_{t-1}|z_t,c)}{p_{\theta_{\text{off}}}(z_{t-1}|z_t,c)}$, so the divergent low-noise terms dominate the sum and destabilize the update. This is the theoretical counterpart of the cliff-like drop in Fig. 2.

B.4 Truncation Removes the Divergent Terms

The truncation strategy of Section 4.3.2 reuses the off-policy latents only for $t \in [t_{\text{off}}, T]$ and regenerates the remaining low-noise steps with the current policy. Since only the reused segment contains off-policy latents, the sequence-level correction is evaluated only over $t \in [t_{\text{off}}, T]$,

$$\log \frac{P_{\pi_{\text{old}}}(\tau_m)}{P_{\pi_{\text{off}}}(\tau_m)} = \sum_{t=t_{\text{off}}}^{T} \log \frac{p_{\theta_{\text{old}}}(z_{t-1} \mid z_t, c)}{p_{\theta_{\text{off}}}(z_{t-1} \mid z_t, c)}, \tag{18}$$

Table 3: Hyperparameters for different tasks.

Task	Replay Buffer Size	KL Weight	Off-policy Hybrid Ratio	Off-policy sample per Group	Decreasing Value of rewards
Text Generation	32	0.01	0.125	1	0.05
Human Preference	64	0.001	0.125	1	0.01
GenEval	32	0.001	0.125	1	0.05

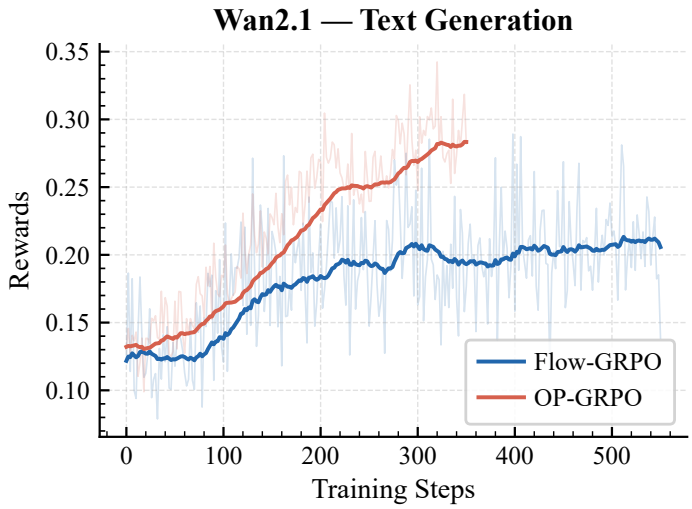


Figure 7: **Training Curves of OP-GRPO and Flow GRPO on the video generation domain.**

so the divergent $t < t_{\text{off}}$ terms with $\sigma_t \to 0$ never enter the objective, leaving only well-conditioned per-step terms bounded by the largest retained $1/\sigma_{t_{\text{off}}}^2$. This motivates the data-driven criterion in Section 4.3.2: we set t_{off} at the onset of the divergent tail, identified as the first step whose per-step log-ratio magnitude exceeds 400% of its stable value. Truncating earlier corrupts the semantic content of the off-policy latents, whereas truncating exactly this ill-conditioned tail removes the unstable terms at negligible cost to quality, as confirmed in Fig. 3 and Section 5.5. Since the truncated steps are nearly deterministic and contribute little perceptual content, regenerating them with the current policy incurs negligible quality loss while restoring numerical stability.

C Algorithm Implementation on Image Generation

We summarize the key hyperparameters of OP-GRPO across three tasks on the image generation model SD3.5-M in Table 3. For the Human Preference Alignment and Visual Text Rendering tasks, OP-GRPO is trained for 1,050 steps, with performance evaluated on the test set every 50 steps. For the more challenging Compositional Image Generation task, OP-GRPO requires 1,500 training steps to converge, compared to over 2,200 steps needed by the original Flow-GRPO algorithm. Regarding the replay buffer update mechanism, we maintain a shadow copy of the reward for each sample in the buffer, which decays by a fixed decrement at every epoch to facilitate timely buffer refresh. The decay schedule is designed such that the highest-reward sample in the buffer is depreciated and eligible for replacement after approximately 10 steps. Concretely, for the Visual Text Rendering and Compositional Image Generation tasks, whose reward ranges are approximately [0.5,1.0], the per-epoch decay is set to 0.05. For the Human Preference Alignment task, whose reward range is narrower at approximately [0.8,1.0], the per-epoch decay is set to 0.01 accordingly. All three experiments are conducted on 32 NVIDIA A100 GPUs.

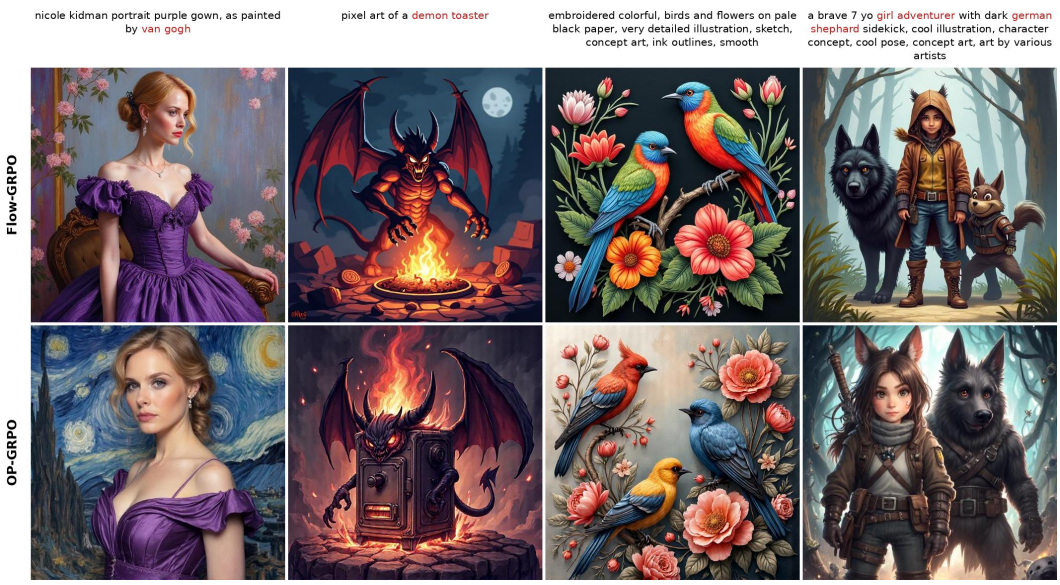


Figure 8: **Qualitative comparison of OP-GRPO and Flow-GRPO on FLUX.1-dev** *Top row*: Flow-GRPO; *bottom row*: OP-GRPO. Each column is annotated with its prompt above the images. OP-GRPO follows the prompts more faithfully and yields sharper, better-composed images.

D More Results on Video Generation

D.1 OP-GRPO on FLUX

To verify that OP-GRPO scales beyond the SD3.5-M backbone used in the main experiments, we further apply it to FLUX.1-dev (Labs, 2024), a substantially larger flow-matching image generation model, on the Human Preference Alignment task with PickScore (Kirstain et al., 2023) as the reward.

Implementation details. We fine-tune FLUX.1-dev with LoRA (rank 64, $\alpha = 128$) at 512×512 resolution. Following the main-text setup, each prompt is sampled with a group size of 8 over 6 denoising steps during training (28 steps for test-set evaluation), with a classifier-free guidance scale of 3.5, a learning rate of 3×10^{-4}, and an EMA on the trainable parameters. OP-GRPO uses a high-quality replay buffer of capacity 32 with the shadow-reward decay mechanism ($\lambda = 0.02$ per epoch, matching the narrow $[0.8, 1.0]$ PickScore reward range), a trajectory truncation of one step, and an off-policy proportion of 25%. The KL coefficient is set to $\beta = 0.01$ and the SDE noise level to 0.9. Both OP-GRPO and the Flow-GRPO baseline share this identical recipe, differing only in the OP-GRPO off-policy machinery.

Training curve. Fig. 9 shows the PickScore training reward of OP-GRPO and Flow-GRPO on FLUX.1-dev. Consistent with the SD3.5-M results reported in the main text, OP-GRPO attains a higher training reward and converges faster throughout training, reaching a smoothed reward of about 0.87 versus roughly 0.82 for Flow-GRPO at the same number of training steps. This confirms that the off-policy replay and trajectory-truncation mechanisms of OP-GRPO remain effective and stable on a much larger backbone, without any task- or model-specific tuning.

Qualitative comparison. We compare the generations of OP-GRPO and Flow-GRPO on FLUX.1-dev at the same training step (720 step) under identical prompts, as shown in Fig. 8. OP-GRPO produces images that adhere more faithfully to the prompt and exhibit finer detail and more coherent composition: it renders the Van Gogh style, the "demon toaster" concept, and the multi-character adventurer-and-companion scene that Flow-GRPO tends to miss or only partially realize. This qualitative gap is consistent with the higher training reward in Fig. 9 and indicates that the faster convergence of OP-GRPO translates into visibly better generations on the larger backbone.

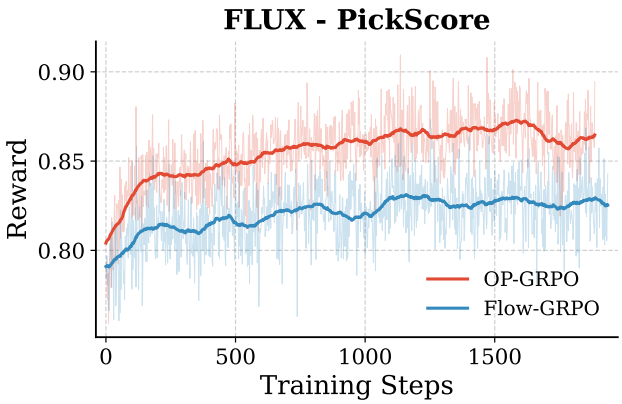


Figure 9: **PickScore training reward of OP-GRPO and Flow-GRPO on FLUX.1-dev.** Light curves denote the raw per-step training reward and dark curves the smoothed trend. OP-GRPO consistently attains a higher reward and converges faster on this larger backbone.

D.2 More Visual Results on Video Generation

Training Curve. We evaluate OP-GRPO in the domain of video generation, utilizing the Wan2.1-1.4B model on the text rendering task. OP-GRPO achieves a score of approximately 0.3 within 450 training steps, whereas Flow-GRPO attains only 0.22 at 550 steps, which further demonstrates the efficiency and superiority of OP-GRPO. The training curve is illustrated in Fig. 7.