TRANSITION-GRPO: IMPROVING FLOW MODEL ALIGNMENT WITH DIRECT TRANSITION TARGETS

Anonymous authors
Paper under double-blind review

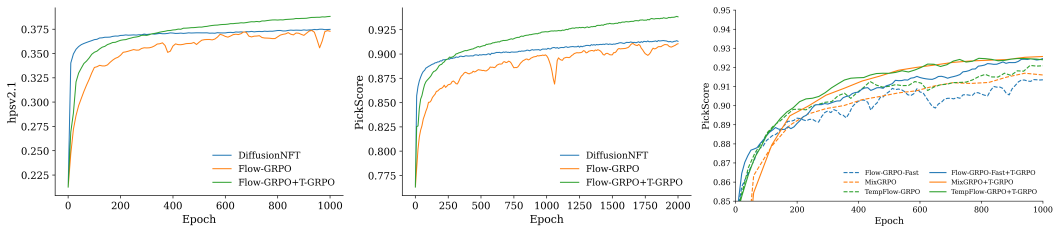

Figure 1: *Left and middle*: Replacing Flow-GRPO with both the loss and initialization of our proposed T-GRPO designs substantially improves reward optimization. On the model-based rewards HPSv2 and PickScore, the enhanced backward-process formulation quickly closes and reverses the gap to the forward-process DiffusionNFT. *Right*: Integrating T-GRPO's components into CPS, MixGRPO, and Flow-GRPO-Fast consistently improves performance, demonstrating that our designs are complementary to existing acceleration strategies.

ABSTRACT

Reinforcement learning has emerged as an effective approach for aligning diffusion and flow-based generative models with downstream preferences. Backward-process methods such as Flow-GRPO optimize stochastic reverse-SDE transitions, but often exhibit slower reward improvement than forward-process alternatives, raising the question of how these transitions can be optimized more effectively. We revisit the Flow-GRPO objective and show that its local update reveals a simple policy-improvement direction in transition-mean space. Building on this observation, we propose **Transition-GRPO (T-GRPO)**, which uses a KL-proximal formulation to derive explicit, advantage-adjusted transition-mean targets and replaces likelihood-ratio optimization with direct regression toward these targets. We further extend this formulation to jointly perform CFG distillation and reward optimization during a short initialization stage, using CFG-generated trajectories to construct advantage-adjusted transition targets for a non-CFG model. Experiments demonstrate that T-GRPO consistently accelerates convergence and achieves higher rewards under a fixed training budget than Flow-GRPO. It also surpasses the forward-process DiffusionNFT baseline on preference-based rewards, while DiffusionNFT remains stronger on structured tasks. Moreover, T-GRPO yields additional gains when integrated with existing acceleration methods.

1 INTRODUCTION

Diffusion and flow-matching-based generative models have emerged as leading paradigms for high-fidelity text-to-image generation. Although these models are typically pretrained with denoising (Ho et al., 2020), score-matching (Song et al., 2020), or flow-matching (Lipman et al., 2022; Liu et al., 2022) objectives, many practical goals, including prompt adherence, aesthetic quality, visual text rendering, compositional reasoning, and safety, are not directly captured by their pretraining losses. This mismatch has motivated growing interest in reinforcement learning (RL) and reward-based post-training, where pretrained generators are further optimized using preference models or task-specific rewards.

A common approach is to view the reverse generation trajectory as a multi-step Markov decision process. Each intermediate latent is treated as a state, the sampled next latent as an action, and

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.

the decoded image receives a terminal reward. Early methods such as DDPO (Black et al., 2023) and DPOK (Fan et al., 2023) applied policy-gradient algorithms to diffusion models under black-box rewards. More recent methods such as Flow-GRPO (Liu et al., 2026) extend this formulation to flow models by converting the deterministic probability-flow ODE into a stochastic reverse SDE. Each denoising step then defines a Gaussian transition policy that can be optimized with group-relative advantages and likelihood ratios (Shao et al., 2024).

Despite its generality, Flow-GRPO often improves reward slowly. Recent methods accelerate it by reducing stochastic timesteps (Li et al., 2025a), reallocating credit across denoising steps (He et al., 2025), improving reverse-SDE sampling (Wang & Yu, 2025), or reusing trajectories more effectively (Li et al., 2025b; Ding & Ye, 2025; Zhang et al., 2026). These approaches modify the rollout pipeline while largely retaining the same ratio-based objective. In contrast, forward-process methods such as AWM (Xue et al., 2025a) and DiffusionNFT (Zheng et al., 2025) add noise to reward-selected generations and optimize a flow-matching objective on the resulting forward-process pairs, achieving much faster reward improvement. This contrast raises a natural question: does the observed gap reflect an inherent limitation of backward-process learning, or does it arise partly from how sampled reverse transitions are optimized?

To better understand how Flow-GRPO learns from sampled reverse-SDE transitions, we revisit its underlying optimization objective. We find that its local update has a simple interpretation: transitions with positive advantages encourage the model to move toward the sampled next latent, while those with negative advantages encourage it to move away. This observation motivates us to directly express this improvement signal as an explicit transition-mean target, providing an alternative to likelihood-ratio optimization.

Motivated by this observation, we propose **Transition-GRPO (T-GRPO)**, which turns the local policy-improvement direction into an explicit learning target. Specifically, we formulate a KL-proximal update around the rollout policy and derive a closed-form transition-mean target by shifting the rollout mean along the advantage-weighted transition residual. We then train the flow model by directly regressing its predicted transition means toward these targets, replacing likelihood-ratio optimization with a variance-normalized regression objective.

The same transition-target formulation also enables a short CFG-guided (Classifier-Free Guidance (Ho & Salimans, 2022)) warm start: a frozen guided model supplies informative transition targets for the non-CFG model before reward optimization begins. The guided model is then removed, and all subsequent rollouts and updates are performed without CFG.

We evaluate T-GRPO on the model-based preference rewards PickScore (Kirstain et al., 2023) and HPSv2.1 (Wu et al., 2023) and the structured objectives GenEval (Ghosh et al., 2023) and OCR (Cui et al., 2025). As summarized in Figure 1, transition-mean regression provides the main improvement over Flow-GRPO, while the CFG-guided warm start further improves its starting point and final results. Together, they consistently increase fixed-budget performance across all four tasks. T-GRPO also catches up with and eventually surpasses the forward-process DiffusionNFT reference on PickScore and HPSv2.1, whereas DiffusionNFT remains stronger on GenEval and OCR. This task-dependent pattern suggests that stochastic backward exploration is particularly useful when the reward can distinguish nearby trajectory variations. Finally, T-GRPO provides additional gains when integrated into representative Flow-GRPO acceleration methods, showing that its objective-level improvement is complementary to advances in the rollout pipeline.

We summarize our contributions as follows:

- We revisit the objective of Flow-GRPO and reveal a simple interpretation of its local policy-improvement direction in transition-mean space. Based on this observation, we derive explicit advantage-adjusted transition targets through a KL-proximal formulation.
- We propose **T-GRPO**, which replaces likelihood-ratio optimization with regression toward advantage-adjusted transition-mean targets. The same formulation naturally supports CFG-guided initialization of a non-CFG model.
- Experiments show that T-GRPO improves convergence and fixed-budget performance over Flow-GRPO and complements existing acceleration methods.

2 Related Work

Reward optimization for diffusion and flow models. Early online methods directly apply policy gradients to the denoising process. DDPO (Black et al., 2023) represents each reverse diffusion step as an action and estimates the gradient of the terminal reward through the transition log-probabilities, providing both score-function and importance-sampled variants. DPOK (Fan et al., 2023) adopts a similar denoising-policy formulation but regularizes the optimized policy toward the pretrained diffusion model through a KL penalty, thereby balancing reward improvement against preservation of the original generator. Flow-GRPO Liu et al. (2026) extends this approach to flow-matching models by converting their deterministic sampling ODE into a marginal-preserving reverse SDE. The resulting Gaussian transitions are optimized with group-normalized terminal rewards, while a denoising-reduction strategy limits the number of transitions used for training. DanceGRPO (Xue et al., 2025b) develops a unified implementation of group-relative policy optimization for diffusion and rectified-flow models and demonstrates the same trajectory-level formulation across image and video generation tasks. Other approaches optimize diffusion models through differentiable reward gradients (Xu et al., 2023; Clark et al., 2024; Prabhudesai et al., 2024) or preference-based objectives (Wallace et al., 2024; Yang et al., 2024a;b; Sun et al., 2026). Several recent methods instead express reward optimization through flow-matching or denoising losses. DiffusionNFT (Zheng et al., 2025) ranks online generations using their rewards, forms positive and negative training signals, and applies forward noising to the generated samples before updating the model with a negative-aware flow-matching objective. Advantage Weighted Matching (Xue et al., 2025a) derives a lower-variance alternative to denoising policy gradients by applying trajectory advantages directly as weights on the standard score- or flow-matching loss. T-GRPO differs from these approaches in that its supervision is constructed from the individual perturbations actually sampled along the reverse-SDE trajectory and applies a loss with an explicit, signed transition-mean target.

Efficient and stable trajectory-based GRPO. A number of extensions improve Flow-GRPO (Liu et al., 2026) by changing where stochastic transitions are introduced. Flow-GRPO-Fast (Liu et al., 2026) first follows a shared deterministic ODE trajectory, branches into a group at a sampled intermediate state, and optimizes only one or two stochastic transitions before returning to deterministic sampling, substantially reducing both group sampling and backpropagation costs. MixGRPO (Li et al., 2025a) generalizes this idea with a sliding temporal window: SDE sampling and GRPO updates are restricted to the window, while the remaining steps use the original ODE; its Flash variant further uses higher-order solvers outside the optimized region. Rather than reducing the stochastic region, Coefficients-Preserving Sampling (Wang & Yu, 2025) revisits the stochastic sampler itself. It attributes degraded rollout quality to excessive noise introduced by conventional reverse-SDE discretization and constructs a DDIM-inspired stochastic update that preserves the deterministic solver coefficients while retaining exploration. OP-GRPO (Zhang et al., 2026) targets the on-policy data bottleneck by storing selected high-quality trajectories in a replay buffer, correcting their distribution shift with sequence-level importance sampling, and excluding late transitions whose off-policy ratios become poorly conditioned.

KL-regularized and regression-based policy improvement. A long line of work stabilizes policy improvement by constraining each update around a behavior or reference policy. REPS (Peters et al., 2010) derives a reward-weighted non-parametric policy under a relative-entropy constraint, while TRPO (Schulman et al., 2015) enforces an explicit trust region and PPO (Schulman et al., 2017) replaces it with a clipped surrogate for practical optimization. MPO (Abdolmaleki et al., 2018b) and relative-entropy-regularized (Abdolmaleki et al., 2018a) policy iteration alternate between KL-constrained nonparametric policy improvement and supervised projection onto a parametric policy, whereas MDPO (Tomar et al., 2020) interprets related updates through mirror descent. A complementary family implements policy improvement directly through regression: AWR (Peng et al., 2019) and AWAC (Nair et al., 2020) fit dataset actions with advantage-dependent weights, CRR (Wang et al., 2020) selects or reweights actions according to critic estimates, and IQL (Kostrikov et al., 2021) extracts a policy through advantage-weighted behavioral cloning. Unlike these methods, which select or reweight sampled actions, T-GRPO uses a signed group-relative advantage to construct a closed-form mean target for each Gaussian reverse transition, moving its mean toward or away from the sampled transition residual.

3 PRELIMINARIES

We review flow-based generation and Flow-GRPO (Liu et al., 2026), focusing on the transition-space representation used by our method. We index sampling steps by $k \in 0, \ldots, N-1$ in generation order, where $k = 0$ denotes the first, high-noise step. Let

$$\tau_0 > \tau_1 > \cdots > \tau_N, \qquad \Delta\tau_k := \tau_{k+1} - \tau_k < 0 \tag{1}$$

denote the corresponding continuous noise levels.

3.1 TRANSITION POLICIES OF FLOW-GRPO

Given a text condition c, a flow model learns a time-dependent velocity field $v_\theta(x_\tau, \tau, c)$, with standard generation following the probability-flow ODE

$$\frac{\mathrm{d}x_\tau}{\mathrm{d}\tau} = v_\theta(x_\tau, \tau, c). \tag{2}$$

The resulting transition is deterministic and therefore does not directly define the stochastic policy required by GRPO (Shao et al., 2024). Flow-GRPO instead performs training rollouts with a marginal-preserving reverse SDE derived from the same velocity field.

Under its discretization, each reverse step can be written as

$$m_{\theta,k} = \alpha_k x_k + \gamma_k v_\theta(x_k, \tau_k, c), \qquad x_{k+1} = m_{\theta,k} + s_k \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, I), \tag{3}$$

where α_k, γ_k, and s_k are determined by the reverse-SDE sampler. Hence each step defines the Gaussian transition policy

$$p_\theta(x_{k+1} \mid x_k, c) = \mathcal{N}\left(x_{k+1}; m_{\theta,k}, s_k^2 I\right). \tag{4}$$

The reverse-SDE construction and its discretization are detailed in Appendix A.1.

This Gaussian representation provides both a tractable transition likelihood and local stochastic exploration around the model-predicted reverse step.

3.2 OPTIMIZATION OF FLOW-GRPO

Flow-GRPO optimizes a terminal image reward by treating the reverse-SDE trajectory as a multi-step policy. For each prompt, a frozen rollout policy $\bar{\theta}$ generates a group of G trajectories. If $R^{(i)}$ is the reward of trajectory i, its group-relative advantage is

$$A^{(i)} = \frac{R^{(i)} - \text{mean}\left(\{R^{(k)}\}_{k=1}^{G}\right)}{\text{std}\left(\{R^{(k)}\}_{k=1}^{G}\right)} \tag{5}$$

Because only the final image is scored, all optimized transitions in the same trajectory share the trajectory-level advantage $A^{(i)}$.

A transition collected from the frozen rollout policy satisfies

$$x_{k+1}^{(i)} = m_{\bar{\theta},k}^{(i)} + s_k \epsilon_k^{(i)}, \qquad \epsilon_k^{(i)} \sim \mathcal{N}(0, I). \tag{6}$$

Flow-GRPO compares its probability under the current and rollout policies:

$$r_k^{(i)}(\theta) = \frac{p_\theta\left(x_{k+1}^{(i)} \mid x_k^{(i)}, c\right)}{p_{\bar{\theta}}\left(x_{k+1}^{(i)} \mid x_k^{(i)}, c\right)}. \tag{7}$$

It then maximizes the PPO-style (Schulman et al., 2017) surrogate

$$\mathcal{J}_{\text{Flow-GRPO}}(\theta) = \mathbb{E}\left[\sum_{i,k} \min\left(r_k^{(i)}(\theta) A^{(i)}, \text{clip}\left(r_k^{(i)}(\theta), 1-\epsilon, 1+\epsilon\right) A^{(i)}\right)\right]. \tag{8}$$

4 TRANSITION-GRPO

We propose **Transition-GRPO (T-GRPO)**, which replaces the likelihood-ratio objective of Flow-GRPO (Liu et al., 2026) in Eq. 8 with direct regression in transition-mean space. The key idea is to extract the local policy-improvement direction encoded by each sampled reverse-SDE transition and convert it into an explicit KL-proximal mean target. We then instantiate this target as a transition-regression objective for reward optimization and extend the same construction to a CFG-guided (Ho & Salimans, 2022) initialization stage.

4.1 TRANSITION INFORMATION IN THE FLOW-GRPO LOSS

We first examine what information a sampled reverse-SDE transition contributes to the Flow-GRPO update. Consider a transition collected from the rollout policy:

$$x_{k+1} = m_{\bar{\theta},k} + s_k \epsilon_k. \tag{9}$$

Define the change in transition mean

$$\delta_k(\theta) := m_{\theta,k} - m_{\bar{\theta},k}. \tag{10}$$

Because the current and rollout policies have the same covariance, their log-likelihood ratio is

$$\log r_k(\theta) = \frac{\langle x_{k+1} - m_{\bar{\theta},k}, \delta_k(\theta) \rangle}{s_k^2} - \frac{|\delta_k(\theta)|_2^2}{2s_k^2} \qquad = \frac{\langle \epsilon_k, \delta_k(\theta) \rangle}{s_k} - \frac{|\delta_k(\theta)|_2^2}{2s_k^2}.$$

Details are provided in Appendix A.2.

At the rollout policy, where $\delta_k = 0$ and $r_k = 1$, the gradient of the unclipped Flow-GRPO term with respect to the transition mean is

$$\begin{aligned} \nabla_{m_{\theta,k}} \left[A^{(i)} r_k(\theta) \right] \Big|_{\theta=\bar{\theta}} &= A^{(i)} \frac{x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)}}{s_k^2}, \\ F_k^{-1} \nabla_{m_{\theta,k}} \left[A^{(i)} r_k(\theta) \right] \Big|_{\theta=\bar{\theta}} &= A^{(i)} \left(x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} \right), \qquad F_k = s_k^{-2} I. \end{aligned} \tag{11}$$

Here, F_k is the Fisher matrix of the Gaussian transition policy with respect to its mean. The first line shows that the sampled transition residual determines the local Flow-GRPO update direction, while the factor s_k^{-2} arises from the Gaussian policy geometry. After Fisher preconditioning, the corresponding mean-space displacement is simply the advantage-weighted transition residual. This observation motivates T-GRPO to construct an explicit transition-mean target from this displacement.

4.2 KL-PROXIMAL TRANSITION TARGETS

The previous subsection identifies

$$A^{(i)} \left(x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} \right) \tag{12}$$

as the mean-space displacement induced by the local Flow-GRPO update. However, this direction alone does not specify a finite update magnitude. Indeed, directly maximizing the corresponding linearized improvement is unbounded. We therefore regularize the update by requiring the new transition policy to remain close to the frozen rollout policy in KL divergence.

Let δ_k denote a candidate displacement of the transition mean from $m_{\bar{\theta},k}^{(i)}$. Using the local Flow-GRPO gradient in Eq. 11, we consider the KL-proximal mean-improvement problem

$$\max_{\delta_k} \quad A^{(i)} \left\langle \frac{x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)}}{s_k^2}, \delta_k \right\rangle - \frac{1}{\eta_k} D_{\mathrm{KL}} \left(\mathcal{N}\left(m_{\bar{\theta},k}^{(i)} + \delta_k, s_k^2 I \right) \middle\| \mathcal{N}\left(m_{\bar{\theta},k}^{(i)}, s_k^2 I \right) \right). \tag{13}$$

The first term encourages improvement in the local Flow-GRPO direction, whereas the KL term prevents the updated transition policy from moving arbitrarily far from the rollout policy. The coefficient $\eta_k > 0$ controls the strength of this proximal constraint.

This objective is strictly concave and has the closed-form solution,

$$\delta_k^\star = \eta_k A^{(i)} \left(x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} \right). \tag{14}$$

We provide the details in Appendix A.3. Therefore, the resulting transition-mean target is

$$\boxed{m_k^{\star,(i)} = m_{\bar{\theta},k}^{(i)} + \eta_k A^{(i)} \left(x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} \right)}. \tag{15}$$

4.3 Reward-Driven Transition Regression

For each sampled transition, the terminal rewards within its rollout group are converted into the trajectory-level advantages $A^{(i)}$ defined in Eq. 5. These advantages are used to construct the transition-mean target $m_k^{\star,(i)}$ in Eq. 15, with a fixed target multiplier $\eta = 10.0$ for all reverse steps.

We train the flow model by regressing its predicted transition mean $m_{\theta,k}^{(i)}$ toward this detached target:

$$\mathcal{L}_{\mathrm{T-GRPO}}(\theta) = \mathbb{E}\left[\sum_{i,k} \frac{\left\| m_{\theta,k}^{(i)} - \mathrm{sg}\left[m_k^{\star,(i)} \right] \right\|_2^2}{2 s_k^2} \right], \tag{16}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient and the factor s_k^{-2} normalizes the squared error by the variance of the corresponding Gaussian transition.

We provide a comparison between Flow-GRPO and our T-GRPO objectives in Appendix A.4.

4.4 CFG-Guided Transition Initialization

Following DiffusionNFT (Zheng et al., 2025) and the official Flow-GRPO implementation, we perform both the main reward optimization and final inference without classifier-free guidance (CFG). This non-CFG setting can substantially improves training efficiency and eliminates the additional unconditional prediction at inference time, but starts from a weaker policy: early non-CFG generations often have much lower quality and provide less informative reward differences.

To improve this initial regime, a natural idea is to use a CFG model to generate higher-quality proposal trajectories from which the non-CFG student can learn. This is difficult to realize within the original Flow-GRPO objective, since guided transitions may lie far from the non-CFG behavior distribution, making their likelihood ratios poorly conditioned. Our transition loss removes this restriction by operating directly in transition space: a guided transition can be treated as a proposal relative to the non-CFG student mean, without requiring it to be sampled from the student policy.

For each prompt, a frozen CFG teacher generates a group of proposal trajectories $\{x_{\bar{\theta},k}^{w,(i)}\}$ with CFG scale w. Let $m_{\theta,k}^{1.0,(i)}$ denote the CFG $= 1.0$ student mean evaluated at the corresponding state. After scoring the final teacher samples and computing the same group-relative advantages $A^{(i)}$, we construct

$$\boxed{m_{k,\mathrm{init}}^{\star,(i)} = m_{\theta,k}^{1.0,(i)} + \eta_k A^{(i)} \left(x_{\bar{\theta},k+1}^{w,(i)} - m_{\theta,k}^{1.0,(i)} \right)}. \tag{17}$$

Thus, the guided trajectory supplies the proposal direction, while the reward determines how that direction is transferred to the non-CFG student. We optimize the same transition-regression objective as Eq. 16, replacing $(m_k^{\star,(i)}, s_k)$ with $(m_{k,\mathrm{init}}^{\star,(i)}, \widehat{s}_k^{(i)})$, where $\widehat{s}_k^{(i)}$ is a proposal-dependent scale estimated from the guided–unguided transition discrepancy. The CFG proposal construction, including its rollout noise, and the scale estimator are detailed in Appendix A.5.

We generate CFG-guided proposals using the same reverse SDE transition formulation, but the teacher trajectories serve only as auxiliary proposals for initialization, their rollout noise scale s_k^w is therefore free to vary. We set $s_k^w = 0$ by default in our experiments. After initialization, the CFG teacher is discarded and the initialized student becomes the non-CFG rollout policy. All subsequent trajectory collection, reward optimization, and inference are performed without CFG.

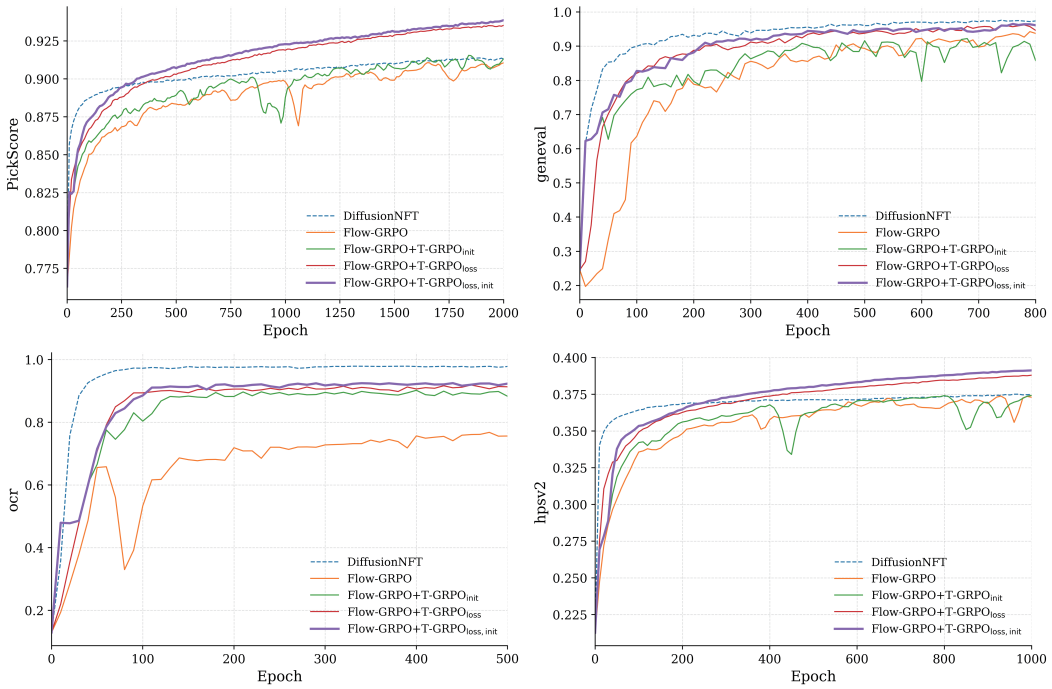

Figure 2: **Optimization trajectories of T-GRPO and its components.** We compare Flow-GRPO with variants using transition-mean regression (T-LOSS), CFG-guided initialization (T-INIT), or both. Results are reported on PickScore, HPSv2.1, GenEval, and OCR for 2,000, 1,000, 800, and 500 epochs, respectively. T-Loss consistently strengthens reward optimization under unchanged reverse-SDE rollouts, while the full T-GRPO achieves the strongest backward-process performance across all four tasks. DiffusionNFT is included as a forward-process reference.

5 EXPERIMENTS

We evaluate T-GRPO to answer three questions: (1) whether transition regression improves Flow-GRPO under controlled rollouts; (2) whether CFG-guided initialization alleviates non-CFG cold start; and (3) whether the proposed components complement existing GRPO accelerations.

5.1 EXPERIMENTAL SETUP

Evaluation protocol. We organize the experiments into a controlled component study and a compatibility study. In the controlled study, *Flow-GRPO* serves as the common backward-process backbone. We compare the original method with *Flow-GRPO+T-Loss*, which replaces only the likelihood-ratio objective with transition-mean regression; *Flow-GRPO+T-Init*, which applies CFG-guided initialization before continuing with the original Flow-GRPO objective; and the full *T-GRPO*, which combines both components. These configurations share the same base model, prompts, rollout process, trajectory budget, optimizer, and evaluation protocol, thereby isolating the effects of the loss and initialization.

We additionally report *DiffusionNFT* (Zheng et al., 2025) as a representative forward-process reference. DiffusionNFT is rerun using its official implementation and task configurations, while Flow-GRPO and its variants are implemented in a common DiffusionNFT-based codebase. This comparison provides context for the forward–backward optimization gap; the primary controlled comparison remains among methods sharing the same Flow-GRPO rollout pipeline.

For the compatibility study, we incorporate the complete T-GRPO formulation into CPS-GRPO (Wang & Yu, 2025), MixGRPO (Li et al., 2025a), and Fast-GRPO (Liu et al., 2026). For each backbone, we retain its original sampler, rollout schedule, and timestep-selection strategy.

Table 1: **Endpoint performance of T-GRPO and its components.** Flow-GRPO variants share the same backward rollout pipeline and differ only in the training loss and initialization. Values in parentheses denote the relative score change from Flow-GRPO. The best and second-best results are shown in bold and underlined, respectively.

Method	PickScore ↑ 2,000 ep.	HPSv2.1 ↑ 1,000 ep.	GenEval ↑ 800 ep.	OCR ↑ 500 ep.
Flow-GRPO	23.675 (–)	0.3737 (–)	0.943 (–)	0.7679 (–)
Flow-GRPO + T-INIT	23.820 (+0.61%)	0.3742 (+0.13%)	0.922 (-2.23%)	0.9021 (+17.48%)
Flow-GRPO + T-LOSS	<u>24.318</u> (<u>+2.72%</u>)	<u>0.3882</u> (<u>+3.88%</u>)	0.962 (+2.01%)	0.9190 (+19.68%)
T-GRPO	**24.406** **(+3.09%)**	**0.3909** **(+4.60%)**	<u>0.965</u> (<u>+2.33%</u>)	<u>0.9289</u> (<u>+20.97%</u>)
DiffusionNFT	23.740 (+0.27%)	0.3750 (+0.35%)	**0.976** **(+3.50%)**	**0.9790** **(+27.49%)**

These experiments test whether T-GRPO is complementary to improvements in trajectory collection and utilization.

Tasks and metrics. We evaluate on four reward fine-tuning tasks. PickScore (Kirstain et al., 2023) and HPSv2.1 (Wu et al., 2023) measure model-based preference alignment. GenEval (Ghosh et al., 2023) evaluates compositional generation, while OCR (Cui et al., 2025) evaluates the correctness of rendered text. We follow the official DiffusionNFT configurations for the prompt datasets, training and test splits, reward functions, and evaluation procedures. Models are optimized on the corresponding training prompts and evaluated on held-out test prompts. We report the mean task-specific score, with higher values indicating better performance.

Implementation details. All backward-process experiments use Stable Diffusion 3.5 Medium (Esser et al., 2024) at 512×512 resolution. Unless otherwise specified, training uses non-CFG 10-step reverse-SDE rollouts, while evaluation uses a non-CFG deterministic solver. T-GRPO uses $\eta = 10$, and CFG-guided initialization is performed for 30 epochs before switching entirely to non-CFG training. Full optimization, sampling, and task-specific configurations are provided in Appendix B.1.

5.2 Main Results and Component Analysis

Figure 2 presents the complete optimization trajectories, while Table 1 reports the corresponding endpoint scores and their relative changes from Flow-GRPO. Selected numerical values along these trajectories are provided in Appendix B.2. Figure 3 complements these quantitative results with generations from the reported endpoint checkpoints. Because reward optimization progresses at different rates across tasks, we report task-specific horizons that are sufficiently long for the characteristic learning dynamics and relative trends to become clear. All endpoints are the latest completed checkpoints rather than retrospectively selected peak scores. In particular, PickScore and HPSv2.1 are still improving near the end of training, so their reported values represent fixed-budget performance rather than converged upper bounds.

Component analysis. Replacing the Flow-GRPO likelihood-ratio objective with T-LOSS improves all four tasks under the same reverse-SDE rollouts and optimization budget, with gains that persist throughout training rather than appearing only at the endpoint. In contrast, T-INIT alone shows a more task-dependent endpoint effect, consistent with its role in improving the starting policy rather than the subsequent optimization dynamics. Combining the two yields the strongest backward-process result on every task and consistently improves upon T-LOSS alone. Notably, on OCR, T-LOSS already improves Flow-GRPO by 19.68%, while the full T-GRPO further increases the gain to 20.97%. These results indicate that transition regression is the primary source of sustained improvement, while CFG-guided initialization provides a complementary advantage through a better non-CFG starting point.

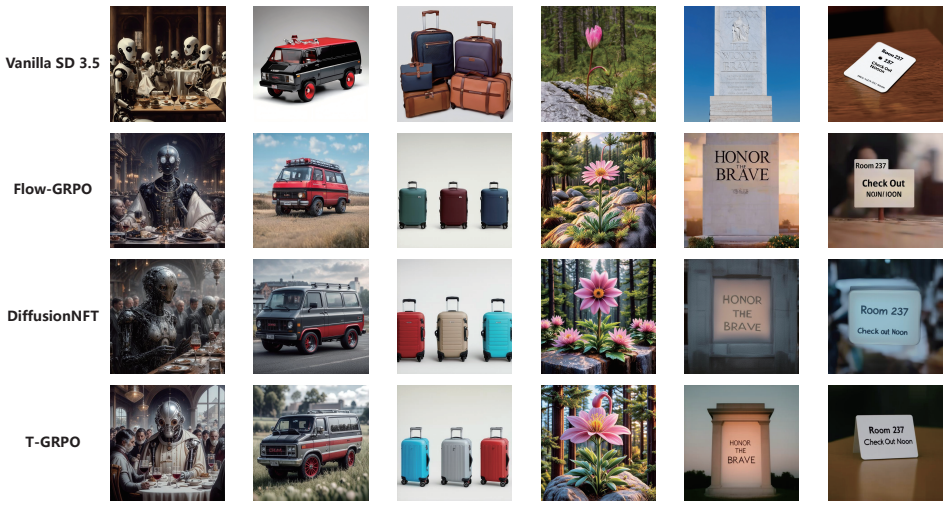

Figure 3: **Qualitative comparison at the reported training endpoints.** We show generations from the compared methods under representative evaluation prompts using PickScore, Geneval, HPS, and OCR rewards.

Comparison with the forward-process reference. DiffusionNFT exhibits a different task profile. T-GRPO performs better on PickScore and HPSv2.1, whose continuous preference scores can distinguish small changes in alignment and visual quality, making stochastic backward exploration and group-relative ranking effective. In contrast, GenEval and OCR rely more on discrete structured correctness, such as object relations, counting, and exact text recognition, and often provide sparse or threshold-like feedback. In such cases, nearby trajectories may receive similar rewards, weakening T-GRPO's relative advantage signal, whereas DiffusionNFT can more directly reinforce successful endpoints through its forward-process objective.

5.3 Compatibility with Existing Flow-GRPO Variants

We next examine whether T-GRPO remains effective when combined with stronger backward-process pipelines. We consider CPS-GRPO, MixGRPO, and Fast-GRPO, and augment each backbone with the complete T-GRPO recipe: CFG-guided initialization followed by transition-mean regression. In each case, we retain the method's original sampler, rollout schedule, and timestep-selection strategy, so the comparison isolates the effect of changing how collected transitions are optimized.

As shown in Figure 1, adding T-GRPO consistently improves all three backbones on PickScore. The gain therefore extends beyond the original Flow-GRPO pipeline and persists across substantially different trajectory designs. CPS-GRPO, MixGRPO, and Fast-GRPO improve how reverse trajectories are generated and where stochastic optimization is applied, whereas T-GRPO improves how the resulting transitions are converted into parameter updates. Their consistent combination indicates that T-GRPO provides an objective-level improvement complementary to advances in sampling and rollout design.

6 Conclusion

We introduced **T-GRPO** to better exploit the information carried by stochastic reverse trajectories. By converting reward-evaluated transition deviations into explicit mean targets, T-GRPO replaces ratio-based updates with a direct transition-level learning signal and naturally supports CFG-guided initialization. Across four reward settings and multiple Flow-GRPO variants, it consistently strengthens backward-process optimization, matching or surpassing the forward-process reference on graded preference rewards within the reported budgets. These findings position the backward process not merely as a vehicle for policy gradients, but as a valuable source of reward-guided local search when paired with an appropriate learning objective.

AI USE STATEMENT

We used generative AI tools solely to improve the linguistic clarity, grammar, and readability of the manuscript. No generative AI tools were used for research-related tasks requiring mandatory disclosure. All AI-assisted edits were carefully reviewed and verified by the authors to ensure that the original meaning and technical accuracy were preserved. The authors take full responsibility for the final content of this work.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Jonas Degrave, Steven Bohez, Yuval Tassa, Dan Belov, Nicolas Heess, and Martin Riedmiller. Relative entropy regularized policy iteration. *arXiv preprint arXiv:1812.02256*, 2018a.

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018b.

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

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/13e8be77982beb73d7ed0bbf122f9f3c-Paper-Conference.pdf.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*, 2025.

Zheng Ding and Weirui Ye. Treegrpo: Tree-advantage grpo for online rl post-training of diffusion models. *arXiv preprint arXiv:2512.08153*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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

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.

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. *arXiv preprint arXiv:2508.04324*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

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.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

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

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*, 2025b.

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

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. *Advances in neural information processing systems*, 38:40783–40818, 2026.

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.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

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

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.

Shikun Sun, Shuo Huang, Yiding Chen, Wen Sun, and Jia Jia. Value-as-return: A two-stage framework to align on the optimal score function. In *Forty-third International Conference on Machine Learning*, 2026.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

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.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models. *arXiv preprint arXiv:2509.25050*, 2025a.

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*, 2025b.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8941–8951. IEEE, 2024a.

Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference. *arXiv preprint arXiv:2402.08265*, 2024b.

Liyu Zhang, Kehan Li, Tingrui Han, Tao Zhao, Yuxuan Sheng, Shibo He, and Chao Li. Op-grpo: Efficient off-policy grpo for flow-matching models. *arXiv preprint arXiv:2604.04142*, 2026.

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

A Additional Derivations

A.1 Reverse-SDE Discretization of Flow-GRPO

Given a text condition c, a flow model learns a time-dependent velocity field $v_\theta(x_\tau, \tau, c)$. Standard generation follows the probability-flow ODE

$$\frac{\mathrm{d}x_\tau}{\mathrm{d}\tau} = v_\theta(x_\tau, \tau, c). \tag{18}$$

After discretization, the next latent is a deterministic function of the current latent and the predicted velocity $v_\theta(x_\tau, \tau, c)$. However, a deterministic ODE step does not provide a non-degenerate transition density from which the likelihood ratios between policies required by GRPO (Shao et al., 2024) can be computed.

Flow-GRPO therefore replaces the deterministic probability-flow ODE in Eq. 18 during rollout with a reverse SDE that preserves the same marginal distribution of x_τ at each time level τ. For the rectified-flow path $x_\tau = (1-\tau)x_0 + \tau x_1$, Flow-GRPO performs rollouts using

$$\mathrm{d}x_\tau = \left[v_\theta(x_\tau, \tau, c) + \frac{g(\tau)^2}{2\tau} \left(x_\tau + (1-\tau) v_\theta(x_\tau, \tau, c) \right) \right] \mathrm{d}\tau + g(\tau)\, \mathrm{d}\overline{W}_\tau, \tag{19}$$

where $\overline{W}_\tau$ denotes a reverse-time Wiener process and

$$g(\tau) = \eta_{\text{noise}} \sqrt{\frac{\tau}{1-\tau}} \tag{20}$$

is the scalar diffusion coefficient, with η_{noise} controlling the stochasticity of the rollout sampler. The stochastic term introduces local exploration and yields tractable transition densities for GRPO.

Because generation proceeds from larger to smaller values of τ, let τ_k and τ_{k+1} denote two consecutive time levels in generation order, with

$$\Delta\tau_k := \tau_{k+1} - \tau_k < 0. \tag{21}$$

Applying the Euler–Maruyama discretization to Eq. 19 gives

$$\begin{aligned} x_{k+1} &= x_k + \left[v_\theta(x_k, \tau_k, c) + \frac{g_k^2}{2\tau_k} \left(x_k + (1-\tau_k) v_\theta(x_k, \tau_k, c) \right) \right] \Delta\tau_k + g_k \sqrt{-\Delta\tau_k}\, \epsilon_k \\ &= \left(1 + \frac{g_k^2}{2\tau_k} \Delta\tau_k \right) x_k + \left(1 + \frac{g_k^2 (1-\tau_k)}{2\tau_k} \right) \Delta\tau_k\, v_\theta(x_k, \tau_k, c) + g_k \sqrt{-\Delta\tau_k}\, \epsilon_k, \end{aligned} \tag{22}$$

where $\epsilon_k \sim \mathcal{N}(0, I)$ and $g_k := g(\tau_k)$.

We denote the deterministic part of Eq. 22 by the transition mean

$$m_{\theta,k} := \alpha_k x_k + \gamma_k v_\theta(x_k, \tau_k, c), \tag{23}$$

where

$$\alpha_k = 1 + \frac{g_k^2}{2\tau_k} \Delta\tau_k, \qquad \gamma_k = \left(1 + \frac{g_k^2 (1-\tau_k)}{2\tau_k} \right) \Delta\tau_k. \tag{24}$$

Defining the transition noise scale as

$$s_k := g_k \sqrt{-\Delta\tau_k}, \tag{25}$$

the discretized reverse-SDE transition can be written compactly as

$$x_{k+1} = m_{\theta,k} + s_k \epsilon_k. \tag{26}$$

Consequently, each reverse step defines the Gaussian transition policy

$$p_\theta(x_{k+1} \mid x_k, c) = \mathcal{N}\left(x_{k+1}; m_{\theta,k}, s_k^2 I \right). \tag{27}$$

A.2 Gaussian Ratio Geometry

We first derive the Gaussian likelihood-ratio identities underlying Eq. 11. Consider a transition collected from the frozen rollout policy,

$$x_{k+1}^{(i)} = m_{\bar{\theta},k}^{(i)} + s_k \epsilon_k^{(i)}, \qquad \epsilon_k^{(i)} \sim \mathcal{N}(0, I), \tag{28}$$

and define the change in transition mean as

$$\delta_k(\theta) = m_{\theta,k}^{(i)} - m_{\bar{\theta},k}^{(i)}. \tag{29}$$

Since the current and rollout policies share the same covariance $s_k^2 I$, their likelihood ratio is

$$r_k^{(i)}(\theta) = \frac{\mathcal{N}\left(x_{k+1}^{(i)}; m_{\theta,k}^{(i)}, s_k^2 I\right)}{\mathcal{N}\left(x_{k+1}^{(i)}; m_{\bar{\theta},k}^{(i)}, s_k^2 I\right)}. \tag{30}$$

Taking logarithms gives

$$\begin{aligned}
\log r_k^{(i)}(\theta) &= -\frac{\left\| x_{k+1}^{(i)} - m_{\theta,k}^{(i)} \right\|_2^2}{2s_k^2} + \frac{\left\| x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} \right\|_2^2}{2s_k^2} \\
&= \frac{\left\langle x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)}, \delta_k(\theta) \right\rangle}{s_k^2} - \frac{\|\delta_k(\theta)\|_2^2}{2s_k^2} \\
&= \frac{\left\langle \epsilon_k^{(i)}, \delta_k(\theta) \right\rangle}{s_k} - \frac{\|\delta_k(\theta)\|_2^2}{2s_k^2}.
\end{aligned} \tag{31}$$

Equivalently, completing the square yields

$$\log r_k^{(i)}(\theta) = -\frac{\left\| m_{\theta,k}^{(i)} - x_{k+1}^{(i)} \right\|_2^2}{2s_k^2} + \frac{\left\| \epsilon_k^{(i)} \right\|_2^2}{2}. \tag{32}$$

Thus, for a fixed sampled transition, the log ratio is a concave quadratic function of the current transition mean, centered at the sampled next latent $x_{k+1}^{(i)}$.

Differentiating with respect to the transition mean gives

$$\nabla_{m_{\theta,k}} \log r_k^{(i)}(\theta) = \frac{x_{k+1}^{(i)} - m_{\theta,k}^{(i)}}{s_k^2}, \tag{33}$$

and therefore

$$\nabla_{m_{\theta,k}} \left[A^{(i)} r_k^{(i)}(\theta) \right] = A^{(i)} r_k^{(i)}(\theta) \frac{x_{k+1}^{(i)} - m_{\theta,k}^{(i)}}{s_k^2}. \tag{34}$$

At the rollout policy, $m_{\theta,k}^{(i)} = m_{\bar{\theta},k}^{(i)}$ and $r_k^{(i)}(\theta) = 1$, so

$$\nabla_{m_{\theta,k}} \left[A^{(i)} r_k^{(i)}(\theta) \right] \Big|_{\theta=\bar{\theta}} = A^{(i)} \frac{x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)}}{s_k^2}. \tag{35}$$

For the Gaussian transition policy, the Fisher matrix with respect to its mean is

$$F_k = s_k^{-2} I. \tag{36}$$

Applying its inverse to Eq. 35 gives

$$F_k^{-1} \nabla_{m_{\theta,k}} \left[A^{(i)} r_k^{(i)}(\theta) \right] \Big|_{\theta=\bar{\theta}} = A^{(i)} \left(x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} \right), \tag{37}$$

which recovers Eq. 11. Hence, in transition-mean space, the local Flow-GRPO improvement direction is precisely the sampled reverse-SDE residual, signed and scaled by the trajectory advantage.

A.3 Derivation of the KL-Proximal Transition Target

We provide the derivation of the closed-form solution to Eq. 13. Since the candidate and rollout transition policies have the same covariance $s_k^2 I$, their KL divergence depends only on the displacement of their means:

$$D_{\mathrm{KL}}\left(\mathcal{N}\left(m_{\bar{\theta},k}^{(i)}+\delta_k, s_k^2 I\right)\middle\|\mathcal{N}\left(m_{\bar{\theta},k}^{(i)}, s_k^2 I\right)\right)=\frac{\|\delta_k\|_2^2}{2s_k^2}. \tag{38}$$

Substituting Eq. 38 into Eq. 13 gives

$$\max_{\delta_k} \quad \frac{1}{s_k^2}\left[A^{(i)}\left\langle x_{k+1}^{(i)}-m_{\bar{\theta},k}^{(i)}, \delta_k\right\rangle-\frac{1}{2\eta_k}\|\delta_k\|_2^2\right]. \tag{39}$$

The objective is strictly concave for $\eta_k > 0$, with gradient

$$\nabla_{\delta_k}=\frac{1}{s_k^2}\left[A^{(i)}\left(x_{k+1}^{(i)}-m_{\bar{\theta},k}^{(i)}\right)-\frac{1}{\eta_k}\delta_k\right]. \tag{40}$$

Setting the gradient to zero yields

$$\delta_k^\star=\eta_k A^{(i)}\left(x_{k+1}^{(i)}-m_{\bar{\theta},k}^{(i)}\right), \tag{41}$$

which recovers Eq. 14. Therefore,

$$m_k^{\star,(i)}=m_{\bar{\theta},k}^{(i)}+\delta_k^\star=m_{\bar{\theta},k}^{(i)}+\eta_k A^{(i)}\left(x_{k+1}^{(i)}-m_{\bar{\theta},k}^{(i)}\right), \tag{42}$$

recovering the transition target in Eq. 15. Notably, the common factor s_k^{-2} cancels in the optimizer, so the target displacement is expressed directly in transition-mean space, while s_k^2 defines the local Gaussian metric.

A.4 Analysis of the T-GRPO Objective

We further analyze how T-GRPO changes the optimization behavior of the Flow-GRPO transition objective. For a sampled transition

$$x_{k+1}^{(i)}=m_{\bar{\theta},k}^{(i)}+s_k\epsilon_k^{(i)}, \tag{43}$$

the gradient of the unclipped Flow-GRPO loss with respect to the current transition mean is

$$\nabla_{m_{\theta,k}}\mathcal{L}_{\text{Flow-GRPO},k}=-A^{(i)}r_k^{(i)}(\theta)\frac{x_{k+1}^{(i)}-m_{\theta,k}^{(i)}}{s_k^2}. \tag{44}$$

Using the Gaussian ratio geometry derived in Appendix A.2, the ratio can be written as

$$r_k^{(i)}(\theta)=\exp\left(\frac{\left\|\epsilon_k^{(i)}\right\|_2^2}{2}-\frac{\left\|m_{\theta,k}^{(i)}-x_{k+1}^{(i)}\right\|_2^2}{2s_k^2}\right). \tag{45}$$

Thus, although the transition residual provides the desired update direction, its magnitude is additionally modulated by the evolving likelihood ratio. When the current mean moves far from the sampled transition, the exponential term can strongly suppress the gradient; PPO-style clipping can further alter the update once the ratio leaves the clipping range.

T-GRPO removes this ratio-dependent scaling by fixing an explicit target at rollout time,

$$m_k^{\star,(i)}=m_{\bar{\theta},k}^{(i)}+\eta_k A^{(i)}\left(x_{k+1}^{(i)}-m_{\bar{\theta},k}^{(i)}\right), \tag{46}$$

and optimizing

$$\mathcal{L}_{\mathrm{T-GRPO},k}=\frac{\left\|m_{\theta,k}^{(i)}-\mathrm{sg}\left[m_k^{\star,(i)}\right]\right\|_2^2}{2s_k^2}. \tag{47}$$

Its gradient is

$$\nabla_{m_{\theta,k}} \mathcal{L}_{\mathrm{T-GRPO},k} = \frac{m_{\theta,k}^{(i)} - m_k^{\star,(i)}}{s_k^2}. \tag{48}$$

At the rollout policy, this gives

$$-\nabla_{m_{\theta,k}} \mathcal{L}_{\mathrm{T-GRPO},k}\big|_{\theta=\bar{\theta}} = \eta_k A^{(i)} \frac{x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)}}{s_k^2}, \tag{49}$$

which is the same local improvement direction as the unclipped Flow-GRPO objective, up to the explicit scale η_k. The distinction therefore lies not in which direction is extracted from the rollout, but in how that direction is followed during optimization: Flow-GRPO continually reweights it through the current likelihood ratio and clipping, whereas T-GRPO regresses toward a fixed finite target.

This view also clarifies the role of stochastic reverse-SDE rollout. Since

$$x_{k+1}^{(i)} - m_{\bar{\theta},k}^{(i)} = s_k \epsilon_k^{(i)}, \tag{50}$$

each sampled transition records a local direction explored around the model-predicted mean. The trajectory advantage evaluates the outcome of that exploration relative to other trajectories in the group, and T-GRPO writes the resulting signed direction back into the transition mean. Under the affine parameterization

$$m_{\theta,k} = \alpha_k x_k + \gamma_k v_\theta(x_k, \tau_k, c), \tag{51}$$

the target equivalently induces the velocity target

$$v_k^{\star,(i)} = v_{\bar{\theta},k}^{(i)} + \eta_k A^{(i)} \frac{s_k}{\gamma_k} \epsilon_k^{(i)}. \tag{52}$$

Hence, transition regression can be viewed as transferring a reward-evaluated stochastic exploration direction into the underlying velocity model, rather than simply imitating every sampled transition.

Finally, the KL geometry gives a direct interpretation of the target scale. Using $m_k^{\star,(i)} - m_{\bar{\theta},k}^{(i)} = \eta_k A^{(i)} s_k \epsilon_k^{(i)}$, the divergence between the target transition and the rollout transition is

$$\begin{aligned} &D_{\mathrm{KL}}\left(\mathcal{N}\left(m_k^{\star,(i)}, s_k^2 I\right) \middle\| \mathcal{N}\left(m_{\bar{\theta},k}^{(i)}, s_k^2 I\right)\right) \\ &= \frac{\eta_k^2 \left(A^{(i)}\right)^2}{2} \left\|\epsilon_k^{(i)}\right\|_2^2. \end{aligned} \tag{53}$$

Therefore, η_k directly controls the scale of the reward-conditioned displacement in the local Gaussian geometry. This also explains its role as the principal target-scale hyperparameter in T-GRPO.

A.5 Details of CFG-Guided Transition Initialization

For each prompt, the frozen CFG teacher generates a group of trajectories. At state $x_k^{w,(i)}$, its transition mean is

$$m_{0,k}^{w,(i)} = \alpha_k x_k^{w,(i)} + \gamma_k v_0^w\left(x_k^{w,(i)}, \tau_k, c\right), \tag{54}$$

where v_0^w denotes the frozen base model evaluated with CFG scale w. The corresponding teacher proposal is

$$x_{k+1}^{w,(i)} = m_{0,k}^{w,(i)} + s_k^w \epsilon_k^{(i)}, \qquad \epsilon_k^{(i)} \sim \mathcal{N}(0, I), \tag{55}$$

where $s_k^w \geq 0$ controls the amount of stochasticity in the teacher trajectory. The construction does not require a particular value of s_k^w, because $y_k^{w,(i)}$ is used as a proposal rather than an on-policy sample. We use $s_k^w = 0$ by default because deterministic CFG rollouts produce higher-reward proposal trajectories than their noisy counterparts.

On the same teacher-visited state, the detached non-CFG student defines the anchor transition mean

$$m_{\bar{\theta},k}^{0,(i)} = \alpha_k x_k^{0,(i)} + \gamma_k v_{\bar{\theta}}\left(x_k^{0,(i)}, \tau_k, c\right). \tag{56}$$

Applying the same target construction as in Eq. 15, we obtain

$$\boxed{m^{\star,(i)}_{k,\text{init}} = m^{0,(i)}_{\bar{\theta},k} + \eta_k A^{(i)} \left(x^{w,(i)}_{k+1} - m^{0,(i)}_{\bar{\theta},k} \right)}. \tag{57}$$

Because $x^{w,(i)}_{k+1}$ is not sampled from the non-CFG reverse-SDE policy, its displacement is not naturally normalized by the student sampler variance. We instead estimate an effective scale directly from the proposal discrepancy:

$$\widehat{s}^{(i)}_k = \max \left(s_{\min}, \text{Std}_{\text{lat}} \left[x^{w,(i)}_{k+1} - m^{0,(i)}_{\bar{\theta},k} \right] \right), \tag{58}$$

where Std_{lat} is computed over the latent dimensions of each sample, with gradients stopped through the estimate.

The initialization objective retains the transition-regression form of Eq. 16:

$$\mathcal{L}_{\text{init}}(\theta) = \mathbb{E} \left[\sum_{i,k} \frac{\left\| m^{0,(i)}_{\theta,k} - \text{sg} \left[m^{\star,(i)}_{k,\text{init}} \right] \right\|_2^2}{2 \, \text{sg} \left[\widehat{s}^{(i)}_k \right]^2} \right]. \tag{59}$$

Compared with the main T-GRPO loss, the sampled non-CFG transition is replaced by the teacher proposal $x^{w,(i)}_{k+1}$, and the sampler scale is replaced by the proposal-dependent scale $\widehat{s}^{(i)}_k$.

After initialization, the CFG teacher is discarded and the initialized student becomes the non-CFG rollout policy. All subsequent trajectory collection, reward optimization, and inference are performed without CFG.

B Additional Experiment Details

B.1 Additional Implementation Details

Unless otherwise specified, all backward-process experiments use Stable Diffusion 3.5 Medium (Esser et al., 2024) and generate images at a resolution of 512×512. One epoch consists of one rollout-collection round followed by optimization on the collected transitions. Each round contains 48 prompts with 24 trajectories per prompt, yielding 1,152 generated training images. Training rollouts are generated without CFG (Ho & Salimans, 2022) using a 10-step reverse-SDE sampler with noise level $\eta_{\text{noise}} = 0.7$. Flow-GRPO and its controlled variants use the same sampled trajectories and optimization budget. Evaluation is performed without CFG on held-out prompts using a 40-step deterministic Euler solver. For transition-mean regression, we set the target scale to $\eta_t = 10$ for all optimized transitions. CFG-guided initialization uses the frozen base model as a deterministic teacher with guidance scale 4.5 and is performed for 30 epochs. The teacher is discarded after initialization, and all subsequent rollout collection and reward optimization are conducted entirely without CFG. Detailed optimization and task-specific configurations are provided in Appendix B.1.

B.2 Numerical Values along Optimization Trajectories

Table 2 provides the numerical values at selected checkpoints from the optimization trajectories reported in the main paper. These values complement the curves in Figures 2 and 1 and are included for detailed quantitative comparison.

B.3 Effect of Target Scale

We study the target scale η, which controls the magnitude of the advantage-weighted displacement in the T-GRPO target. As shown in Figure 4(a), a small value of $\eta = 1$ leads to substantially slower reward improvement, indicating that overly conservative target shifts underutilize the transition-level exploration signal. Increasing η to 5 noticeably accelerates optimization, while $\eta = 10$ and 20 achieve similarly strong performance throughout training. The gain largely saturates beyond $\eta = 10$, and we therefore use $\eta = 10$ as the default in all main experiments.

Table 2: **Numerical values along the optimization trajectories.** We report selected checkpoints from the training curves shown in Figures 2 and 1. For the controlled study, we compare Flow-GRPO, its T-LOSS and T-INIT variants, the full T-GRPO, and DiffusionNFT across the four reward settings. The "30 (init)" column denotes the checkpoint immediately after CFG-guided initialization for methods using T-INIT; for the remaining methods, it reports the corresponding training epoch. Higher is better.

GenEval	30 (init)	200	400	600	800
Flow-GRPO	0.245	0.806	0.878	0.923	0.943
DiffusionNFT	0.772	0.935	0.956	0.971	0.976
Flow-GRPO+T-GRPO$_{\text{loss}}$	0.567	0.888	0.934	0.951	0.962
Flow-GRPO+T-GRPO$_{\text{init}}$	0.646	0.818	0.908	0.916	0.922
Flow-GRPO+T-GRPO$_{\text{loss}}$+T-GRPO$_{\text{init}}$	0.646	0.882	0.945	0.951	0.965
PickScore	30 (init)	250	500	1000	2000
Flow-GRPO	0.8145	0.8684	0.8834	0.8981	0.9106
DiffusionNFT	0.8720	0.8948	0.8989	0.9049	0.9131
Flow-GRPO+T-GRPO$_{\text{loss}}$	0.8400	0.8880	0.9030	0.9189	0.9353
Flow-GRPO+T-GRPO$_{\text{init}}$	0.8260	0.8797	0.8893	0.9012	0.9162
Flow-GRPO+T-GRPO$_{\text{loss}}$+T-GRPO$_{\text{init}}$	0.8260	0.8944	0.9071	0.9230	0.9387
OCR	30 (init)	100	150	250	500
Flow-GRPO	0.3794	0.6582	0.6807	0.7204	0.7679
DiffusionNFT	0.8852	0.9730	0.9778	0.9778	0.9790
Flow-GRPO+T-GRPO$_{\text{loss}}$	0.4791	0.8940	0.9014	0.9084	0.9190
Flow-GRPO+T-GRPO$_{\text{init}}$	0.4860	0.8305	0.8835	0.8971	0.9021
Flow-GRPO+T-GRPO$_{\text{loss}}$+T-GRPO$_{\text{init}}$	0.4860	0.8856	0.9127	0.9211	0.9289
HPSv2	30 (init)	150	300	500	1000
Flow-GRPO	0.2862	0.3383	0.3559	0.3616	0.3737
DiffusionNFT	0.3543	0.3668	0.3696	0.3713	0.3750
Flow-GRPO+T-GRPO$_{\text{loss}}$	0.3208	0.3586	0.3726	0.3777	0.3882
Flow-GRPO+T-GRPO$_{\text{init}}$	0.2888	0.3437	0.3605	0.3678	0.3742
Flow-GRPO+T-GRPO$_{\text{loss}}$+T-GRPO$_{\text{init}}$	0.2888	0.3574	0.3709	0.3795	0.3909
PickScore	100	250	500	750	1000
MixGRPO	0.8634	0.8956	0.9048	0.9120	0.9170
MixGRPO+T-GRPO	0.8544	0.8997	0.9156	0.9226	0.9257
Flow-GRPO-Fast	0.8817	0.8935	0.9074	0.9084	0.9136
Flow-GRPO-Fast+T-GRPO	0.8841	0.8989	0.9109	0.9179	0.9247
TempFlow-GRPO	0.8803	0.8982	0.9108	0.9123	0.9211
TempFlow-GRPO+T-GRPO	0.8788	0.9036	0.9170	0.9242	0.9247

B.4 EFFECT OF CFG-GUIDED INITIALIZATION

We further examine the effect of the rollout noise used by the CFG teacher during initialization. Recall that the guided trajectory serves only as an auxiliary proposal for constructing transition targets; it is not a behavior trajectory whose likelihood must be matched by the student. The teacher rollout noise scale can therefore be chosen independently of the subsequent non-CFG T-GRPO training.

Figure 4(b) compares initialization with noise levels 0, 0.2, 0.4, and 0.7. All settings rapidly reach nearly identical reward levels, showing that CFG-guided initialization is insensitive to the teacher rollout stochasticity over the tested range. We use zero noise by default, which provides a simple deterministic proposal trajectory without introducing unnecessary sampling variance. After initialization, the CFG teacher is discarded and all settings follow the same non-CFG T-GRPO optimization procedure; hence, the teacher noise can affect the second stage only through the initialized policy.

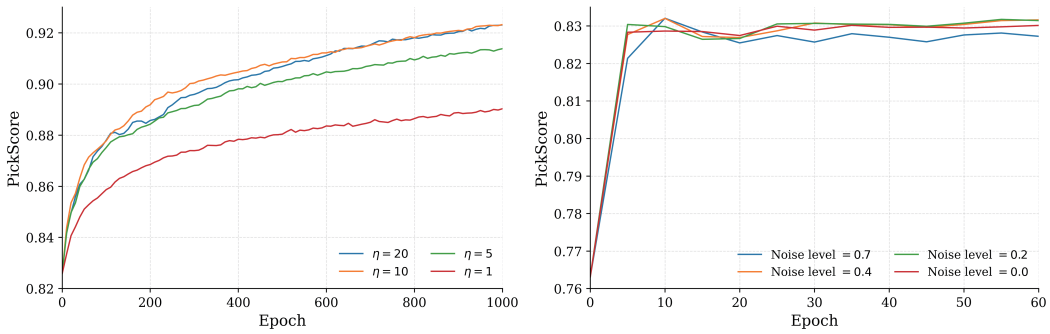

Figure 4: **Ablations of T-GRPO target scale and CFG-teacher rollout noise.** **(a)** Effect of the target scale η on reward optimization. Small target shifts substantially slow learning, while $\eta = 10$ and $\eta = 20$ achieve similar performance, indicating saturation beyond a moderate target scale. **(b)** Effect of the CFG-teacher rollout noise during initialization. Noise levels from 0 to 0.7 converge to nearly identical reward levels, showing that the initialization is largely insensitive to the stochasticity of the auxiliary guided proposals.

B.5 Cross-Reward Trade-offs

Our experiments primarily study optimization efficiency for a given reward. Accordingly, we use a small reference-regularization coefficient $\beta = 10^{-4}$, allowing the optimized policy to move relatively freely toward the target reward. We therefore additionally examine whether the faster optimization of T-GRPO introduces stronger degradation on rewards that are not used for training.

Table 3 compares Flow-GRPO with two T-GRPO checkpoints: an earlier checkpoint chosen to approximately match the target reward achieved by Flow-GRPO, and the final T-GRPO checkpoint after further optimization. At matched target-reward levels, we do not observe additional cross-reward degradation from T-GRPO. When optimizing PickScore, T-GRPO reaches a comparable target score after 720 epochs versus 2,490 epochs for Flow-GRPO, while attaining higher ImageReward and HPSv2 scores. A similar pattern holds for GenEval: the matched T-GRPO checkpoint achieves comparable GenEval performance while maintaining similar or higher scores on the other evaluated rewards.

Cross-reward trade-offs become more visible when T-GRPO is allowed to continue optimizing beyond the level reached by Flow-GRPO. On GenEval, increasing the target score from 0.9362 at the matched checkpoint to 0.9601 at the final checkpoint is accompanied by decreases in PickScore, ImageReward, and HPSv2. For PickScore optimization, this effect is considerably milder: the final checkpoint slightly decreases ImageReward and HPSv2 relative to the matched T-GRPO checkpoint, while both remain above the corresponding Flow-GRPO scores. These results suggest that T-GRPO does not inherently amplify cross-reward degradation at a matched optimization level; rather, its stronger optimization can move the policy further along the underlying reward trade-off.

Mitigating such reward trade-offs is outside the scope of this work, which focuses on improving the efficiency of reward optimization. Stronger reference regularization or multi-reward objectives could be incorporated when cross-reward preservation is itself an optimization objective.

B.6 Text Prompts for Qualitative Comparisons

The qualitative examples in Figure 3 correspond to the following prompts, listed from left to right:

1. **PickScore:** “Robots, cyborgs, priests, eating and drinking wine, busy restaurant, art by Antonello da Messina.”
2. **PickScore:** “A-team GMC 1982 van, black, red, grey, red alloy wheels.”
3. **GenEval:** “A photo of three suitcases.”
4. **HPSv2.1:** “A pink flower with a long neck and sticking out of a crack on a rock among pine trees in a forest blooming.”

Table 3: **Cross-reward evaluation at matched target-reward and final checkpoints.** For each training objective, we compare the Flow-GRPO checkpoint with an earlier T-GRPO checkpoint selected to achieve a similar target-reward value (*matched*), as well as the final T-GRPO checkpoint. The optimized reward is highlighted in gray. Higher is better for all metrics.

Method	Epoch	PickScore ↑	ImageReward ↑	HPSv2 ↑	GenEval ↑
Optimizing PickScore					
Flow-GRPO	2490	0.9140	1.3298	0.3266	–
T-GRPO (matched)	720	0.9166	1.3843	0.3413	–
T-GRPO (final)	2490	0.9434	1.3591	0.3402	–
Optimizing GenEval					
Flow-GRPO	1620	0.8427	0.9286	0.2499	0.9315
T-GRPO (matched)	1260	0.8641	0.9292	0.2602	0.9362
T-GRPO (final)	1560	0.8406	0.8145	0.2444	0.9601

5. **OCR:** "A grand marble monument stands tall, its surface meticulously etched with the words 'Honor the Brave'."
6. **OCR:** "A close-up of a hotel key card sleeve on a wooden desk, with the text 'Room 237 Check Out Noon' clearly visible. The background shows a blurred hotel room, emphasizing the key card sleeve."