BEYOND THE KL BRAKE: REWARD-LEVEL ADAPTIVE ANCHORING FOR DIFFUSION POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are difficult to directly align with specific preferences. While Reinforcement Learning (RL) improves alignment, it suffers from sparse rewards. A common method is to assign the only final reward to all timesteps, but this weakens credit assignment. When rewards are identical across steps, action-level differences are obscured, making it hard to correct erroneous denoising behaviors and increasing the risk of distributional shift. Kullback–Leibler (KL) divergence is widely used to prevent distributional collapse, yet the KL-as-loss paradigm has limitations. First, it only indirectly regularizes parameters and cannot directly suppress harmful actions. Local drift caused by such actions can accumulate through diffusion's iterative process, a problem observed in UNet-based RL and amplified in Flow RL with SDE noise. Second, under KL-as-loss, the model must continually overcome the optimization resistance introduced by a static regularization term while pursuing higher rewards. As a result, more time is required to reach the same level of performance as training without KL. Third, a fixed KL weight creates a trade-off between preserving the pretrained distribution (exploitation) and efficient alignment (exploration): small weights cause rapid distribution drift, while large weights over-anchor the model to the pretrained distribution, weakening preference optimization. To address the above issues, we propose StepKL, which converts KL-as-loss into fine-grained step rewards, making distributional deviation an explicit per-step feedback signal during online policy optimization. This suppresses drift accumulation and mitigates reward sparsity. By jointly modeling exploration intensity and reward trends, StepKLadaptively adjusts anchoring strength, preserving pretrained advantages while improving alignment.

1 INTRODUCTION

Pre-trained diffusion models only focus on recovering the data distribution and are therefore difficult to directly align with specific preferences Ma et al. (2025). Reinforcement learning (RL)Schulman et al. (2017) can improve alignmentBlack et al. (2023); Deng et al. (2024), but faces challenges such as reward sparsity and reward hacking. Reward sparsity arises since feedback is provided on the final image, making RL training difficult Zhang et al. (2024b). A compromise is to uniformly propagate the final reward to all preceding steps Franceschelli & Musolesi (2024); Yan et al. (2026).

Although reward backfilling stabilizes training, it weakens action–reward attribution. RL evaluates behavior based on reward differences induced by different actions. When all timesteps receive identical rewards, the model cannot distinguish which actions truly improve generation quality. If a denoising action introduces harmful distributional drift, diffusion's iterative process propagates and accumulates this bias across subsequent steps, which is a key cause of reward hacking Yan et al. (2026). To date, KL-as-loss Fan et al. (2023) remains the most widely adopted method for mitigating hacking. By constraining RL updates to stay close to the pretrained generative distribution, it prevents the model from collapsing into a few modes or shortcut solutions. For diffusion models, such reference constraints are particularly important, since the pretrained distribution encodes rich visual priors and diverse generative patterns. Excessive deviation from this distribution can severely harm realism and diversity. Detailed ***Related Works*** is provided in the Appendix.

This paper identifies key limitations of the current KL paradigm. First, KL-as-loss constrains parameter updates only through a global objective and does not directly participate in step-level reward

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.

or advantage construction. As a result, it cannot promptly suppress harmful denoising actions at specific timesteps or correct the misattribution introduced by reward propagation. Although it alleviates distributional shift at a global level, it lacks immediate feedback for local drift. Second, recent studies show that adding KL as a loss term requires more time to reach the same performance level as training without KL Liu et al. (2026). When KL acts as an additional penalty, the model must continuously counteract the static regularization term while maximizing rewards, which slows optimization. Beyond these issues, existing KL methods typically rely on a fixed weighting coefficient, creating a trade-off between preserving the original distribution (exploitation) and improving alignment (exploration). If the KL weight is small, the model may deviate from the pretrained distribution and suffer reward hacking. If it is large, the model becomes overly anchored to the original distribution, leading to insufficient alignment improvement.

Building on the above insights, we propose StepKL, a reward-level adaptive anchoring RL approach. StepKL converts the reference constraint into a fine-grained step reward, called Anchoring Reward. As a result, the model receives explicit and differentiated feedback at each timestep based on how much it deviates from the pretrained distribution. In this way, the reference constraint is no longer an external regularization term independent of the reward, but directly contributes to the advantage/action estimation in RL. This design alleviates reward sparsity and suppresses the accumulation of harmful drift. Furthermore, StepKL jointly models exploration intensity and reward trends to adaptively determine the strength of reference anchoring. When exploration remains strong and the reward trend continues to improve, the method allows the policy to perform more sufficient preference optimization near the pretrained distribution. This leads to exploration-dominant training. When exploration rises, but reward trends deteriorate, it indicates harmful exploration. In such cases, the anchoring reward is strengthened to suppress ineffective perturbations, yielding exploitation-dominant training. StepKL dynamically adjusts constraint strength using training dynamics and reward trends. It preserves the pretrained distribution while improving alignment performance.

Our method is important for Flow RL. Specifically, in the Flow RL paradigm, increasing SDE noise enhances exploration and improves the chance of discovering better policies. However, stronger noise raises the risk of distributional drift. KL-as-loss provides a coarse-grained, parameter-level constraint that pulls the model back globally, which can become ineffective when noise is too large. In this work, we convert the reference constraint into a fine-grained step reward and combine it with an adaptive mechanism. This design penalizes high-deviation exploration when it brings little improvement. Meanwhile, it preserves exploration that leads to beneficial outcomes. As a result, the model can search for better alignment solutions under larger SDE noise without losing control.

This paper makes three main contributions: (1) We revisit the KL mechanism in diffusion RL and identify three key limitations of the KL-as-loss paradigm. (2) We propose Anchoring Reward, which integrates the reference constraint into the reward modeling of each denoising step. This fine-grained supervision alleviates reward sparsity and prevents the propagation and accumulation of harmful drift. (3) By jointly modeling exploration intensity and reward trends, we adaptively adjust the anchoring strength. This enables more effective alignment exploration while preserving the advantages of the pretrained distribution, thereby improving both reward acquisition and training efficiency.

2 METHOD

To address the limitations of KL-as-loss, we propose StepKL see Fig.1. In Sec.2.2, we construct Anchoring Reward to provide differentiated feedback for suppressing harmful distributional drift. In Sec.2.3, we introduce an adaptive trade-off mechanism to enhance preference alignment while preserving the advantages of the pretrained distribution.

2.1 DIFFUSION MODELS WITH RL

Diffusion Models. Given a clean sample $\mathbf{x}_0$, the forward process gradually injects Gaussian noise as $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}\right)$, where $\alpha_t = 1 - \beta_t$ and β_t denotes the noise variance at timestep t. Equivalently, the one-step forward transition can be written as $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. By recursively applying this process, one obtains a latent trajectory $\mathbf{x}_0 \to \cdots \to \mathbf{x}_T$, where $\mathbf{x}_T$ approaches a standard Gaussian distribution. Let $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. The marginal distribution from $\mathbf{x}_0$ to $\mathbf{x}_t$ is given by $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\xi}$. The

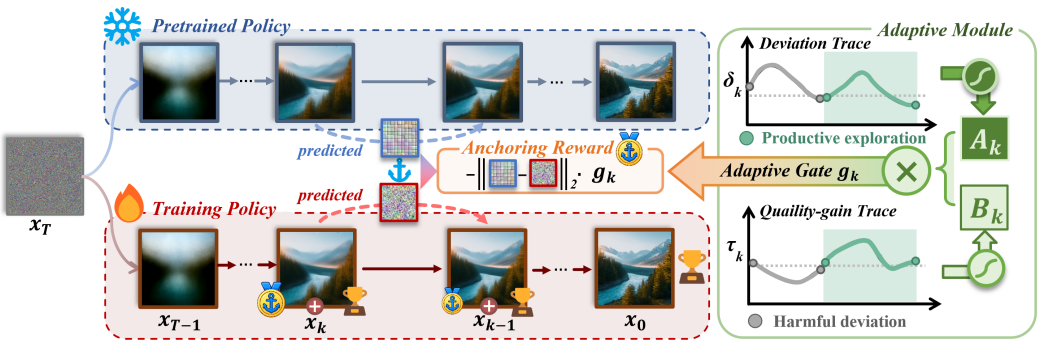


Figure 1: StepKL Framework.

reverse process is parameterized by a denoising model that learns to approximate $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$ and progressively reconstructs clean samples from $\mathbf{x}_T$ under condition $\mathbf{c}$. In the common noise-prediction parameterization, a neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$ predicts the noise component contained in $\mathbf{x}_t$. Based on this predicted noise, the estimation of the clean sample can be recovered as:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t, t, \mathbf{c}) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}) \right). \tag{1}$$

2.1.1 *Denoising as Markov Decision Process.*

We use $p_\theta(\mathbf{x}_{0:T} \mid \mathbf{c})$ to define as the diffusion model who performs text-to-image generation tasks. The prompt $\mathbf{c}$ follows the distribution $p(\mathbf{c})$. The diffusion sampling process can be formulated as an MDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R \rangle$, where $\mathbf{s}_t \triangleq (\mathbf{c}, \mathbf{x}_{T-t}) \in \mathcal{S}$ denotes the state composed of the condition and the current intermediate variable, $\mathbf{a}_t \triangleq \mathbf{x}_{T-t-1} \in \mathcal{A}$ denotes the action corresponding to the next denoised variable, $\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) \triangleq p_\theta(\mathbf{x}_{T-t-1} \mid \mathbf{x}_{T-t}, \mathbf{c})$ denotes the denoising policy determined by the model prediction and sampling noise, $P_0(\mathbf{s}_0) \triangleq p(\mathbf{c})\mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ denotes the initial state distribution, $P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \triangleq \delta_{(\mathbf{c}, \mathbf{x}_{T-t-1})}$ denotes the deterministic transition after selecting $\mathbf{a}_t$, and R denotes the reward function:

$$R(\mathbf{s}_t, \mathbf{a}_t) \triangleq \begin{cases} r(\mathbf{s}_{t+1}) = r(\mathbf{x}_0, \mathbf{c}) & \text{if } t = T-1, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The RL objective is to maximize the expected task reward: $\max_\theta \mathbb{E}_{p(\mathbf{c})} \mathbb{E}_{p_\theta(\mathbf{x}_0|\mathbf{c})}[r(\mathbf{x}_0, \mathbf{c})]$. Extreme task-reward sparsity makes single-reward RL training unstable for denoising-policy learning. Therefore, most studies Fan et al. (2023); Black et al. (2023); Franceschelli & Musolesi (2024); Chen et al. (2024c); Uehara et al. (2024) adopt reward backfilling in practice, assigning the sole reward equally to all preceding denoising steps; the resulting RL reward is:

$$R_{\text{real}}(\mathbf{s}_t, \mathbf{a}_t) \triangleq r(\mathbf{x}_0, \mathbf{c}). \tag{3}$$

This compromise reduces instability, yet uniform global rewards cause attribution ambiguity by blurring timestep action-outcome correspondence.

2.2 Anchoring Reward

To mitigate action-reward misattribution in Eq. 3 and turn loss-level distribution constraints into step-wise drift feedback, we introduce Anchoring Reward as $R_{\text{an}}(\mathbf{s}_t, \mathbf{a}_t)$. It measures timestep-wise prediction deviation between the trainable and reference models to provide differentiated one-step rewards, directly suppressing harmful denoising actions and their drift accumulation. Specifically, let $\boldsymbol{\epsilon}_{\theta^*}$ and $\boldsymbol{\epsilon}_\theta$ denote the frozen reference and trainable models. At timestep t, their predicted noise values are: $\mathbf{e}_t^{\text{ref}} = \boldsymbol{\epsilon}_{\theta^*}(\mathbf{x}_t, t, \mathbf{c})$, $\mathbf{e}_t^{\text{pre}} = \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c})$. The predicted noise difference $\mathcal{D}$ between $\boldsymbol{\epsilon}_{\theta^*}$ and $\boldsymbol{\epsilon}_\theta$ at timestep t is formulated as $\mathcal{D}^{(t)} = \left\| \mathbf{e}_t^{\text{pre}} - \mathbf{e}_t^{\text{ref}} \right\|_2^2$. $\mathcal{D}^{(t)}$ measures timestep-wise predictive consistency: Lower values mean the fine-tuned distribution better matches the pretrained

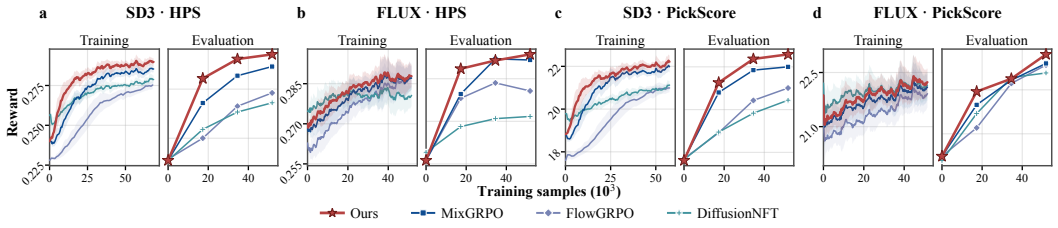


Figure 2: **Reward optimization on the Pick-a-Pic prompt set.** Training and held-out evaluation rewards are reported for HPSv2 and PickScore objectives using SD3.5 and FLUX.1-dev.

distribution. We therefore build an intermediate reward from this discrepancy to provide fine-grained step-wise guidance and suppress excessive RL-induced drift. The $R_{\text{an}}(\mathbf{s}_t, \mathbf{a}_t)$ is computed as:

$$R_{\text{an}}(\mathbf{s}_t, \mathbf{a}_t) = -\mathcal{D}^{(t)} = -\left\|\mathbf{e}_t^{\text{ref}} - \mathbf{e}_t^{\text{pre}}\right\|_2^2. \tag{4}$$

The negative sign enforces discrepancy minimization. By providing differentiated feedback in denoising, $R_{\text{an}}(\mathbf{s}_t, \mathbf{a}_t) = -\mathcal{D}^{(t)}$ mitigates reward sparsity and attribution ambiguity. For Flow-matching models, we instead measure velocity-prediction discrepancy. Let v_{θ^*} and v_θ denote the frozen reference and trainable velocity predictors. At timestep t, their predicted velocities are: $\mathbf{v}_t^{\text{ref}} = v_{\theta^*}(\mathbf{x}_t, t, c)$, $\mathbf{v}_t^{\text{pre}} = v_\theta(\mathbf{x}_t, t, c)$. Then, the flow-matching anchoring reward:

$$R_{\text{an}}^{\text{FM}}(\mathbf{s}_t, \mathbf{a}_t) = -\left\|\mathbf{v}_t^{\text{ref}} - \mathbf{v}_t^{\text{pre}}\right\|_1. \tag{5}$$

KL-as-loss cannot promptly correct step-level drift. $R_{\text{an}}(\mathbf{s}_t, \mathbf{a}_t) = -\mathcal{D}^{(t)}$ penalizes deviations and provides differentiated dense feedback to reduce reward sparsity and action-reward misattribution.

2.3 Adaptive Distribution-Preference Trade-off

The Anchoring Reward in Eq. 4 measures the step-wise prediction deviation of the training model from the frozen reference model. However, applying the same penalty to all deviations may also suppress beneficial exploration that improves generation quality. To address this, we leverage the historical statistics of the current denoising trajectory to adaptively determine whether a deviation should be penalized. We strengthen the anti-drift constraint only when the deviation at a step becomes larger than its historical level and does not bring sufficient generation quality improvement.

Deviation-Quality Aware. Consider a sampled trajectory with N denoising steps, indexed as $k = 0, \ldots, N-1$. Let t_k denote the diffusion timestep corresponding to the k^{th} denoising step. The prediction deviation of the training model from the reference model at this step is $d_k \triangleq \mathcal{D}^{(t_k)}$. The trajectory history at step k is $\mathcal{H}_k^d = \{0, 1, \ldots, k-1\}$. For $k \geq 1$, the intra-trajectory deviation is

$$\begin{aligned} m_k^d &= \frac{1}{|\mathcal{H}_k^d|} \sum_{j \in \mathcal{H}_k^d} d_j, \\ s_k^d &= \sqrt{\frac{1}{|\mathcal{H}_k^d|} \sum_{j \in \mathcal{H}_k^d} (d_j - m_k^d)^2}, \\ \delta_k &= m_k^d + s_k^d. \end{aligned} \tag{6}$$

When $d_k > \delta_k$, the current deviation exceeds the adaptive upper bound, which is determined by the mean and standard deviation of historical deviations along the same trajectory.

The magnitude of deviation alone cannot distinguish harmful drift from beneficial exploration. Thus, we also evaluate its actual contribution to the generation quality. At step k, the estimated clean sample obtained using Eq. 1 is given by $\widehat{\mathbf{x}}_0^{(k)} \triangleq \widehat{\mathbf{x}}_0(\mathbf{x}_{t_k}, t_k, \mathbf{c})$. Let the image quality be evaluated by a scoring model $\mathcal{Q}(\cdot, \mathbf{c})$. The image quality score at step k and its corresponding quality gain are defined as $q_k = \mathcal{Q}(\widehat{\mathbf{x}}_0^{(k)}, \mathbf{c})$, $\Delta q_k = q_k - q_{k-1}$, $\mathcal{H}_k^q = \{1, \ldots, k-1\}$. When $k > 1$, the mean and

Table 1: **Multi-metric evaluation after GenEval optimization on SD3.5.**

Method	Preference			Fidelity			Diversity			Image Quality			#Top1
	PS↑	IR↑	HPS↑	GenEval↑	CLIP↑	FID↓	C-Dist.↑	IS↑	TCE↑	CLIP-IQA↑	AES↑	TOPIQ-NR↑	
SD3.5	20.555	-0.423	0.259	0.282	0.253	–	0.371	22.784	37.546	0.581	5.024	0.565	0
FlowGRPO	20.529	-0.583	0.255	0.324	0.255	97.580	0.364	21.740	37.482	0.566	4.963	0.526	0
MixGRPO	<u>21.467</u>	<u>0.194</u>	<u>0.270</u>	0.453	0.254	22.612	<u>0.401</u>	<u>28.922</u>	<u>38.425</u>	<u>0.633</u>	<u>5.195</u>	0.602	0
DiffusionNFT	21.119	0.067	0.263	0.487	0.252	<u>14.629</u>	0.393	23.974	37.776	0.617	5.010	0.584	0
FlashGRPO	21.184	-0.087	0.260	<u>0.515</u>	**0.260**	50.385	0.374	17.771	38.036	0.593	4.907	<u>0.624</u>	1
Ours	**22.485**	**0.936**	**0.282**	**0.761**	<u>0.256</u>	**9.233**	**0.412**	**30.527**	**39.517**	**0.668**	**5.281**	**0.721**	**11**

standard deviation of historical quality gains, along with the minimum gain threshold, are

$$\begin{aligned} m_k^q &= \frac{1}{|\mathcal{H}_k^q|} \sum_{j \in \mathcal{H}_k^q} \Delta q_j, \\ s_k^q &= \sqrt{\frac{1}{|\mathcal{H}_k^q|} \sum_{j \in \mathcal{H}_k^q} (\Delta q_j - m_k^q)^2}, \\ \tau_k &= m_k^q - s_k^q. \end{aligned} \tag{7}$$

When $\Delta q_k < \tau_k$, the current quality gain falls below the adaptive lower bound determined by the mean and standard deviation of historical gains along the same trajectory.

Adaptive Weight. To avoid unstable trajectory-level statistics when early samples are insufficient, we delay the activation of adaptive gating. Specifically, when $k \geq K_{\min}$, where $K_{\min}$ is a hyperparameter we construct a deviation gate A_k and a low-gain gate B_k, and multiply them to obtain the adaptive weight. For $k < K_{\min}$, we set $g_k = 0$ as a warm-up strategy:

$$\begin{aligned} A_k &= \max\left(0,\, 2\sigma\left(\frac{d_k - \delta_k}{s_k^d + \varepsilon}\right) - 1\right), \\ B_k &= \max\left(0,\, 2\sigma\left(\frac{\tau_k - \Delta q_k}{s_k^q + \varepsilon}\right) - 1\right), \\ g_k &= A_k B_k. \end{aligned} \tag{8}$$

Here, $\sigma(\cdot)$ denotes the sigmoid function, and ε is a small numerical constant introduced to prevent division by zero. The coefficient $g_k \in [0, 1)$ serves as an adaptive factor to modulate the constraint reward. With g_k incorporated, the final constraint reward is defined as:

$$R_{\text{an}}^k(\mathbf{s}_{t_k}, \mathbf{a}_{t_k}) \triangleq g_k R_{\text{an}}(\mathbf{s}_{t_k}, \mathbf{a}_{t_k}). \tag{9}$$

With adaptive regulation, we enable more fine-grained anti-drift control during fine-tuning. When deviations remain within a normal range, exploration is not additionally suppressed. When deviations are large but bring substantial preference-alignment gains, excessive penalties on beneficial exploration are avoided. In contrast, when deviations are large and quality gains are insufficient, the anti-drift constraint is significantly strengthened. This adaptive mechanism avoids the limitation of KL-as-loss methods that uniformly penalize all deviations regardless of their effectiveness. Thus, it prevents excessive drift from the original distribution while improving preference alignment.

Adaptive Weight. After incorporating the adaptive mechanism, the total reward for RL training is

$$R_{\text{total}}(\mathbf{s}_t, \mathbf{a}_t) = \underbrace{r\left(\mathbf{x}_0, \mathbf{c}\right)}_{\text{Task Reward}} + \lambda \cdot \underbrace{R_{\text{an}}^k(\mathbf{s}_{t_k}, \mathbf{a}_{t_k})}_{\text{Anchoring Reward}}. \tag{10}$$

Since Anchoring Reward adaptively varies across timesteps based on local reference deviation, the final reward is no longer a uniform scalar backfilled to all denoising steps. Unlike standard reward backfilling, where actions share identical feedback, our design provides step-specific rewards, reducing interference in policy optimization caused by assigning the same signal to different actions.

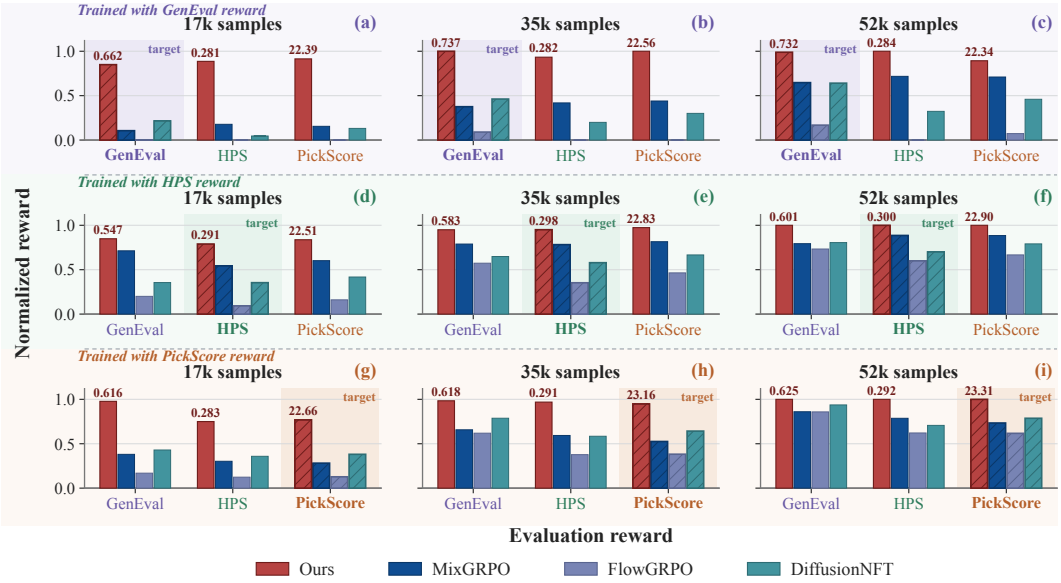


Figure 3: **Target- and cross-reward evaluation on SD3.5.** Models trained with GenEval, HPSv2, or PickScore are evaluated with all three rewards at matched sample budgets. Scores are normalized within each training–evaluation pair.

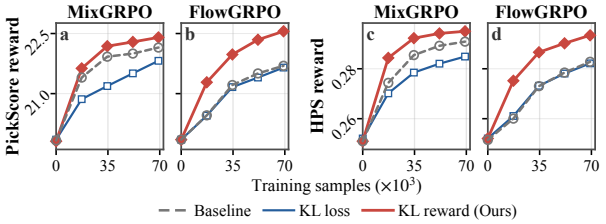


Figure 4: Reward-level anchoring versus KL-as-loss on SD3.5 and Pick-a-Pic dataset.

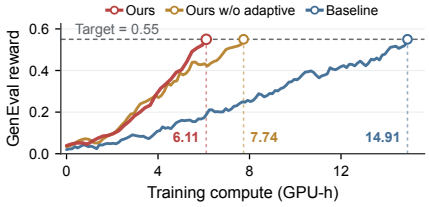


Figure 5: **Computational efficiency on SD3.5 with GenEval.** Compute required to reach the target reward 0.55.

3 Experimental Evaluation

3.1 Experiment Setting

Datasets. We evaluate on four complementary prompt sets: Simple Animals Black et al. (2023); Pick-a-Pic Kirstain et al. (2023) captures diverse human preferences; the photorealistic HPSv2 subset Wu et al. (2023a) emphasizes realistic scenes; and GenEval Ghosh et al. (2023) tests compositional constraints involving objects, counting, colors, attributes, and spatial relations.

Rewards and Metrics. During training, we optimize Aesthetic Score Schuhmann et al. (2022), PickScore Kirstain et al. (2023), HPSv2 Wu et al. (2023a), and GenEval Ghosh et al. (2023). At evaluation, we report metrics along four axes aligned with Table 1. **Preference:** PickScore (PS), ImageReward (IR) Xu et al. (2023), and HPSv2 (HPS). **Fidelity:** GenEval, CLIPScore (CLIP) Hessel et al. (2021), and Fréchet Inception Distance (FID). **Diversity:** mean pairwise cosine distance between CLIP image embeddings (C-Dist.) Radford et al., Inception Score (IS) Salimans et al. (2016), and Truncated CLIP Entropy (TCE) Ibarrola & Grace (2024). **Image quality:** CLIP-IQA Wang et al. (2023), Aesthetic Score (AES), and TOPIQ-NR Chen et al. (2024a). Higher values indicate better performance except for FID, where lower is better.

Models. We evaluate StepKL on U-Net diffusion backbones SD1.4, SD1.5, and SD2.1 Rombach et al. (2022), as well as flow-matching backbones SD3.5 Esser et al. (2024) and FLUX.1-dev Black Forest Labs (2024), covering diverse model scales and generative parameterizations.

Table 2: **Multi-reward scores at 35k samples.**

Model	Anti-drift	Gate g_k	Gen Eval	PS	HPS	Aes.	CLIP
Baseline	–	–	0.427	0.824	0.268	5.171	0.861
V1	✓	–	0.694	0.846	0.270	4.996	0.935
V2	✓	shuffle	0.543	0.830	0.272	5.110	0.886
Ours	✓	✓	**0.737**	**0.868**	**0.282**	**5.279**	**0.939**

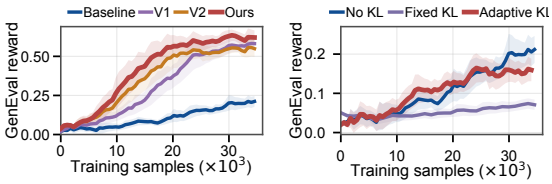


Figure 6: **Ablation curves on SD3.5–GenEval.** Left: adaptive anti-drift reward. Right: adaptive loss-level KL.

3.2 Experiments on Pre-defined RL Objectives

We first evaluate direct reward optimization across flow-matching backbones and preference objectives. Figure 2 reports training and held-out evaluation on Pick-a-Pic using SD3.5 and FLUX.1-dev, with HPSv2 or PickScore as the training objective. StepKL achieves the highest held-out target reward in all settings, outperforming MixGRPO, FlowGRPO, and DiffusionNFT. These results demonstrate consistent effectiveness across flow-matching backbones and preference rewards. Figure 4 further isolates the reference-constraint formulation. Across both MixGRPO and FlowGRPO, reward-level anchoring consistently outperforms KL-as-loss and the unregularized baseline on held-out PickScore and HPSv2. Figure 3 shows that StepKL consistently improves both target and cross-reward performance, demonstrating robust transfer beyond the optimized objective.

3.3 Reward Hacking Analysis

A higher target reward does not necessarily indicate better generation quality, because a model may increase the optimized score while drifting from the pretrained distribution. We therefore analyze reward hacking from three complementary perspectives: feature-distribution preservation, qualitative behavior at comparable OCR rewards, and improvements in independent quality dimensions.

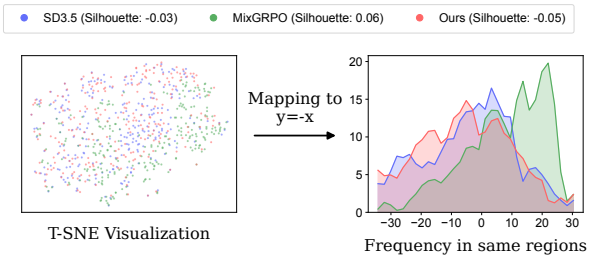


Figure 7: **Feature-distribution analysis on SD3.5.** t-SNE embeddings (left) and distributions after projection onto $y = -x$ (right); silhouette scores are reported in the legend.

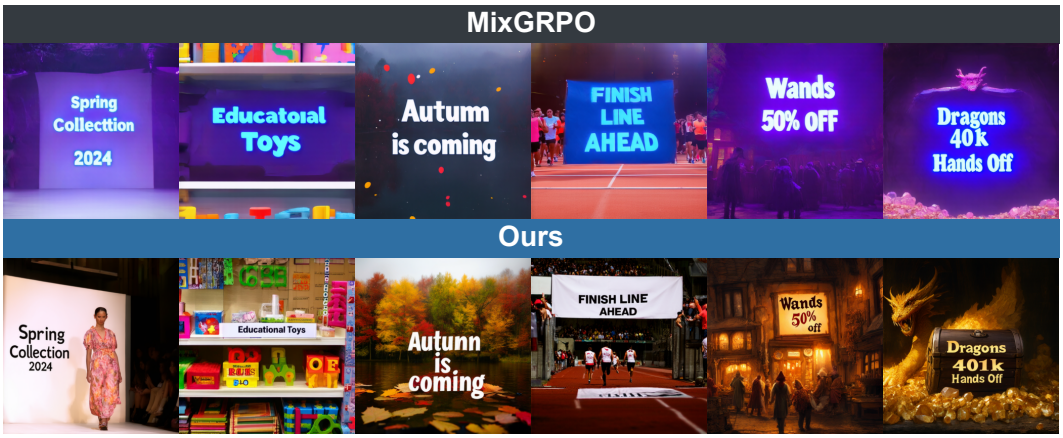
Feature-Distribution Preservation. Figure 7 compares the geometry of image features after fine-tuning. StepKL largely overlaps with pretrained SD3.5, whereas MixGRPO concentrates in a narrower projected region, indicating less feature-space contraction under StepKL.

Matched-Reward Qualitative Comparison. At comparable online OCR rewards, MixGRPO enlarges the target text at the expense of scene content, whereas StepKL preserves both text legibility and scene structure (Figure 8). This contrast indicates less severe OCR reward hacking with StepKL.

Multi-Metric Evaluation. We evaluate models with metrics covering preference, fidelity, diversity, and image quality. Under matched budgets, StepKL ranks first on 11 metrics spanning all evaluation dimensions (Table 1). It also preserves scene content and visual diversity while satisfying compositional constraints (Figure 20), indicating gains without mode collapse or reward hacking.

3.4 Adaptiveness to Varying Exploration

In flow-matching diffusion RL, higher SDE noise enhances exploration but increases distributional drift and instability. We compare StepKL with KL-as-loss at low, medium, and high noise levels. As shown in Figure 9, at low and medium noise levels, StepKL reaches comparable rewards with substantially fewer samples and benefits more from stronger exploration. Thus, under the same regularization strength, it converts trajectory stochasticity into efficient reward learning more effectively than KL-as-loss. Under high noise, KL-as-loss becomes unstable and collapses, whereas StepKL suppresses harmful deviations while retaining useful exploration. This adaptive trajectory control enables effective training under over-exploration and broadens the usable exploration range.



Method	OCR Reward ↑	OO Epoch ↑	Scene
MixGRPO	0.6839	120	Lost
Ours	**0.7679**	**> 180**	**Preserved**

Figure 8: **OCR reward-hacking comparison.** (top) Matched-reward samples. (bottom) Online OCR reward and the first over-optimization (OO) epoch; $>$ 180 denotes no observed over-optimization during training.

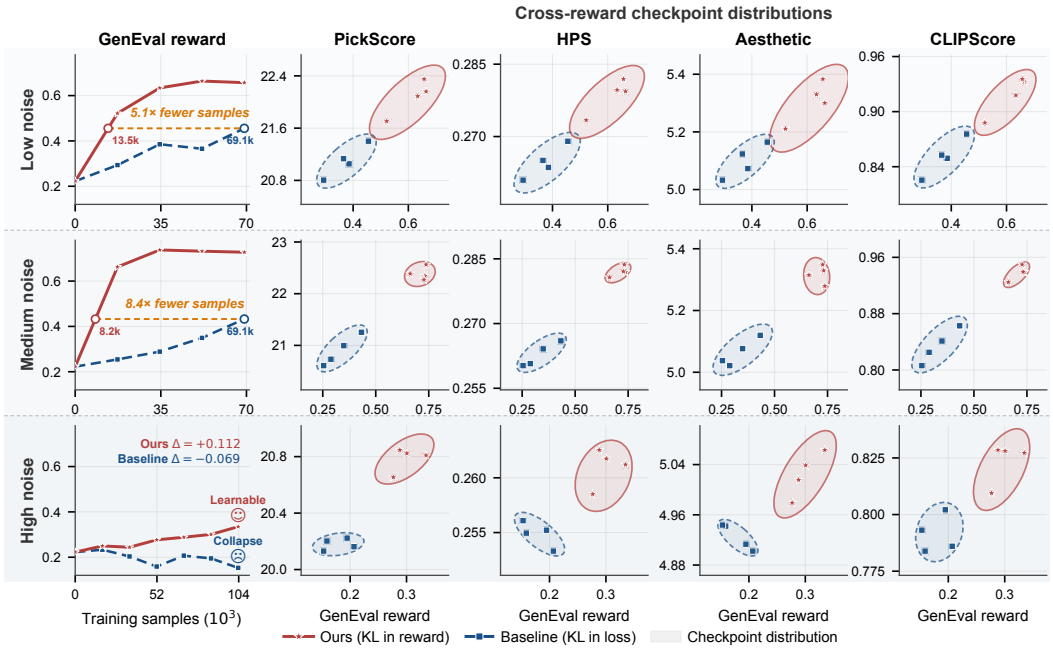


Figure 9: **Adaptiveness to varying exploration.** Rows show results with different noise levels. The first column reports GenEval against training samples; annotations give the samples used by StepKL to match the final baseline score. Other columns compare matched checkpoints across GenEval and independent metrics; shaded regions trace checkpoints rather than confidence intervals.

Moreover, the higher target rewards of our method are accompanied by gains in all dimensions. Its advantage therefore reflects comprehensive quality improvements rather than reward hacking.

3.5 Plug-and-Play Integration Experiments

As shown in Figure 10, we apply kl-as-reward to both MixGRPO and FlowGRPO while optimizing either PickScore or HPSv2. Under the same training-sample budget, kl-as-reward consistently outperforms both no KL and KL-as-loss.

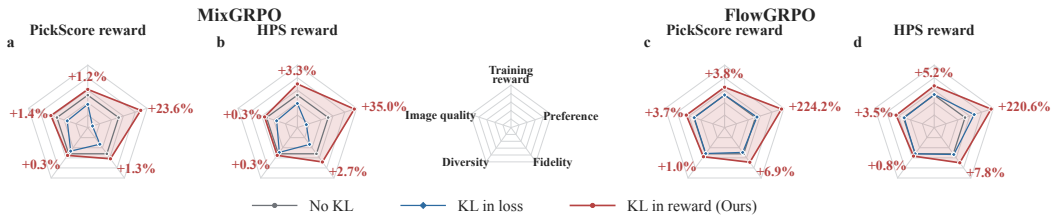


Figure 10: **Reward-in-KL as a general plug-in.** At a matched budget of 17,280 generated training samples, we compare no KL, KL in loss, and KL in reward with MixGRPO and FlowGRPO under PickScore or HPSv2 optimization. The five spokes report the optimized reward, preference (ImageReward), fidelity (MUSIQ), diversity (TCE), and image quality (Aesthetic Score). Radial distances encode signed log-scaled changes relative to no KL within each setting; red annotations show the exact relative gains of KL in reward.

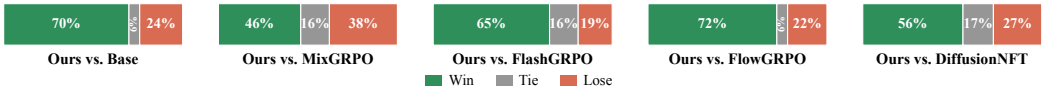


Figure 11: **Blind VLM preference evaluation on GenEval (SD3.5).**

3.6 Computation Efficiency

We compare the training compute required to reach a shared reward target under identical hardware, excluding startup and evaluation. Figure 5 shows that StepKL requires substantially less compute than the non-adaptive variant and FlowGRPO, demonstrating more efficient reward learning.

3.7 Ablation Study

Adaptive Anti-Drift Reward. Figure 6(Left) isolates the effects of the anti-drift reward and its adaptive gate. The ungated variant verifies the benefit of introducing step-wise anti-drift feedback, while the shuffled-gate variant preserves the gate-value distribution but breaks its correspondence with the deviation and quality signals. StepKL consistently performs best across reported metrics. The results show that effective anti-drift control requires step-wise feedback, but also assigning the appropriate constraint strength to each denoising step according to its training dynamics.

Adaptive Loss-Level KL. Figure 6(Right) applies the same adaptive weighting strategy to conventional KL regularization. Under matched average regularization strength, adaptive KL consistently outperforms fixed KL. This result demonstrates that allocating regularization according to sample- and timestep-specific signals is more effective than applying a uniform penalty, further validating the general effectiveness of the proposed adaptive weighting strategy.

3.8 VLM Preference Evaluation

To further assess generation quality from a preference perspective, we conduct pairwise comparisons on the GenEval prompt set with SD3.5, using checkpoints trained under matched sample counts. A vision-language model (VLM) serves as the judge and compares StepKL against each baseline for every prompt under a blind setting. In Figure 11, StepKL is preferred over every baseline and shows a clear advantage. This confirms that an independent VLM judge consistently recognizes StepKL's improvements, rather than limiting them to training rewards or individual automatic metrics.

4 Conclusion

We presented StepKL, a reward-level adaptive anchoring approach for diffusion RL. By transforming KL regularization into step-wise rewards, StepKL distinguishes beneficial exploration from distributional drift. Extensive experiments demonstrate superior performance, robustness, and efficiency, establishing reward-level anchoring as a compelling alternative to KL-as-loss.

AI USE STATEMENT

In this work, generative AI tools were mainly utilized to assist with language polishing.

REFERENCES

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

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

Black Forest Labs. FLUX. `https://blackforestlabs.ai/announcing-black-forest-labs/`, 2024.

Yuanjiang Cao, Quan Z Sheng, Julian McAuley, and Lina Yao. Reinforcement learning for generative ai: A survey. *arXiv preprint arXiv:2308.14328*, 2023.

Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. TOPIQ: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33:2404–2418, 2024a.

Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. Opportunities and challenges of diffusion models for generative ai. *National Science Review*, 11(12):nwae348, 2024b.

Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. *arXiv preprint arXiv:2404.07771*, 2024c.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Fei Deng, Qifei Wang, Wei Wei, Tingbo Hou, and Matthias Grundmann. Prdp: Proximal reward difference prediction for large-scale reward finetuning of diffusion models. In *Computer Vision and Pattern Recognition*, pp. 7423–7433, 2024.

Xin Ding, Lei Yu, Xin Li, Zhijun Tu, Hanting Chen, Jie Hu, and Zhibo Chen. Rass: Improving denoising diffusion samplers with reinforced active sampling scheduler. In *Computer Vision and Pattern Recognition Conference*, pp. 12923–12933, 2025.

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

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

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Giorgio Franceschelli and Mirco Musolesi. Reinforcement learning for generative ai: State of the art, opportunities and open research challenges. *Journal of Artificial Intelligence Research*, 79: 417–446, 2024.

Yaozhong Gan, Renye Yan, Zhe Wu, and Junliang Xing. Reflective policy optimization. *arXiv preprint arXiv:2406.03678*, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Zhaolin Gao, Jonathan Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, Drew Bagnell, Jason D Lee, and Wen Sun. Rebel: Reinforcement learning via regressing relative rewards. *Advances in Neural Information Processing Systems*, 37: 52354–52400, 2024.

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

QIANYUE HAO, XIAOQIAN QI, YUAN YUAN, ZEFANG ZONG, HONGYI CHEN, KEYU ZHAO, SHENGYUAN WANG, YUNKE ZHANG, JIAN YUAN, and YONG LI. Reinforcement learning in the era of large language models: Challenges and opportunities. 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, 2021.

Francisco Ibarrola and Kazjon Grace. Measuring diversity in co-creative image generation. *arXiv preprint arXiv:2403.13826*, 2024.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in Neural Information Processing Systems*, 31, 2018.

Dongzhi Jiang, Guanglu Song, Xiaoshi Wu, Renrui Zhang, Dazhong Shen, Zhuofan Zong, Yu Liu, and Hongsheng Li. Comat: Aligning text-to-image diffusion model with image-to-text concept matching. *Advances in Neural Information Processing Systems*, 37:76177–76209, 2024.

Shyamgopal Karthik, Huseyin Coskun, Zeynep Akata, Sergey Tulyakov, Jian Ren, and Anil Kag. Scalable ranked preference optimization for text-to-image generation. *arXiv preprint arXiv:2410.18013*, 2024.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. *arXiv preprint arXiv:2501.05803*, 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*, 2023.

Kyungmin Lee, Sangkyung Kwak, Kihyuk Sohn, and Jinwoo Shin. Direct consistency optimization for robust customization of text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 37:103269–103304, 2024.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *Advances in Neural Information Processing Systems*, 37:24897–24925, 2024.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024.

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.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PmLR.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sushil Sikchi, Joey Hejna, Brad Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *Advances in Neural Information Processing Systems*, 37:126207–126242, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29, 2016.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 2022.

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

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Inference-time alignment of diffusion models with direct noise optimization. *arXiv preprint arXiv:2405.18881*, 2024.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

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 *Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Jianyi Wang, Kelvin C. K. Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2555–2563, 2023.

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*, 2023a.

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*, 2023b.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *International Conference on Computer Vision*, pp. 2096–2105, 2023c.

Xin Xie and Dong Gong. Dymo: Training-free diffusion model alignment with dynamic multi-objective scheduling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13220–13230, 2025.

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

Renye Yan, Jikang Cheng, Shikun Sun, Yi Sun, You Wu, Wei Peng, Zongwei Wang, Ling Liang, Junliang Xing, and Yimao Cai. Do less, achieve more: Do we need every-step optimization for rl fine-tuning of diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16561–16571, 2026.

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 *Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024a.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024b.

Sixian Zhang, Bohan Wang, Junqiang Wu, Yan Li, Tingting Gao, Di Zhang, and Zhongyuan Wang. Learning multi-dimensional human preference for text-to-image generation. In *Computer Vision and Pattern Recognition*, pp. 8018–8027, 2024a.

Ziyi Zhang, Sen Zhang, Yibing Zhan, Yong Luo, Yonggang Wen, and Dacheng Tao. Confronting reward overoptimization for diffusion models: A perspective of inductive and primacy biases. In *International Conference on Machine Learning*, pp. 60396–60413. PMLR, 2024b.

A Related Works

A.1 RL in Diffusion Model

Diffusion models Chen et al. (2024b); Jiang et al. (2024) have demonstrated outstanding performance in text-to-image generation tasks and have become the mainstream paradigm for high-quality image synthesis Ding et al. (2025); Betker et al. (2023); Clark et al. (2023); Lin et al. (2024); Esser et al. (2024); Lee et al. (2024); Li et al. (2024). However, the original training objective of diffusion models mainly focuses on accurately recovering the data distribution and lacks the ability to directly align with specific preferences (such as style, aesthetics, or semantic orientation). When generating images that satisfy specific criteria, the generalization capability of pretrained models is often limited Yang et al. (2024b); Ma et al. (2025); Gao et al. (2024). Reinforcement learning provides an effective solution for preference optimization Franceschelli & Musolesi (2024); Wallace et al. (2024); Clark et al. (2023); Chen et al. (2024c); Fan et al. (2024; 2023) by explicitly modeling preferences through reward functions. Specifically, after introducing reinforcement learning-based fine-tuning, the denoising process of diffusion models is redefined as a multi-step decision sequence: the noise prediction at each denoising step is regarded as an action selection in reinforcement learning. After completing the entire denoising process, the quality of the final generated image or its score regarding predefined preferences evaluated by an assessment model is used as a reward signal for policy optimization. Through this approach, the model can make preference-oriented decisions during training, progressively guiding the generation process from the initial noise to the final high-quality image, and optimizing high-level preference objectives such as image style and semantics Karthik et al. (2024); Li et al. (2022); Wu et al. (2023c;b); Zhang et al. (2024a).

A.2 Sparse Reward and Reward Hacking

Sparse Reward: In traditional reinforcement learning Gan et al. (2024), sparse rewards refer to the situation where feedback is only provided when the final objective is achieved Wallace et al. (2024); Yang et al. (2024a); Xie & Gong (2025); Kim et al. (2025); Tang et al. (2024). This makes it difficult for the model to adjust the behaviors of intermediate steps during training, thereby affecting optimization efficiency and stability. This issue also exists in the fine-tuning of diffusion models. Since the denoising process is a multi-step sequence, the prediction at each step directly affects the final result. However, reinforcement learning rewards are usually only provided after the image generation process is completed, causing intermediate denoising steps to lack reward signals. This not only reduces training stability but also increases the difficulty of model optimization. To address this issue, existing methods generally adopt a simple compromise: directly backfilling the reward obtained from the final image equally to all preceding denoising steps Black et al. (2023); Franceschelli & Musolesi (2024); Chen et al. (2024c); Uehara et al. (2024). Although this approach enables reinforcement learning training to proceed normally, it inevitably eliminates the contribution differences among intermediate steps, resulting in reward attribution mismatch. Moreover, this mismatch is gradually amplified as the distance from the final image increases. This makes it difficult for the model to establish a stable action–outcome correspondence, thereby weakening the optimization effectiveness and causing reward hacking Ibarz et al. (2018); Skalse et al. (2022); Fan et al. (2023); Gao et al. (2023); Rafailov et al. (2024); Franceschelli & Musolesi (2024); HAO et al. (2024); Cao et al. (2023). In contrast, StepKL introduces fine-grained feedback based on the noise discrepancy between the pre-trained and post-trained models along the denoising trajectory, enabling each step to receive distinct reward signals and significantly alleviating reward attribution mismatch.

Reward Hacking: Reward hacking is an imbalance phenomenon in reinforcement learning fine-tuning of diffusion models. It manifests as the model deviating from the generation patterns and data distribution while pursuing higher reward scores. For example, in human preference optimization tasks, the model may satisfy the reward model by exaggerating colors or applying excessive sharpening, rather than genuinely improving perceptual quality. The root cause lies in the mismatch between the reward signals and the structure of the diffusion process. Reinforcement learning rewards are usually calculated only after the final image generation, while diffusion generation is a long-horizon, multi-step decision-making process. When reinforcement signals are gradually propagated back along the denoising trajectory, the model has difficulty accurately determining which intermediate predictions truly contribute to the final quality, and may therefore learn spurious correlated features unrelated to the preference. The consequence of this mismatch is that the model

gradually converges toward the "false high-reward regions" preferred by the reward model, resulting in degradation phenomena such as semantic drift and structural anomalies. Existing methods usually suppress excessive deviation through KL constraints or distribution regularization. However, these global constraints cannot timely correct local errors in intermediate denoising steps, making deviation accumulation more likely. In contrast, StepKL progressively applies penalty rewards constructed from the predicted noise discrepancy between the pre-trained and post-trained models during the diffusion dynamics, enabling the model to adaptively correct distribution shifts at each step and thereby avoiding the amplification of reward hacking.

B Additional Experiments

B.1 Detailed Implementation

Table 3 lists the core hyperparameters of the flow-matching experiments in this paper. Table 4 lists the core hyperparameters of the UNet experiments in the Appendix.

Table 3: Hyper-parameters used in our flow-matching experiments.

Name	Description	Value (SD3.5)	Value (FLUX)
Input resolution	Generated image resolution	512×512	512×512
Mixed precision	Numerical precision used during training	FP16	BF16
Batch size	Training batch size per GPU	9	3
Gradient accumulation	Number of gradient accumulation steps	8	6
GRPO group size K	Number of images sampled for each prompt	18	24
Denoising steps	Number of denoising steps during training	10	6
Guidance scale	Classifier-free guidance scale	1.0	3.5
SDE noise level	Noise scale used for stochastic exploration	0.8	0.8
Optimizer	Optimization algorithm	AdamW	AdamW
Learning rate	Initial learning rate	3×10^{-4}	3×10^{-4}
Weight decay	Weight decay coefficient	1×10^{-4}	1×10^{-4}
LoRA rank	Rank of the LoRA adapters	32	64

B.2 Experiments on UNet-based Diffusion

The main paper has validated the effectiveness of StepKL on two flow-matching models, SD3.5 and FLUX.1-dev. To examine the generalization capability of the method across different types of generative models, this section further provides additional experiments on the U-Net-based model. We select three representative U-Net reinforcement learning fine-tuning methods, DDPO Black et al. (2023), DPOK Fan et al. (2023), and D3PO Yang et al. (2024a), as baselines, and investigate the generalization capability of StepKL from three dimensions: backbone networks, prompt distributions, and reward types.

Backbone Generalization. Figure 12 compares three backbones, SD1.4, SD1.5, and SD2.1, under the Simple Animals prompt set and the Aesthetic reward. Although the three models exhibit significant differences in model capacity and pre-training distributions, StepKL consistently achieves the highest target reward under the same training sample budget, demonstrating that its benefits are not tied to a specific U-Net scale or pre-training distribution.

Prompt Set Generalization. Figure 14 fixes SD1.5 and the Aesthetic reward, and replaces the prompt set with Pick-a-Pic and HPSv2, which are closer to the real user distribution. On both prompt sets, StepKL consistently outperforms all baselines, demonstrating that the method is insensitive to the semantic distribution of training prompts.

Reward Generalization. Figure 15 further investigates the optimization of underlying image attribute objectives unrelated to semantic preferences: the compression and incompressibility rewards with opposite optimization directions on SD1.4. StepKL achieves the best performance on both opposite objectives, demonstrating that its adaptive anti-drift mechanism is reward-agnostic and is not a specialized design for a specific type of preference model.

Table 4: Hyper-parameters used in our U-Net (DDIM) experiments.

Name	Description	Value
Input resolution	Generated image resolution	512×512
Mixed precision	Numerical precision used during training	FP16
Sample batch size	Sampling batch size per GPU	16
Batch size	Training batch size per GPU	2
Gradient accumulation	Number of gradient accumulation steps	8
Denoising steps	Number of denoising steps during training	50
Guidance scale	Classifier-free guidance scale	5.0
DDIM η	Stochasticity of the DDIM sampler	1.0
PPO clip range	Clipping range of the policy probability ratio	1×10^{-4}
Optimizer	Optimization algorithm	AdamW
Learning rate	Initial learning rate	3×10^{-4}
Weight decay	Weight decay coefficient	1×10^{-4}
LoRA rank	Rank of the LoRA adapters	4

In summary, combined with the flow-matching experiments in the main paper, StepKL consistently achieves improvements across two generation paradigms, five backbones (SD1.4/1.5/2.1, SD3.5, and FLUX.1-dev), four prompt sets, and various rewards covering human preferences, compositional constraints, and underlying image attributes. This demonstrates that StepKL can serve as a general diffusion model reinforcement learning fine-tuning framework that is decoupled from both backbones and rewards.

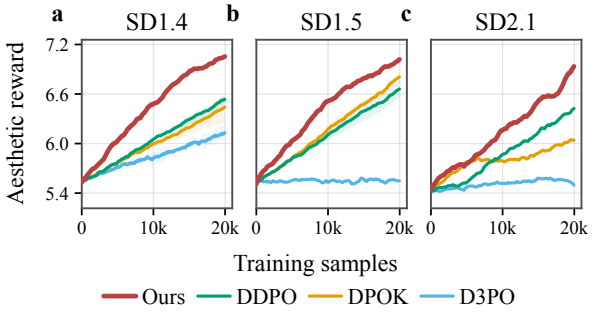


Figure 12: **Backbone generalization on Simple Animals with Aesthetic reward.** Training reward curves for (a) SD1.4, (b) SD1.5, and (c) SD2.1 under matched sample budgets. StepKL consistently achieves the highest Aesthetic reward across all three U-Net backbones, demonstrating that its effectiveness is not tied to a specific model scale or pre-training distribution.

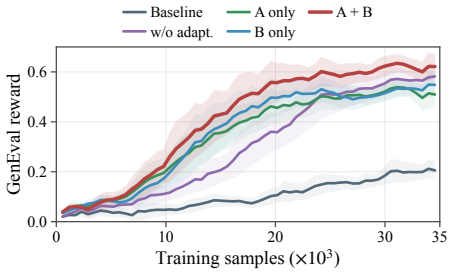


Figure 13: **Ablation of the Adaptive Gating Components.** A denotes the deviation gate, which is used to identify prediction deviations that abnormally increase relative to the historical trajectory level; B denotes the low-gain gate, which is used to identify denoising steps that do not bring sufficient quality improvements.

B.3 Computational Cost and Optimization Efficiency

As shown in Figure 17, compared with MixGRPO, non-adaptive StepKL requires additional forward computations of the frozen reference policy on the sampled intermediate states, and constructs self-correction rewards based on the prediction discrepancy between the reference policy and the current policy, thereby increasing the computational cost during training. Furthermore, the adaptive module additionally reconstructs and decodes the intermediate states, uses the reward model to evaluate their quality, and calculates adaptive weights at each timestep, which also introduces certain computational overhead.

Although the designs increase the computational cost per training iteration, the more fine-grained reward signals significantly reduce the number of training iterations required to achieve the same reward level. Meanwhile, the adaptive mechanism further distinguishes the optimization value of

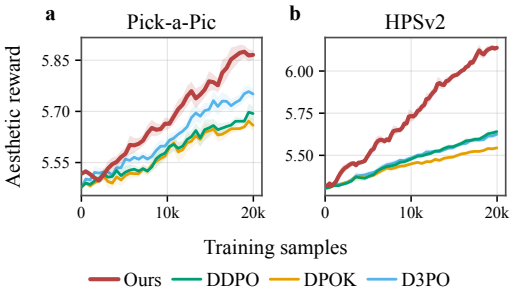


Figure 14: **Prompt-set generalization on SD1.5 with Aesthetic reward.** We replace the prompt distribution with (a) Pick-a-Pic and (b) HPSv2, which are closer to real user queries. StepKL outperforms all baselines on both sets, demonstrating robustness to the semantic distribution of training prompts.

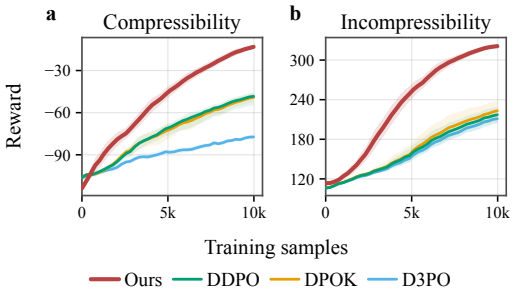


Figure 15: **Reward generalization on SD1.4 with opposing optimization objectives.** (a) Compressibility and (b) Incompressibility represent structurally opposite image-attribute rewards. StepKL achieves the best performance on both, confirming that its adaptive anchoring mechanism is reward-agnostic and not specialized for a particular preference model.

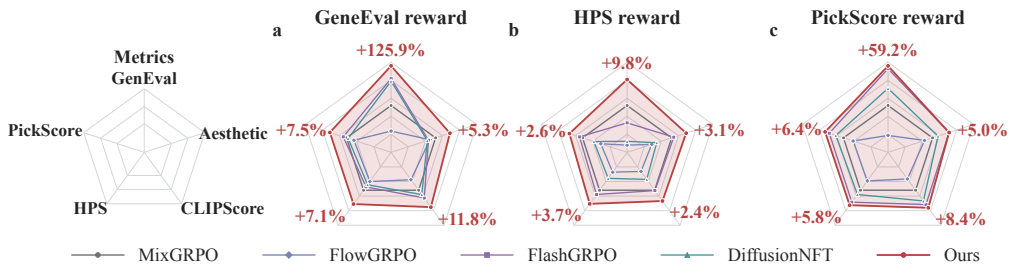


Figure 16: **Efficient reward optimization and cross-metric generalization under the same training sample budget.** On SD3.5, we train using the GenEval prompt set and align the evaluation according to the cumulative number of generated training samples. Evaluation is conducted when the number of generated training samples reaches 17,280. Subfigures **a**–**c** present the evaluation results using GenEval as the prompt set and GenEval, HPSv2, and PickScore as the training rewards, respectively. The radial distance of the radar chart represents the logarithmic-scale variation relative to MixGRPO, where a farther radial position indicates a higher relative score; the red percentages denote the improvement of StepKL over MixGRPO. StepKL consistently improves both the target metrics and non-target metrics under all three reward optimization settings, demonstrating that under the same training budget, StepKL can more efficiently learn generation policies that achieve target reward optimization while maintaining non-target quality. Its performance gains do not merely result from over-optimization of a single reward, but instead reflect more stable cross-metric generalization and comprehensive improvement in generation quality.

different denoising steps, making the training more focused on effective updates. Therefore, the additional computational overhead per iteration is offset by faster reward convergence, ultimately resulting in higher optimization efficiency and achieving the goal of alleviating reward hacking.

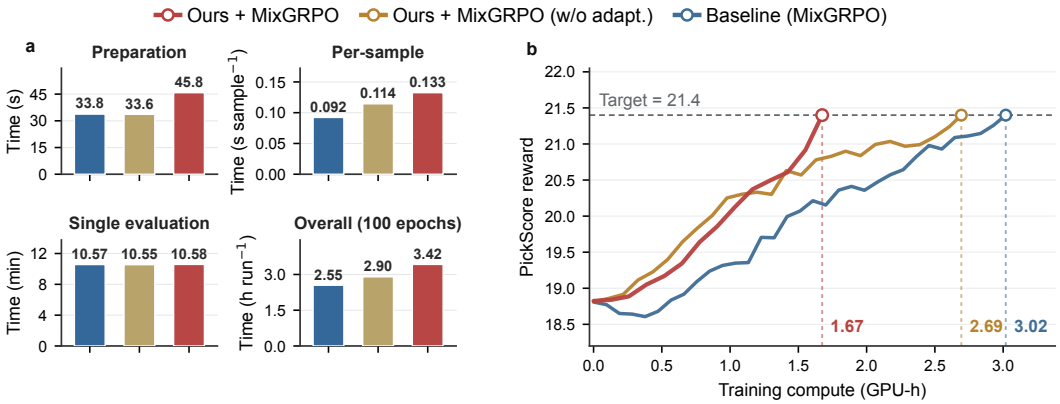


Figure 17: Computational Cost and Optimization Efficiency of SD3.5 on the Pick-a-Pic Dataset with PickScore as the Reward. The introduction of ***our method significantly improves the efficiency of reward acquisition***.

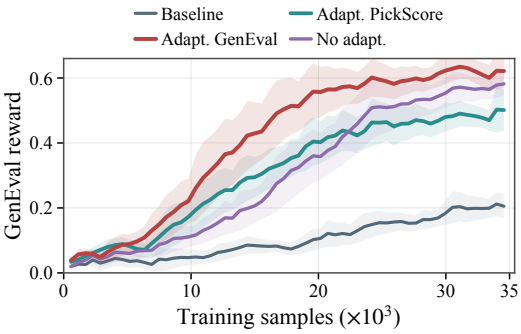


Figure 18: **Ablation Experiment of the Adaptive Intermediate Reward Model.** All variants are trained on SD3.5 using GenEval as the external reward. Adapt. GenEval uses an adaptive internal reward consistent with the target reward, while Adapt. PickScore replaces the intermediate-step reward model with PickScore.

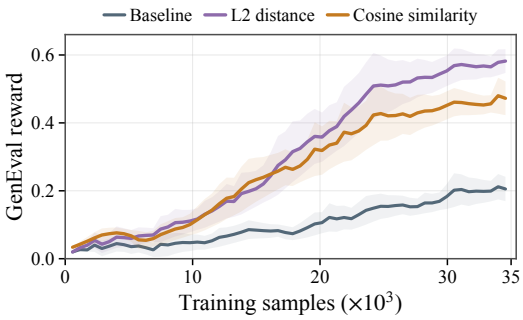


Figure 19: **Impact of Deviation Measurement Methods.** Comparing the effects of L_2 distance and cosine similarity on the step-wise anti-drift reward during training on SD3.5 with GenEval as the target reward.

B.4 More Visual Results

B.4.1 Effect of Reward Hacking.

We further analyze the capability of StepKL in alleviating reward hacking. As shown in Figure 21 22, in the animal generation task, DPOK (***kl-as-loss***) tends to suffer from issues such as category confusion, abnormal subject structures, and semantic inconsistency. In the OCR reward optimization task, MixGRPO (***kl-as-loss***) tends to repeatedly generate highly similar neon text patterns while ignoring the specific scenes required by the prompts. In contrast, StepKL (***kl-as-reward***) can better preserve the subject category, structural integrity, and scene semantics while achieving comparable rewards.

B.4.2 Effect of Reward Learning.

StepKL not only alleviates reward hacking but also significantly improves the efficiency of reward acquisition without sacrificing image diversity. As shown in Figure 20, under the same training sample budget, the images generated by our method better conform to the prompt descriptions. Figure 16 further demonstrates that, under the same training sample budget, regardless of which

reward is used for optimization, StepKL not only significantly improves the corresponding target metrics but also consistently achieves gains on non-target metrics. We further examine whether reward-level KL regularization can serve as a general plug-in across optimization algorithms and reward objectives.

B.5 More Ablation Experiments

B.5.1 Impact of the Adaptive Reward Model.

As shown in Figure 18, using PickScore for adaptive intermediate-step reward computation can also bring faster performance improvements in the early stage of training, but it is inferior to the version without the adaptive mechanism in the later stage. In contrast, using GenEval for the corresponding adaptive computation achieves better overall performance. This indicates that the adaptive reward model needs to be consistent with the evaluation dimensions of the target task in order to accurately distinguish the optimization value of different denoising steps and continuously guide effective updates in the later stage of training.

B.5.2 Ablation of Velocity Difference Calculation Methods.

As shown in Figure 19, we further compare the effects of L2 distance and cosine distance in the anchoring reward. Both distance metrics outperform the baseline method, indicating that explicitly characterizing the deviation between the current policy and the reference policy helps improve generation quality. It should be noted that, to more intuitively demonstrate the impact of the distance metric, we consistently remove the influence of the adaptive module.

B.5.3 Ablation of the Adaptive Mechanism.

As shown in Figure 13, we further analyze the roles of the two components in the adaptive gating factor $g_k = A_k B_k$. Removing the deviation gate A_k makes the model more susceptible to harmful policy drift, while removing the quality gain gate B_k reduces the model's ability to retain effective exploration. The complete dual-gating design achieves a better balance between reward improvement and training stability, validating the complementarity of the two types of gating signals.

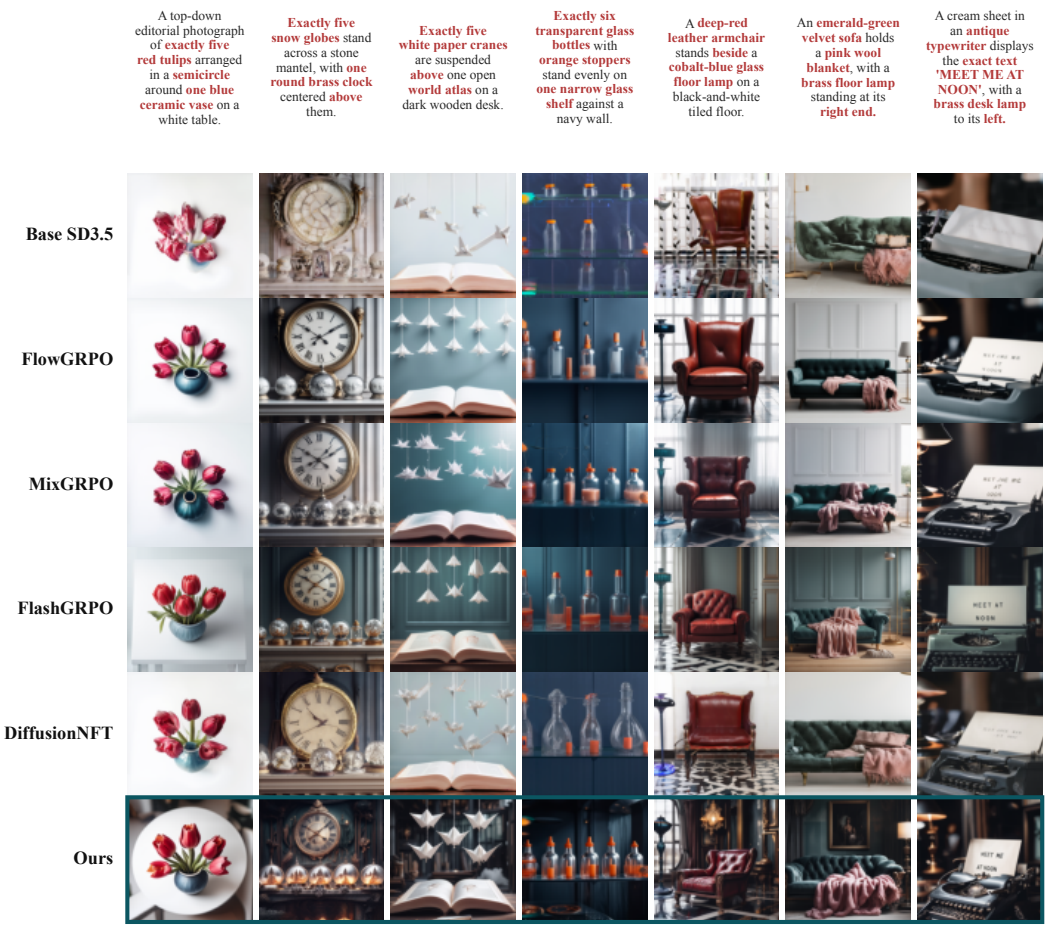


Figure 20: **Qualitative Comparison of Different Methods on Complex Compositional Prompts.** Each column examines fine-grained constraints, including object quantity, color, material, and compositional relationships, while each row presents the generation results of the base SD3.5, FlowGRPO, MixGRPO, FlashGRPO, DiffusionNFT, and StepKL.

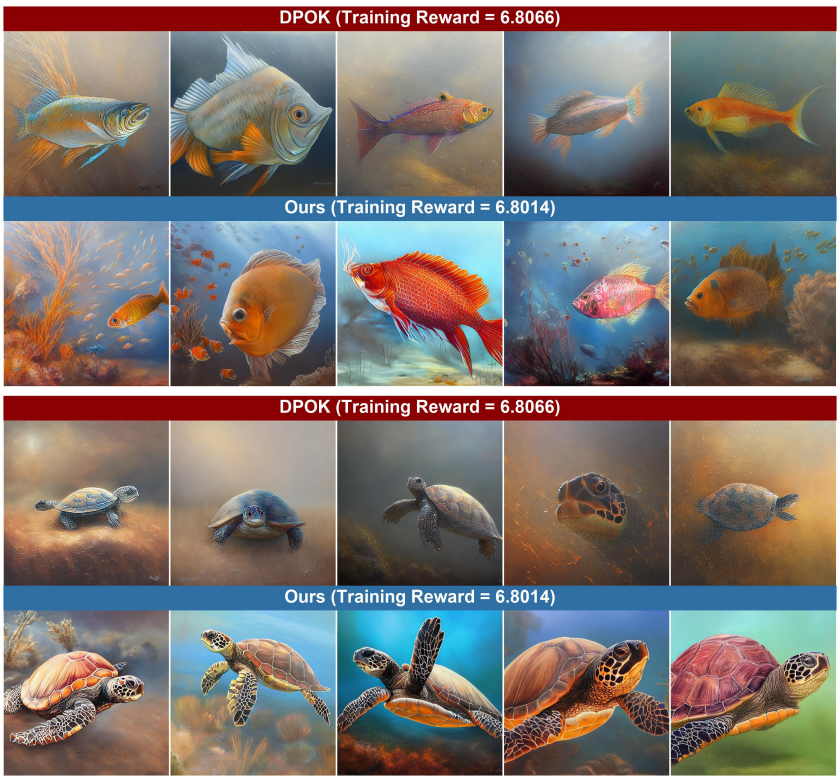


Figure 21: **Comparison of Generation Results between DPOK and StepKL under the Aesthetic Reward.** Despite having nearly identical training rewards, DPOK (kl-as-loss) is more likely to produce abnormal subject structures or weakened underwater scene semantics. Meanwhile, DPOK exhibits typical reward hacking, where turtle generations tend to converge to similar background patterns (yellow grassland). In contrast, StepKL more stably preserves complete animal structures and rich scene content.

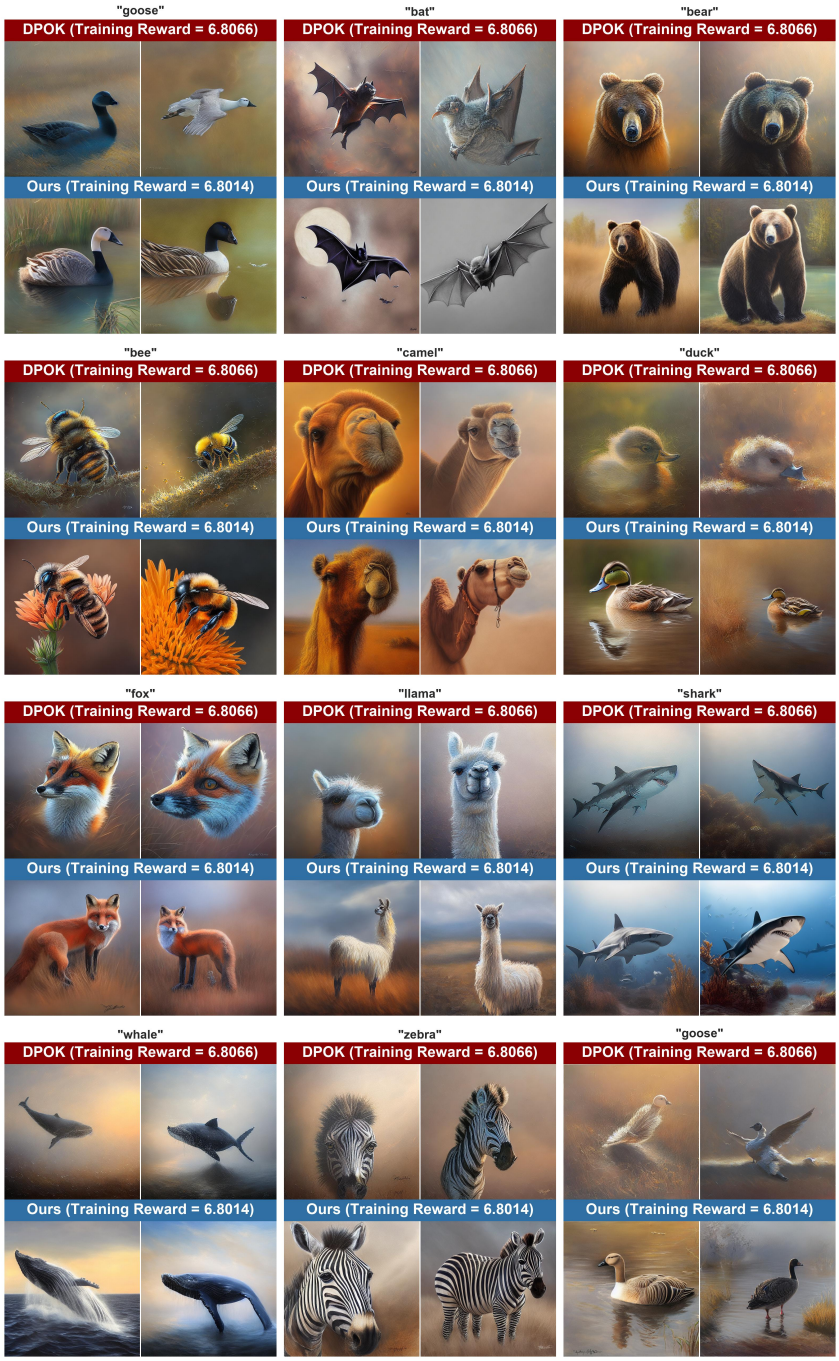


Figure 22: **Comparison of Generation Results between DPOK and StepKL under the Aesthetic Reward.** DPOK is more likely to generate animal images with category confusion, incomplete structures, background distortion, or only prominent local features. In contrast, StepKL can more stably preserve the animal category specified by the prompt, background details, and complete subject structures, indicating that it can effectively alleviate over-optimization of the reward model.

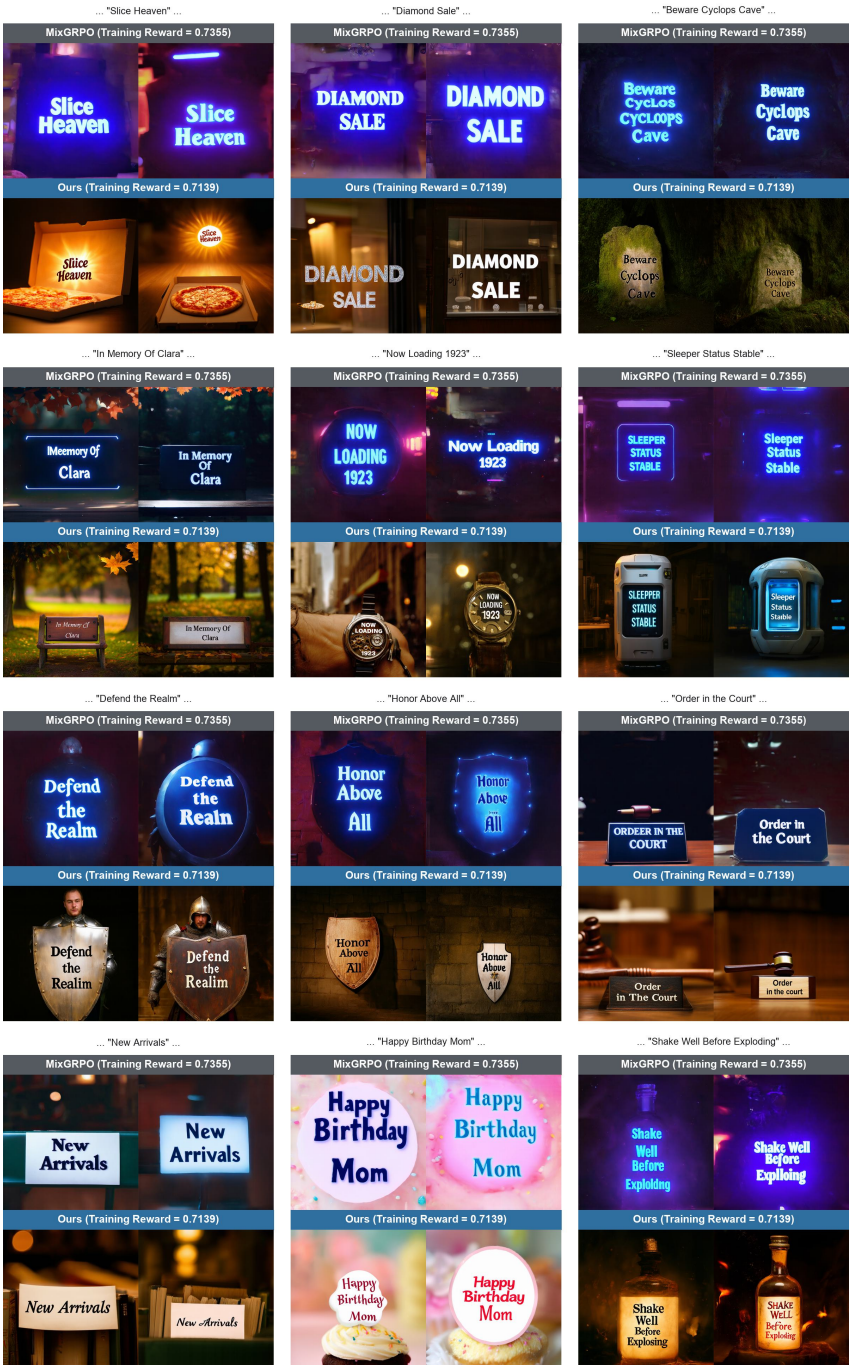


Figure 23: **Comparison of Generation Results between MixGRPO and StepKL under the OCR Reward.** The results of MixGRPO tend to directly present the target text as high-contrast signs, weakening the scene semantics specified in the prompt. In contrast, StepKL can naturally integrate the text into the corresponding objects and scenes, demonstrating that it effectively alleviates excessive reward hacking of the reward model.