EXPLORATION–EXPLOITATION DILEMMA INTRODUCED BY DIFFUSION RL FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite advances in image generation, diffusion models still struggle to align directly with specific preferences, while reinforcement learning (RL) offers an effective solution but introduces challenges in balancing exploration and exploitation. Over-exploration reduces learning efficiency, while over-exploitation traps the policy in local optima and may even cause reward hacking. This dilemma is further aggravated in diffusion model fine-tuning by sparse rewards and long-horizon denoising. Against this backdrop, we present a systematic theoretical analysis of the coordination between exploration and exploitation in diffusion fine-tuning. First, we derive an analytical expression for the optimal policy under an uncertainty-driven exploration term. We then derive three coordination cases describing how exploration and exploitation evolve across the generation process. Based on these theories, we formalize an adaptive framework E_2Ada. Extensive **24** experiments show that our method consistently outperforms state-of-the-art approaches in diversity (8.32%), alignment (10.30%), and speed (88.64%).

1 INTRODUCTION

Diffusion models have progressed in image generation, yet their distributional-reconstruction training paradigm lacks direct alignment with specific preferences. Reinforcement learning (RL) can improve preference alignment, but introduces exploration–exploitation coordination challenges. In RL, exploration discovers high-value behaviors, whereas exploitation improves expected returns. Despite advances in RL algorithms, the relative intensities of exploration and exploitation remain difficult to coordinate adaptively. Excessive exploration reduces learning efficiency and destabilizes policy convergence, while overly deterministic exploitation risks suboptimal convergence Ahmed et al. (2019). For diffusion model fine-tuning, sparse rewards and long-horizon denoising intensify this challenge and often cause reward hacking Skalse et al. (2022); Eisenstein et al. (2023).

To understand reward hacking, we analyze reward structure. In prevailing paradigms, preference rewards are available only at the final denoising step, making stable policy learning from direct single-reward optimization challenging (see Appendix 5.4 for experimental demonstration) Karnewar et al. (2023). Existing methods therefore backfill the final preference reward to all denoising steps Franceschelli & Musolesi (2024); Chen et al. (2024). Although this stabilizes training, it biases optimization toward over-exploiting the final reward and weakens exploration, thereby causing the policy to pursue step-agnostic shortcuts for high final reward rather than improve generation quality across steps, ultimately triggering reward hacking. *ppendix D discusses detailed **related work**.*

Clearly, the imbalance between exploration and exploitation in diffusion RL is an important cause of reward hacking Franceschelli & Musolesi (2024). To mitigate this issue, we analyze the exploration–exploitation trade-off in diffusion fine-tuning using policy entropy to measure exploration, thereby clarifying how the two should be coordinated during training. First, we establish a theoretical link between policy entropy and exploratory behavior. Subsequently, we propose a novelty-seeking reward term to achieve a reasonable adjustment in policy entropy. Furthermore, we derive three forms of adaptive exploration–exploitation coordination during fine-tuning, characterizing the dynamic evolution of exploration and exploitation during training.

Based on the theory, we propose E_2Ada, an RL framework that generalizes to traditional diffusion and flow-matching models. E_2Ada's adaptive factor balances exploration and exploitation with generation progress and dynamically terminates trajectories per input, reducing computation 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.

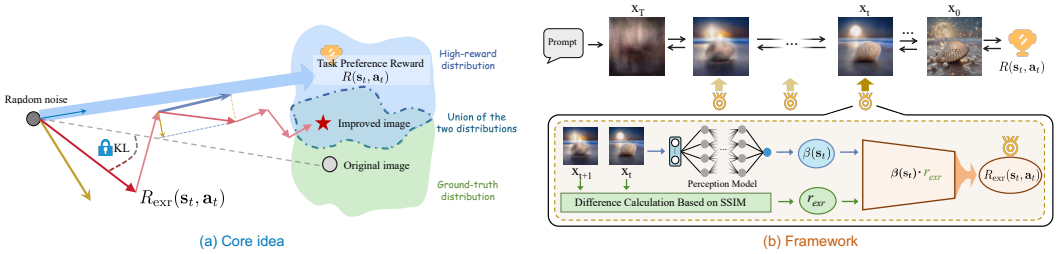

Figure 1: **Idea & Framework**. We adopt an adaptive exploration–exploitation trade-off: early training ensures diversity preventing premature convergence; later training maximizes preference reward, enhancing both *diversity and preference objectives*.

over-optimization risk. We evaluate on **6** diffusion models, **4** datasets, and **10** baselines over **24** experiments, where our method outperforms the baselines. Overall, our contributions are threefold: (1) a theoretical analysis of the exploration–exploitation trade-off in RL fine-tuning, yielding three adaptive coordination scenarios; (2) an intermediate exploration reward that addresses sparse rewards and improves generation diversity; and (3) E_2Ada, a theory-driven adaptive framework that balances exploration and exploitation to mitigate suboptimal traps and reward hacking.

2 Method

2.1 Reinforcement Learning in Diffusion Models

Diffusion Models. Let $\mathbf{x}_0$ be a sample and $\mathbf{x}_T \sim p(\mathbf{x}_T) = \mathcal{N}(0, I)$ a noise prior. The model learns reverse transitions $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$, which approximate the posterior $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$. Let $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$, where $\alpha_t := 1 - \beta_t, \quad \bar{\alpha}_t := \prod_{k=1}^{t} \alpha_k, \quad (\bar{\alpha}_0 := 1)$. Let $\epsilon_\theta(\mathbf{x}_t, t)$ denote the learned noise-prediction function. The induced estimate of the clean sample $\hat{\mathbf{x}}_0(\mathbf{x}_t)$ can be obtained as Xie & Gong (2025):

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

Conditioned on $\mathbf{z}$, e.g., a textual description, the reverse process becomes $p_\theta(\mathbf{x}_{0:T} \mid \mathbf{z}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{z})$.

Denoising as Entropy-Aware Markov Decision Process. We formulate denoising as a Markov Decision Process. $\mathbf{s}_t$ represents image states (latents) during the diffusion model's denoising process. Each state $\mathbf{s}_t$ corresponds to the intermediate state $\mathbf{x}_{T-t}$ in the denoising sequence, $\mathbf{s}_t = (\mathbf{z}, \mathbf{x}_{T-t})$. The action $\mathbf{a}_t$ includes both the noise prediction result and the newly sampled noise at the timestep t, i.e., $\mathbf{a}_t = \mathbf{x}_{T-t-1}$. It determines the transition from the current state $\mathbf{x}_{T-t}$ to the next state $\mathbf{x}_{T-t-1}$. π_θ denotes the parameterized policy $\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) = p_\theta(\mathbf{x}_{T-t-1} \mid \mathbf{x}_{T-t}, \mathbf{z})$. P_0 is the initial state distribution. $\delta_{(\mathbf{z}, \mathbf{x_{T-t-1}})}$ represents the Dirac distribution centered at $(\mathbf{z}, \mathbf{x_{T-t-1}})$. $P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ denotes the *state transition distribution*, which defines the conditional probability distribution of the next state $\mathbf{s}_{t+1}$ given the current state $\mathbf{s}_t$ and the action $\mathbf{a}_t$, $P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) = \delta_{(\mathbf{z}, \mathbf{x_{T-t-1}})}$. Regarding the specific form of state transitions, we adopt the Entropy-Aware MDP framework as formulated in Eq. 2.

$$p(\mathbf{x}_{T-t-1} | \mathbf{x}_{T-t}) = \mathcal{N}\left(\mathbf{x}_{T-t-1}; \mu_\theta(\mathbf{x}_{T-t}, T-t), \sigma^2_{\theta, T-t}\mathbf{I}\right), \tag{2}$$

Appendix B.4 discusses the relation of μ_θ, σ_θ and ϵ_θ. The preference reward of current RL is

$$R(\mathbf{s}_t, \mathbf{a}_t) \triangleq \begin{cases} r(\mathbf{s}_{t+1}) = r(\mathbf{x}_0, \mathbf{z}) & \text{if } t = T-1, \\ 0 & \text{otherwise.} \end{cases}. \tag{3}$$

$R(\mathbf{s}_t, \mathbf{a}_t)$ denotes the predefined preference reward , i.e., the exploitation reward. The mainstream reward paradigm thus lacks an explicit reward term that encourages exploration.

2.2 EXPLORATION–EXPLOITATION THEORY

(1) Exploration: explores the generative space to find high-value trajectories. (2) Exploitation: the model generates outputs by exploiting known high-reward generative patterns. Existing methods backfill final rewards, promoting over-exploitation and thereby triggering reward hacking. To address this, we introduce diffusion-specific exploration rewards, defined as:

$$r_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t) = 1 - \text{SSIM}_{t,t+1}. \tag{4}$$

For a detailed explanation of Eq. 4, please refer to Sec. 2.3. With exr/ext denoting 'explore'/'exploit', $R(\mathbf{s}_t, \mathbf{a}_t)$ is the preference (exploitation) reward. We define the objective as:

$$\max_{\theta} \mathbb{E}_{p(\mathbf{z})} \mathbb{E}_{\pi_\theta} [\sum_t (R(\mathbf{s}_t,\mathbf{a}_t) + \beta r_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t) + \alpha H(\pi_\theta(\cdot|\mathbf{s}_t)))], \tag{5}$$

where $r_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t)$ is incorporated into policy optimization as a ***purposeful exploration*** term. We introduce a regularization term of an appropriate magnitude, denoted $H(\pi_\theta(\cdot|\mathbf{s}_t)) = -\sum_a \pi(a|\mathbf{s}_t) \log \pi(a|\mathbf{s}_t)$, which acts as a distributional regularizer. Since RL tends to rapidly drive generation toward a single high-reward image (i.e., a point) Liu et al. (2025), this term preserves the distributional nature of generation. β and α denote coefficients.

2.2.1 *Solvability Proof under exploration reward.*

Based on Eq. 5, we prove that incorporating exploration rewards does not compromise solvability. Specifically, the following theorem demonstrates the existence of an optimal policy.

Theorem 1. *Given the objective, the optimal policy π_{θ^*} has the following expression:*

$$\pi_{\theta^*}(\mathbf{a}_t|\mathbf{s}_t) \propto \exp\{\frac{Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_t)}{\alpha}\},$$

where $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_t) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_t) + Q_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$, $Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_t) = R(\mathbf{s}_t, \mathbf{a}_t) + \alpha \cdot \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q_{\text{ext}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\}$, *and* $Q_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t) = \beta r_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t) + \alpha \cdot \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q_{\text{exr}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\}$.

With the exploration reward, our proposed algorithm is solvable, and an optimal policy influenced by both preference and exploration rewards exists (proof in Appendix B.1).

2.2.2 *Impact of exploration reward on Entropy.*

In RL, entropy measures exploration De Paola et al. (2025). We establish the relationship between exploration rewards and entropy. Based on Theorem 1, define total entropy under exploitation and exploration rewards as $H(\pi_{\text{total}}(\cdot|\mathbf{s})) = -\int_{\mathbf{a}_i} \frac{e^{Q_i+\delta_i}}{\int_{\mathbf{a}_j} e^{Q_j+\delta_j} d\mathbf{a}_j} \log \frac{e^{Q_i+\delta_i}}{\int_{\mathbf{a}_j} e^{Q_j+\delta_j} d\mathbf{a}_j} d\mathbf{a}_i$, where $Q_i = Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_i)$ and $\delta_i = Q_{\text{exr}}(\mathbf{s}, \mathbf{a}_i)$. Similarly, define the entropy $H(\pi_{\text{ext}}(\cdot|\mathbf{s}))$ of the extrinsic policy π_{ext}, defined by $H(\pi_{\text{ext}}(\cdot|\mathbf{s})) = -\int_{\mathbf{a}_i} \frac{e^{Q_i}}{\int_{\mathbf{a}_j} e^{Q_j} d\mathbf{a}_j} \log \frac{e^{Q_i}}{\int_{\mathbf{a}_j} e^{Q_j} d\mathbf{a}_j} d\mathbf{a}_i$, where $Q_i = Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_i)$. In the following, we present the relationship between $H(\pi_{\text{ext}}(\cdot|\mathbf{s}))$ and $H(\pi_{\text{total}}(\cdot|\mathbf{s}))$.

Lemma 2.1. *For any two actions $\mathbf{a}_i$ and $\mathbf{a}_j$ such that $Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_i) > Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_j)$, it holds that $0 < Q_{\text{exr}}(\mathbf{s}, \mathbf{a}_j) - Q_{\text{exr}}(\mathbf{s}, \mathbf{a}_i) < Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_i) - Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_j)$. Then, for any state $\mathbf{s}$, we have $H(\pi_{\text{ext}}(\cdot|\mathbf{s})) < H(\pi_{\text{total}}(\cdot|\mathbf{s}))$.*

It shows that adding $r_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t)$ increases entropy and broadens exploration. Fig. 27 visualizes the effect of *only the exploration reward*, supporting Lemma 2.1. Its proof is in Appendix B.2.

2.2.3 *Theory of Exploration–Exploitation Coordination.*

In RL, exploration uncovers high-value policies, while exploitation stabilizes policy convergence. Excessive exploration harms learning efficiency, while deterministic exploitation risks suboptimal convergence. Most existing RL fine-tuning methods focus on exploitation but fail to encourage exploration, resulting in an exploration–exploitation imbalance, which is a major cause of reward hacking. To achieve adaptive exploration–exploitation, we relax the fixed coefficient β in Eq. 5 and

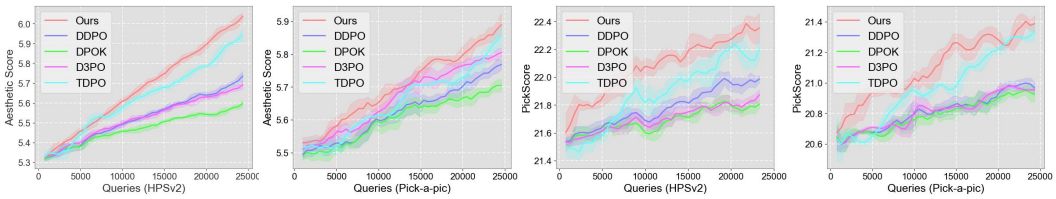

Figure 2: **Trained with Multiple Datasets and Target Rewards**. HPSv2 and Pick-a-Pic prompts using Aesthetic Score and PickScore as objectives. *Notably, TDPO incurs reward hacking during training (see Fig. 15), but our method avoids it.*

replace it with a state-dependent variable $\beta(s_t) \in [0, 1]$, allowing entropy to increase or decrease adaptively. Accordingly, Eq. 5 is updated to:

$$\max_{\theta} \mathbb{E}_{p(\mathbf{z})} \mathbb{E}_{\pi_\theta} [\sum_t (R(\mathbf{s}_t, \mathbf{a}_t) + \beta(s_t) r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t) + \alpha H(\pi_\theta(\cdot|\mathbf{s}_t)))]. \tag{6}$$

As shown in Lemma 2.1, *with fixed* β, introducing $r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$ increases entropy and exploration. In Theorem 2, we use an *adaptive* $\beta(\mathbf{s}_t)$ to replace the fixed β to modulate $r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$, allowing policy entropy to ***increase or decrease***, thereby achieving adaptive exploration-exploitation.

Theorem 2. *Under conditions of Lemma 2.1 and different values of β function, for any s_t, policy optimization with the above Eq. 6 can have either $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) \leq H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$ or $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) > H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$.*

Proof. From Theorem 1, we have $\delta(\mathbf{s}_t, \mathbf{a}_t) \triangleq \beta(\mathbf{s}_t) r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t) + \alpha \cdot \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q_{\text{exr}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\}$. Depending on $\beta(\mathbf{s}_t)$, policy optimization admits three cases regarding the exploration-exploitation trade-off. For example, consider two actions $\mathbf{a}_1$ and $\mathbf{a}_2$ from action space.

Case I - Exploration Dominant: for any $\mathbf{s}_t$, when $\beta(\mathbf{s}_t) \equiv 1$, $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_1) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) + \delta(\mathbf{s}_t, \mathbf{a}_1)$, and $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_2) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_2) + \delta(\mathbf{s}, \mathbf{a}_2)$. Because $\beta(\mathbf{s}_t) \equiv 1$, from Lemma 2.1, $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) < H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$, i.e., the entropy increases. In this case, the strength of exploration reaches its peak; our method exhibits stronger exploratory ability than methods without exploration rewards.

Case II - Adaptive Exploration-Exploitation: if $Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) \geq Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a})$, for any $\mathbf{a}$, and $\beta(\mathbf{s}_t)$ satisfies $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_1) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) + \delta(\mathbf{s}_t, \mathbf{a}_1)$ and $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_2) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_2)$, we have $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) > H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$, i.e., the optimal policy probability increases while total entropy decreases. In this case, the degree of exploitation increases while the level of exploration decreases.

Case III - Exploitation Dominant: for any $\mathbf{s}$, when $\beta(\mathbf{s}_t) \equiv 0$, we have $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_1) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) + C$ and $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_2) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_2) + C$, where C is constant. Entropy is unchanged, i.e., $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) = H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$. Exploration is zeroed out, and exploitation dominates. □

Theorem shows our proposed $\beta(\mathbf{s}_t)$ can adaptively increase or decrease policy entropy, dynamically adjusting training strategy to balance exploration and exploitation.

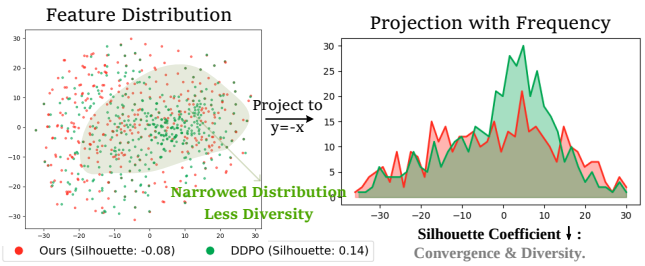

Figure 3: Visualization of feature diversity.

Table 2: The high-frequency energy&ratio in generated images.

Method	Energy	Ratio (%)
Base	2.23×10^6	1.90×10^{-4}
DDPO	2.51×10^6	2.07×10^{-4}
DPOK	1.79×10^6	1.65×10^{-4}
TDPO	5.80×10^6	4.77×10^{-4}
Ours	1.49×10^6	1.52×10^{-4}

Table 1: **Reward Hacking Evaluations with Multiple Cross-Metrics.** Metrics use SDv15 as backbone and AES as target reward. Bold/underlining mark best/second-best results; backbone (gray) entries excluded from comparison. *At matched AES, Ours consistently outperforms others in Fidelity, Diversity, and Richness, indicating limited hacking by the target reward.*

Method	Preference				Fidelity			Diversity			Richness			#Top2
	AES	PS↑	IR↑	HPS↑	FID↓	CLIP↑	iFS↑	LPIPS↑	IS↑	TCE↑	BRI↓	NIQE↓	SE↑	
SDv15	5.514	21.29	0.916	0.280	-	0.243	-	0.646	30.31	40.23	8.02	4.131	11.34	-
DDPO	6.004	20.99	0.897	0.266	95.15	0.239	0.494	0.601	26.17	38.98	11.01	6.967	10.98	0
DPOK	5.959	21.12	0.793	0.271	63.31	0.242	0.689	0.643	28.18	39.51	9.07	4.795	11.30	3
D3PO	5.970	21.10	0.829	0.276	72.37	0.240	0.696	0.640	27.97	39.95	8.69	5.003	11.37	2
B2Diff	6.029	21.09	0.965	**0.279**	53.32	0.241	0.777	0.647	26.21	39.79	8.32	5.652	11.58	5
DR	6.011	21.41	0.723	0.273	74.25	**0.244**	0.316	0.548	27.89	39.37	7.66	5.030	11.72	4
TDPO	6.072	21.31	0.676	0.261	101.4	0.232	0.217	0.517	25.95	37.16	8.08	6.050	11.52	0
Ours	6.009	**21.59**	**1.021**	**0.279**	**50.19**	**0.244**	**0.857**	**0.651**	**30.29**	**40.07**	**5.35**	**3.932**	**11.97**	**12**

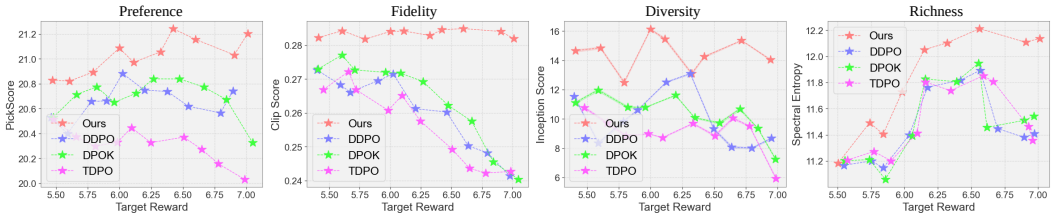

Figure 4: **Evaluating Reward Hacking** with Different Levels of Target Reward. AES is deployed as the *target reward (x-axis)*, and other unoptimized metrics are used for reward hacking evaluations.

2.3 Theoretical Formalization Framework

In Sec. 2.2, we provided a systematic theoretical analysis of exploration–exploitation. In this subsection, based on the above theory, we formalize an adaptive framework named E_2Ada.

Differential Step Reward. Diffusion RL requires step-wise denoising optimization, yet final-only preference rewards leave intermediate steps feedback-free, creating sparse rewards that cause policy optimization to fail (see Appendix Fig. 17). Backfilling the final reward to all steps yields (1) *Policy learning accuracy decline* from identical rewards for distinct actions; and (2) *Over-exploitation*, where repeated reinforcement of the same reward over-exploitation it, narrows the generation distribution, reduces diversity, and induces reward hacking. We address both with a diversity-based exploration reward that supplies differentiated real-time rewards across steps. We compute the exploration reward using the Structural Similarity Index Measure (SSIM) Wang et al. (2004), its formula is provided in the appendix. First, using Eq. 1, we obtain the predictions $\hat{\mathbf{x}}_0(\mathbf{s}_t)$ and $\hat{\mathbf{x}}_0(\mathbf{s}_{t+1})$ corresponding to the neighboring latent steps $\mathbf{s}_t$ and $\mathbf{s}_{t+1}$. Then, we define the exploration reward as: $r_{\text{exr}}(t) = 1 - \text{SSIM}_{t,t+1}$, where $\text{SSIM}_{t,t+1} = \text{SSIM}(\hat{\mathbf{x}}_0(\mathbf{s}_t), \hat{\mathbf{x}}_0(\mathbf{s}_{t+1}))$. For the ability of $r_{\text{explore}}(t)$ to increase trajectory diversity during training, we provide qualitative visual evidence in Fig. 27. We further report the performance through various diversity metrics in Tab. 1&4, and Fig. 4. In particular, Fig. 3 visually demonstrates the increased diversity of the generated image distribution. To quantitatively support the observations in Fig. 3, we further provide the corresponding results in Tab. 2, which show that Ours has even lower high-frequency energy and high-frequency ratio, indicating that the increased diversity arises from semantic variation rather than high-frequency noise. *To achieve increased diversity while maintaining fidelity, we constrain the generative distribution during training using KL divergence*, which is a common practice in the field Liu et al. (2025).

Adaptive Mechanism. We enable adaptive exploration–exploitation coordination using an adaptive coefficient predictor $\psi(\cdot)$. trained by labeling the clean image $\mathbf{x}_0$ as $y = 1$ and the initial noisy image $\mathbf{x}_T$ as $y = 0$. Its loss: $\mathcal{L}_{\text{adaptive}} = -\frac{1}{N}\sum_{i,t}^{N,T}[y_t[i]\log\psi(\mathbf{s}_t[i]) + (1-y_t[i])\log(1-\psi(\mathbf{s}_t[i]))]$, where $t \in \{1,\ldots,T+1\}$ denotes the diffusion denoising timestep, and $i \in \{1,\ldots,N\}$ indexes different image samples. Here, $y_i \in \{0,1\}$ denotes the corresponding noise-level label, and $\psi(\mathbf{s}_t)[i] \in [0,1]$ represents the pre-

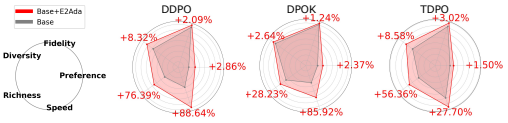

Figure 5: Deploying E2Ada (red) as a **plugin** to methods (gray). We evaluate percentage improvements across five dimensions: PS, CLIP, LPIPS, SE, and training speed.

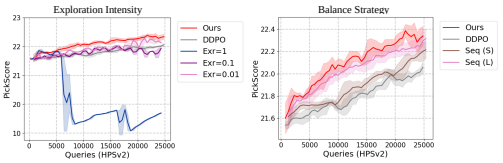

Figure 6: Ablation study with multiple Exploration (Exr) intensities and ***adaptive strategies***. This experiment verifies that *excessive exploration impairs stable convergence*.

dicted noise probability. After training, given a state $\mathbf{s}_t$, the predictor outputs an indicator reflecting the current noise level, denoted as $\beta(\mathbf{s}_t) = 1 - \psi(\mathbf{s}_t)$. We use $\beta(\mathbf{s}_t)$ as the adaptive factor for exploration–exploitation. Therefore, the total reward during training for our method can be expressed as: $R_{\text{total}}(\mathbf{s}_t, \mathbf{a}_t) = R(\mathbf{s}_t, \mathbf{a}_t) + \beta(\mathbf{s}_t)\, r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t) \triangleq R(\mathbf{s}_t, \mathbf{a}_t) + R_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$, where $R(\mathbf{s}_t, \mathbf{a}_t)$ is the exploitation (preference) reward and $r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$ is the exploration reward. For clarity, we represent $r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$ with the adaptive factor $\beta(\mathbf{s}_t)$ collectively as $R_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t)$. Appendix.A.5 provides details on the adaptive mechanism and network.

Adaptive Termination Strategy (ATS). When $\psi(\mathbf{s}_t) = 1$, i.e. ,$\beta(\mathbf{s}_t) = 0$ indicates prompt-wise denoising completion and terminates RL training. With structure largely stable late in diffusion, further RL is futile and may overconcentrate limited adjustment capacity on local details, amplifying biases and harming generation quality Ma et al. (2024). ATS therefore adaptively allocates training length per prompt, reducing cost and over-optimization without compromising image quality.

3 Experiments

3.1 Implementation Details

Datasets. We utilize four popular datasets for the fine-tuning: *HPSv2* Wu et al. (2023), *Pick-a-Pic* Kirstain et al. (2023), *GenEval* Ghosh et al. (2023), and Simple Animals Black et al. (2023).

Rewards and Metrics. The training rewards in this paper include Aesthetic Score (AES) Schuhmann et al. (2022), PickScore (PS) Kirstain et al. (2023), JPEG compressibility, incompressibility, and multi-objective (AES+PS). To quantify reward hacking, we introduce metrics including: *Aesthetic Preference:* AES, PS, IR, and HPSv2 Wu et al. (2023). *Image Fidelity:* ClipScore Radford et al. (2021) (Clip), FID Heusel et al. (2017), and improved F1 Score (iFS) Kynkäänniemi et al. (2019). *Generative Diversity:* LPIPS Zhang et al. (2018), TCE Ibarrola & Grace (2024), and Inception Score Salimans et al. (2016) (IS). *Compositional Richness:* NIQE Mittal et al. (2012b), BRISQUE (BRI) Mittal et al. (2012a), and Spectral Entropy (SE) Liu et al. (2014).

Backbones. To evaluate robustness across architectures, we adopt two benchmarks: (1) *Unet* Diffusion models, including v1.4 (SDv1.4), v1.5 (SDv1.5), v2.1-turbo (SDv2.1), and XL 1.0 (SDXL). (2) *Flow-matching* models such as SD 3.5-medium (SD3.5) and Flux (see Fig. 9, 11, 24 and 26).

Baselines. Our method is benchmarked against state-of-the-art approaches under both Unet and Flow-matching diffusion frameworks. Due to architectural differences, comparisons are conducted separately within each framework. The Unet baselines include DDPO Black et al. (2023), DPOK Fan et al. (2024), D3PO Yang et al. (2024), TDPO Zhang et al. (2024), DenseReward Chan et al. (2024) (DR), and B2DiffusionRL Hu et al. (2025) (B2Diff), while the flow-matching baselines include GRPO (Flow-GRPO), PPO (DDPO), SFT (Flow-SFT), and DPO (Diffusion-DPO) Wallace et al. (2024). Notably, our method can be integrated with any of these baselines, though for simplicity, we refer to the combination of DDPO with our method as Ours, unless specified otherwise. The shadows in the curve figures represent the standard deviations with three seeds.

3.2 In Practice: Multiple Settings

Results Trained with Multiple Target Rewards. To evaluate reward learning, we conduct experiments on PickaPic and HPSv2 datasets, using PickScore and Aesthetic Score as reward objectives.

As shown in Fig. 2, our method consistently demonstrates a faster learning rate. This improvement can be attributed to the balance between exploration and exploitation, which enables our approach to discover better policies, ultimately leading to faster and more stable convergence. More results can be found in *Appendix*.

Results in supplementary reward objectives. We conducted experiments with three supplementary reward objectives: compressibility, incompressibility, and multi-objective settings, and on the GenEval auxiliary dataset. Furthermore, we evaluated against two additional baselines, *i.e.*, DR and B2Diff. See Appendix Fig. 14 for detailed results.

Results Trained on Multiple Backbones. To further validate the generality of our proposed exploration-exploitation coordination policy, we conducted extensive evaluations across a broader range of base models, including SDv14, SDv15, SDXL, Flux, and SD3.5. Due to the architectural discrepancies inherent to Flux and SD3.5 that result in performance variations, we present the corresponding results separately in Sec. 3.4. As shown in Appendix Fig. 16, our method consistently achieves optimal performance across diverse backbones. *On robust large-parameter models like SDXL, only Ours can effectively learn the reward signal*.

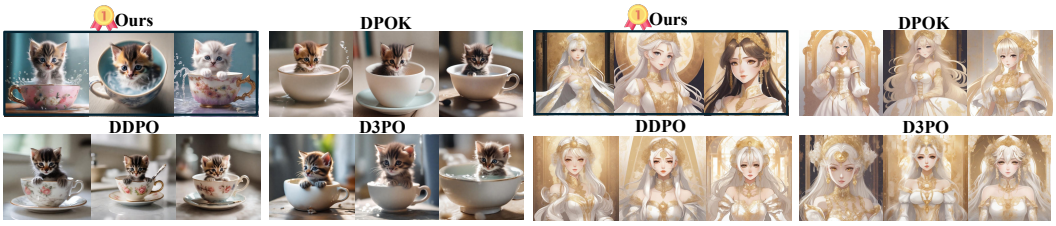

Figure 7: **Generation Diversity**. The baseline collapses to similar facial poses and cup–cat configurations after RL training due to reward hacking, whereas our method preserves generation diversity.

3.3 Performance on Mitigating Reward Hacking.

Multi-metrics Results under Fixed Target Reward. To fairly evaluate reward hacking across different methods, all methods were trained until reaching an aesthetics score of nearly 6.0, after which we compared their overall performance on other metrics. In Tab. 1, we also report the original SDv1.5 model as a reference. When AES is fixed as the training metric, our method consistently ranks among the top two across all evaluation metrics, indicating that E2Ada is not excessively hacked by the training metric at the cost of performance on the evaluation metrics.

Results under multi-intensities Target Reward. In addition to evaluating final generation results, we designed experiments to assess the extent of reward hacking under different reward learning settings across various baselines. In Fig. 4, our method remains virtually unaffected by hacking when AES is below 7. Although a slight performance degradation is observed in final training stages, which is difficult to avoid due to eventual reward hacking with over-optimization, our approach consistently outperforms all baselines.

Visualization of Reward Hacking Mitigation. With exploration and adaptive mechanism, effectively alleviating degeneration toward a single generation mode caused by reward hacking (see Appendix Figs. 15 & 7).

3.4 Effectiveness as Plugin

Given that our proposed exploration-exploitation balance is a high-level guiding principle, we consider implementing our method across different baseline approaches, as well as under various RL policies and diffusion model architectures.

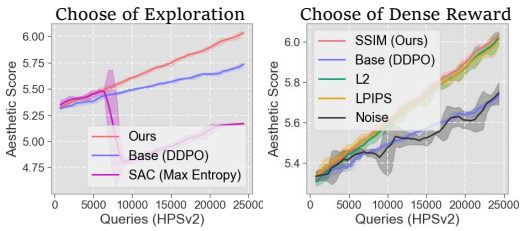

Figure 8: Exploration & Reward Design. We compare purposeful exploration against aimless alternatives and evaluate different formulations of the exploration reward.

Table 3: **Reward Learning & Hacking on SD3.5 and FLUX.** All baselines are trained for the same 240 epochs with PS as the optimization reward. The remaining metrics are unseen during training and evaluated after training.

Methods	PS↑	CLIP↑	LPIPS↑	NIQE↓	#Top2
SD3.5	21.65	**0.238**	**0.641**	4.969	2
+Flow-GRPO	21.91	0.232	0.623	4.712	2
+Ours	**22.29**	0.236	0.637	**4.547**	**4**
FLUX	21.51	**0.237**	0.644	4.012	3
+Flow-GRPO	21.59	0.234	0.622	4.179	1
+Ours	**21.75**	**0.237**	**0.645**	**3.832**	**4**

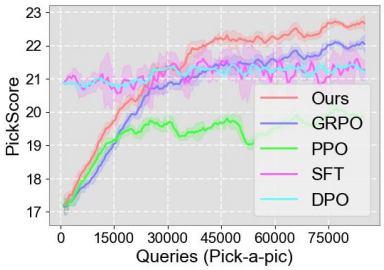

Figure 9: Results on SD3.5 with different RL policies. See Appendix for experiment on FLUX.

Table 4: **Ablation on Each Proposed Component.** Ablation results on *HPSv2*, with PS as the target reward. Time denotes training hours to reach PS=22.2, and TpS denotes time per sample in seconds.

Variants	PS	AES↑	CLIP↑	IS↑	SE↑	Time↓	TpS↓	#Top2
Base	22.16	5.310	0.239	21.91	11.01	10.41	3.01	1
Exr-fixed	22.19	5.401	0.241	23.52	11.91	8.79	3.17	1
Seq (S)	22.20	5.671	0.241	22.51	11.57	8.95	3.60	0
Seq (L)	22.20	5.695	0.243	22.79	11.51	6.76	3.63	1
w/o ATS	22.23	5.696	**0.244**	22.89	11.92	7.17	3.89	3
Ours	22.21	**5.743**	0.243	**23.71**	**12.01**	**5.44**	**2.82**	6

Results Implemented on Different Baselines. We integrate E2Ada into DDPO, DPOK, and TDPO, and conduct training on the Animal dataset using SDv15 until reaching a target reward of AES = 6.0. We then evaluate performance across five dimensions: HPS, CLIP, LPIPS, SE, and training speed. Fig. 5 shows that incorporating E2Ada improves overall performance across all methods.

Results Implemented on Flow-Matching Models. To further validate applicability of E2Ada, we evaluate its integration with GRPO on two Flow-Matching models, SD3.5 and Flux, and compare results with those obtained using various reinforcement learning strategies, including GRPO (Flow-GRPO), PPO (DDPO), SFT (Flow-SFT), and DPO (Diffusion-DPO). In Fig. 9, our method consistently demonstrates improved performance, further substantiating its potential for practical application and its generalizability.

Inference Results on Flow-Matching Models. In Tab. 3, we also evaluate the trained models (SD3.5 and FLUX) on the test set based on pick-a-pic prompt set. It can be observed that the proposed E2Ada can effectively enhance the target reward while maintaining other quality perspectives.

3.5 Ablation Study

Exploration-Exploitation Trade-off. We theoretically analyze the importance of the exploration–exploitation trade-off. To empirically investigate this, we first compare the effects of varying exploration intensities and different balancing strategies on training performance. In Fig. 6, we present results under three levels of exploration intensity, as well as two balancing strategies: static sequential scheduling (Seq) with Sigmoid as mapping function (S) and Linear as mapping function (L). Firstly, excessive exploration leads to model collapse. Regarding the balancing strategies, while the introduction of a balance allows the model to learn, using fixed sequential strategies results in significant training fluctuations and less promising results. Overall, our proposed dynamic balancing strategy achieves the best performance. Subsequently, we analyze the multi-dimensional generation quality of the trained models. As shown in Tab. 4, introducing exploration effectively enhances



Figure 10: Computation cost evaluation.

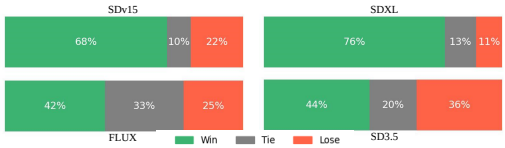

Figure 11: User study on multiple backbones.

image diversity and pixel-level richness, whereas balancing exploration and exploitation yields the best overall performance. In terms of computational cost, incorporating exploration and exploitation introduces only minimal additional overhead per training sample. Moreover, the Adaptive Termination Strategy (ATS) we propose further reduces the training time, resulting in lower overall cost compared to the baseline.

Purposeful vs. Aimless Exploration We compare our purposeful exploration strategy, which explicitly encourages compositional diversity, with two aimless exploration approaches: maximum-entropy exploration and direct noise augmentation. As shown in Fig. 8 Left, our method achieves stable improvement throughout training. In contrast, indiscriminately maximizing entropy readily destabilizes or even collapses training, while the noise-based approach performs worse than the baseline. Results demonstrate that purposeful exploration is more effective than aimless exploration.

Designs of the Exploration Reward. In Fig. 8 Right, we investigate different formulations of exploration rewards. Distance-based metrics (e.g., L2 and LPIPS) achieve reasonably similar behavior, whereas pure random noise fails to model structural differences and is ineffective. These results suggest that the design of exploration rewards is meaningful, with SSIM emerging as a simple yet effective choice for r_{exr}.

3.6 Computation Experiments

We report the wall-clock time and GPU time required by different methods to reach a target reward value (AES = 6) on the Pick-a-Pic prompt set, as well as the per-sample training time. Fig. 10 shows ours achieves the target reward in significantly shorter time, suggesting a more efficient learning of the reward function. Furthermore, our method demonstrates competitive performance in terms of per-sample training time. Overall, the computational cost analysis confirms the superior training efficiency of our approach.

3.7 Subjective Evaluation

Qualitative Showcases. In this paper, we evaluate the generative quality qualitatively from two main perspectives, *i.e.*, diversity and alignment. In Fig. 7, we first compare generation diversity among different methods. It can be observed that our results consistently exhibit diverse posture, color, and details. Then, in Fig. 25, we compare the image-text alignment performance with three different settings. The results also demonstrate our improved performance.

User Study. Here, we recruit subjects to judge the win-loss between images (same seeds&prompts) with or without our method on different backbones. As shown in Fig. 11, our method consistently exhibits superior performance on both SDE-based and flow-matching-based diffusion models.

4 Conclusion

In this paper, we propose to establish an adaptive exploration–exploitation theory for RL fine-tuning of diffusion models, with an applicable implementation termed E_2Ada. Specifically, we propose an adaptive-based exploration reward that aims to alleviate sparse rewards and reward hacking in diffusion RL. Extensive experiments show that early exploration improves diversity, whereas late exploitation enhances preference reward, achieving both with lower computation.

AI USE STATEMENT

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

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pp. 151–160. PMLR, 2019.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

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

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Alex James Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. Dense reward for free in reinforcement learning from human feedback. 2024.

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*, 2024.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9): 10850–10869, 2023.

Bo Dai, Albert E. Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. SBEED: convergent reinforcement learning with nonlinear function approximation. In *Proceedings of International Conference on Machine Learning,*, 2018.

Vincenzo De Paola, Riccardo Zamboni, Mirco Mutti, and Marcello Restelli. Enhancing diversity in parallel agents: A maximum state entropy exploration story. *arXiv preprint arXiv:2505.01336*, 2025.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

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.

Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Björn Ommer. Diffusion models and representation learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

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

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2426–2436, 2023.

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

Zijing Hu, Fengda Zhang, Long Chen, Kun Kuang, Jiahui Li, Kaifeng Gao, Jun Xiao, Xin Wang, and Wenwu Zhu. Towards better alignment: Training diffusion models with reinforcement learning against sparse rewards. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23604–23614, 2025.

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

Animesh Karnewar, Andrea Vedaldi, David Novotny, and Niloy J Mitra. Holodiffusion: Training a 3d diffusion model using 2d images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18423–18433, 2023.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24174–24184, 2024.

Amirhossein Kazerouni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical image analysis*, 88:102846, 2023.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

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.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Buhua Liu, Shitong Shao, Bao Li, Lichen Bai, Zhiqiang Xu, Haoyi Xiong, James T Kwok, Sumi Helal, and Zeke Xie. Alignment of diffusion models: Fundamentals, challenges, and future. *ACM Computing Surveys*, 58(9):1–37, 2026.

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

Lixiong Liu, Bao Liu, Hua Huang, and Alan Conrad Bovik. No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image communication*, 29(8):856–863, 2014.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15762–15772, 2024.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012a.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012b.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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, 2021.

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, 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.

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

Michel Tokic. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual conference on artificial intelligence*, pp. 203–210. Springer, 2010.

Mark Towers, Ariel Kwiatkowski, John Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun Kg, Rodrigo Perez-Vicente, et al. Gymnasium: A standard interface for reinforcement learning environments. *Advances in Neural Information Processing Systems*, 38, 2026.

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.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

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

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David Fleet. Controlling space and time with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 95989–96012, 2025.

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.

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

RenYe Yan, Jikang Cheng, Yaozhong Gan, Shikun Sun, You Wu, Yunfan Yang, Liang Ling, Jinlong Lin, Yeshuang Zhu, Jie Zhou, Jinchao Zhang, Junliang Xing, Yimao Cai, and Ru Huang. Entropy-adaptive diffusion policy optimization with dynamic step alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1924–1934, October 2025.

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, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

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, 2024.

Appendix

A Additional Detailed Experiments and Analysis

A.1 Experimental List Overview

We conduct a total of **24** experiments. **Due to the large experimental scale**, we summarize the experiments in the following list to help readers clearly understand the intent of each experiment.

- **(1) Training Effect with *Dataset* & *Reward Switching* Comparison Experiment.** See Fig. 2. Comparing the training effects of different methods under different datasets and reward functions.
- **(2) Visualization Experiment of Reward Hacking.** See Fig. 15. By comparing the images generated by different methods, this experiment visually shows the generation distribution shrinkage and the decline of text–image alignment caused by reward hacking, and verifies that our method can alleviate this phenomenon.
- **(3) Curve Quantitative Experiment of Reward Hacking.** See Fig. 4. Using AEScore as the training objective and simultaneously monitoring the changes of other evaluation metrics during the training process to determine whether different methods are hijacked by the training reward.
- **(4) Table Quantitative Experiment of Reward Hacking.** See Tab. 1. All methods use AES as the training objective, and when all methods reach the same aesthetic score (about 6), they are evaluated using a total of 12 metrics from four dimensions: Preference, Fidelity, Diversity, and Richness, to measure the extent to which each method is hijacked by the aesthetic reward; the more severe the hijacking, the worse its performance on other metrics under the same aesthetic score.
- **(5) *Backbone Switching* Experiment.** See Fig. 16. We conduct this experiment to verify that the advantages of our method remain unaffected when the backbone to be fine-tuned is replaced.
- **(6) *Plug-in* Experiment.** See Fig. 5. This experiment proves that our method is easy to plug in and that plugging it in leads to significant improvements over the original baseline.
- **(7) Curve-based Ablation Study.** See Fig. 6. We verify the impact of exploration strength and the exploration–exploitation adaptive mechanism on generation performance.
- **(8) Table-based Ablation Study.** See Tab. 4. We verify the impact of exploration strength and the exploration–exploitation adaptive mechanism on generation performance.
- **(9) Reward Learning and Hacking Performance on *Flow-matching* Diffusion Models.** See Tab. 3. This experiment proves that E_2Adastill has a significant effect on alleviating reward hacking and comprehensively improving generation quality for flow-matching based diffusion models.
- **(10) Training Effect in *Flow-matching* Diffusion Models.** See Fig. 9. Compare the training effects of multiple methods under flow-matching.
- **(11) *Generation Diversity* Comparison Experiment.** See Fig. 7. Under the same prompt, we compare the generative diversity of different methods.
- **(12) *Computation Efficiency* Experiment.** See Fig. 10. We systematically compare the computational cost of different methods from two dimensions. The results show that, due to the proposed Adaptive Termination Strategy, our method achieves significantly lower computational cost than various baselines.
- **(13) Controlled Generation & OCR & Complex Instruction Alignment based on Flow-Matching Models.** See Fig. 25. We compare the alignment capability of different methods for different task objectives.
- **(14) Task Objective Switching Experiment.** See Fig. 12. We provide this experiment to verify that the advantages of our method are not affected by replacing the backbone to be fine-tuned.

- **(15) Comparing Training Effectiveness with More SOTA Approaches.** See Fig. 14. In this experiment, we introduce as many advanced methods commonly used in the field as possible to compare the training performance.
- **(16) Reward Backfilling Visualization.** See Fig. 17.
- **(17) Quantitative Evaluation of Adaptive Termination's Impact on Performance.** See Tab. 6. This experiment demonstrates that the adaptive early stopping mechanism does not only have no negative impact on the optimization performance, but also improves the score while reducing computational overhead. This is because, under the influence of the adaptive balancing mechanism, the exploration encouragement in the early stage increases the likelihood of discovering better generation paths, while the exploitation encouragement in the later stage helps stabilize the convergence of the high-value trajectories already discovered.
- **(18) Qualitative Analysis of the Impact of Adaptive Termination on Performance.** See Fig. 21. From a visualization standpoint, this experiment shows that Adaptive Termination has no impact on the quality of image generation.
- **(19) Generation Results of SD3.5 with E_2Ada**. See Fig. 28.
- **(20) Generation Results of SDXL with E_2Ada**. See Fig. 29.
- **(21) Control Generation Based on SDXL**. See Fig. 22.
- **(22) Complex Prompt Generation Based on SDXL**. See Fig. 23.
- **(23) Control Generation Based on SD3.5**. See Fig. 24.
- **(24) Complex Prompt Generation Based on SD3.5**. See Fig. 26.

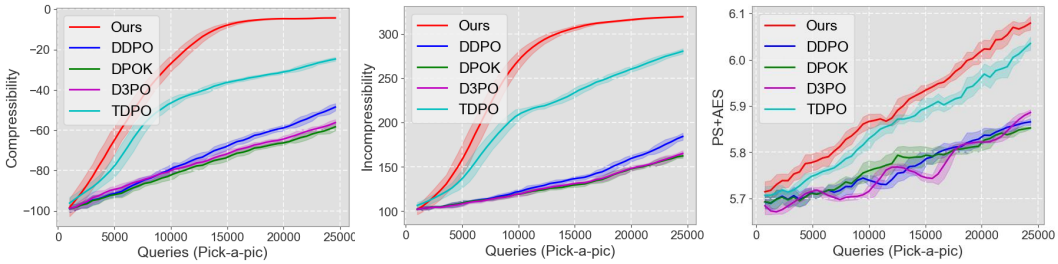

Figure 12: Results with more optimizing objectives.

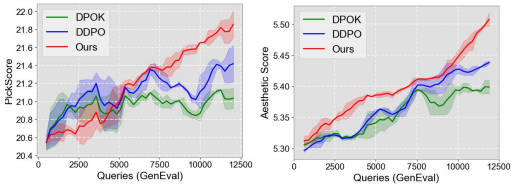

Figure 13: Results on supplemented GenEval dataset.

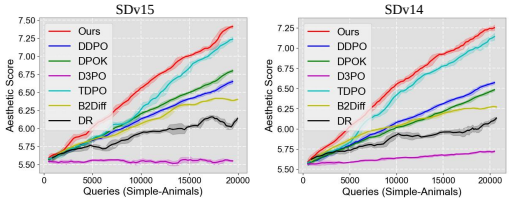

Figure 14: Results with more baselines.

A.2 Additional Quantative Experiments

Here, we provide the supplementary results with more datasets, more objectives, and more baselines. In Fig. 13, we supplement the results on GenEval with two objectives. Fig. 12 shows the results on Compressibility, Incompressibility, and multi-objective (PS+AES). Fig. 14 adds two baselines (*i.e.*, B2Diff and DenseReward) for comparisons. It can be observed that our method is consistently effective across these various settings, which further demonstrates its plug-and-play performance.



Figure 15: **Visualizing Reward hacking**. Existing methods neglect exploration and rely on reward backfilling, overusing preference rewards, which narrows the generation distribution, causing *similar backgrounds* or *reduced text-image alignment*.

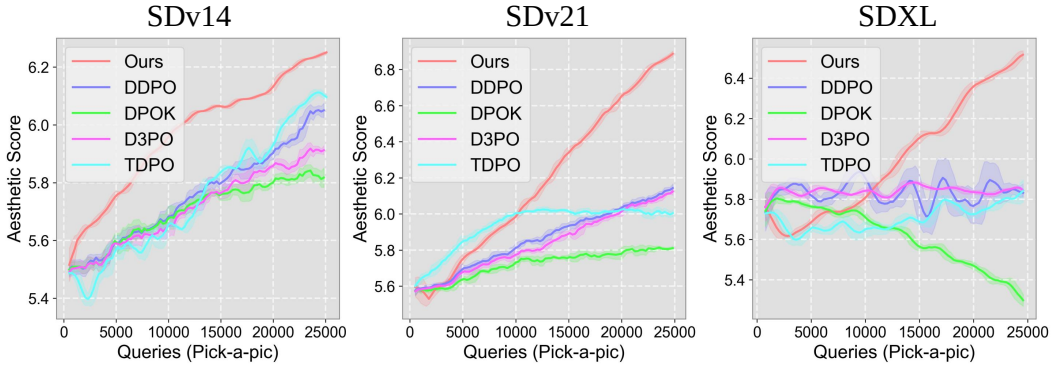

Figure 16: Results on **widely used diffusion backbones**. See Fig. 9 for **flow-matching-based** backbones comparisons.

A.3 Additional Qualitative Experiments

First, in Fig. 16, we shows the learning performance comparisons on SD14 SD15 and SDXL. We present several qualitative results from different models, highlighting the effectiveness of SDXL and SD3.5 in various tasks.

First, we show the overall impreesive quality of our results based on SD-XL (Fig. 29) and SD3.5 (Fig. 28) Then, Fig. 29 shows the qualitative results obtained using the SDXL model. These results demonstrate the model's generative capabilities, visually evaluated based on the realism and coherence of the outputs generated from the GenEval dataset. In Fig. 22, we present a qualitative comparison on controlled generation using the SDXL model. The results illustrate the model's proficiency in generating outputs that align closely with specific prompts from the GenEval dataset, showcasing its controlled generation ability. Fig. 23 evaluates the SDXL model's performance in terms of semantic alignment and overall generative quality. This figure highlights the model's ability to not only generate realistic images but also to ensure that the generated images are semantically aligned with the input prompts. Finally, Fig. 24 compares the controlled generation capabilities of the SD3.5 model. The figure reveals that SD3.5, like SDXL, produces high-quality images from the GenEval dataset but exhibits different characteristics in terms of color handling and visual consistency.

Fig. 26 evaluates the semantic alignment and overall generative quality of the SD3.5 model. It demonstrates that SD3.5 can generate semantically consistent images that closely match the given prompts, further showcasing its capability in both quality and alignment tasks.

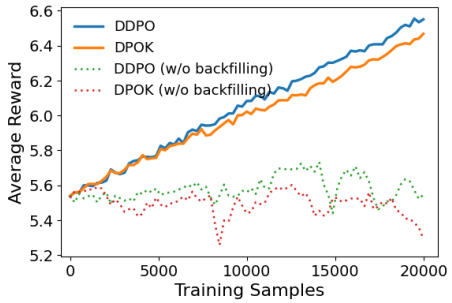

Figure 17: **Reward Backfilling.** A single reward cannot perform policy optimization (dashed line). Thus prior work backfills it to all earlier denoising steps (solid line), which enables optimization but inevitably causes action–reward mismatch.

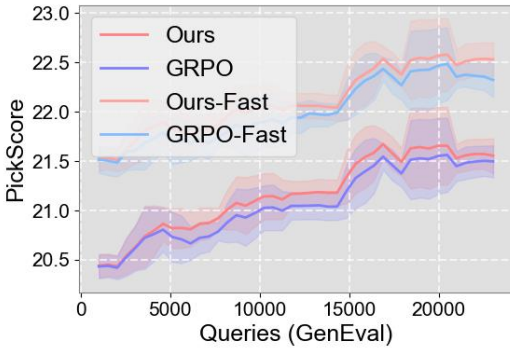

Figure 18: Results on FLUX1.0 model with exactly aligned and unchanged hyper-parameters.

Table 5: Investigation on the influnce of introducing Adaptivec Termination Strategy to the generated image quality. Results demonstrate adaptive termination can better maintain image quality.

Method	Aesthetic Score	CLIP Score	Inception Score	PickScore
DDPO (32)	0.6675	0.2819	16.27	20.98
DPOK (32)	0.6537	0.2931	16.15	21.40
Ours (ada.)	**0.6812**	**0.2939**	**17.07**	**21.62**

A.4 Reward Backfilling Induced Reward Hacking and Policy Misoptimization

A.4.1 Existing Problems:

In existing RL formulations for diffusion models, rewards are usually obtained only at the final step of the denoising process. As a result, all preceding denoising steps execute actions but receive no immediate reward feedback, making it impossible to evaluate the quality of each action and preventing policy optimization from proceeding normally (as shown by the dashed line in Fig. 2).

To alleviate the sparse reward problem caused by relying solely on the final-step reward, existing methods adopt a compromise strategy by equally backfilling the final reward to all preceding denoising steps (as shown by the solid line in Fig. 2). Although this restores policy optimization, it introduces the following problems:

- *(1) Action–reward mismatch.* Equal backfilling causes actions at different timesteps to receive the same reward, preventing reasonable evaluation of single-step contributions, leading to credit assignment failure and misleading policy optimization. As a result, the learned policy may not truly improve generation quality, yielding limited reward improvement in the task (see Fig. 2, 16, 9).
- *(2) Overexploitation and diversity degradation.* Since all timesteps optimize the same final reward, it easily leads to overfitting to this reward, i.e., the overexploitation discussed in this paper. This narrows the generation distribution, breaks the exploration–exploitation balance, and reduces diversity, and may even cause reward hacking in severe cases (see Fig. 15, 7).

A.4.2 Our Solution Approach:

- *(1) Addressing Action–Reward Mismatch.* To address the action–reward attribution error problem, we innovatively introduce an exploration reward that measures generation diversity based on the differences between images corresponding to adjacent latents. This reward assigns different reward signals to each intermediate step during the denoising pro-

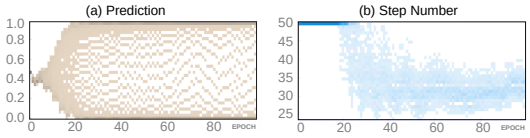

Figure 19: ATS Robustness

cess, thereby avoiding attribution bias caused by backfilling the same reward. Through this mechanism, we eliminate the action–reward attribution error problem and provide a clear evaluation score of exploration strength for each intermediate denoising action. Specifically, the credit-assignment issue is **determined by the total rewards**. After introducing the exploration reward in Eq.4, although the exploitation reward remains shared across timesteps, the transition-dependent exploration reward varies over timesteps, thereby producing **distinguishable total rewards** and alleviating the sparse-reward problem.

- *(2) Overexploitation and diversity degradation*. To achieve adaptive coordination between exploration and exploitation across different denoising stages, we propose an adaptive coordination mechanism. This mechanism encourages exploration of more diverse denoising paths in the early stage, while emphasizing stable convergence to high-reward image distributions in the later stage. Through this adaptive exploration–exploitation coordination, we avoid the problem that existing methods prematurely converge to a single generation mode due to overexploitation, and effectively alleviate the diversity loss caused by reward hacking.

A.5 Justification of the Adaptive Mechanism and Adaptive Network

A.5.1 Noise-level with Adaptive.

In the early stage, image noise is high, the main structure has not yet stabilized, and compositional uncertainty is large. Therefore, the generative policy has a broader space for exploration, and exploration should be particularly strengthened. In the later stage, as the noise decreases, the generation pattern becomes stable and uncertainty decreases significantly, causing the effective space available for exploration to shrink. Meanwhile, as denoising steps gradually approach the final exploitation reward, the training objective should shift from exploration to exploitation. Using noise prediction as an adaptive coefficient can realize the above motivation: high noise corresponds to the early stage of diffusion, where exploration should be enhanced; low noise corresponds to the tail end of denoising and is closer to the task (exploitation) reward, where exploitation should be strengthened. In RL, the core of the exploration–exploitation trade-off is to adjust exploration intensity according to uncertainty. In diffusion, the noise level is precisely a reflection of environmental uncertainty; therefore, it is reasonable to use noise to regulate exploration intensity.

A.5.2 Predictor Robustness & Prompt Complexity Effects.

Our predictor achieves robustness to complex prompts for two reasons: 1) Diffusion denoising is highly staged, where semantic alignment and major structures are mainly completed in early denoising (see Fig. 20), while ATS operates in the late denoising stage, making it less sensitive to prompt complexity. 2) The predictor achieves high accuracy across diverse prompts due to training on prompts with varying categories and complexities. In Tab. 6, we analyze ATS step of prompts consisting of 0-10 words, 10-20 words, and 20++ words. The results show that our method can adaptively adjust based on the prompt complexity, with a stable variation. Overall, predictor-controlled ATS truncates only RL training, not inference denoising, so occasional errors do not impair generation.

A.6 Why Does ATS Not Compromise Image Quality?

ATS truncates only the RL optimization, not the remaining denoising steps.

(1) *Diffusion*: denoising process is highly staged Ma et al. (2024) (see Fig. 20): early steps form global semantics, middle steps refine core details, and late steps mainly adjust marginal textures.

Table 6: ATS Step (Diff. Words)

Metric	[1,10)	[10,20)	20++
Mean	28.7	31.1	32.6
std	2.81	3.12	2.96

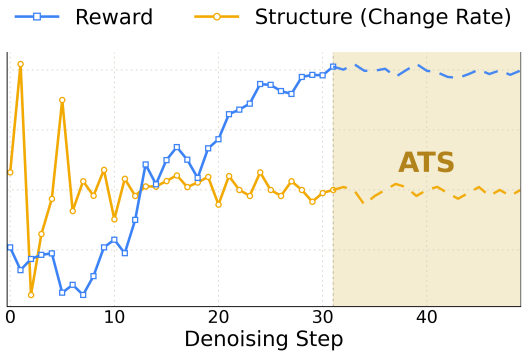

Figure 20: Reward&Structure change during denoising.

Since images are nearly converged in late denoising, truncating several RL steps causes negligible quality degradation.

(2) *Uncertainty*: RL learns by distinguishing different behaviors based on the different returns they yield. Once late denoising nearly converges, next-step generation becomes highly deterministic, limiting meaningful strategy differences for RL optimization. Continued optimization is therefore ineffective and may overfit local details, ***similar to repeatedly forcing a student who already knows the answer to solve the same problem again***.

B PROOF

B.1 PROOF OF THEOREM 1

Theorem 1 *Given the objective, the optimal policy p_{θ^*} has the following expression:*

$$\pi_{\theta^*}(\mathbf{a}_t|\mathbf{s}_t) \propto \exp\{\frac{Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_t)}{\alpha}\},$$

where

$$Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_t) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_t) + Q_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t),$$

$$Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_t) = R(\mathbf{s}_t, \mathbf{a}_t) + \alpha \cdot \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q_{\text{ext}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\},$$

and

$$Q_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t) = \beta r_{\text{exr}}(\mathbf{s}_t, \mathbf{a}_t) + \alpha \cdot \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q_{\text{exr}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\}.$$

Proof. According to Dai et al. (2018), we have

$$\pi_{\theta^*}(\mathbf{a}_t|\mathbf{s}_t) \propto \exp\{\frac{Q(\mathbf{s}_t, \mathbf{a}_t)}{\alpha}\},$$

where $Q(\mathbf{s}_t, \mathbf{a}_t) = R(\mathbf{s}_t, \mathbf{a}_t) + \alpha \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\}$.
From Q, we know that when $t \neq T-1$,

$$\exp\{Q(\mathbf{s}_t, \mathbf{a}_t))/\alpha\} = \exp\{\beta r_{exr}(\mathbf{s}_t, \mathbf{a}_t))/\alpha\} \cdot \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}))}{\alpha}\}$$

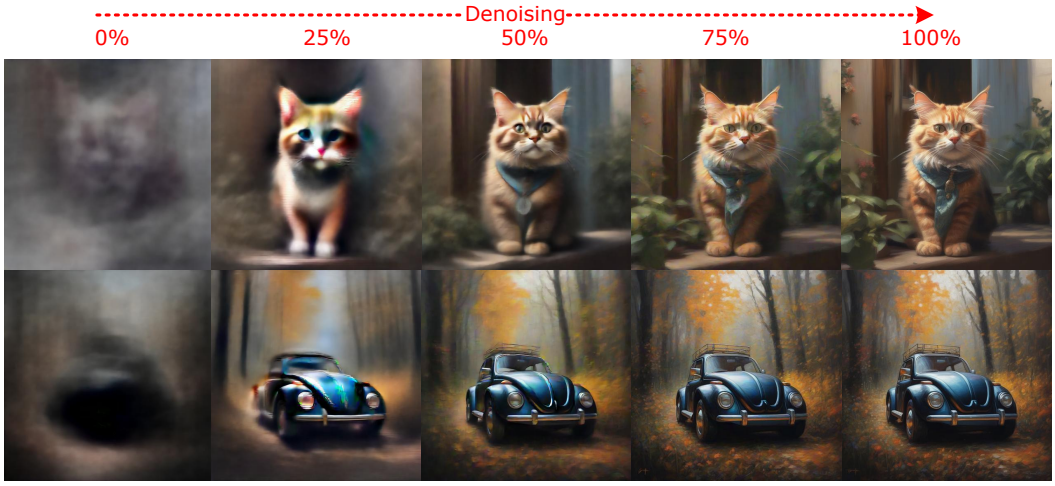

Figure 21: It can be seen that during the RL training process, when the denoising progress reaches around 75%, the image has largely converged and no longer undergoes significant changes. Therefore, early termination at this stage does not negatively impact image generation. Our adaptive early termination mechanism is implemented at this stage, effectively preventing excessive optimization of images that are close to convergence.

.

And when $t = T - 1$,

$$\exp\{Q(\mathbf{s}_{T-1}, \mathbf{a}_{T-1})/\alpha\} = \exp\{\beta R(\mathbf{s}_{T-1}, \mathbf{a}_{T-1})/\alpha\}$$

Therefore,

$$\begin{aligned}
&\exp\{Q(\mathbf{s}_t, \mathbf{a}_t)/\alpha\} \\
=&\exp\{\beta r_{exr}(\mathbf{s}_t, \mathbf{a}_t)/\alpha\} \cdot \sum_{\mathbf{a}_{t+1}} \exp\{\beta r_{exr}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})/\alpha\} \\
&\cdots \sum_{\mathbf{a}_{T-1}} \exp\{\beta r_{exr}(\mathbf{s}_{T-1}, \mathbf{a}_{T-1})/\alpha\} \cdot \exp\{\beta(\mathbf{s}_{T-1}, \mathbf{a}_{T-1})/\alpha\} \\
=&\exp\{Q_{ext}(\mathbf{s}_t, \mathbf{a}_t)/\alpha\} \cdot \exp\{Q_{exr}(\mathbf{s}_t, \mathbf{a}_t)/\alpha\}
\end{aligned}$$

It is easy to draw $Q(\mathbf{s}_t, \mathbf{a}_t) = Q_{ext}(\mathbf{s}_t, \mathbf{a}_t) + Q_{exr}(\mathbf{s}_t, \mathbf{a}_t)$.

□

B.2 Proof of Lemma 2.1

Lemma 2.1 *For any two actions $\mathbf{a}_i$ and $\mathbf{a}_j$ such that $Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_i) > Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_j)$, it holds that $0 < Q_{\text{exr}}(\mathbf{s}, \mathbf{a}_j) - Q_{\text{exr}}(\mathbf{s}, \mathbf{a}_i) < Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_i) - Q_{\text{ext}}(\mathbf{s}, \mathbf{a}_j)$, for any state $\mathbf{s}$, then we have*

$$H(\pi_{\text{ext}}(\cdot|\mathbf{s})) < H(\pi_{\text{total}}(\cdot|\mathbf{s})).$$

Proof. Consider a family of parameterized distributions

$$\pi_\theta(\mathbf{a}|\mathbf{s}) = \frac{\exp\{Q_{ext}(\mathbf{s}, \mathbf{a}) + \theta Q_{exr}(\mathbf{s}, \mathbf{a})\}}{Z(\theta)}, \quad \theta \in [0, 1],$$

where $Z(\theta) = \int_{\mathbf{a}} \exp\{Q_{ext}(\mathbf{s}, \mathbf{a}) + \theta Q_{exr}(\mathbf{s}, \mathbf{a})\} d\mathbf{a}$. Clearly, $\pi_0 = \pi_{ext}$ and $\pi_1 = \pi_{total}$. Define the entropy function

$$H(\theta) = H(\pi_\theta(\cdot|s)) = -\int \pi_\theta(\mathbf{a}|\mathbf{s}) \log \pi_\theta(\mathbf{a}|\mathbf{s}) \, d\mathbf{a}.$$

It suffices to prove that $H(\theta)$ is non-decreasing on $[0, 1]$, i.e., $H'(\theta) \geq 0$.

For brevity, denote $Q_{\mathbf{a}} = Q_{ext}(\mathbf{s}, \mathbf{a})$, $\delta_{\mathbf{a}} = Q_{exr}(\mathbf{s}, \mathbf{a})$, and let $\mathbb{E}_\theta[\cdot]$ represent the expectation under π_θ. First, compute the derivative of the distribution with respect to θ:

$$\frac{d\pi_\theta(\mathbf{a}|\mathbf{s})}{d\theta} = \pi_\theta(\mathbf{a}|\mathbf{s})\big(\delta_{\mathbf{a}} - \mathbb{E}_\theta[\delta_{\mathbf{a}}]\big).$$

The derivative of the entropy is then

$$\begin{aligned} H'(\theta) &= -\int \frac{d\pi_\theta(\mathbf{a}|\mathbf{s})}{d\theta} \log \pi_\theta(\mathbf{a}|\mathbf{s})\, d\mathbf{a} \\ &= -\int \pi_\theta(\mathbf{a}|\mathbf{s})\big(\delta_{\mathbf{a}} - \mathbb{E}_\theta[\delta_{\mathbf{a}}]\big) \log \pi_\theta(\mathbf{a}|\mathbf{s})\, d\mathbf{a} \\ &= -\mathbb{E}_\theta\big[\delta_{\mathbf{a}} \log \pi_\theta\big] + \mathbb{E}_\theta[\delta_{\mathbf{a}}] \cdot \mathbb{E}_\theta[\log \pi_\theta]. \end{aligned}$$

Since $\log \pi_\theta(\mathbf{a}|\mathbf{s}) = Q_a + \theta\delta_{\mathbf{a}} - \log Z(\theta)$, we have

$$\begin{aligned} \mathbb{E}_\theta[\log \pi_\theta] &= \mathbb{E}_\theta[Q_a] + \theta\mathbb{E}_\theta[\delta_{\mathbf{a}}] - \log Z(\theta), \\ \mathbb{E}_\theta[\delta_{\mathbf{a}} \log \pi_\theta] &= \mathbb{E}_\theta[\delta_{\mathbf{a}} Q_a] + \theta\mathbb{E}_\theta[\delta_{\mathbf{a}}^2] - \log Z(\theta)\mathbb{E}_\theta[\delta_{\mathbf{a}}]. \end{aligned}$$

Substituting these into the expression for $H'(\theta)$ and canceling the terms involving $\log Z(\theta)$ yields

$$\begin{aligned} H'(\theta) &= -\mathbb{E}_\theta[\delta_{\mathbf{a}} Q_a] - \theta\mathbb{E}_\theta[\delta_{\mathbf{a}}^2] + \mathbb{E}_\theta[\delta_{\mathbf{a}}]\mathbb{E}_\theta[Q_a] + \theta\big(\mathbb{E}_\theta[\delta_{\mathbf{a}}]\big)^2 \\ &= -\Big(\mathbb{E}_\theta[\delta_{\mathbf{a}}(Q_a + \theta\delta_{\mathbf{a}})] - \mathbb{E}_\theta[\delta_{\mathbf{a}}] \cdot \mathbb{E}_\theta[Q_a + \theta\delta_{\mathbf{a}}]\Big) \\ &= -\operatorname{Cov}_{\pi_\theta}\big(\delta_{\mathbf{a}},\, Q_a + \theta\delta_{\mathbf{a}}\big). \end{aligned}$$

Thus, establishing $H'(\theta) \geq 0$ is equivalent to showing

$$\operatorname{Cov}_{\pi_\theta}\big(\delta_{\mathbf{a}},\, Q_a + \theta\delta_{\mathbf{a}}\big) \leq 0, \qquad \forall\theta \in [0, 1].$$

From the assumption $0 \leq \delta_{\mathbf{a}_j} - \delta_{\mathbf{a}_i} \leq Q_{\mathbf{a}_i} - Q_{\mathbf{a}_j}$, it follows that if $Q_{\mathbf{a}_i} > Q_{\mathbf{a}_j}$, then $\delta_{\mathbf{a}_i} \leq \delta_{\mathbf{a}_j}$; hence $\delta_{\mathbf{a}}$ is monotonically decreasing in $Q_{\mathbf{a}}$. Moreover, for any $\theta \in [0, 1]$,

$$Q_{\mathbf{a}_i} - Q_{\mathbf{a}_j} \geq \delta_{\mathbf{a}_j} - \delta_{\mathbf{a}_i} \geq \theta(\delta_{\mathbf{a}_j} - \delta_{\mathbf{a}_i}),$$

which rearranges to $Q_{\mathbf{a}_i} + \theta\delta_{\mathbf{a}_i} \geq Q_{\mathbf{a}_j} + \theta\delta_{\mathbf{a}_j}$. Therefore, $Q_{\mathbf{a}} + \theta\delta_{\mathbf{a}}$ is monotonically increasing in $Q_{\mathbf{a}}$. Consequently, the functions $\delta_{\mathbf{a}}$ and $Q_{\mathbf{a}} + \theta\delta_{\mathbf{a}}$ are anti-monotonic with respect to $Q_{\mathbf{a}}$. By the covariance inequality (a Chebyshev's inequality) for any positive weight distribution, the covariance of two anti-monotonic functions is non-positive. Hence,

$$\operatorname{Cov}_{\pi_\theta}\big(\delta_{\mathbf{a}},\, Q_a + \theta\delta_{\mathbf{a}}\big) \leq 0,$$

and therefore $H'(\theta) \geq 0$. This implies that $H(\theta)$ is monotonically non-decreasing on $[0, 1]$, so

$$H(1) = H\big(\pi_{total}(\cdot|\mathbf{s})\big) \geq H(0) = H\big(\pi_{ext}(\cdot|\mathbf{s})\big).$$

□

B.3 PROOF OF THEOREM 2

Theorem 2 *Under conditions of Lemma 2.1 and different values of β function, for any s_t, policy optimization with the above Eq. 6 can have either*

$$H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) \leq H(\pi_{\text{total}}(\cdot|\mathbf{s}_t)) \text{or} H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) > H(\pi_{\text{total}}(\cdot|\mathbf{s}_t)).$$

Proof. According to Theorem 1, we have

$$\delta(\mathbf{s}_t,\mathbf{a}_t) \triangleq \beta(\mathbf{s}_t) r_{\text{exr}}(\mathbf{s}_t,\mathbf{a}_t) + \alpha \cdot \log \sum_{\mathbf{a}_{t+1}} \exp\{\frac{Q_{\text{exr}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})}{\alpha}\}.$$

Depending on $\beta(\mathbf{s}_t)$, the policy optimization can accommodate all three cases regarding the exploration-exploitation trade-off. For example, we take two actions $\mathbf{a}_1$ and $\mathbf{a}_2$ from the action space.

Case I - Exploration Dominant: for any $\mathbf{s}_t$, when $\beta(\mathbf{s}_t) \equiv 1$, $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_1) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) + \delta(\mathbf{s}_t, \mathbf{a}_1)$, and $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_2) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_2) + \delta(\mathbf{s}, \mathbf{a}_2)$. Because $\beta(\mathbf{s}_t) \equiv 1$, from Lemma 2.1, we know $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) < H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$, i.e., the entropy increases. In this case, the strength of exploration reaches its peak; our method exhibits stronger exploratory ability than methods without exploration rewards.

Case II - Adaptive Exploration-Exploitation: if $Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) \geq Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a})$, for any $\mathbf{a}$, and $\beta(\mathbf{s}_t)$ satisfies $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_1) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) + \delta(\mathbf{s}_t, \mathbf{a}_1)$ and $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_2) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_2)$, we have $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) > H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$, i.e., the probability of the optimal policy increases while total entropy starts to decrease. In this case, the degree of exploitation increases while the level of exploration decreases.

Case III - Exploitation Dominant: for any $\mathbf{s}$, when $\beta(\mathbf{s}_t) \equiv 0$, we have $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_1) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_1) + C$ and $Q_{\text{total}}(\mathbf{s}_t, \mathbf{a}_2) = Q_{\text{ext}}(\mathbf{s}_t, \mathbf{a}_2) + C$, where C is constant. The entropy remains unchanged, i.e., $H(\pi_{\text{ext}}(\cdot|\mathbf{s}_t)) = H(\pi_{\text{total}}(\cdot|\mathbf{s}_t))$. In this case, exploration is zeroed out, and exploitation dominates. □

B.4 THE DETAIL IMPLEMENTATION OF EQUATION EQUATION 2

We adopt the DDIM formulation and parameterize the step-wise variance, allowing us to construct an entropy-aware MDP, unlike previous works that rely on a fixed policy entropy. Specifically: Our step-wise mean and variance are parameterized as:

$$\begin{aligned}
\mu_\theta(\mathbf{x}_t, t) &= \sqrt{\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\, \epsilon_\theta^{(t)}(\mathbf{x}_t)}{\sqrt{\alpha_t}} \right) \\
&+ \sqrt{1 - \alpha_{t-1} - \eta_\theta^2 \frac{(1-\alpha_{t-1})}{(1-\alpha_t)} \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)}\, \epsilon_\theta^{(t)}(\mathbf{x}_t) \\
\sigma_{\theta,t} &= \eta_\theta \sqrt{\frac{(1-\alpha_{t-1})}{(1-\alpha_t)} \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)},
\end{aligned} \tag{7}$$

where $\eta_\theta = \eta_\theta(\mathbf{x}_t, t, \epsilon_\theta^t(\mathbf{x}_t))$ is a function that predicting the entropy based on the current step and denoising state.

C COMPUTATION DETAILS OF STRUCTURAL SIMILARITY INDEX (SSIM)

To quantitatively measure the structural similarity between two images, we adopt the Structural Similarity Index (SSIM) Wang et al. (2004). Given two aligned image patches x and y, SSIM evaluates their similarity by jointly comparing luminance, contrast, and structural information.

Formally, SSIM is defined as

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{8}$$

where μ_x and μ_y denote the mean intensities of patches x and y, respectively. σ_x^2 and σ_y^2 represent their variances, while σ_{xy} denotes the covariance between the two patches.

These statistics are computed as

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i,$$

$$\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i,$$

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2,$$

$$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2,$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y),$$

where N denotes the number of pixels in the local window.

The constants C_1 and C_2 are introduced to stabilize the division when the denominator is close to zero:

$$C_1 = (K_1 L)^2, \qquad C_2 = (K_2 L)^2, \tag{9}$$

where L is the dynamic range of pixel values (typically $L = 255$ for 8-bit images), and $K_1 = 0.01$, $K_2 = 0.03$.

Following the standard protocol Wang et al. (2004), SSIM is computed over local windows using a sliding Gaussian kernel (typically 11×11 with standard deviation 1.5), and the final SSIM score is obtained by averaging the SSIM values over all spatial locations:

$$\text{SSIM}(X, Y) = \frac{1}{M}\sum_{j=1}^{M} \text{SSIM}(x_j, y_j), \tag{10}$$

where M denotes the number of local windows. The resulting SSIM score ranges from -1 to 1, where larger values indicate higher structural similarity.

D Related Works

D.1 Text-to-Image Diffusion Models

Text-to-image diffusion models Croitoru et al. (2023); Kingma et al. (2021); Karras et al. (2024); Carlini et al. (2023); Gandikota et al. (2023) aim to generate semantically consistent and visually realistic images based on natural-language prompts. Existing methods are typically built upon large-scale paired image-text data, learning to progressively recover target images from random noise through conditional denoising training, while injecting linguistic semantics into the generation process via text encoders, cross-attention mechanisms, or conditional guidance strategies. This training paradigm enables models to effectively fit the visual distribution of the training data and establish basic correspondences between textual descriptions and image content Peebles & Xie (2023); Kazerouni et al. (2023); Watson et al. (2025); Fuest et al. (2026); Valevski et al. (2025). However, standard diffusion training essentially optimizes noise prediction or score-matching objectives, with supervisory signals derived from static image-text data rather than explicit human preferences regarding image quality, aesthetic style, structural plausibility, or specific task objectives. Consequently, although such models can generate images that are semantically reasonable overall, they struggle to directly optimize for particular preferences, such as improving visual aesthetics, strengthening responsiveness to complex prompts, enhancing text-rendering quality, or achieving stable trade-offs among multiple quality objectives. This limitation has prompted subsequent research to explore post-training methods for diffusion models based on preference learning, reward modeling, and reinforcement learning.

D.2 Reinforcement Learning in Diffusion Models

Reinforcement learning Gan et al. (2024); Yan et al. (2026); Towers et al. (2026); Aggarwal & Welleck (2025); Agrawal et al. (2025); Liu et al. (2026) has been increasingly adopted to fine-tune diffusion models for improving alignment with human preferences or task-specific objectives. Most existing approaches treat the diffusion model as a conditional generative policy and optimize it using preference or task rewards computed from the final generated image. This paradigm enables preference alignment without explicit environment interaction and has demonstrated promising empirical performance.

Due to the long-horizon nature of the denoising process, directly optimizing diffusion models with terminal rewards often suffers from severe reward sparsity and training instability. To address this issue, many methods propagate the final reward to all denoising steps Fan et al. (2023); Black et al. (2023); Franceschelli & Musolesi (2024); Chen et al. (2024); Uehara et al. (2024); Yan et al. (2025), allowing intermediate actions to receive learning signals during optimization. This reward backpropagation strategy has become a common practice and plays a key role in making RL-based diffusion fine-tuning feasible in practice.

However, existing RL-based diffusion methods largely rely on fixed reward propagation and optimization schemes, implicitly assuming uniform contributions of different denoising steps. As a result, the dynamic behavior of exploration and exploitation during denoising is rarely analyzed Ahmed et al. (2019); Schulman et al. (2015; 2017); Bellemare et al. (2016); Tokic (2010). In particular, how exploration should be encouraged in early stages and gradually coordinated with exploitation as denoising progresses remains underexplored. This limitation motivates a more principled investigation into exploration–exploitation coordination tailored to diffusion model fine-tuning.

D.3 Reward Hacking

Reward hacking Skalse et al. (2022); Eisenstein et al. (2023) has long been recognized as a critical issue in reinforcement learning, where policies exploit imperfections or shortcuts in reward functions rather than accomplishing the intended objectives. Prior studies have shown that such behavior often emerges when optimization overly emphasizes reward maximization without sufficient constraints on policy behavior.

In the context of diffusion model fine-tuning, reward hacking manifests in a distinct and structured manner. Unlike standard RL settings, diffusion models rely on long-horizon denoising trajectories, where rewards are typically defined only at the final generation step. To alleviate reward sparsity, most existing methods propagate terminal rewards to all intermediate denoising steps. While effective for stabilizing training, this design implicitly removes distinctions among denoising actions and encourages step-agnostic optimization behaviors.

As a consequence, policies may converge to degenerate generation patterns that achieve high reward scores while sacrificing diversity or fidelity. Recent works have empirically observed such phenomena in diffusion fine-tuning, particularly under preference-based optimization, yet these behaviors are often treated as empirical artifacts rather than symptoms of a deeper optimization imbalance. A systematic understanding of reward hacking from the perspective of exploration–exploitation coordination in diffusion models remains largely unexplored.

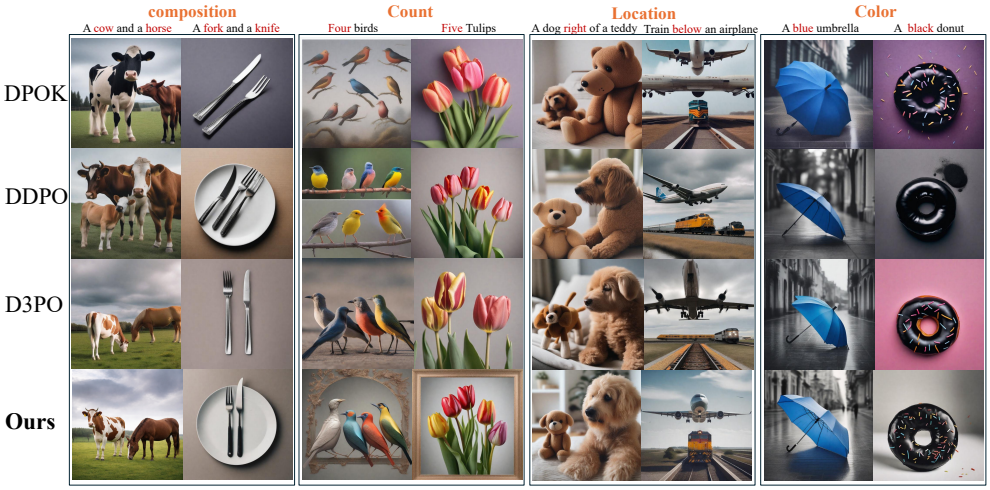

Figure 22: **Qualitative Comparison on Controlled Generation. (SDXL)** All results are obtained via prompts from the GenEval dataset.

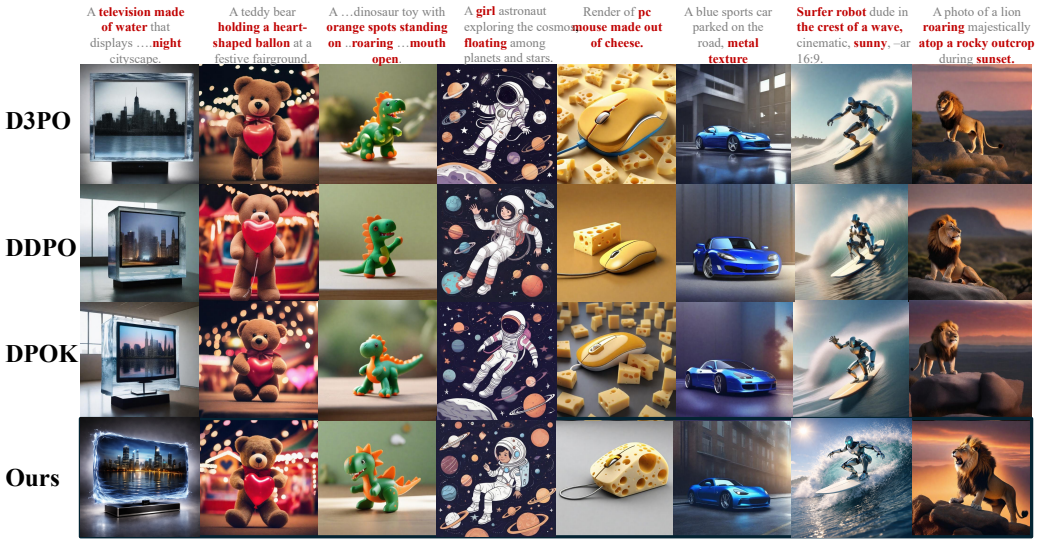

Figure 23: Qualitative evaluation on semantic alignment and overall generative quality with SD-XL.



Figure 24: **Qualitative Comparison on Controlled Generation. (SD3.5)** All results are obtained via prompts from the GenEval dataset.

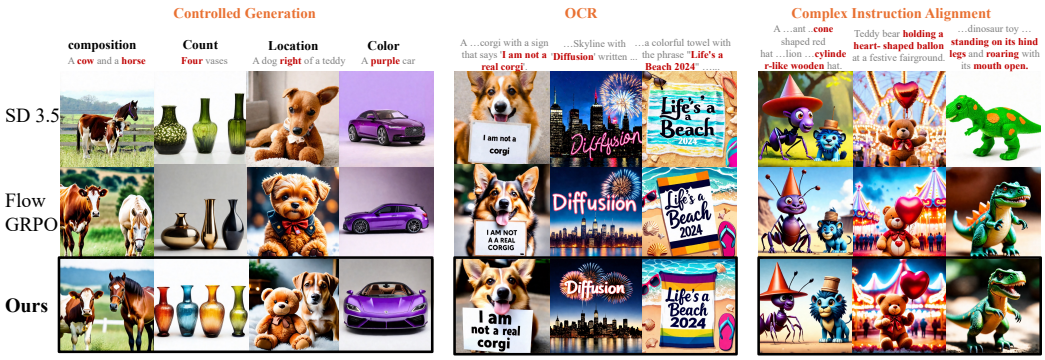

Figure 25: **Controlled Generation, OCR, Complex Instruction Alignment**.



Figure 26: Qualitative evaluation on semantic alignment and overall generative quality with SD3.5.



Figure 27: We visualized the changes in images as the training epochs increased, starting from the same initial image and optimized by different methods. Compared to the baseline method, our approach performs better in terms of object pose and color diversity, demonstrating stronger adaptability during training. It effectively mitigates reward hacking and identifies improved optimization results.



Figure 28: **Qualitative Results on SD35 Model.**

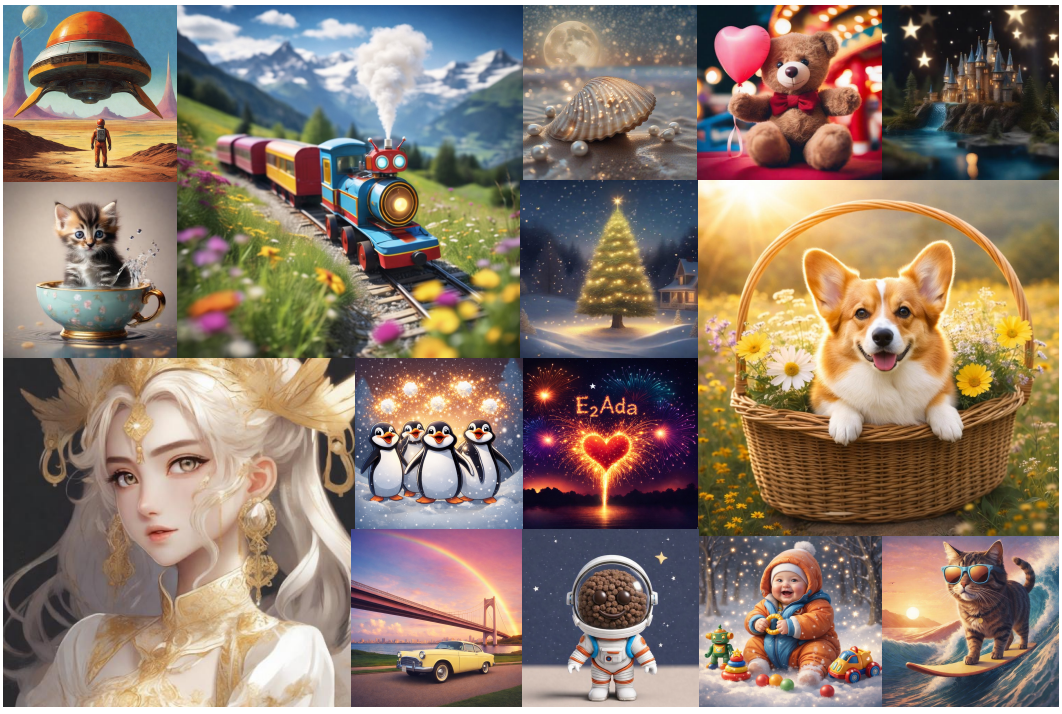

Figure 29: **Qualitative Results on SDXL Model.**