Beyond the Dirac Delta: Mitigating Diversity Collapse in Reinforcement Fine-Tuning for Image Generation

Jinmei Liu[1] **Haoru Li**[1] **Zhenhong Sun**[2] **Chaofeng Chen**[3] **Yatao Bian**[4]
Hongdong Li[2] **Bo Wang**[1] **Daoyi Dong**[5] **Zhi Wang**[1*]

[1] Nanjing University [2] Australian National University [3] Wuhan University
[4] National University of Singapore [5] University of Technology Sydney

jmliu@smail.nju.edu.cn zhiwang@nju.edu.cn

Abstract

Reinforcement learning (RL) has emerged as a paradigm for fine-tuning large-scale generative models, such as diffusion and flow models, to align with complex human preferences and user-specified tasks. A fundamental limitation remains *the curse of diversity collapse*, where the objective formulation and optimization landscape inherently collapse the policy to a Dirac delta distribution. To address this challenge, we propose **DRIFT** (**D**ive**R**sity-**I**ncentivized Reinforcement **F**ine-**T**uning for Versatile Image Generation), an innovative framework that systematically incentivizes output diversity throughout the on-policy fine-tuning process, reconciling strong task alignment with high generation diversity to enhance versatility essential for applications that demand diverse candidate generations. We approach the problem across three representative perspectives: i) **sampling** a reward-concentrated subset that filters out reward outliers to prevent premature collapse; ii) **prompting** with stochastic variations to expand the conditioning space, and iii) **optimization** of the intra-group diversity with a potential-based reward shaping mechanism. Experimental results show that DRIFT exhibits clear Pareto dominance in task alignment and generation diversity, achieving 3.88%$\sim$81.42% higher diversity at matched alignment and 14.38%$\sim$56.45% higher alignment at matched diversity. Our code is available at https://github.com/NJU-RL/DRIFT.

1 Introduction

With advancements in large-scale models trained on massive data, image generation models (e.g., diffusion or flow models) have demonstrated success in realistic image synthesis under diverse forms of guidance [Lipman et al., 2023, Dombrowski et al., 2025], including image-based [Zhang et al., 2024a, Sun et al., 2026], classifier-based [Dhariwal and Nichol, 2021], text-based guidance [Rombach et al., 2022], and structure-aware generation [Sun et al., 2024, Wang et al., 2024]. Despite these advances, they are often deployed in scenarios that are not directly aligned with their likelihood-based training objectives. As a result, even state-of-the-art models [Podell et al., 2024, Betker et al., 2023] frequently exhibit misalignment with user prompts [Jiang et al., 2024] or human preferences [Xu et al., 2023]. Recent studies explore using reinforcement learning (RL) to fine-tune image generation models by formulating the multi-step denoising process as a sequential decision-making task [Fan et al., 2023, Black et al., 2024], achieving effective optimization for downstream tasks using only a black-box reward function [Wallace et al., 2024, Xue et al., 2025, Liu et al., 2025a].

*Correspondence to Zhi Wang <zhiwang@nju.edu.cn>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

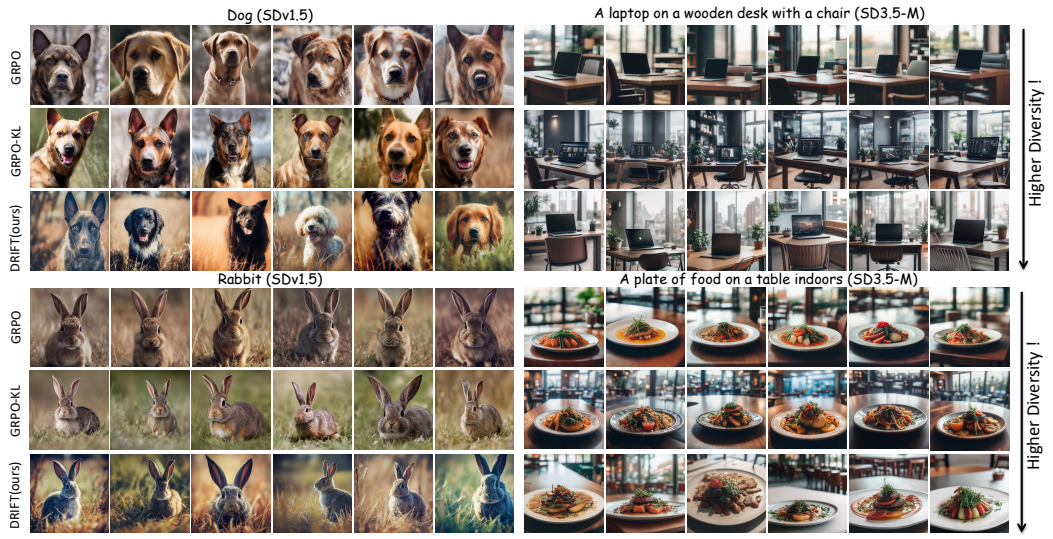

Figure 1: Image generation models fine-tuned via RL often suffer from *diversity collapse*, resulting in repetitive outputs with similar subject appearances and scene compositions. Instead, DRIFT maintains both high fidelity and diversity. The three methods used identical prompts and random seeds for generation, with SDv1.5 and SD3.5-M fine-tuned using PickScore as the reward function.

However, a fundamental limitation of RL fine-tuning methods is *diversity collapse*, where the fine-tuned model achieves high rewards but loses output diversity [Barceló et al., 2024], producing monotonous single-pattern images [Jena et al., 2025, Hu et al., 2026]. This phenomenon is not only a consequence of the imperfect reward design, but an inherent "curse" of the RL fine-tuning paradigm, as it continuously increases the likelihood of high-reward regions while substantially narrowing the overall coverage of the model [Yue et al., 2025, Cui et al., 2025, Yan et al., 2025]. A natural remedy is to incorporate Kullback-Leibler (KL) regularization between the fine-tuned and pre-trained models, thereby discouraging over-optimization [Fan et al., 2023, Liu et al., 2025a]. However, simply constraining the policy to the base model limits versatility in learning new content, resulting in suboptimal alignment to diverse downstream tasks [Hong et al., 2026]. Recent studies [Jena et al., 2025, Sorokin et al., 2025, Gandikota and Bau, 2025, Sadat et al., 2024] explore various sampling strategies to balance alignment and diversity, but rely on hand-tuned inference-time heuristics and do not address diversity collapse during RL fine-tuning. These limitations underscore the need for effective solutions to mitigate diversity collapse and enable RL for versatile image generation.

In this paper, we propose **DRIFT** (**D**ive**R**sity-**I**ncentivized Reinforcement **F**ine-**T**uning for Versatile Image Generation), an innovative framework that systematically injects diversity incentives into RL fine-tuning to achieve both strong task alignment and high generation diversity, promoting versatility essential for applications that demand diverse candidate generations. We first analyze the *curse of diversity collapse* in the mainstream paradigm, showing that the objective and optimization landscape inevitably collapse the policy to a Dirac delta distribution. This motivates explicitly embedding diversity incentives into RL fine-tuning, enabling broader exploration of diverse generations during on-policy learning. To this end, we approach the problem from three key dimensions: i) **sampling**, we sample a reward-concentrated subset for policy update, filtering out reward outliers to prevent premature collapse; ii) **prompting**, we introduce stochastic variations to prompt inputs, expanding the conditioning space from which image groups are generated; and iii) **optimization**, we formulate the intra-group diversity as an intrinsic reward and design a potential-based reward shaping scheme to preserve optimal policy invariance. We anticipate that this study will provide a coherent view for further research into diversity-preserving techniques for versatile image generation.

Experimental results show that DRIFT consistently achieves superior Pareto dominance regarding task alignment and diversity across various reward models. Compared to competitive baselines, DRIFT yields a 3.88%$\sim$81.42% increase in diversity at matched alignment and a 14.38%$\sim$56.45% increase in alignment at matched diversity. Comprehensive analysis and case studies reveal that DRIFT mitigates diversity collapse from multiple perspectives.

2 Related Work

RL Fine-Tuning of Generative Models. Methods can be roughly categorized into three kinds according to the reward function. The first kind uses differentiable reward functions and backpropagates the reward function gradient through the full sampling procedure [Prabhudesai et al., 2023, Clark et al., 2024, Jia et al., 2025]. The second uses preference-based reward functions derived from human comparison data to align with human preferences, such as DPOK [Fan et al., 2023], Diffusion-DPO [Wallace et al., 2024], and Diffusion-KTO [Li et al., 2024]. The third uses a general form of black-box reward functions. DDPO [Black et al., 2024] formulates a policy gradient algorithm to optimize diffusion models, followed by B^2-DiffuRL [Hu et al., 2025] that tackles the sparse-reward challenge. Flow-GRPO [Liu et al., 2025a] first integrates online RL into flow matching models, with an ODE-to-SDE conversion to enable statistical sampling for exploration. DanceGRPO [Xue et al., 2025] further scales RL to large and diverse prompt sets for visual generation tasks. The expanding studies underscore the efficacy and practical utility of RL fine-tuning for large-scale generative models [Uehara et al., 2024, Chen et al., 2024, Ye et al., 2024, Li et al., 2025, Zheng et al., 2025].

Diversity in Image Generation. Maintaining generation diversity has long been a central challenge in the field of image synthesis [Zhang et al., 2024b, Dombrowski et al., 2025, Li et al., 2026]. The standard remedy to preserve diversity is using KL regularization to constrain the deviation from the base model [Zhai et al., 2025, Ye et al., 2025], which can degrade the model's versatility in learning new content [Hong et al., 2026]. Barceló et al. [2024] propose a hierarchical approach to preserve high-level diversity by only fine-tuning the low-level features at later denoising timesteps. Miao et al. [2024] fine-tune diffusion models using only a diversity reward to enhance generation diversity, while requiring a set of unbiased images for reference. Another line approaches the problem by adopting various sampling strategies at inference time, such as condition-annealed sampling [Sadat et al., 2024], annealed importance guidance [Jena et al., 2025], and combined generation [Sorokin et al., 2025]. However, these methods rely on hand-tuned heuristics for guided sampling and fail to address the fundamental problem of diversity collapse during RL fine-tuning.

DiverseGRPO [Liu et al., 2026] and GARDO [He et al., 2025] are representative concurrent efforts that address diversity preservation in RL fine-tuning. DiverseGRPO preserves diversity via spectral clustering and stronger early-stage KL constraints, while GARDO mitigates reward hacking using gated KL penalties and diversity-weighted advantage reweighting. In contrast, we delve into the curse of diversity collapse within the KL-constrained paradigm, tackling the challenge through a systematic framework spanning sampling, prompting, and optimization.

3 Preliminaries

Denoising as Sequential Decision-Making. Diffusion/flow models transform a data distribution $p(\mathbf{x}_0|\mathbf{c})$ over a dataset of samples $\mathbf{x}_0$ and prompts $\mathbf{c}$ through a sequential Markovian forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ that iteratively adds noise to data. Sampling from a trained model θ begins with drawing a random $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and following the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$ to produce a trajectory $\{\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, ..., \boldsymbol{x}_0\}$ ending with the clean $\boldsymbol{x}_0$. Following common practice [Rafailov et al., 2023, Liu et al., 2025a], RL fine-tuning of a pre-trained model is formulated as a KL-constrained reward maximization problem as

$$J(\theta) = \mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_0 \sim p_\theta(\boldsymbol{x}_0|\boldsymbol{c})} \left[r(\boldsymbol{x}_0, \boldsymbol{c}) \right] - \beta \cdot \mathbb{D}_{\text{KL}} \left[\pi_\theta(\boldsymbol{x}_0|\boldsymbol{c}) \| \pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c}) \right], \tag{1}$$

where $r(\boldsymbol{x}_0, \boldsymbol{c})$ is the reward defined over samples and prompts, and β controls the deviation from a reference policy π_{ref}, usually the initial pre-trained model. The iterative denoising process is mapped to an MDP as $\boldsymbol{s}_t \triangleq (\boldsymbol{x}_t, \boldsymbol{c}, t)$, $\boldsymbol{a}_t \triangleq \boldsymbol{x}_{t-1}$, $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t) \triangleq p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$, $P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \triangleq (\delta_{\boldsymbol{c}}, \delta_{t-1}, \delta_{\boldsymbol{x}_{t-1}})$, $R(\boldsymbol{s}_t, \boldsymbol{a}_t) \triangleq \mathbb{1}(t=0) \cdot r(\boldsymbol{x}_0, \boldsymbol{c})$, where $\boldsymbol{s}_t / \boldsymbol{a}_t$ denotes state/action, π is the policy, and δ_y is the Dirac delta distribution with nonzero density only at y. The generative model serves as the policy network π_θ and parameterizes the transition kernel P.

Group Relative Policy Optimization (GRPO). We build upon GRPO [Shao et al., 2024, Xue et al., 2025, Liu et al., 2025a] to estimate policy gradients for the objective in Eq. 1. Given a prompt $\boldsymbol{c}$, a group of G images $\{\boldsymbol{x}_0^1, ..., \boldsymbol{x}_0^G\}$ is generated from the sampling distribution $p_{\theta_{\text{old}}}(\boldsymbol{x}_0|\boldsymbol{c})$ with previous parameters θ_{old}, yielding rewards $\{r_1, ..., r_G\}$. The advantage for each sample $\boldsymbol{x}_0^i$ is calculated from

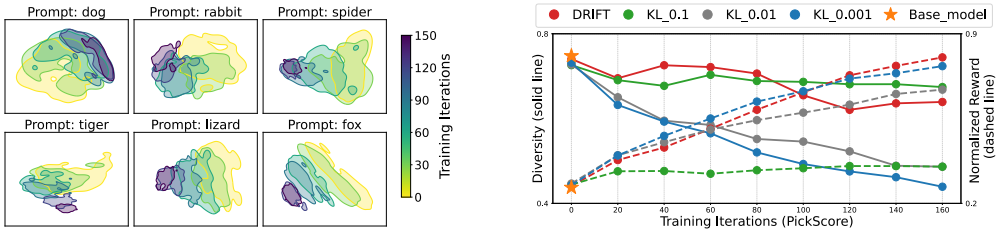

Figure 2: **Left**: Diversity collapse of the policy distribution $\pi_\theta(\boldsymbol{x}_0|\boldsymbol{c})$ during GRPO fine-tuning, visualized by kernel density estimation contours of DreamSim embeddings, where the support area shrinks monotonically over training. **Right**: Comparison of reward-diversity tradeoffs, where DRIFT outperforms GRPO-KL by preserving significantly higher diversity at equivalent reward levels.

group-level comparisons as

$$A_i = \frac{r_i - \text{mean}\left(\{r_1, r_2, \cdots, r_G\}\right)}{\text{std}\left(\{r_1, r_2, \cdots, r_G\}\right)}. \tag{2}$$

The policy π_θ is then updated by maximizing:

$$\mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_0^i \sim p_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^{G} \sum_{t=0}^{T} \text{CLIP}\left(\rho_{t,i} A_i\right) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta || \pi_{\text{ref}}] \right],$$

where $\rho_{t,i} = \frac{\pi_\theta(\mathbf{a}_{t,i}|\mathbf{s}_{t,i})}{\pi_{\theta_{\text{old}}}(\mathbf{a}_{t,i}|\mathbf{s}_{t,i})}$ is the importance sampling ratio at timestep t in sample $\boldsymbol{x}_0^i$. The clipped objective $\text{CLIP}(\rho_{t,i} A_i) = \min\left(\rho_{t,i} A_i, \text{clip}\left(\rho_{t,i}, 1-\epsilon, 1+\epsilon\right) A_i\right)$ ensures stable updates within the trust region [Schulman et al., 2017], where ϵ is a preset threshold constraining the discrepancy between the target policy π_θ and previous $\pi_{\theta_{\text{old}}}$.

4 The Curse of Diversity Collapse

Using a general derivation from reward-weighted regression [Peters and Schaal, 2007], the optimal solution to the reward maximization objective in Eq. 1 takes the form:

$$\pi^*(\boldsymbol{x}_0|\boldsymbol{c}) = \frac{1}{Z(\boldsymbol{c})} \pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c}) \exp\left(\frac{1}{\beta} r(\boldsymbol{x}_0, \boldsymbol{c})\right), \tag{3}$$

where $Z(\boldsymbol{c}) = \sum_{\boldsymbol{x}_0} \pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c}) \exp\left(\frac{1}{\beta} r(\boldsymbol{x}_0, \boldsymbol{c})\right)$ is the partition function. Appendix B.1 presents a complete derivation.

A large β leads to under-optimization, where the model becomes too conservative to effectively explore the reward landscape, ultimately failing to achieve significant gains in task alignment. Lowering β induces a rapid sharpening of the π^* distribution, as the exponential weighting term amplifies minor reward variations into massive probability gaps. When β approaches zero, i.e., the objective in Eq. 1 reduces to unconstrained reward maximization. Consequently, the model suffers from *total diversity collapse* as the policy converges to the singular maximum-reward mode as

$$\lim_{\beta \to 0^+} \pi_\beta^*(\boldsymbol{x}_0|\boldsymbol{c}) = \delta\left(\boldsymbol{x}_0 - \arg\max_{\boldsymbol{x}_0} r(\boldsymbol{x}_0, \boldsymbol{c})\right), \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta distribution that collapses all probability mass onto the origin (for details, see Appendix B.2). From the optimization perspective, the policy gradient for the objective in Eq. 1 is derived as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta \cdot \left(r - \beta \log \frac{\pi_\theta}{\pi_{\text{ref}}}\right)\right] = \mathbb{E}_{\pi_\theta}[\underbrace{\nabla_\theta \log \pi_\theta \cdot r}_{\text{Reward Pull}} - \underbrace{\beta \nabla_\theta \log \pi_\theta \cdot \log \frac{\pi_\theta}{\pi_{\text{ref}}}}_{\text{Diversity Push-back}}]. \tag{5}$$

Since the *on-policy gradient* is calculated as an expectation over the current policy π_θ, as soon as π_θ shifts slightly toward a high-reward region, the model starts sampling from that region more frequently. The KL constraint provides a push-back signal only for samples generated by π_θ; once low-reward

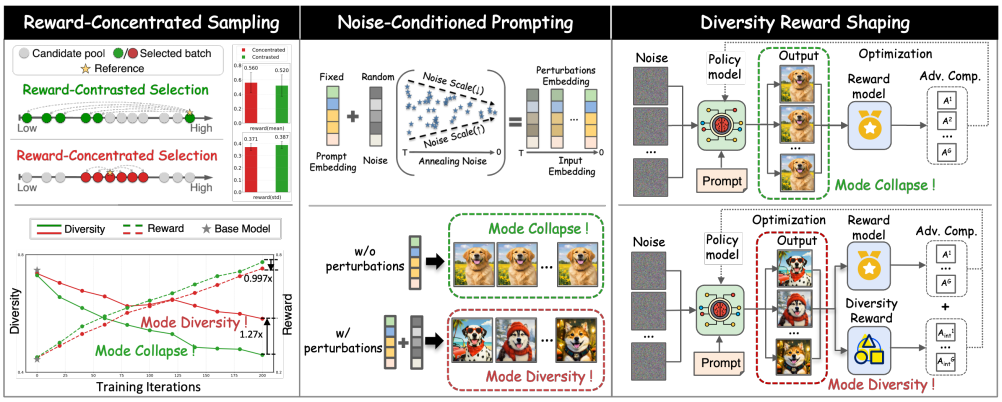

Figure 3: Overview of DRIFT, a unified framework for mitigating diversity collapse across sampling, prompting, and optimization.

yet diverse regions are no longer sampled, their gradient signal vanishes, consistent with observations in [Ye et al., 2025]. As an on-policy method, RL further drives the model toward out-of-distribution regions where π_{ref} offers insufficient coverage and weak regularization, an effect amplified by multi-step Markovian sampling as discrepancies accumulate along the denoising trajectory. This analysis shows that diversity collapse arises from (i) a fundamental limitation of the objective formulation (Eqs. 3–4) and (ii) local minima in the optimization landscape (Eq. 5), as empirically validated in Figure 2, making its mitigation essential for scaling RL compute toward versatile image generation. Here, "Beyond the Dirac Delta" refers to mitigating diversity collapse during on-policy fine-tuning, rather than guaranteeing its complete elimination. As Eq. 4 shows, assuming a unique reward maximizer within the reference policy's support, the optimal policy still converges to a Dirac Delta distribution in the theoretical limit $\beta \to 0^+$.

5 Mitigating Diversity Collapse

In this section, we present DRIFT, a framework for versatile image generation that embeds diversity incentives into on-policy fine-tuning from three dimensions: sampling, prompting, and optimization.

5.1 Reward-Concentrated Sampling

As shown in Eq. 3, RL fine-tuning shifts the reference policy toward an optimal distribution through the exponential weighting term $\exp(\frac{1}{\beta} r(\boldsymbol{x}_0, \boldsymbol{c}))$. In a GRPO-based setting, substantial reward variance within a group can be amplified into significant probability gaps in the updated policy, thereby exacerbating diversity collapse. Motivated by this insight, we update the policy using a concentrated subset that suppresses outliers which would otherwise over-sharpen the policy distribution.

Specifically, we adopt a simple selective sampling regime during fine-tuning. At each training iteration, the policy generates a pool of $2G$ candidate outputs for each prompt, from which G samples are selected to construct the training batch. For each candidate, we compute its cumulative distance to its $G-1$ nearest neighbors under a sample-level distance metric. The sample yielding the minimal sum is identified as the reference point, and the training subset is constructed by grouping this reference with its $G-1$ nearest neighbors. This procedure selects a locally coherent group and empirically produces a more concentrated, yet still informative, reward distribution within the selected group. The details are provided in Appendix F.

5.2 Noise-Conditioned Prompting

Beyond sample selection, we investigate how inducing stochasticity at the prompting stage influences output diversity during RFT. Specifically, we introduce controlled noise perturbations to the prompt embeddings to induce a broader distribution of generated images. For a given prompt $\boldsymbol{c}$, we add Gaussian noise to its embedding $\boldsymbol{e}$ at each denoising timestep t, resulting in a time-dependent

perturbed prompt embedding $\tilde{\boldsymbol{e}}_t$ as

$$\tilde{\boldsymbol{e}}_t = \sqrt{w(t)}\,\boldsymbol{e} + \eta \cdot \sqrt{1 - w(t)}\,\boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{6}$$

where η controls the overall noise scale and $w(t) \in [0, 1]$ is an annealing function that decreases with timestep t. In this manner, larger perturbations are applied at earlier timesteps and gradually reduced as the denoising process proceeds. The annealing schedule aligns prompt-level stochasticity with the diffusion dynamics, encouraging diverse global structures while preserving fine-grained semantic consistency at later stages. Details are provided in Appendix G.

5.3 Diversity Optimization via Reward Shaping

We incorporate the output diversity as an auxiliary objective, enabling its direct optimization throughout the training process. A straightforward approach is to include the intra-group diversity as an additional reward to guide fine-tuning. However, such naive reward shaping can shift the optimal policy, potentially misleading the model toward learning suboptimal policies [Ng et al., 1999]. Therefore, we employ a potential-based reward shaping scheme to allow for the inclusion of the intrinsic reward without distorting the original objective [Wang et al., 2023, Müller and Kudenko, 2025]. Formally, we formulate the **intrinsic reward** R_{int} as the difference between the diversities of adjacent states as

$$R_{\text{int}}(s_t, a_t, s_{t+1}) = \gamma d(s_{t+1}) - d(s_t), \tag{7}$$

where the diversity function $d(\cdot)$ is exactly the potential function over states $s \in \mathcal{S}$. In the setting of the denoising MDP in Sec. 3, the specific intrinsic reward becomes

$$R_{\text{int}}\big((\boldsymbol{x}_t^i, \boldsymbol{c}, t), \boldsymbol{x}_{t-1}^i, (\boldsymbol{x}_{t-1}^i, \boldsymbol{c}, t-1)\big) = \gamma d(\boldsymbol{x}_{t-1}^i) - d(\boldsymbol{x}_t^i),$$

where $i = 1, ..., G$. Since our method is built upon GRPO, which calculates policy gradients at the trajectory level, the intrinsic reward for a complete denoising process $\tau = (\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, ..., \boldsymbol{x}_0)$ is computed as

$$\begin{aligned} R_{\text{int}}(\tau^i) &= \sum\nolimits_{t=0}^{T-1} \gamma^t R_{\text{int}}(s_t, a_t, s_{t+1}) = \sum\nolimits_{t=0}^{T-1} \gamma^t \big[\gamma d(\boldsymbol{x}_{T-t-1}^i) - d(\boldsymbol{x}_{T-t}^i)\big] \\ &= \gamma^T d(\boldsymbol{x}_0^i) - d(\boldsymbol{x}_T^i) = \gamma^T d(\boldsymbol{x}_0^i). \end{aligned} \tag{8}$$

Here, we assume that the diversity of initial noise $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is zero, i.e., $d(\boldsymbol{x}_T^i) = 0$. The diversity of the final output $d(\boldsymbol{x}_0^i)$ is defined as the intra-group dissimilarity with respect to other samples within the same generation group. Appendix C provides details on $d(\boldsymbol{x}_0^i)$ computation.

The diversity-incentivized intrinsic reward enables the model to maintain higher output diversity without compromising the primary objective of task alignment. This mechanism effectively prevents the fine-tuned model from collapsing into narrow solution modes, promoting a balanced tradeoff between generation quality and diversity. Accordingly, the new reward function is defined as

$$\tilde{R}(\tau^i) = R(\tau^i) + \lambda \cdot R_{\text{int}}(\tau^i), \tag{9}$$

where λ is the shaping ratio that balances between quality and diversity. Finally, we substitute the augmented reward function $\tilde{r}_i = \tilde{R}(\tau^i)$ for the original one $r_i = R(\tau^i)$ in Eq. 2 to compute the advantage under the GRPO framework.

Incorporating the intrinsic reward R_{int} transforms the original MDP $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ into $\tilde{M} = (\mathcal{S}, \mathcal{A}, P, \tilde{R}, \gamma)$, where $\tilde{R} = R + \lambda R_{\text{int}}$. As we optimize a policy for the transformed MDP $\tilde{M}$ with the intention of deploying it within the original MDP M, it is critical to ensure that reward shaping does not bias the agent toward suboptimal solutions. Theorem 1 establishes that optimal policy invariance is preserved when incorporating intra-group diversity as an intrinsic reward, providing a theoretical foundation for our reward shaping mechanism. Appendix A presents a detailed proof.

Theorem 1 (Optimal Policy Invariance)**.** *Let $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ denote the MDP for the task of fine-tuning diffusion models with RL. $d(\cdot) : S \mapsto \mathbb{R}$ is a real-valued function that computes the intra-group diversity $d(s)$ of the state s within a group of generation samples. We formulate $R_{int}(\cdot) : S \times A \times S \mapsto \mathbb{R}$ as an intrinsic reward function that is the difference between the diversities of adjacent states, such that for all $s \in S, a \in A, s' \in S$, $R_{int}(s, a, s') = \gamma d(s') - d(s)$. Then, with any constant balancing ratio λ, every optimal policy in the transformed MDP $\tilde{M} = (\mathcal{S}, \mathcal{A}, P, R + \lambda R_{int}, \gamma)$ will also be an optimal policy in M, and vice versa.*

The ingenuity of our reward-shaping design lies in that, by defining the shaping reward as the diversity difference between adjacent states (Eq. 7), the intrinsic reward for a complete denoising trajectory is derived as the diversity of the clean image, as in Eq. 8. This elegant formulation bypasses the need to evaluate intermediate noised images, which significantly reduces computation while minimizing the influence of noisy factors on the diversity calculation.

Clipped Reward Shaping. Although DRIFT preserves optimal policy invariance under reward shaping, the model may over-exploit the diversity bonus at the expense of the primary objective, i.e., the *reward hacking* phenomenon, where an RL agent exploits flaws or ambiguities in the reward function to achieve high rewards without truly solving the intended task. Because diversity rewards are high-variance and outlier-sensitive, noisy or low-quality outputs off the target manifold can yield exaggerated diversity signals, steering optimization toward dispersed but sub-optimal samples. To address this issue, we clip the diversity reward as $R_{\text{int}}(\tau^i) = \text{clip}\big(R_{\text{int}}(\tau^i),\, 0,\, \sigma\big)$, where σ is an upper bound that prevents excessive exploitation of the shaping reward.

Decoupled Advantage Computation. During training, both the primary reward R and the intrinsic reward R_{int} evolve continuously as the policy improves, yet they typically operate on disparate and dynamically shifting scales. The magnitude of the intrinsic reward remains stable, since it is calculated based on the relative distance between samples within the group. However, the magnitude of the primary reward can vary substantially across different training stages and reward models. This complicates the development of a flexible weighting scheme for aggregating the two rewards as in Eq. 9. To circumvent this issue, we instead compute the advantages for each reward component independently, and merge them at the advantage level as $\tilde{A} = A + \lambda \cdot A_{\text{int}}$. By decoupling advantage estimation, each reward signal is normalized relative to its own baseline before aggregation. This design mitigates sensitivity to scale mismatch and temporal fluctuations, facilitating more robust and interpretable control over diversity regularization.

6 Experiments

Models. Following prior works Black et al. [2024], Liu et al. [2025a], we adopt Stable Diffusion v1.5 (SDv1.5) Rombach et al. [2022] and Stable Diffusion 3.5 Medium (SD3.5-M) Esser et al. [2024] as the backbone models. See Appendix I for implementation details.

Evaluation Measures. We evaluate alignment reward and generation diversity on a fixed prompt set. Diversity is measured using DreamSim Diversity, ClipScore Diversity, Generalized Recall, and Vendi Score. Since DreamSim is used during training, the other three metrics provide complementary evaluations. We further evaluate distributional fidelity using Fréchet Inception Distance (FID), with images generated by the pretrained base model as the reference. See Appendix D for more details.

Baselines. We compare DRIFT with DDPO (on SDv1.5), GRPO, GRPO-KL, Nabla-GFlowNet, GARDO, and DiverseGRPO under the corresponding backbone and reward settings. We also include the pretrained base model and the inference-time method CADS as references. See Appendix E for more details. Fine-tuning is evaluated separately with two reward functions, **PickScore** Kirstain et al. [2023] and **HPSv2** Wu et al. [2023], to align with human preferences and text relevance. Rewards are normalized to $[0, 1]$. See Appendix J for reward details.

6.1 Study of Diversity-Aware Reward Shaping

We evaluate DRIFT, which directly optimizes intra-group diversity via potential-based reward shaping, and show its superiority over competitive baselines. Figures 4 and 11 (Appendix K) present Pareto frontiers for fine-tuning SDv1.5 and SD3.5-M with PickScore and HPSv2. GRPO-KL partially alleviates diversity loss through KL regularization, while Nabla-GFlowNet seeks to preserve diversity by learning to sample from a reward-reweighted pretrained distribution. GARDO and DiverseGRPO address diversity preservation through diversity-aware policy updates. We also include CADS, an inference-time strategy, for reference. While CADS outperforms both the base model and DRIFT on most diversity metrics, its lower primary reward suggests a less favorable balance between diversity and alignment. In contrast, DRIFT achieves a favorable tradeoff between reward maximization and diversity preservation. The decline in diversity during RL fine-tuning is substantially reduced, confirming that DRIFT effectively mitigates the trend of diversity collapse.

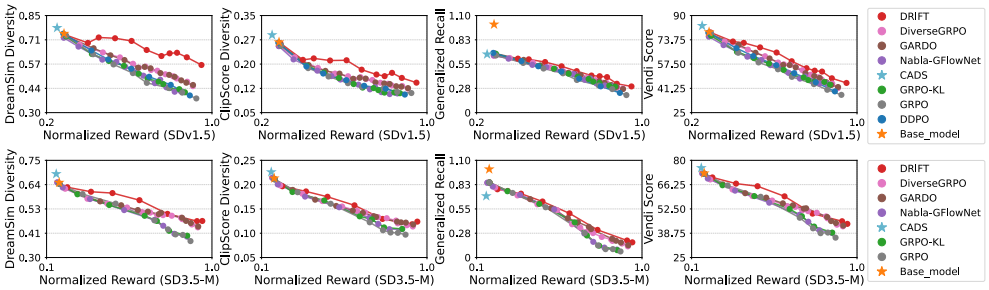

Figure 4: Reward-diversity comparison between DRIFT and baselines, with SDv1.5 and SD3.5-M fine-tuned using PickScore reward.

Table 1: Comparison of diversity and quality metrics under different reward functions and backbones. Diversity is evaluated using DreamSim, ClipScore Diversity, Generalized Recall, and Vendi Score, while quality is measured by the normalized reward. Percentage improvements represent the average relative gains of DRIFT over the respective baselines in each group under identical settings.

Backbone	Algorithm	PickScore					HPSv2				
		DreamSim (↑)	CLIP (↑)	Recall (↑)	Vendi (↑)	Reward (↑)	DreamSim (↑)	CLIP (↑)	Recall (↑)	Vendi (↑)	Reward (↑)
SDv1.5	Base model	0.7476 ±0.0121	0.2678 ±0.0041	1.0000 ±0.0000	78.96 ±0.9665	0.2653 ±0.01064	0.7476 ±0.0121	0.2678 ±0.0041	1.0000 ±0.0000	78.96 ±0.9665	0.3016 ±0.01810
	CADS	0.7805 ±0.0054	0.2899 ±0.0077	0.6614 ±0.0080	83.15 ±0.3352	0.2323 ±0.00648	0.7805 ±0.0054	0.2899 ±0.0077	0.6614 ±0.0080	83.15 ±0.3352	0.2629 ±0.01251
	DDPO	0.3974 ±0.0030	0.1058 ±0.0034	0.2233 ±0.0105	39.23 ±0.1550	0.5851 ±0.00886	0.4267 ±0.0041	0.1076 ±0.0021	0.3303 ±0.0072	41.53 ±0.2605	0.5400 ±0.00408
	GRPO	0.3804 ±0.0077	0.1105 ±0.0026	0.1989 ±0.0170	36.90 ±0.4305	0.5927 ±0.01331	0.4245 ±0.0038	0.1102 ±0.0024	0.2883 ±0.0090	40.79 ±0.3253	0.5409 ±0.01487
	GRPO-KL	0.4153 ±0.0086	0.1100 ±0.0047	0.2945 ±0.0095	43.73 ±0.4253	0.5503 ±0.00739	0.4342 ±0.0072	0.1120 ±0.0043	0.3496 ±0.0154	44.14 ±0.3083	0.5961 ±0.01160
	Nabla-GFlowNet	0.4186 ±0.0089	0.1092 ±0.0023	0.2962 ±0.0072	44.24 ±0.2283	0.5568 ±0.01314	0.4570 ±0.0036	0.1165 ±0.0015	0.3702 ±0.0087	45.89 ±0.3651	0.5974 ±0.00936
	GARDO	0.4567 ±0.0120	0.1256 ±0.0046	0.2671 ±0.0090	42.13 ±0.5144	0.6176 ±0.00845	0.4914 ±0.0074	0.1312 ±0.0019	0.3730 ±0.0041	45.24 ±0.0557	0.6473 ±0.00618
	DiverseGRPO	0.4499 ±0.0024	0.1243 ±0.0020	0.2735 ±0.0182	41.39 ±0.1833	0.6525 ±0.01147	0.4889 ±0.0049	0.1265 ±0.0030	0.3703 ±0.0125	**48.66** ±0.1529	0.6457 ±0.00588
	DRIFT	**0.5698** ±0.0090	**0.1426** ±0.0043	**0.2969** ±0.0141	**44.89** ±0.2927	**0.9187** ±0.01030	**0.5737** ±0.0076	**0.1439** ±0.0037	**0.3734** ±0.0081	45.93 ±0.3822	**0.9252** ±0.00908
	Improve (avg.)	+36.32%	+25.39%	+17.17%	+9.20%	+55.59%	+26.89%	+23.29%	+8.53%	+3.88%	+56.45%
SD3.5-M	Base model	0.6471 ±0.0107	0.2131 ±0.0030	1.0000 ±0.0000	72.72 ±0.7306	0.1726 ±0.01050	0.6468 ±0.0107	0.2121 ±0.0030	1.0000 ±0.0000	72.74 ±0.7306	0.2322 ±0.00529
	CADS	0.6871 ±0.0075	0.2258 ±0.0043	0.6960 ±0.0124	75.71 ±0.3338	0.1574 ±0.00379	0.6871 ±0.0075	0.2258 ±0.0043	0.6960 ±0.0124	75.71 ±0.3338	0.2167 ±0.00422
	GRPO	0.3761 ±0.0060	0.0970 ±0.0016	0.0689 ±0.0140	36.24 ±0.2217	0.7354 ±0.00652	0.3868 ±0.0068	0.1418 ±0.0026	0.0686 ±0.0153	33.79 ±0.3017	0.7722 ±0.00395
	GRPO-KL	0.4010 ±0.0066	0.1088 ±0.0025	0.0832 ±0.0099	39.08 ±0.2585	0.7412 ±0.00950	0.4028 ±0.0036	0.1405 ±0.0036	**0.0898** ±0.0122	35.72 ±0.1278	0.7908 ±0.00749
	Nabla-GFlowNet	0.3968 ±0.0064	0.1082 ±0.0028	0.0900 ±0.0389	38.93 ±0.3870	0.7110 ±0.01152	0.4010 ±0.0070	0.1488 ±0.0023	0.0683 ±0.0141	34.61 ±0.2501	0.7304 ±0.00710
	GARDO	0.4436 ±0.0058	0.1179 ±0.0024	0.1385 ±0.0075	43.05 ±0.3879	0.8538 ±0.00874	0.4099 ±0.0070	0.1457 ±0.0045	0.0745 ±0.0194	31.67 ±0.3388	0.8217 ±0.00734
	DiverseGRPO	0.4396 ±0.0051	0.1138 ±0.0029	0.1265 ±0.0243	42.62 ±0.5161	0.8591 ±0.00734	0.4072 ±0.0067	0.1479 ±0.0011	0.0736 ±0.0143	31.47 ±0.4291	0.8250 ±0.00333
	DRIFT	**0.4692** ±0.0073	**0.1238** ±0.0031	**0.1719** ±0.0126	**44.04** ±0.2197	**0.9130** ±0.00954	**0.4462** ±0.0042	**0.1590** ±0.0034	0.0880 ±0.0070	**38.97** ±0.2164	**0.8995** ±0.00628
	Improve (avg.)	+14.50%	+13.92%	+81.42%	+10.59%	+17.79%	+11.17%	+9.76%	+18.56%	+16.78%	+14.38%

Table 1 summarizes the primary quantitative results. DRIFT achieves positive average relative gains over the RL fine-tuning baselines across all four diversity metrics, ranging from 3.88% to 81.42% across backbone and reward settings. On SDv1.5, the average DreamSim and Generalized Recall gains with PickScore are 36.32% and 17.17%, respectively; the average DreamSim and ClipScore Diversity gains with HPSv2 are 26.89% and 23.29%. The average reward gains are 55.59% and 56.45% on SDv1.5, and 17.79% and 14.38% on SD3.5-M, for PickScore and HPSv2, respectively. These averages assign equal weight to each RL fine-tuning baseline within a backbone–reward group and exclude the base model and CADS. Complementing these quantitative findings, qualitative results in Figure 1 and Figures 12, 13, and 14 (Appendix K) further show that DRIFT produces diverse generations without compromising visual quality. Overall, DRIFT improves alignment reward and generation diversity on average over the RL fine-tuning baselines, although individual baselines remain stronger on some diversity metrics. We evaluate DRIFT on expanded sets of 100 and 200 prompts for SDv1.5 and SD3.5-M, respectively. DRIFT achieves average diversity and reward gains of 35.01% and 20.74%, respectively, over the original RL fine-tuning baselines, extending

the improvements beyond the original evaluation prompts. These gains are accompanied by small average relative FID increases on SDv1.5 and reductions on SD3.5-M (Appendix K.2).

6.2 Study of Reward-Concentrated Sampling

We verify the effectiveness of *reward-concentrated sampling* by comparing it with *reward-contrasted sampling*, an inverse strategy that updates the policy using a subset with larger reward variance. This comparison reveals how sampling strategies govern the tradeoff between reward optimization and diversity preservation. Figure 5 shows the Pareto frontiers of the two strategies during fine-tuning, where reward-concentrated sampling consistently achieves a more favorable frontier. At matched reward levels, it preserves substantially higher diversity across all four metrics by filtering reward outliers that would otherwise over-sharpen the policy and hinder exploration, promoting semantically meaningful behaviors for task alignment. As illustrated in Figure 6, reward-concentrated sampling yields a broader distribution of high-fidelity generations (left), which is quantitatively confirmed (right) by 8.56% $\sim$ 30.77% diversity gains and a 37.74% reward gain.

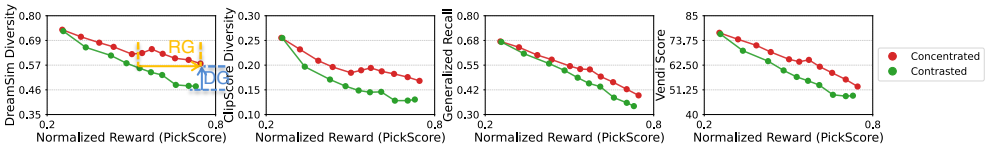

Figure 5: Comparison of reward-diversity tradeoff between reward-concentrated and reward-contrasted sampling, with SDv1.5 fine-tuned using PickScore reward. The record points on the Pareto frontier are collected from checkpoints at intervals during fine-tuning. **DG** denotes **D**iversity **G**ain at equivalent reward levels and **RG** denotes **R**eward **G**ain at equivalent diversity levels (Details in Appendix H). Both metrics are reported in the histograms (e.g., Figure 6) and tables (e.g., Table 1).



Figure 6: **Left**: Qualitative results show that reward-concentrated sampling preserves fidelity while improving diversity when fine-tuning SDv1.5 with PickScore. **Right**: Quantitative results show higher diversity and reward gain than reward-contrasted sampling.

6.3 Study of Noise-Conditioned Prompting

To comprehensively evaluate the effect of noise-conditioned prompting, we investigate four ablations: 1) add noise to prompt embeddings at both the training and testing stages, 2) add noise at training only, 3) add noise at testing only, and 4) no noise. The noise schedule remains consistent across training and testing, as described in Sec. 5.2. Figure 7 summarizes the reward–diversity tradeoffs for four ablations. Injecting noise into prompt embeddings during training consistently improves generation diversity by broadening the conditioning space, whereas adding comparable noise only at test time yields no noticeable gains. These results show that prompt-level stochasticity is effective only during training, where it shapes the latent space and exploration dynamics, enabling persistent diversity even without test-time noise. Additional quantitative evidence is provided in Figure 8, reporting 2.86%$\sim$15.38% diversity gains and an 11.11% reward gain.

6.4 Analysis

We conduct ablation studies on the weighting coefficient λ in decoupled advantage aggregation and the clipping range σ in clipped reward shaping. Figure 9 shows that overly large λ impedes convergence of the primary reward, as excessive emphasis on diversity regularization interferes with task optimization, while very small λ leads to insufficient diversity preservation. These results suggest that λ can effectively balance reward optimization and diversity maintenance.

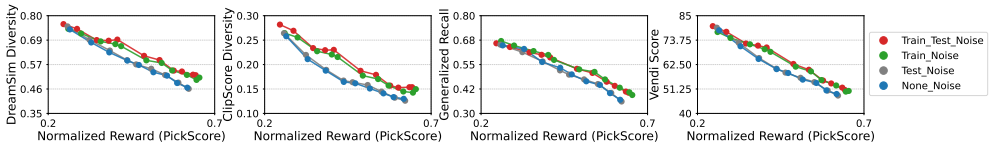

Figure 7: Comparison of reward-diversity tradeoff for prompting with and without noise, with SDv1.5 fine-tuned using PickScore reward.



Figure 8: **Left**: Qualitative results show that training with prompting noise preserves fidelity while improving diversity in PickScore fine-tuning of SDv1.5. **Right**: Quantitative results show higher diversity and reward gain than training without noise.

Figure 10 examines the effect of the clipping range σ on training dynamics. When σ is too small, the diversity reward saturates at the clipping boundaries in early training, weakening effective diversity signals and causing rapid diversity decay. Combined with per-prompt normalization, this increases the variance of the diversity advantage in later stages, introducing gradient noise that slows reward convergence. A moderate clipping range alleviates these issues by suppressing extremes while preserving informative variation. Compared to a larger range (e.g., σ=2), σ=1 better controls outliers in the normalized diversity advantage, yielding more stable training. Overall, σ affects training stability more than final performance, serving as a practical mechanism for robust optimization. We further conduct a component-wise ablation study to examine the interactions among the proposed components. The results demonstrate their complementary contributions, with the full configuration consistently outperforming partial variants across the reported metrics (Appendix K.3).

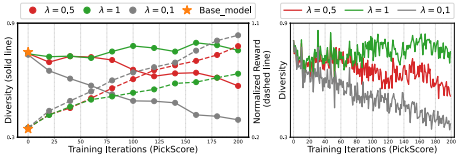

Figure 9: Effect of the coefficient λ in DRIFT. **Left**: Evolution of the reward–diversity tradeoff. **Right**: Diversity during training.

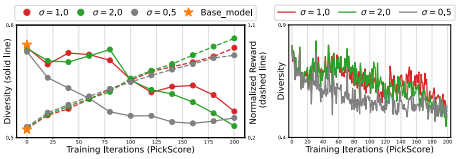

Figure 10: Effect of the clipping term σ in DRIFT. **Left**: Evolution of the reward–diversity tradeoff. **Right**: Diversity during training.

7 Conclusion

In this paper, we study the challenge of diversity collapse in RL fine-tuning of generative models and introduce DRIFT, a principled framework that incentivizes output diversity for versatile image generation. Our method mitigates diversity collapse from three complementary dimensions: reward-concentrated sampling, noise-conditioned prompting, and diversity-aware reward shaping. Extensive experimental results demonstrate that DRIFT consistently achieves more favorable reward-diversity Pareto frontiers, reconciling high generation diversity with strong task alignment. These results suggest that diversity preservation is compatible with reward optimization and can contribute to robust and controllable generative modeling. We hope this analysis and framework provide a foundation for future diversity-preserving RL fine-tuning of generative models.

DRIFT primarily targets tasks with multiple valid outputs and incurs additional computational overhead from candidate generation and diversity evaluation. Future work will explore efficient diversity estimation using internal model representations and extend DRIFT to video and 3D generation.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62376122 & 72394363), in part by the Fundamental Research Funds for the Central Universities (Grant No. 011814380048), in part by the Ministry of Education, Singapore (Grant No. T1 251RES2607), and in part by the NUS Artificial Intelligence Institute (NAII), National University of Singapore (Grant No. NAII-SG-2025-36).

References

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proceedings of International Conference on Learning Representations*, 2023.

Mischa Dombrowski, Weitong Zhang, Sarah Cechnicka, Hadrien Reynaud, and Bernhard Kainz. Image generation diversity issues and how to tame them. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3029–3039, 2025.

Weitong Zhang, Chengqi Zang, Liu Li, Sarah Cechnicka, Cheng Ouyang, and Bernhard Kainz. Stability and generalizability in SDE diffusion models with measure-preserving dynamics. In *Advances in Neural Information Processing Systems*, volume 37, pages 81606–81644, 2024a.

Zhenhong Sun, Yifu Wang, Yonhon Ng, Yongzhi Xu, Daoyi Dong, Hongdong Li, and Pan Ji. T^3-S2S: Training-free triplet tuning for sketch to scene synthesis in controllable concept art generation. *Transactions on Machine Learning Research*, 2026.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Zhenhong Sun, Junyan Wang, Zhiyu Tan, Daoyi Dong, Hailan Ma, Hao Li, and Dong Gong. Eggen: Image generation with multi-entity prior learning through entity guidance. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6637–6645, 2024.

Junyan Wang, Zhenhong Sun, Zhiyu Tan, Xuanbai Chen, Weihua Chen, Hao Li, Cheng Zhang, and Yang Song. Towards effective usage of human-centric priors in diffusion models for text-based human image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8446–8455. IEEE, 2024.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *Proceedings of International Conference on Learning Representations*, 2024.

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. Technical report, OpenAI, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 37, pages 76177–76209, 2024.

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. In *Advances in Neural Information Processing Systems*, volume 36, pages 15903–15935, 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. In *Advances in Neural Information Processing Systems*, volume 36, pages 79858–79885, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *Proceedings of International Conference on Learning Representations*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

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*, 2025a.

Roberto Barceló, Cristóbal Alcázar, and Felipe Tobar. Avoiding mode collapse in diffusion models fine-tuned with reinforcement learning. *arXiv preprint arXiv:2410.08315*, 2024.

Rohit Jena, Ali Taghibakhshi, Sahil Jain, Gerald Shen, Nima Tajbakhsh, and Arash Vahdat. Elucidating optimal reward-diversity tradeoffs in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 232–242. IEEE, 2025.

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. In *Proceedings of International Conference on Learning Representations*, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *Advances in Neural Information Processing Systems*, 2025.

Jiwoo Hong, Sayak Paul, Noah Lee, Kashif Rasul, James Thorne, and Jongheon Jeong. Margin-aware preference optimization for aligning diffusion models without reference. In *AAAI Conference on Artificial Intelligence*, 2026.

Dmitrii Sorokin, Maksim Nakhodnov, Andrey Kuznetsov, and Aibek Alanov. Imagerefl: Balancing quality and diversity in human-aligned diffusion models. *arXiv preprint arXiv:2505.22569*, 2025.

Rohit Gandikota and David Bau. Distilling diversity and control in diffusion models. *arXiv preprint arXiv:2503.10637*, 2025.

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling. In *Proceedings of International Conference on Learning Representations*, 2024.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *Proceedings of International Conference on Learning Representations*, 2024.

Zhiwei Jia, Yuesong Nan, Huixi Zhao, and Gengdai Liu. Reward fine-tuning two-step diffusion models via learning differentiable latent-space surrogate reward. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12922, 2025.

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

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23604–23614, 2025.

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.

Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Enhancing diffusion models with text-encoder reinforcement learning. In *Proceedings of European Conference on Computer Vision*, pages 182–198, 2024.

Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. DreamReward: Text-to-3d generation with human preference. In *Proceedings of European Conference on Computer Vision*, pages 259–276. Springer, 2024.

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

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

Yinan Zhang, Eric Tzeng, Yilun Du, and Dmitry Kislyuk. Large-scale reinforcement learning for diffusion models. In *Proceedings of European Conference on Computer Vision*, pages 1–17, 2024b.

Haoru Li, Jinmei Liu, Zhiyong Wang, Xiaoming Li, Zhenhong Sun, Daoyi Dong, Chunlin Chen, and Zhi Wang. Many ways to succeed: Diversity-driven rl fine-tuning for VLA generalization. *arXiv preprint arXiv:2610.09943*, 2026.

Kevin Zhai, Utsav Singh, Anirudh Thatipelli, Souradip Chakraborty, Anit Kumar Sahu, Furong Huang, Amrit Singh Bedi, and Mubarak Shah. MIRA: Towards mitigating reward hacking in inference-time alignment of T2I diffusion models. *arXiv preprint arXiv:2510.01549*, 2025.

Haotian Ye, Kaiwen Zheng, Jiashu Xu, Puheng Li, Huayu Chen, Jiaqi Han, Sheng Liu, Qinsheng Zhang, Hanzi Mao, Zekun Hao, et al. Data-regularized reinforcement learning for diffusion models at scale. *arXiv preprint arXiv:2512.04332*, 2025.

Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10844–10853, 2024.

Henglin Liu, Huijuan Huang, Jing Wang, Chang Liu, Xiu Li, and Xiangyang Ji. Diversegrpo: Mitigating mode collapse in image generation via diversity-aware grpo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1864–1873, 2026.

Haoran He, Yuxiao Ye, Jie Liu, Jiajun Liang, Zhiyong Wang, Ziyang Yuan, Xintao Wang, Hangyu Mao, Pengfei Wan, and Ling Pan. GARDO: Reinforcing diffusion models without reward hacking. *arXiv preprint arXiv:2512.24138*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741, 2023.

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

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

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of International Conference on Machine Learning*, pages 745–750, 2007.

Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of International Conference on Machine Learning*, pages 278–287, 1999.

Yiming Wang, Ming Yang, Renzhi Dong, Binbin Sun, Furui Liu, et al. Efficient potential-based exploration in reinforcement learning using inverse dynamic bisimulation metric. In *Advances in Neural Information Processing Systems*, volume 36, pages 38786–38797, 2023.

Henrik Müller and Daniel Kudenko. Improving the effectiveness of potential-based reward shaping in reinforcement learning. In *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*, pages 2684–2686, 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 *Proceedings of International Conference on Learning Representations*, 2024.

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. In *Advances in Neural Information Processing Systems*, volume 36, pages 36652–36663, 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*, 2023.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. DreamSim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, et al. Openclip. *Zenodo*, 2021.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

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

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhen Liu, Tim Xiao, Weiyang Liu, Yoshua Bengio, and Dinghuai Zhang. Efficient diversity-preserving diffusion alignment via gradient-informed gflownets. In *International Conference on Learning Representations*, 2025b.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proceedings of International Conference on Learning Representations*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of International Conference on Learning Representations*, 2021b.

A Optimal Policy Invariance in DRIFT

Following the classical reward shaping study [Ng et al., 1999], we give the proof of Theorem 1, which guarantees the optimal policy invariance when incorporating the intra-group diversity as an intrinsic reward.

Theorem 1 (Optimal Policy Invariance)**.** *Let $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ denote the MDP for the task of fine-tuning diffusion models with RL. $d(\cdot) : S \mapsto \mathbb{R}$ is a real-valued function that computes the intra-group diversity $d(s)$ of the state s within a group of generation samples. We formulate $R_{int}(\cdot) : S \times A \times S \mapsto \mathbb{R}$ as an intrinsic reward function that is the difference between the diversities of adjacent states, such that for all $s \in S, a \in A, s' \in S$, $R_{int}(s, a, s') = \gamma d(s') - d(s)$. Then, with any constant balancing ratio λ, every optimal policy in the transformed MDP $\tilde{M} = (\mathcal{S}, \mathcal{A}, P, R + \lambda R_{int}, \gamma)$ will also be an optimal policy in M, and vice versa.*

Proof. For the original MDP M, we know that its optimal Q-function Q^*_M satisfies the Bellman optimality equation [Sutton and Barto, 2018]:

$$Q^*_M(s, a) = \mathbb{E}_{s'} \left[R(s, a, s') + \gamma \max_{a' \in A} Q^*_M(s', a') \right]. \tag{10}$$

With some simple algebraic manipulation, we can get:

$$Q^*_M(s, a) - \lambda d(s) = \mathbb{E}_{s'} \left[R(s, a, s') + \lambda \Big(\gamma d(s') - d(s) \Big) + \gamma \max_{a' \in A} \Big(Q^*_M(s', a') - \lambda d(s') \Big) \right]. \tag{11}$$

If we now define $\hat{Q}_{\tilde{M}}(s, a) \triangleq Q^*_M(s, a) - \lambda d(s)$ and substitute that and $R_{\text{int}}(s, a, s') = \gamma d(s') - d(s)$ into the previous equation, we can get:

$$\begin{aligned} \hat{Q}_{\tilde{M}}(s, a) &= \mathbb{E}_{s'} \left[R(s, a, s') + \lambda R_{\text{int}}(s, a, s') + \gamma \max_{a' \in A} \hat{Q}_{\tilde{M}}(s', a') \right] \\ &= \mathbb{E}_{s'} \left[R'(s, a, s') + \gamma \max_{a' \in A} \hat{Q}_{\tilde{M}}(s', a') \right], \end{aligned} \tag{12}$$

which is exactly the Bellman optimality equation for the transformed MDP $\tilde{M}$, where $\tilde{R} = R + \lambda R_{\text{int}}$ is the reward function for M'. Thus, $Q^*_{\tilde{M}}(s, a) = \hat{Q}_{\tilde{M}}(s, a) = Q^*_M(s, a) - \lambda d(s)$, and the optimal policy for M' therefore satisfies:

$$\begin{aligned} \pi^*_{\tilde{M}}(s) &= \arg\max_{a \in A} Q^*_{\tilde{M}}(s, a) \\ &= \arg\max_{a \in A} \Big[Q^*_M(s, a) - \lambda d(s) \Big] \\ &= \arg\max_{a \in A} Q^*_M(s, a), \end{aligned} \tag{13}$$

and is therefore also optimal in M. To show every optimal policy in M is also optimal in $\tilde{M}$, simply apply the same proof with the roles of M and $\tilde{M}$ interchanged (and using $-R_{\text{int}}$ as the intrinsic reward). This completes the proof. □

B Mathematical Derivations

B.1 Deriving the Optimum of the KL-Constrained Reward Maximization Objective

In this appendix, we will derive Eq. 3. Analogous to Eq. 1, we maximize the KL-constrained objective:

$$J(\theta) = \mathbb{E}_{\boldsymbol{c}, \boldsymbol{x}_0 \sim \pi(\boldsymbol{x}_0|\boldsymbol{c})} \left[r(\boldsymbol{x}_0, \boldsymbol{c}) \right] - \beta \cdot \mathbb{D}_{\text{KL}} \left[\pi(\boldsymbol{x}_0|\boldsymbol{c}) || \pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c}) \right], \tag{14}$$

under any reward function $r(\boldsymbol{x}_0, \boldsymbol{c})$, reference model π_{ref} and a general non-parametric policy class. The objective is reformulated as

$$\begin{aligned}
&\max_{\pi} \mathbb{E}_{\boldsymbol{c},\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0|\boldsymbol{c})}\left[r(\boldsymbol{x}_0,\boldsymbol{c})\right] - \beta\mathbb{D}_{\text{KL}}\left[\pi(\boldsymbol{x}_0|\boldsymbol{c})||\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\right] \\
=&\max_{\pi} \mathbb{E}_{\boldsymbol{c}}\mathbb{E}_{\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0|\boldsymbol{c})}\left[r(\boldsymbol{x}_0,\boldsymbol{c}) - \beta\log\frac{\pi(\boldsymbol{x}_0|\boldsymbol{c})}{\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})}\right] \\
=&\min_{\pi} \mathbb{E}_{\boldsymbol{c}}\mathbb{E}_{\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0|\boldsymbol{c})}\left[\log\frac{\pi(\boldsymbol{x}_0|\boldsymbol{c})}{\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})} - \frac{1}{\beta}r(\boldsymbol{x}_0,\boldsymbol{c})\right] \\
=&\min_{\pi} \mathbb{E}_{\boldsymbol{c}}\mathbb{E}_{\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0|\boldsymbol{c})}\left[\log\frac{\pi(\boldsymbol{x}_0|\boldsymbol{c})}{\frac{1}{Z(\boldsymbol{c})}\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\exp\left(\frac{1}{\beta}r(\boldsymbol{x}_0,\boldsymbol{c})\right)} - \log Z(\boldsymbol{c})\right],
\end{aligned} \tag{15}$$

where the partition function is:

$$Z(\boldsymbol{c}) = \sum_{\boldsymbol{x}_0}\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\exp\left(\frac{1}{\beta}r(\boldsymbol{x}_0,\boldsymbol{c})\right). \tag{16}$$

Note that the partition function is a function of only $\boldsymbol{c}$ and the reference policy π_{ref}, but does not depend on the policy π. We can now define

$$\pi^*(\boldsymbol{x}_0|\boldsymbol{c}) = \frac{1}{Z(\boldsymbol{c})}\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\exp\left(\frac{1}{\beta}r(\boldsymbol{x}_0,\boldsymbol{c})\right), \tag{17}$$

which is a valid probability distribution as $\pi^*(\boldsymbol{x}_0|\boldsymbol{c}) \geq 0,\ \forall \boldsymbol{x}_0$ and $\sum_{\boldsymbol{x}_0}\pi^*(\boldsymbol{x}_0|\boldsymbol{c}) = 1$. Next, we can re-organize the objective in Eq. 15 as

$$\min_{\pi} \mathbb{E}_{\boldsymbol{c}}\mathbb{E}_{\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0|\boldsymbol{c})}\left[\log\frac{\pi(\boldsymbol{x}_0|\boldsymbol{c})}{\frac{1}{Z(\boldsymbol{c})}\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\exp\left(\frac{1}{\beta}r(\boldsymbol{x}_0,\boldsymbol{c})\right)} - \log Z(\boldsymbol{c})\right] \tag{18}$$

$$= \min_{\pi} \mathbb{E}_{\boldsymbol{c}}\left[\mathbb{E}_{\boldsymbol{x}_0\sim\pi(\boldsymbol{x}_0|\boldsymbol{c})}\left[\log\frac{\pi(\boldsymbol{x}_0|\boldsymbol{c})}{\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})}\right] - \log Z(\boldsymbol{c})\right] \tag{19}$$

$$= \mathbb{E}_{\boldsymbol{c}}\left[\mathbb{D}_{\text{KL}}[\pi(\boldsymbol{x}_0|\boldsymbol{c})||\pi^*(\boldsymbol{x}_0|\boldsymbol{c})] - \log Z(\boldsymbol{c})\right] \tag{20}$$

$$= \mathbb{E}_{\boldsymbol{c}}\left[\mathbb{D}_{\text{KL}}(\pi(\boldsymbol{x}_0|\boldsymbol{c})||\pi^*(\boldsymbol{x}_0|\boldsymbol{c}))\right]. \tag{21}$$

Eq. 19 comes from that $Z(\boldsymbol{c})$ is not a function of $\boldsymbol{x}_0$, and Eq. 21 comes from that $Z(\boldsymbol{c})$ does not depend on π. Gibbs' inequality establishes that $\mathbb{D}_{\text{KL}}(\pi||\pi^*) \geq 0$, with equality held if and only if the distributions are identical. Hence, we have the optimal solution:

$$\pi(\boldsymbol{x}_0|\boldsymbol{c}) = \pi^*(\boldsymbol{x}_0|\boldsymbol{c}) = \frac{1}{Z(\boldsymbol{c})}\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\exp\left(\frac{1}{\beta}r(\boldsymbol{x}_0,\boldsymbol{c})\right), \tag{22}$$

for all $\boldsymbol{c}$. This completes the derivation.

B.2 Deriving the Optimal Policy When KL Coefficient Approaches Zero

In this appendix, we will derive Eq. 4 that shows the behavior of the optimal policy π^* as $\beta \to 0^+$. Let $\boldsymbol{x}_0^* = \arg\max_{\boldsymbol{x}_0} r(\boldsymbol{x}_0, \boldsymbol{c})$ be the unique global maximum of the reward function r. We can reformulate the distribution to characterize the density at any point $\boldsymbol{x}_0$ relative to the density at the global maximum $\boldsymbol{x}_0^*$ as

$$\frac{\pi^*(\boldsymbol{x}_0|\boldsymbol{c})}{\pi^*(\boldsymbol{x}_0^*|\boldsymbol{c})} = \frac{\pi_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})}{\pi_{\text{ref}}(\boldsymbol{x}_0^*|\boldsymbol{c})}\exp\left(\frac{r(\boldsymbol{x}_0,\boldsymbol{c}) - r(\boldsymbol{x}_0^*,\boldsymbol{c})}{\beta}\right). \tag{23}$$

As $\beta \to 0^+$, we have

- If $\boldsymbol{x}_0 = \boldsymbol{x}_0^*$, the exponent is 0, and the ratio is 1.
- If $r(\boldsymbol{x}_0, \boldsymbol{c}) < r(\boldsymbol{x}_0^*, \boldsymbol{c})$, then $r(\boldsymbol{x}_0, \boldsymbol{c}) - r(\boldsymbol{x}_0^*, \boldsymbol{c})$ is a negative constant, leading to the fraction $\frac{r(\boldsymbol{x}_0,\boldsymbol{c})-r(\boldsymbol{x}_0^*,\boldsymbol{c})}{\beta} \to -\infty$ and the ratio $\frac{\pi^*(\boldsymbol{x}_0|\boldsymbol{c})}{\pi^*(\boldsymbol{x}_0^*|\boldsymbol{c})} \to 0$.

This implies that for any $\boldsymbol{x}_0$ where the reward is not maximal, the relative probability mass drops to zero as

$$\lim_{\beta \to 0^+} \frac{\pi^*(\boldsymbol{x}_0|\boldsymbol{c})}{\pi^*(\boldsymbol{x}_0^*|\boldsymbol{c})} = 0 \quad \text{for all } \boldsymbol{x}_0 \neq \boldsymbol{x}_0^*. \tag{24}$$

Because the total probability must integrate to 1, but the density at all non-maximal points vanishes, the entire probability mass must pile up at the location of the maximum reward. Formally, for any smooth test function $f(\boldsymbol{x}_0)$:

$$\lim_{\beta \to 0^+} \int \pi^*(\boldsymbol{x}_0) f(\boldsymbol{x}_0) d\boldsymbol{x}_0 = f(\boldsymbol{x}_0^*). \tag{25}$$

This is the defining property of the Dirac delta distribution $\delta(\boldsymbol{x}_0 - \boldsymbol{x}_0^*)$. This completes the proof.

C Quantifying Intra-Group Diversity

Given a pair of output samples $(\boldsymbol{x}_0^i, \boldsymbol{x}_0^j)$ generated from the same prompt, we project them into a latent embedding space using an encoder $\mathcal{E}$ and calculate their dissimilarity as the squared Euclidean distance between the corresponding embeddings:

$$d\left(\boldsymbol{x}_0^i, \boldsymbol{x}_0^j\right) = \left\| \mathcal{E}(\boldsymbol{x}_0^i) - \mathcal{E}(\boldsymbol{x}_0^j) \right\|^2. \tag{26}$$

For a group of samples $\left\{\mathbf{x}_0^1, \mathbf{x}_0^2, \ldots, \mathbf{x}_0^G\right\}$, we compute their pairwise dissimilarities using the diversity metric in Eq. 26, resulting in a $G \times G$ matrix where the element of row i and column j denotes the diversity between sample $\boldsymbol{x}_0^i$ and $\boldsymbol{x}_0^j$. The diversity of sample $\mathbf{x}_0^i$ within the group is obtained by simply averaging its dissimilarities to all other outputs, i.e., averaging across the i-th row in the diversity matrix as

$$d\left(\mathbf{x}_0^i\right) = \frac{1}{G-1} \sum\nolimits_{i \neq j}^{G} d\left(\boldsymbol{x}_0^i, \boldsymbol{x}_0^j\right). \tag{27}$$

A higher value of $d(\mathbf{x}_0^i)$ indicates that the i-th sample is less similar to the other outputs within the group, reflecting greater intra-group diversity.

The most straightforward choice for the pre-trained encoder $\mathcal{E}$ in Eq. 26 is a variational auto-encoder (VAE), which serves as the de facto standard for obtaining latent representations in modern diffusion models [Rombach et al., 2022]. However, such encoders may fail to capture meaningful perceptual or semantic variations in generated images. Instead, we utilize DreamSim [Fu et al., 2023], a model trained by concatenating CLIP, OpenCLIP, and DINO embeddings, and subsequently fine-tuned on human perceptual judgments. DreamSim is particularly well-suited for RFT that typically addresses downstream objectives like text-image alignment and human-centric evaluation. To this end, we design a simple, easy-to-implement metric to quantify image diversity using image-level representations extracted from an off-the-shelf pre-trained encoder. As a general framework, our method is also compatible with any other diversity metrics.

D Computation of Evaluation Metrics

We construct the generation image set using a fixed set of prompts and a controlled sampling protocol. Specifically, we select 40 representative prompts from the evaluation set, ensuring consistency across all methods evaluated, including both the base model and the fine-tuned models. For each prompt, we generate 40 images using identical random seeds across different models, resulting in a total of 1,600 generated images per method.

DreamSim Diversity uses the DreamSim library Fu et al. [2023] to compute DreamSim diversity as the mean pairwise squared distance between DreamSim embeddings of 40 generations for a given prompt, averaged across 40 prompts. Namely,

$$\text{DreamSim_diversity} = \frac{1}{40} \sum_{k=1}^{40} \frac{2}{40 \cdot 39} \sum_{1 \leq i < j \leq 40} \left\| \text{DreamSim}\left(o_i^k\right) - \text{DreamSim}\left(o_j^k\right) \right\|^2, \tag{28}$$

where o_i^k denotes the i-th generation for the k-th prompt. A higher DreamSim Diversity value indicates greater variation among generated samples, reflecting higher generative diversity.

ClipScore Diversity uses the `open_clip` library Ilharco et al. [2021] to extract CLIP image embeddings. Following DreamSim Diversity, we compute the mean pairwise squared Euclidean distance among 40 generations per prompt, averaged across 40 prompts.

Generalized Recall Sajjadi et al. [2018] introduces the classic concepts of precision and recall to the study of generative models, providing distinct conclusions that are otherwise confounded by metrics like FID. While high precision implies a higher degree of realism and fidelity of the images compared to the base distribution, high recall implies higher coverage of the data distribution by the generator, signifying diversity. The approach proposed (e.g., Kynkäänniemi et al. [2019]) forms non-parametric representations of the data manifolds using overlapping hyperspheres defined by the k-Nearest Neighbors (kNN) technique. Following this method, binary assignments are used to compute the recall. We specifically compute the recall of the generated distribution with respect to the distribution of the base model, which covers multiple modes of the dataset. For our experiments, the neighborhood size is set to $k = 10$.

Vendi Score (Friedman and Dieng [2022]) is a diversity metric that quantifies the effective number of distinct modes captured by a set of images. Unlike distance-based metrics, the Vendi Score provides a single value that represents the effective rank or intrinsic dimensionality of the generated feature distribution. It operates by first extracting high-dimensional feature embeddings from the images. These embeddings are then used to construct a kernel matrix K, where each entry K_{ij} measures the similarity between samples i and j. The score is computed by applying a mathematical function (specifically, the effective rank calculation) to the eigenvalues λ_i of this normalized kernel matrix, often using the exponential of the Shannon entropy of the normalized eigenvalues:

$$\text{Vendi}(\mathbf{E}) = \exp\left(-\sum_i p_i \ln p_i\right), \quad \text{where } p_i = \frac{\lambda_i}{\sum_j \lambda_j}. \tag{29}$$

A higher Vendi Score indicates greater diversity, as it implies that the images are more spread out in the feature space, corresponding to a higher effective number of captured modes. Comparisons use the same sample count within each evaluation setting, since empirical Vendi scores can depend on sample size.

Fréchet Inception Distance (FID) [Heusel et al., 2017] is a metric that measures the discrepancy between generated and reference image distributions. We use images sampled from the corresponding pretrained base model as the reference distribution. For each evaluation setting, we use identical prompts, sample counts, and preprocessing across different methods. Specifically, we extract 2048-dimensional features from the pool3 layer of an FID-compatible Inception-v3 network. Let $(\mu_{\text{ref}}, \Sigma_{\text{ref}})$ and $(\mu_{\text{gen}}, \Sigma_{\text{gen}})$ denote the empirical means and covariance matrices of the reference and generated image features, respectively. FID is calculated as

$$\text{FID} = \|\mu_{\text{ref}} - \mu_{\text{gen}}\|_2^2 + \text{Tr}\left(\Sigma_{\text{ref}} + \Sigma_{\text{gen}} - 2(\Sigma_{\text{ref}}\Sigma_{\text{gen}})^{1/2}\right). \tag{30}$$

For numerical stability, we compute the covariance cross-term using singular values of the centered feature matrices. Lower FID indicates smaller distributional deviation from the base model.

E Baselines

We compare DRIFT with representative baselines for image generation, including the pretrained base model, CADS, DDPO, GRPO, GRPO-KL, GARDO, DiverseGRPO, and Nabla-GFlowNet. We summarize each baseline as follows:

- **Base model** Rombach et al. [2022]: pretrained base models used in all experiments (SDv1.5 or SD3.5-M).
- **CADS** [Sadat et al., 2024]: an inference-time diversity baseline using timestep-dependent conditioning perturbations.
- **DDPO** Black et al. [2024]: formulates a policy gradient algorithm to optimize diffusion models.
- **GRPO** Shao et al. [2024]: computes advantages from group-level comparisons to estimate policy gradients.

- **GRPO-KL** Liu et al. [2025a]: extends GRPO with KL regularization to prevent over-optimization.
- **GARDO** [He et al., 2025]: a diversity-preserving RL baseline with gated KL penalties and diversity-weighted advantage reweighting.
- **DiverseGRPO** [Liu et al., 2026]: mitigates diversity collapse through spectral clustering and stronger early-stage KL regularization.
- **Nabla-GFlowNet** [Liu et al., 2025b]: leverages reward gradients for diversity-preserving diffusion model alignment.

F Details of Sample Selection

F.1 Sample Selection Strategies

This appendix provides implementation details of the sample selection strategies used to construct training batches in our GRPO fine-tuning experiments. For each prompt, the policy generates a pool of $2G$ candidate outputs $\{\mathbf{o}_1, \ldots, \mathbf{o}_{2G}\}$, from which G samples are selected to form the effective training batch. We compute the pairwise distances in a perceptual embedding space. Specifically, each output $\mathbf{o}_i$ is encoded by a DreamSim encoder $\mathcal{E}(\cdot)$, and the pairwise distances D_{ij} are given by:

$$D_{ij} = \|\mathcal{E}(\mathbf{o}_i) - \mathcal{E}(\mathbf{o}_j)\|_2. \tag{31}$$

This results in a perceptual distance matrix $\mathbf{D} \in \mathbb{R}^{2G\times 2G}$. Although the distances are computed in the perceptual embedding space rather than directly from scalar rewards, we empirically find that perceptually concentrated groups tend to exhibit higher reward means and lower, yet non-degenerate, reward variances than perceptually contrasted groups. This avoids directly collapsing the group in reward space while still reducing reward outliers in the selected batch.

Reward-Concentrated Selection. This strategy selects G samples that are close to each other in the perceptual distance by using a nearest-neighbor criterion. For each candidate i, we retrieve the indices of its $G-1$ nearest neighbors and compute the corresponding distance sum

$$s_i = \sum_{j\in\min(i;G-1)} D_{ij}, \tag{32}$$

where $\min(i; G-1)$ denotes the set of indices of the $G-1$ smallest distances in row i, excluding the self-distance D_{ii}. We choose the reference index $i^\star = \arg\min_i s_i$ and form the training batch by including this reference sample together with its $G-1$ nearest neighbors.

Reward-Contrasted Selection. This strategy selects G samples that are well separated in the perceptual distance by using a farthest-neighbor criterion. For each candidate i, we retrieve its $G-1$ farthest neighbors and compute the corresponding distance sum

$$t_i = \sum_{j\in\max(i;G-1)} D_{ij}, \tag{33}$$

where $\max(i; G-1)$ denotes the set of indices of the $G-1$ largest distances in row i, excluding the self-distance D_{ii}. We choose the reference index $i^\star = \arg\max_i t_i$ and form the training batch by including this reference sample together with its $G-1$ farthest neighbors.

All other training components are kept identical across settings, with the two strategies differing only in how G samples are selected from the $2G$ candidate pool.

F.2 Computational Overhead and Efficiency

We evaluate the computational overhead of RCS on SDv1.5 with PickScore. Compared with training without RCS, the end-to-end training time increases from 400.79 to 512.68 seconds per epoch, corresponding to an overhead of 27.92%. The additional cost primarily arises from generating $2G$ candidates and extracting their perceptual embeddings, while task rewards are evaluated and policy updates are performed only on the selected G samples. Pairwise-distance computation and nearest-neighbor selection introduce approximately 0.01 seconds per epoch, indicating that the selection operation itself incurs negligible computational overhead.

Table 2: Comparison between RCS and full sampling under the same candidate-generation budget on SDv1.5 with PickScore. Higher values indicate better performance.

Method	DreamSim	CLIP	Recall	Vendi	Reward
Full sampling	0.4680	0.1285	0.3639	49.71	0.5363
RCS	**0.5822**	**0.1682**	**0.3969**	**52.75**	**0.7297**
Gain (%)	+24.40	+30.89	+9.07	+6.12	+36.06

To further examine whether the performance improvement arises merely from generating additional candidates, we compare RCS with full sampling under the same candidate-generation budget. Both methods generate 16 candidates per prompt. Full sampling uses all 16 candidates for policy optimization, whereas RCS selects eight candidates for subsequent reward evaluation and policy updates. As shown in Table 2, RCS consistently outperforms full sampling across all evaluation metrics, achieving an average diversity gain of 17.62% and a reward gain of 36.06%. These results suggest that selecting informative training groups, rather than simply increasing the number of generated candidates, contributes to the improved reward-diversity trade-off.

G Annealed Prompt Embedding Noise

Our noise-conditioned prompting strategy is inspired by the Condition-Annealed Diffusion Sampler (CADS) Sadat et al. [2024], which introduces timestep-dependent perturbations to conditioning signals during diffusion inference to promote generative diversity. CADS applies stronger perturbations at early diffusion steps and gradually reduces them as inference progresses, aligning stochasticity with the diffusion dynamics.

Following this principle, we inject timestep-annealed noise into the prompt embeddings during training-time generation. Let e denote the clean prompt embedding. At the diffusion timestep t, we construct a perturbed prompt embedding $\tilde{e}_t$ as

$$\tilde{e}_t = \sqrt{w(t)}\, e + \eta \cdot \sqrt{1 - w(t)}\, n, \quad n \sim \mathcal{N}(0, I), \tag{34}$$

where η controls the overall noise scale and $w(t) \in [0, 1]$ is an annealing function that decreases with the diffusion timestep. We adopt a piecewise linear annealing schedule for $w(t)$:

$$w(t) = \begin{cases} 1, & t \le \tau_1, \\ \frac{\tau_2 - t}{\tau_2 - \tau_1}, & \tau_1 < t < \tau_2, \\ 0, & t \ge \tau_2, \end{cases} \tag{35}$$

where τ_1 and τ_2 are user-defined thresholds that determine the transition from high to low noise. Since diffusion inference proceeds backward in time, this schedule applies stronger perturbations at earlier timesteps and gradually removes noise as generation converges.

Adding noise to the prompt embeddings alters their mean and variance, which can lead to instability when the noise scale is large. To mitigate this effect, we apply a rescaling operation similar to that used in CADS. Specifically, given a noisy embedding $\tilde{e}_t$ with empirical mean and standard deviation $\mu(\tilde{e}_t)$ and $\sigma(\tilde{e}_t)$, we compute a rescaled embedding

$$\tilde{e}_t^{\text{rescaled}} = \frac{\tilde{e}_t - \mu(\tilde{e}_t)}{\sigma(\tilde{e}_t)} \sigma_e + \mu_e, \tag{36}$$

where μ_e and σ_e denote the mean and standard deviation of the clean prompt embeddings. The final prompt embedding is then obtained by mixing the rescaled and unrescaled embeddings:

$$\tilde{e}_t^{\text{final}} = \psi\, \tilde{e}_t^{\text{rescaled}} + (1 - \psi)\, \tilde{e}_t, \tag{37}$$

where $\psi \in [0, 1]$ controls the strength of rescaling. This rescaling strategy improves numerical stability under high noise levels while preserving sufficient stochasticity for diversity enhancement. Unlike CADS, which applies condition annealing purely at inference time, our formulation integrates timestep-annealed prompt embedding noise into reinforcement learning fine-tuning, explicitly diversifying the conditioning space explored during optimization and helping mitigate diversity collapse. In our experiments, the noise scale is set to $\eta = 0.05$, the annealing thresholds are $\tau_1 = 0.4$ and $\tau_2 = 1.0$ (with timesteps normalized to $[0, 1]$), and the rescaling mixing factor is $\psi = 1$, which we found to work well in practice.

H Diversity Gain and Reward Gain

When measuring RG, we maintain approximately equivalent diversity levels using DreamSim Diversity as the reference, as illustrated by the yellow line. Conversely, for DG, we maintain approximately equivalent reward levels, as indicated by the blue line. Since it is infeasible to obtain checkpoints that induce exactly the same reward or diversity values for different methods, we adopt a stringent comparison protocol by selecting checkpoints for DRIFT that slightly exceed the baseline's coordinates. This approach confirms that our reported improvements represent a lower bound of the actual performance gap.

I Implementation Details

Following prior work Black et al. [2024], Liu et al. [2025a], we use both Stable Diffusion v1.5 (SDv1.5) Rombach et al. [2022] and Stable Diffusion 3.5 Medium (SD3.5-M) Esser et al. [2024] as the backbone models. Fine-tuning is performed using LoRA Hu et al. [2022] applied to the attention layers of the UNet Ronneberger et al. [2015] or Transformers, which significantly reduces training overhead while maintaining generation quality.

For SDv1.5, we employ the DDIM sampler with 50 sampling steps under an SDE-based stochastic formulation Song et al. [2021a]. Noise is injected during the process to enable stochastic sampling and likelihood estimation, with a classifier-free guidance (CFG) scale of 5. For SD3.5-M, we follow the methodology of Liu et al. [2025a] but modify the sampling strategy during training. We increase the number of sampling timesteps to 16, utilizing a hybrid strategy of 14 stochastic SDE steps followed by 2 deterministic ODE steps Song et al. [2021b], with a classifier-free guidance scale of 4.5. Notably, training is restricted to the initial 14 steps. We find that this configuration enhances the training efficacy and yields superior sampling quality. For evaluation, we utilize a deterministic ODE solver with 40 timesteps to generate samples.

We perform training on 45 common animals using two reward functions on SDv1.5, PickScore and HPSv2. For SD3.5-M, we train on a set of more complex compositional prompts involving objects, scenes, and indoor/outdoor contexts. The hyperparameter settings for each reward are listed in Table 3. All experiments are conducted on a system with two NVIDIA RTX 4090 GPUs, using FP16 mixed-precision training with automatic mixed precision (AMP) and gradient scaling for improved computational efficiency and reduced memory consumption. The KL ratio β is set to 0.001 for GRPO-KL. We use LoRA with $\alpha = 16$ and $r = 8$ for SDv1.5 and $\alpha = 64$ and $r = 32$ for SD3.5-M.

Table 3: List of hyperparameter configurations for PickScore and HPSv2.

Hyperparameters	**PickScore(SDv1.5)**	**HPSv2(SDv1.5)**	**PickScore(SD3.5-M)**	**HPSv2(SD3.5-M)**
Random seed	42	42	42	42
Denoising timesteps (T)	50	50	16	16
group size (G)	8	8	24	24
Guidance scale	5.0	5.0	4.5	4.5
Policy learning rate	1×10^{-4}	1×10^{-4}	3×10^{-4}	3×10^{-4}
Policy clipping range	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Maximum gradient norm	1.0	1.0	1.0	1.0
Optimizer	AdamW	AdamW	AdamW	AdamW
Optimizer weight decay	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
Optimizer β_1	0.9	0.9	0.9	0.9
Optimizer β_2	0.999	0.999	0.999	0.999
Optimizer ϵ	1×10^{-8}	1×10^{-8}	1×10^{-8}	1×10^{-8}
Sampling batch size	16	16	6	6
Samples per epoch	256	256	288	288
Training batch size	4	4	4	4
Gradient accumulation steps	16	16	36	36
Gradient updates per epoch	4	4	2	2
noise level α	1.0	1.0	0.7	0.7
shaping ratio λ	0.5	0.5	0.6	0.6
predetermined upper bound σ	1.0	1.0	1.0	1.0

J Reward Models

We use two human-preference-based reward models, HPSv2 and PickScore, to provide external reward feedback during training. Both models are trained to predict pairwise preferences between images generated from the same prompt.

HPSv2 is trained on a dataset containing approximately 434k images organized into pairwise comparisons. Each comparison consists of two images generated by different models using the same prompt and is annotated with a binary preference choice provided by a single annotator. The prompts are collected from DrawBench and DiffusionDB, where prompts from DiffusionDB are further sanitized using ChatGPT to reduce biases introduced by stylistic trigger words and to lower the overall NSFW score. This preprocessing results in a more controlled prompt distribution for training the reward model.

PickScore is trained on the Pick-a-Pic dataset, which is collected through a web-based application. Users are allowed to freely write prompts and are presented with two generated images per prompt. They are asked to select their preferred image or indicate a tie if no strong preference exists. While moderation is applied to remove users who generate NSFW content or exhibit low-quality annotation behavior (e.g., extremely rapid or random choices), the dataset still contains a notable amount of NSFW prompts. As a result, fine-tuning with PickScore may lead to an increased tendency to generate NSFW images, even when the input prompts are not explicitly NSFW.

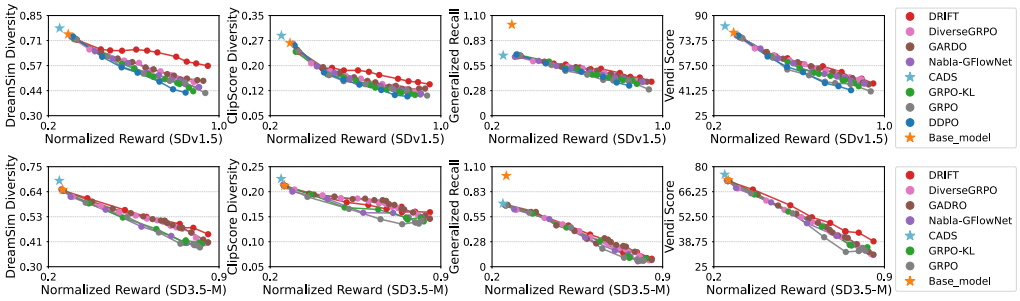

Figure 11: Reward-diversity comparison between DRIFT and baselines, with SDv1.5 and SD3.5-M fine-tuned using HPSv2 reward.

K Additional Experimental Results

K.1 Reward-Diversity and Qualitative Comparisons

Due to space constraints, additional results for Section 6.1 are provided in this appendix. Figure 11 presents Pareto frontiers for fine-tuning SDv1.5 and SD3.5-M with HPSv2. Figure 12, Figure 13, and Figure 14 present additional qualitative comparisons between DRIFT and baseline fine-tuning methods. While all models are fine-tuned on SDv1.5 or SD3.5-M using HPSv2 or PickScore as the reward function, clear differences in diversity emerge. Baseline methods frequently exhibit diversity collapse, generating visually similar samples that repeat the same breeds, poses, and background compositions across different generations. Although individual samples may achieve high reward scores, the overall output distribution is narrow and lacks meaningful variation.

In contrast, DRIFT consistently produces samples with substantially greater diversity while maintaining high visual fidelity. The generated images display richer variation in object appearance, pose, and scene composition, without introducing noticeable artifacts or degradation in image quality. These results suggest that DRIFT effectively mitigates reward-driven mode collapse and encourages exploration of diverse yet coherent solutions within the learned generative space. Importantly, the improved diversity observed in DRIFT is not limited to superficial texture or color changes. Instead, it reflects genuine structural and compositional variation, indicating that DRIFT promotes diversity at a semantic level rather than relying on low-level perturbations to increase perceptual distance.

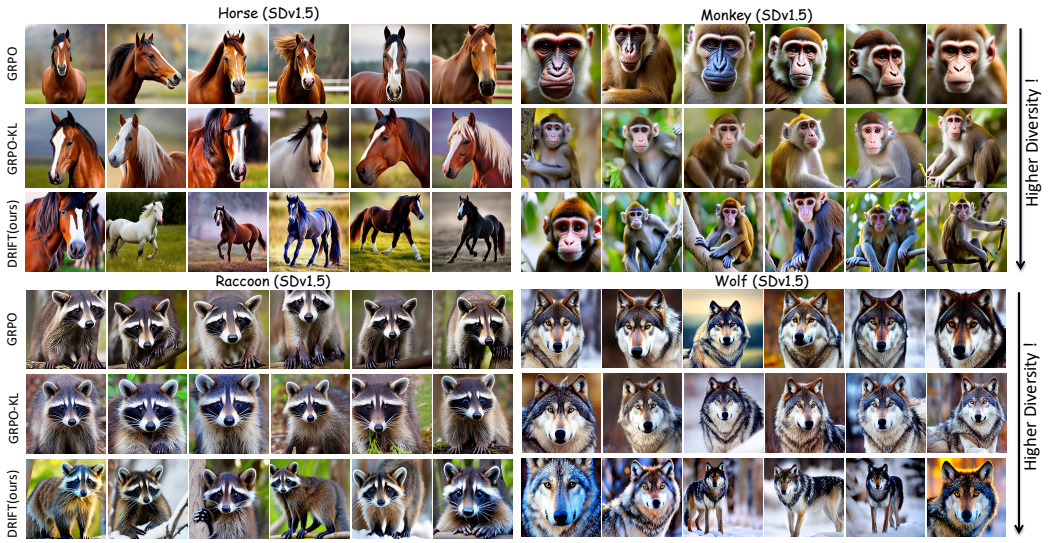

Figure 12: Qualitative diversity comparisons show that baseline methods suffer from diversity collapse, producing repetitive samples with similar breeds, poses, and backgrounds, whereas DRIFT maintains high fidelity with substantially greater diversity. All models are fine-tuned on SDv1.5 using HPSv2 as the reward function.

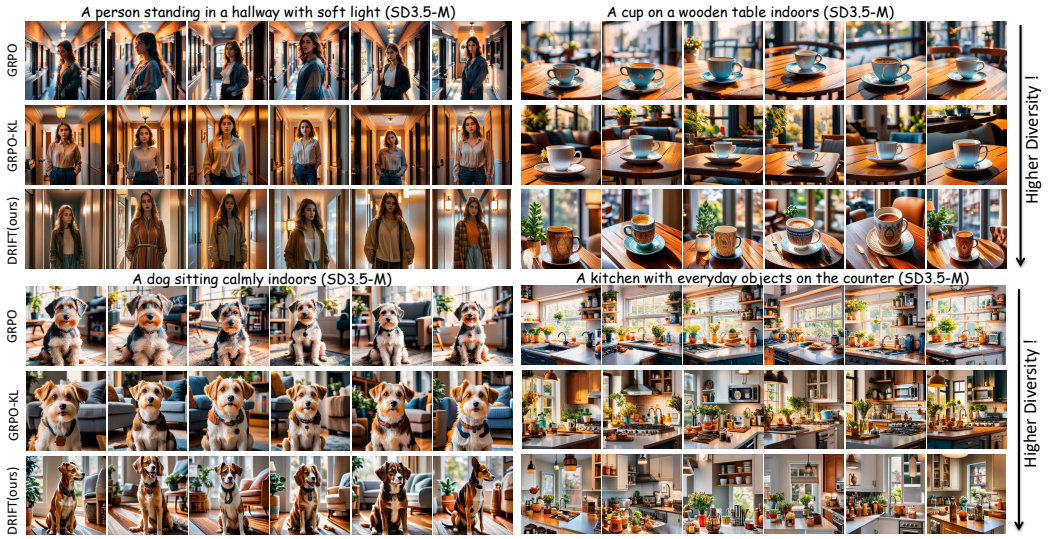

Figure 13: Qualitative diversity comparisons show that baseline methods suffer from diversity collapse, producing repetitive samples with similar visual attributes and scene compositions, whereas DRIFT maintains high fidelity with substantially greater diversity. All models are fine-tuned on SD3.5-M using HPSv2 as the reward function.

K.2 Evaluation on Expanded Prompt Sets

To further evaluate the robustness of DRIFT beyond the original 40-prompt evaluation set, we expand the evaluation to 100 prompts for SDv1.5 and 200 compositional prompts for SD3.5-M. The expanded evaluation covers more diverse generation conditions and assesses the reward-diversity trade-off across broader prompt sets. We additionally report Fréchet Inception Distance (FID), using images generated by the corresponding pretrained base model as the reference distribution. As shown in Table 4, DRIFT achieves average diversity and reward gains of 35.01% and 20.74%, respectively, over the original fine-tuning baselines. These results indicate that the improvements in reward

Table 4: Evaluation on expanded prompt sets with 100 prompts for SDv1.5 and 200 prompts for SD3.5-M. Each entry reports PickScore/HPSv2, with mean and standard deviation on separate lines. Higher values indicate better performance except for FID. Percentage improvements are averaged over the original fine-tuning baselines, excluding the base model and CADS.

Algorithm	DreamSim (↑)	CLIP (↑)	Recall (↑)	Vendi (↑)	Reward (↑)	FID (↓)
SDv1.5 (100 prompts)						
Base model	0.7508/0.7508 ±0.0081/ ± 0.0081	0.2753/0.2753 ±0.0036/ ± 0.0036	1.0000/1.0000 ±0.0000/ ± 0.0000	94.11/94.11 ±0.7093/ ± 0.7093	0.2969/0.2755 ±0.01237/ ± 0.00763	0.00/0.00 ±0.0000/ ± 0.0000
CADS	0.7771/0.7771 ±0.0032/ ± 0.0032	0.2958/0.2958 ±0.0031/ ± 0.0031	0.6532/0.6532 ±0.0046/ ± 0.0046	97.36/97.36 ±0.4540/ ± 0.4540	0.2787/0.2400 ±0.01275/ ± 0.00585	8.26/8.26 ±0.3935/ ± 0.3935
DDPO	0.4266/0.4255 ±0.0074/ ± 0.0078	0.1345/0.1222 ±0.0039/ ± 0.0027	0.2201/0.2919 ±0.0100/ ± 0.0149	47.74/46.49 ±0.5386/ ± 0.6237	0.6363/0.6577 ±0.01185/ ± 0.00761	24.21/22.35 ±0.9564/ ± 0.7033
GRPO	0.4143/0.4324 ±0.0047/ ± 0.0056	0.1345/0.1255 ±0.0023/ ± 0.0021	0.2201/0.2781 ±0.0063/ ± 0.0129	43.67/47.49 ±0.3499/ ± 0.3248	0.7030/0.6711 ±0.00883/ ± 0.00258	23.71/20.03 ±0.9456/ ± 1.2741
GRPO-KL	0.4722/0.4826 ±0.0042/ ± 0.0052	0.1393/0.1396 ±0.0024/ ± 0.0029	0.3363/0.3878 ±0.0072/ ± 0.0125	54.97/55.48 ±0.2102/ ± 0.2941	0.6666/0.6533 ±0.00964/ ± 0.00414	18.07/16.77 ±1.0315/ ± 0.9940
Nabla-GFlowNet	0.4592/0.4853 ±0.0068/ ± 0.0038	0.1357/0.1407 ±0.0047/ ± 0.0011	0.3351/0.3911 ±0.0074/ ± 0.0151	55.90/55.54 ±0.2042/ ± 0.5645	0.8909/0.6888 ±0.00627/ ± 0.00831	18.95/16.82 ±1.2421/ ± 1.0339
GARDO	0.4665/0.5274 ±0.0040/ ± 0.0042	0.1450/0.1595 ±0.0016/ ± 0.0033	0.2637/0.4082 ±0.0168/ ± 0.0059	48.76/60.67 ±0.1175/ ± 0.5981	0.8181/0.7955 ±0.00616/ ± 0.00616	24.10/20.15 ±1.0256/ ± 0.7522
DiverseGRPO	0.5127/0.5355 ±0.0026/ ± 0.0049	0.1578/0.1606 ±0.0011/ ± 0.0018	0.3389/0.4196 ±0.0106/ ± 0.0079	55.64/62.16 ±0.3879/ ± 0.2952	0.8545/0.8666 ±0.00210/ ± 0.00709	20.92/15.78 ±0.9218/ ± 0.9127
DRIFT	0.6268/0.6428 ±0.0063/ ± 0.0022	0.1948/0.1890 ±0.0023/ ± 0.0008	0.3387/0.4691 ±0.0188/ ± 0.0036	56.48/65.59 ±0.4966/ ± 0.1201	0.9393/0.9288 ±0.00969/ ± 0.00871	22.14/18.78 ±1.0670/ ± 0.5776
Improve (avg.)	+37.35%/+34.55%	+38.48%/+35.19%	+22.99%/+32.84%	+11.46%/+21.52%	+25.39%/+30.09%	-3.65%/-2.27%
SD3.5-M (200 prompts)						
Base model	0.6485/0.6486 ±0.0034/ ± 0.0034	0.2138/0.2138 ±0.0016/ ± 0.0013	1.0000/1.0000 ±0.0000/ ± 0.0000	81.18/81.19 ±0.4022/ ± 0.3986	0.2875/0.2133 ±0.00474/ ± 0.00083	0.00/0.00 ±0.0000/ ± 0.0000
CADS	0.6913/0.6926 ±0.0041/ ± 0.0042	0.2312/0.2316 ±0.0026/ ± 0.0028	0.5984/0.5694 ±0.0115/ ± 0.0058	84.76/85.12 ±0.5927/ ± 0.6536	0.2737/0.1990 ±0.00247/ ± 0.00326	5.75/5.67 ±0.0985/ ± 0.2142
GRPO	0.3612/0.3479 ±0.0001/ ± 0.0011	0.1016/0.1199 ±0.0003/ ± 0.0002	0.0807/0.0234 ±0.0046/ ± 0.0039	36.81/31.68 ±0.0683/ ± 0.1193	0.8025/0.7371 ±0.00470/ ± 0.00263	47.26/54.65 ±0.5443/ ± 0.5982
GRPO-KL	0.3939/0.3728 ±0.0009/ ± 0.0028	0.1115/0.1312 ±0.0008/ ± 0.0019	0.1173/0.0381 ±0.0026/ ± 0.0033	40.23/40.23 ±0.1577/ ± 0.2267	0.8112/0.7523 ±0.00480/ ± 0.00209	42.36/47.86 ±0.4311/ ± 0.5490
Nabla-GFlowNet	0.4472/0.3364 ±0.0008/ ± 0.0026	0.1218/0.1149 ±0.0025/ ± 0.0011	0.1496/0.0127 ±0.0031/ ± 0.0057	49.74/27.56 ±0.1649/ ± 0.1504	0.7900/0.8638 ±0.00049/ ± 0.00299	31.49/60.32 ±0.5943/ ± 0.6574
GARDO	0.4485/0.4040 ±0.0012/ ± 0.0032	0.1151/0.1456 ±0.0014/ ± 0.0017	0.1051/0.0509 ±0.0044/ ± 0.0096	46.81/33.25 ±0.1583/ ± 0.2347	0.8437/0.8657 ±0.00292/ ± 0.00275	37.60/43.54 ±0.6173/ ± 0.6362
DiverseGRPO	0.4333/0.4062 ±0.0025/ ± 0.0025	0.1147/0.1463 ±0.0009/ ± 0.0011	0.0917/0.0553 ±0.0029/ ± 0.0090	45.73/33.68 ±0.2433/ ± 0.0814	0.8650/0.8580 ±0.00515/ ± 0.00080	36.54/42.30 ±0.5869/ ± 0.5315
DRIFT	0.4591/0.4404 ±0.0023/ ± 0.0014	0.1335/0.1685 ±0.0008/ ± 0.0004	0.1807/0.0638 ±0.0085/ ± 0.0046	50.20/40.11 ±0.1872/ ± 0.1825	0.9350/0.9219 ±0.00354/ ± 0.00240	36.28/44.53 ±0.3872/ ± 0.6907
Improve (avg.)	+10.93%/+18.61%	+18.62%/+29.30%	+73.55%/+136.64%	+15.81%/+22.31%	+13.81%/+13.66%	+5.32%/+8.82%

and generation diversity extend beyond the original evaluation prompts. The overall diversity gain averages the four diversity metrics across the four backbone–reward settings with equal weight; the overall reward gain averages the four settings. Compared with fine-tuning baselines, DRIFT achieves average FID reductions of 5.32% and 8.82% on SD3.5-M under PickScore and HPSv2, respectively, but increases of 3.65% and 2.27% on SDv1.5. Thus, reward and diversity gains do not always imply closer agreement with the pretrained model's output distribution.

K.3 Component Interaction Ablation

We conduct a comprehensive ablation study on SDv1.5 with PickScore to investigate the individual and combined effects of Reward-Concentrated Sampling (RCS), Noise-Conditioned Prompting (NCP), and Reward Shaping (RS). We evaluate all seven nonempty combinations of the three components, as reported in Table 5. Each component individually improves reward and generation diversity, while pairwise combinations generally yield further improvements. Notably, NCP + RCS + RS outperforms every pairwise combination across all reported metrics, demonstrating the complementary contributions of the three components.

To complement the quantitative ablation, we provide a qualitative comparison using the same prompt to illustrate the effects of progressively integrating the proposed components. As shown in Fig. 15, we successively add RS, RCS, and NCP to the GRPO baseline. All configurations use the same prompt, sampling seeds, and inference settings, with ten selected samples shown for each configuration. The

displayed samples suggest increasing variation in appearance, pose, and background while remaining consistent with the prompt, in line with the quantitative results in Table 5.

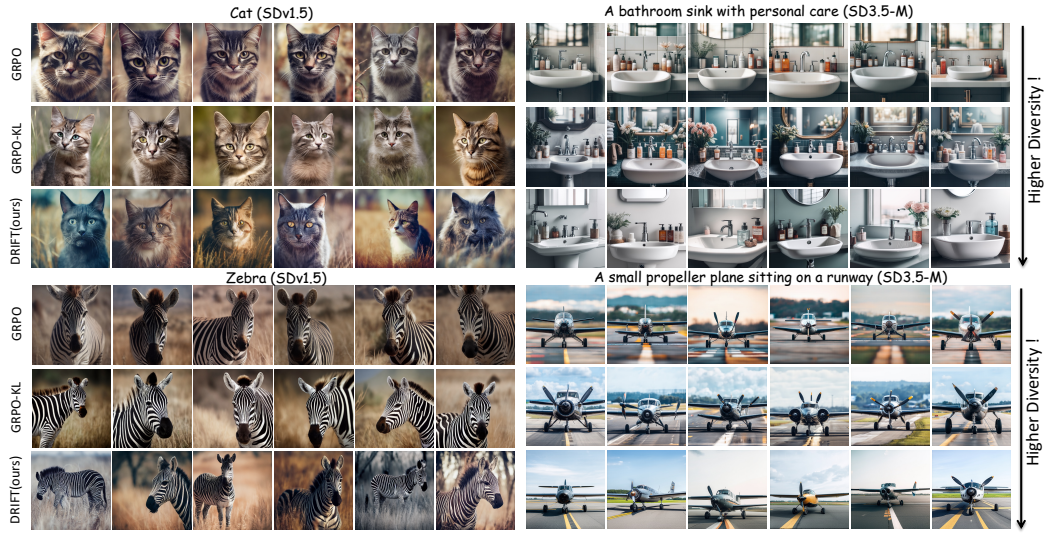

Figure 14: Qualitative diversity comparisons show that baseline methods suffer from diversity collapse, producing repetitive samples with similar visual attributes and scene compositions, whereas DRIFT maintains high fidelity with substantially greater diversity. All models are fine-tuned on SDv1.5 and SD3.5-M using PickScore as the reward function.

Table 5: Component interaction ablation on SDv1.5 with PickScore. Values report relative improvements (%) in diversity and reward under the experimental comparison protocol.

Components	DreamSim	CLIP	Recall	Vendi	Reward
NCP	+8.51	+15.38	+5.41	+2.86	+11.11
RCS	+24.24	+29.48	+7.32	+7.78	+45.76
RS	+49.79	+29.05	+49.27	+21.65	+55.00
NCP + RCS	+30.38	+30.68	+26.70	+12.66	+39.95
NCP + RS	+59.91	+50.72	+26.46	+17.50	+81.18
RCS + RS	+65.17	+59.10	+49.02	+27.60	+79.84
NCP + RCS + RS	**+83.52**	**+86.88**	**+49.62**	**+31.31**	**+126.99**

L Impact Statement

This paper presents DRIFT, a reinforcement fine-tuning framework designed to mitigate diversity collapse when optimizing preference/quality rewards for image generation models, improving alignment while preserving output variability and coverage. The method can increase the practicality and robustness of generative systems in creative design, content production, data augmentation, and human–AI interaction: better-preserved diversity reduces repetitive outputs and improves coverage of long-tail concepts and styles. Potential risks include that a stronger and more diverse generation may lower the barrier for large-scale synthesis of misleading or deceptive imagery, intensify copyright and likeness-right disputes, or amplify biases present in training data (e.g., stereotypical depictions of demographic groups). Moreover, if the reward model is biased, RL fine-tuning may further entrench or amplify such bias. To mitigate these risks in deployment, we recommend combining content safety filtering with watermarking/provenance mechanisms; auditing reward models and data for bias; restricting generation and re-training in high-risk domains (e.g., political figures, minors, sensitive contexts); and reporting both alignment and diversity/fairness metrics to avoid unintended consequences driven by a single reward signal.

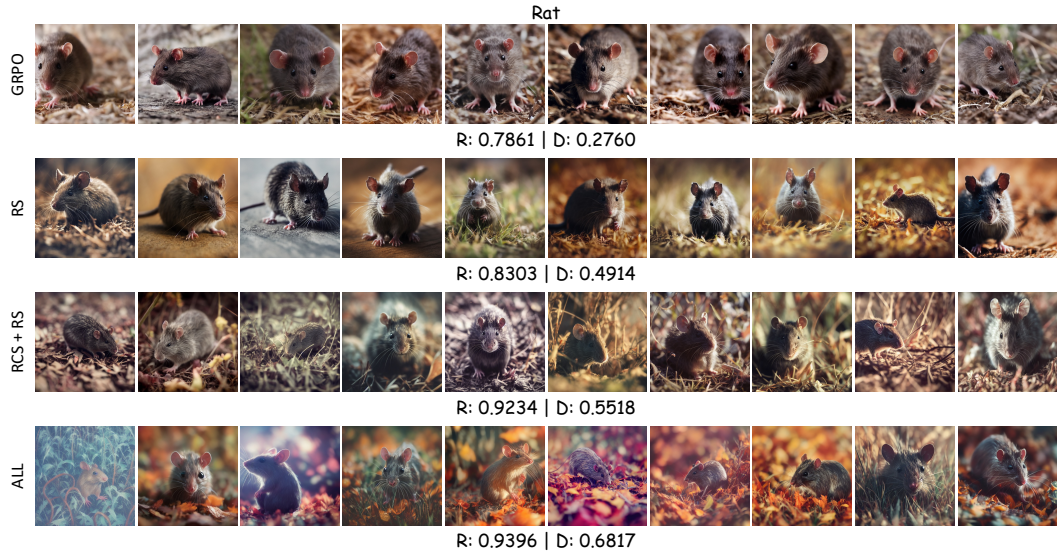

Figure 15: Qualitative comparison of the progressive integration of the proposed components on SDv1.5 using PickScore as the reward function. The prompt is "Rat." From top to bottom, the configurations are GRPO, GRPO + RS, GRPO + RCS + RS, and GRPO + RCS + RS + NCP (ALL). All configurations are evaluated using the same prompt, sampling seeds, and inference settings. R denotes the mean normalized PickScore reward, and D denotes the intra-group DreamSim diversity. Ten selected samples are shown for each configuration.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to the Abstract and Section 1: Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Section 7 for Conclusions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper includes theoretical results with explicit assumptions and complete proofs. Theorem 1 states the assumptions for optimal policy invariance under potential-based reward shaping, and its proof is provided in Appendix A; additional derivations for the KL-constrained optimum and the limiting Dirac-delta behavior are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the key methodological and experimental details needed to reproduce the main results, including the model backbones, reward models, evaluation metrics, baselines, training configurations, optimizer settings, and hyperparameters in Section 6, Appendix I, and Table 3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The training implementation of DRIFT is publicly available at https://github.com/NJU-RL/DRIFT. The released code supports reproduction of the proposed method, while the experimental settings, evaluation protocol, and hyperparameters are detailed in Section 6 and the appendices.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the main experimental settings, including backbone models, reward functions, evaluation metrics, baselines, training and sampling configurations, optimizer settings, and hyperparameters. These details are provided in Section 6, Appendix I, and Table 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tables 1 and 4 report means and standard deviations. We do not report formal statistical significance tests.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the computational setup used for the experiments, including the GPU configuration and mixed-precision training details, in Appendix I. The main training hyperparameters required to reproduce the reported experiments are also provided in Table 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The paper does not involve human-subject studies, private data collection, or the release of high-risk datasets or models. Potential societal risks of improved image generation, including misuse, bias amplification, and copyright or likeness concerns, are discussed in Appendix L, together with possible mitigation strategies.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Appendix L.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new datasets, model checkpoints, or deployed image-generation systems. Potential misuse risks of the method are discussed in Appendix L.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: The paper uses existing models, reward models, and evaluation encoders, including Stable Diffusion, PickScore, HPSv2, DreamSim, and OpenCLIP. These assets are credited through citations in the main text and appendices. We use them only for research purposes and follow their publicly released licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We publicly release the training implementation of DRIFT, with accompanying documentation and configurations to support reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects. As a result, there is no need to provide instructions given to participants, screenshots, or details about compensation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects. Therefore, there are no potential risks to disclose, and no Institutional Review Board (IRB) or equivalent approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methods of this research do not involve Large Language Models (LLMs) as important, original, or non-standard components. As such, there is no need to provide a declaration regarding the usage of LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.