

Reward Fine-Tuning Two-Step Diffusion Models via Learning Differentiable Latent-Space Surrogate Reward

Zhiwei Jia, Yuesong Nan, Huixi Zhao, Gengdai Liu

Zoom Communications

{firstname.lastname}@zoom.com

Abstract

*Recent research has shown that fine-tuning diffusion models (DMs) with arbitrary rewards, including non-differentiable ones, is feasible with reinforcement learning (RL) techniques, enabling flexible model alignment. However, applying existing RL methods to step-distilled DMs is challenging for ultra-fast (≤ 2 -step) image generation. Our analysis suggests several limitations of policy-based RL methods such as PPO or DPO toward this goal. Based on the insights, we propose fine-tuning DMs with learned differentiable surrogate rewards. Our method, named **LaSRO**, learns surrogate reward models in the latent space of SDXL to convert arbitrary rewards into differentiable ones for effective reward gradient guidance. LaSRO leverages pre-trained latent DMs for reward modeling and tailors reward optimization for ≤ 2 -step image generation with efficient off-policy exploration. LaSRO is effective and stable for improving ultra-fast image generation with different reward objectives, outperforming popular RL methods including DDPO [2] and Diffusion-DPO [76]. We further show LaSRO’s connection to value-based RL, providing theoretical insights. See our webpage [here](#).*

1. Introduction

Diffusion models (DMs) [19, 66, 68, 69] have significantly transformed generative modeling for continuous data, delivering remarkable results across various data modalities. In particular, recent advances [56, 57, 59] demonstrate their capabilities of generating diverse and high-fidelity images according to free-form input text prompts, albeit at the cost of a slow iterative sampling process. Various methods are proposed to accelerate such a sampling process [39, 41, 43, 67], with the most recent distillation-based methods [42, 60, 61, 70] pushing the envelope to achieve extreme few-step generation (≤ 2 steps).

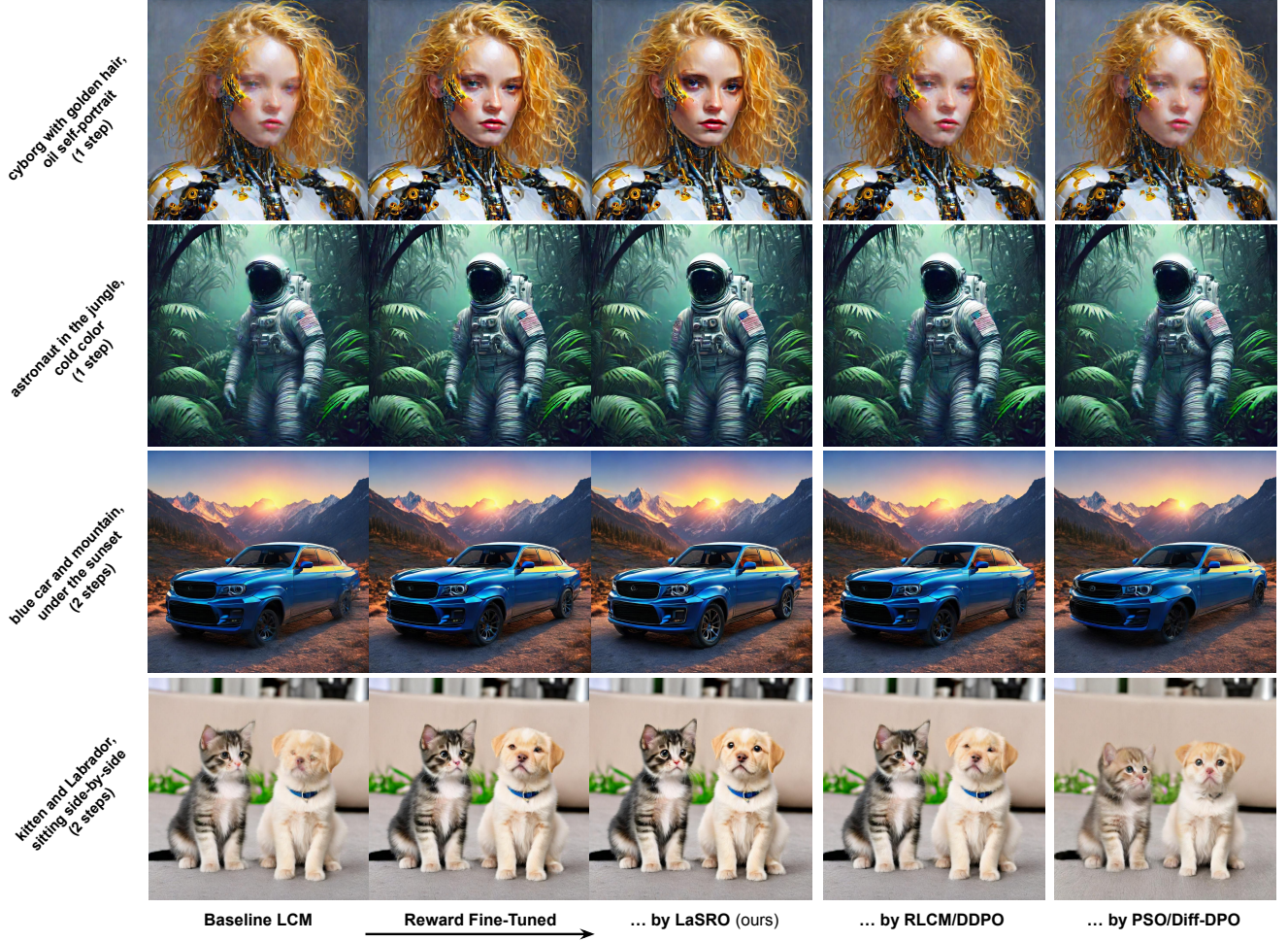
On the other hand, state-of-the-art image diffusion models pre-trained on Internet-scale datasets (e.g. LAION-5B

[62]) usually incorporate a fine-tuning stage that guides the model’s behavior towards specific human preferences (aesthetics, debiased, etc.) This is achieved either by supervised learning (SFT) with curated human preference datasets [29, 80, 84] or through reinforcement learning (RL) to leverage a diverse set of reward signals, including those not naturally differentiable such as incompressibility, or potentially the ones involving large language models (LLMs) or vision-language models (VLMs) [24, 44, 74].

However, it is non-trivial to fine-tune time-distilled DMs for ≤ 2 -step image generation with policy-based RL (e.g., PPO [64] or DPO [54]). We detailedly analyze the emerged optimization challenges negligible for non-distilled DMs. Most RL objectives are *ineffective* for ≤ 2 -step DMs due to the deterministic 2nd sampling step. The mapping of the distilled 2-step sampler from noisy images to clean ones (and thus to most reward signals) is *highly non-smooth*, making policy gradient estimation challenging. RL methods relying on the denoising diffusion loss (such as Diffusion-DPO [76]) are incompatible with such a property, leading to blurred images. Moreover, the 2-step mapping lacks enough stochasticity, posing a *hard exploration* problem. These conditions make most RL methods infeasible or ineffective for fine-tuning two-step DMs.

To address these limitations, we propose to learn a surrogate reward model in the latent space (of SDXL [50]) that converts an arbitrary reward signal into a differentiable one. Reward fine-tuning with this learned surrogate avoids policy gradient estimation as it directly and efficiently provides reward gradient guidance. Specifically, we propose to use a pre-trained latent diffusion model as the backbone of the surrogate model, which shows superior memory & speed efficiency and better generalizability compared to the alternatives based on VLMs (CLIP [53] and BLIP [35]). Additionally, our method achieves efficient exploration for 2-step image DMs with off-policy samples. The overall pipeline consists of two stages: the pre-training stage for learning the surrogate reward and the reward optimization stage that alternates between improving the DMs and online adapting the surrogate. We call our method **LaSRO**, i.e., **Latent-**

Figure 1. Generated images of resolution 1024^2 with ≤ 2 steps via LCM-SSD-1B [45] (baseline) and those fine-tuned with Image Reward [84]. 1st column: baseline results. 2nd and 3rd: results during and after fine-tuning the baseline via our method LaSRO. 4th: fine-tuned via RLCM [48] (a variant of DDPO [2]). 5th: fine-tuned via PSO [46] (a variant of Diffusion-DPO [76]). Ours significantly improves the visual quality of ≤ 2 -step image generation while other strong RL methods fail due to training instability and inefficiency.



space **Surrogate Reward Optimization**. LaSRO effectively fine-tunes two-step DMs across different reward objectives (including non-differentiable ones) and outperforms popular RL methods (DDPO [2], Diffusion-DPO [76], etc.). We further show a connection of LaSRO to value-based RL [78] to provide theoretical insights behind our approach. In summary, our contributions are:

- We analyze the key challenges in applying RL to improve ≤ 2 -step image diffusion models.
- We propose a framework, named LaSRO, that uses latent-space surrogate rewards to optimize arbitrary reward signals with efficient off-policy exploration.
- We establish LaSRO’s connection to value-based RL, demonstrate its advantages over popular policy-based RL methods, and validate its design with ablation studies.

2. Related Work

Diffusion Models (DMs) Diffusion models [19, 66, 68, 69] have become the dominant generative models for various continuous data modalities [5, 20, 21, 26, 31, 40, 51, 55, 58, 86]. Particularly, recent advances [56, 57, 59] make text-to-image generation vastly popular. A major drawback of DMs is their lengthy iterative denoising process during inference, often requiring many sampling steps.

Step-Distilled Diffusion Models Recent research strives to achieve ≤ 2 step generation through step-distillation of a teacher diffusion model. Latent Consistency Models (LCMs) [45] apply Consistency Distillation [70] to learn to map any point along the sampling trajectories of the teachers (latent DMs) to their endpoints with one sampling step. Progressive Distillation [60] incrementally distills a diffu-

sion model into a faster one. InstaFlow [42] apply distillation after modifying the teacher DM’s probability flow to be smoother. Adversarial Diffusion Distillation [61] distills a teacher into a faster student via adversarial training.

Fine-tuning DMs with Arbitrary Reward Signals Reward fine-tuning DMs with RL becomes promising for aligning model outputs with arbitrary reward signals, including non-differentiable ones. The simplest form of RL comes in as reward-weighted regression (RWR) [49], including [9, 24, 33]. Others [8, 76, 85] based on direct preference optimization, DPO [54], or its variant [37], can be seen as RWR-based approaches that fall into this category. Policy-based RL methods [2, 4, 11–13] relying on policy gradient or its equivalence handle arbitrary rewards by using large quantities of online samples. Value-based RL methods [78] have been largely under-explored for fine-tuning DMs since they can struggle with long-horizon MDPs with delayed rewards [72]. On the other hand, supervised fine-tuning methods [7, 52, 81, 84] align DMs by back-propagating the reward gradient through the denoising process to update the model parameters. While effective, they are constrained to scenarios where specifically labeled preference datasets are available [29, 79, 80, 84] or the reward signal is differentiable. Many recent rewards [24, 44, 74] are not naturally differentiable as they involve VLMs [35] or LLMs [1].

Reward Fine-tuning \leq 2-Step Step-Distilled DMs Fine-tuning DMs with policy-based RL remains a great challenge for ≤ 2 -step generation. The distilled diffusion sampler lacks stochasticity and the underlying mapping is highly non-smooth, making methods based on either policy gradient estimation or reward-weighted regression ineffective (see Sec. 4 for details). Very recently, RLCM [48] and PSO [46] were proposed to reward fine-tuning step-distilled DMs with policy-based RL, and RG-LCD [36] was proposed to use reward guidance in the step-distillation process of LCMs. However, they are ineffective or inefficient in improving ≤ 2 -step DMs with arbitrary rewards.

3. Preliminaries

3.1. Diffusion Models (DMs)

We consider the conditional denoising diffusion probabilistic models [19, 66], which represent an image generation distribution $p(\mathbf{x}_0|\mathbf{c})$ over a sample image \mathbf{x}_0 given a text prompt \mathbf{c} . The distribution is modeled as the reverse of a Markovian diffusion process $q(\mathbf{x}_{t+1}|\mathbf{x}_t)$ that progressively adds Gaussian noise Z_t to (a scaled version of) \mathbf{x}_0 . To reverse the diffusion, a time-conditioned neural network, parametrized with ϵ -prediction as $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$, is trained by maximizing a variational lower bound on the log-likelihood

of the training data, or equivalently, minimizing the loss:

$$\mathcal{L}_{\text{ddpm}}(\theta) = \mathbb{E} [\|Z_t - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})\|^2] \quad (1)$$

where the expectation is taken over all training samples $(\mathbf{x}_0, \mathbf{c})$, timestep $t \in \{0, \dots, T\}$, and the corresponding noisy samples $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ with the added noise Z_t . For simplicity, we ignore the scaling terms and denote the predicted output (rather than the noise) as $\mu_\theta(\mathbf{x}_t, t, \mathbf{c})$.

Sampling from a diffusion model starts with Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively generates $\mathbf{x}_{T-1}, \mathbf{x}_{T-2}$ all the way to \mathbf{x}_0 . With some pre-set monotonically decreasing noise schedules $\{\sigma_t^2\}$, a common choice to parametrize the reverse denoising step given a timestep t is:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1} | \mu_\theta(\mathbf{x}_t, t, \mathbf{c}), \sigma_t^2 \mathbf{I}) \quad (2)$$

3.2. Latent Consistency Models (LCMs)

Aimed for fast image generation, the latent consistency model (LCM) [45], a type of step-distilled DMs, was proposed to perform Consistency Distillation [70] in the latent space of the teacher DMs (e.g., SDXL [50]). We select LCMs as our test bed due to their simplicity, efficiency, and training stability (our approach is *not* tied to LCMs, see Tab. 1). For simplicity, we abuse the notation to use \mathbf{x}_t to also refer to the latent code. LCM learns a function $f_\theta(\mathbf{x}_t, t, \mathbf{c})$ that maps any sample \mathbf{x}_t on a denoising trajectory to the endpoint \mathbf{x}_0 by “compressing” the trajectory induced by the teacher DMs (see Fig. 2 upper right). It enforces its output given \mathbf{x}_t to be close to its output given \mathbf{x}_{t-1} , while overriding $f_\theta(\mathbf{x}_0, 0, \mathbf{c}) = \mathbf{x}_0$ for the boundary condition. The (simplified) LCM training loss is:

$$\mathcal{L}_{\text{lcm}}(\theta) = \mathbb{E} \left[d \left(f_\theta(\mathbf{x}_t, t, \mathbf{c}), f_\theta(\hat{\mathbf{x}}_{t-1}^\phi, t-1, \mathbf{c}) \right) \right] \quad (3)$$

where $d(\cdot, \cdot)$ is a metric function; the expectation is taken over \mathbf{x}, \mathbf{c} and t ; stop gradient is applied to the 2nd f_θ in $d(\cdot, \cdot)$; $\hat{\mathbf{x}}_{t-1}^\phi$ is a sample obtained from \mathbf{x}_t with one denoising step using the teacher diffusion model ϕ given \mathbf{c} .

While LCMs produce images with only one step, by re-injecting noise back to the results of each LCM step (except for the last one), LCMs benefit from a multi-step sampling procedure similar to regular DMs. Namely, similar to Eq. 2, each LCM sampling step (except for the last one) follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1} | f_\theta(\mathbf{x}_t, t, \mathbf{c}), \sigma_t^2 \mathbf{I}) \quad (4)$$

Again, scaling terms are ignored. Usually, only up to 8 steps are taken for LCMs (much fewer than regular DMs).

3.3. RL Formulation of Fine-tuning DMs & LCMs

We formulate RL fine-tuning of DMs using Markov Decision Process (MDP) [72]. Similar to [2], we map a diffusion model to the following finite-horizon MDP

$(\mathcal{S}, \mathcal{A}, P, R, \mu, H)$ with policy $\pi(\mathbf{a}_t | \mathbf{s}_t)$ and a task reward $r(s, \mathbf{c})$ that evaluates the image given the text prompt \mathbf{c} :

$$\begin{aligned} \mathbf{s}_t &\triangleq (\mathbf{x}_{\tau_t}, \tau_t, \mathbf{c}) \in \mathcal{S} \quad \mathbf{a}_t \triangleq \mathbf{x}_{\tau_{t+1}} \in \mathcal{A} \\ \pi(\mathbf{a}_t | \mathbf{s}_t) &\triangleq p_\theta(\mathbf{x}_{\tau_{t+1}} | \mathbf{x}_{\tau_t}, \mathbf{c}) \\ P(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) &\triangleq (\delta_{\mathbf{x}_{\tau_{t+1}}}, \delta_{\tau_{t+1}}, \delta_{\mathbf{c}}) \\ R(\mathbf{s}_t, \mathbf{a}_t) &= \begin{cases} r(\mathbf{a}_t, \mathbf{c}) & \text{if } t = H - 1 \\ 0 & \text{otherwise} \end{cases} \\ \mu &\triangleq (p(\mathbf{c}), \delta_{\tau_0}, \mathcal{N}(\mathbf{0}, \mathbf{I})) \end{aligned}$$

where δ_y is the Dirac δ -distribution and $p_\theta(\mathbf{x}_{\tau_{t+1}} | \mathbf{x}_{\tau_t}, \mathbf{c})$ is defined according to Eq. 2 or 4. Note that, in MDP formulation (and for the rest of this paper), in contrast to the previous notation, the initial timestep is denoted τ_0 , and after H denoising steps, the last timestep is denoted τ_H .

For LCMs (f_θ from Sec. 3.2), since the intermediate sampling result lies in the image/latent space where a non-zero reward is defined, the MDP formulation differs by:

$$R(\mathbf{s}_t, \mathbf{a}_t) = r(f_\theta(\mathbf{x}_{\tau_t}, \tau_t, \mathbf{c}), \mathbf{c}), \forall t \quad (5)$$

4. What are the RL Obstacles?

We present an insightful analysis of three major challenges.

4.1. Hard Exploration for Two-Step Image DMs

Efficient and effective exploration in the state space is key to all online RL methods. Fine-tuning two-step DMs poses a hard exploration problem for RL. Diffusion models, expressed as stochastic policies $p_\theta(\mathbf{x}_{\tau_H} | \mathbf{x}_{\tau_0}, \mathbf{c})$, explore the image/latent space by repeatedly injecting Gaussian noise during the sampling process (Eq. 2 or 4). While regular DMs incrementally add noise to render diverse images, step-distilled DMs add noise in significantly fewer increments, leading to less diverse images, an observation we visualize in Fig. 2 (left). Given a fixed initial noise \mathbf{x}_{τ_0} and a prompt \mathbf{c} , one-step DMs (i.e., LCMs) are *deterministic* functions, and, by *injecting noise only once*, two-step LCMs allow *little exploration*:

$$\mathbf{x}_{\tau_H} \sim f_\theta(f_\theta(\mathbf{x}_{\tau_0}, \tau_0, \mathbf{c}) + Z, \tau_{H/2}, \mathbf{c}) \quad (6)$$

where scaling terms are ignored; Z is the injected noise at timestep $\tau_{H/2}$ as the only source of stochasticity. This limited exploration hinders on-policy RL methods.

4.2. Degenerated RL Objectives for Two-Step DMs

Typically, policy-based RL optimizes the log-likelihood of online samples to enhance the policy’s expected reward. However, most policy-based methods have their objectives degenerated when fine-tuning two-step DMs, substantially impeding their effectiveness. For instance, a simple form of DDPO [2] to fine-tune DMs has its objective and gradient:

$$\mathcal{J}_{\text{ddpo}}(\theta) = \mathbb{E}_{\mathbf{c} \sim p(\mathbf{c}), \mathbf{x}_{\tau_H} \sim p_\theta(\mathbf{x}_{\tau_H} | \mathbf{c})} [r(\mathbf{x}_{\tau_H}, \mathbf{c})] \quad (7)$$

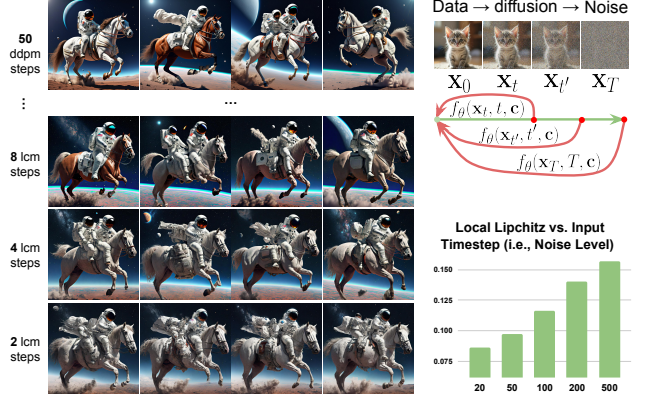


Figure 2. (left) Given *astronaut riding a horse* as \mathbf{c} , fixed initial noise \mathbf{x}_{τ_0} and guidance scale, the reduction of noise-injecting steps leads to less diverse images generated from SSD-1B [17] and its LCM, and thus harder exploration for $p_\theta(\mathbf{x}_{\tau_H} | \mathbf{x}_{\tau_0}, \mathbf{c})$. (right) Illustration of the LCM mapping f_θ . We show the growing empirical local Lipschitz of the LCM mapping from a noisy image to a clean image and to a generic image quality score (details in Appendix A). The x-axis is the input’s noise level (t as in \mathbf{x}_t).

$$\nabla_\theta \mathcal{J}_{\text{ddpo}} = \mathbb{E} \left[\sum_{t=0}^{H-1} \nabla_\theta \log p_\theta(\mathbf{x}_{\tau_{t+1}} | \mathbf{x}_{\tau_t}, \mathbf{c}) r(\mathbf{x}_{\tau_H}, \mathbf{c}) \right] \quad (8)$$

For two-step LCMs, since the 2nd step is deterministic, $p_\theta(\mathbf{x}_{\tau_H} | \mathbf{x}_{\tau_{H/2}}, \mathbf{c})$ does not exist; Eq. 7 degenerates to:

$$\mathcal{J}_{\text{ddpo}}^{\text{deg}} = \mathbb{E} [\nabla_\theta \log p_\theta(\mathbf{x}_{\tau_{H/2}} | \mathbf{x}_{\tau_0}, \mathbf{c}) r(\mathbf{x}_{\tau_H}, \mathbf{c})] \quad (9)$$

This objective only accounts for the *first half* of the sampling process, making DDPO ineffective. Similar issues can be found in other policy-based methods such as DPO [46].

4.3. Non-Smooth Mappings of Two-Step DMs

The non-smoothness of the mapping that underlies two-step DMs makes it challenging to fine-tune them with methods based on either policy gradient estimation or reward-weighted regression (RWR). Specifically, with a distilled two-step diffusion sampler, the underlying mapping from a noisy image to a clean image is highly non-smooth (i.e., has a great Lipschitz constant). Thus, slight changes in actions could drastically change most rewards. The same holds for two-step latent DMs: an LCM step $f_\theta(\mathbf{x}_{\tau_t}, \tau_t, \mathbf{c})$ denoising an input with a higher noise level (τ_t closer to τ_0) requires greater model expressiveness, resulting in a higher (local) Lipschitz constant [38]. Note that, the 2nd step of a two-step LCM handles a very high noise level from $\mathbf{x}_{\tau_{H/2}}$, empirically verified in Fig. 2 (right).

This non-smoothness makes it difficult to estimate policy gradients, leading to training instability of PPO [64]. In fact, many online RL methods that rely on policy gradient theorem [73] suffer from high variance gradient estima-

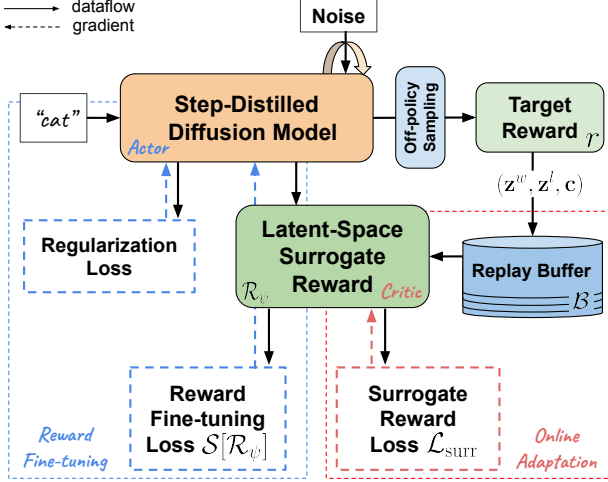


Figure 3. Training pipeline of the fine-tuning stage of LaSRO. This stage alternates between reward fine-tuning the DM and on-line adapting the surrogate reward. Since connected to value-based RL, we present LaSRO in the style of actor-critic methods [30].

tion due to the non-smoothness. Previously, “sharpness”, a closely related concept, was adopted in the study of robustness of supervised learning, generative modeling, and RL [14, 27, 28, 32]. In contrast, many-step diffusion sampling “divides” the high Lipschitz so that policy gradient estimation becomes much easier for each step.

Methods based on RWR [49] also suffer severely from the non-smoothness. RWR mitigates high-variance policy gradient estimation by instead performing re-weighted regression of online samples to favor actions of higher rewards. Methods based on DPO [54] fall into this category. So does PRDP [8], which applies RWR adaptive to the reward difference between good and bad samples. While RWR utilizes a re-weighted variant of the vanilla diffusion loss (Eq. 1), [46] shows that using Eq. 1 to fine-tune step-distilled DMs leads to blurry images by destroying the mapping induced by the step-distillation process. We argue it is because the loss does not account for the substantially increased non-smoothness of step-distilled DMs (particularly two-step DMs) compared to many-step DMs. PSO [46] utilizes additional constraints in response, yet is ineffective for two-step DMs.

5. Clearing the RL Obstacles with LaSRO

In this paper, we focus on reward fine-tuning 2-step DMs, targeting ultra-fast image generation (please see App. D for the inference speed). Formally, we introduce LaSRO, an RL training framework consisting of two stages: (1) pre-training the latent-space surrogate reward (Alg. 1). (2) reward fine-tuning the two-step DMs (Alg. 2). We visualize the pipeline of the latter stage in Fig. 3.

5.1. Latent-space Surrogate Reward Learning

We start with surrogate reward learning. Based on the insights from Sec. 4, we avoid both policy gradient estimation and reward-weighted regression. Instead, we propose to learn a *differentiable* surrogate reward that efficiently provides accurate reward gradient guidance for two-step image generation (guidance *also* provided for the second deterministic step). Direct reward gradient enabled by surrogate reward learning significantly improves exploration as noise injection is no longer the only way to explore when fine-tuning DMs. Similar techniques (i.e., differentiable physics) were proposed in RL for robotics [16, 23, 25].

We leverage large pre-trained multimodal models to learn a surrogate reward that provides gradients generalizable over unseen prompts or images. We evaluate different multimodal backbones including CLIP [53] (used as rewards in [36]), BLIP [35] (used as rewards in [84]), and the UNet encoder of SDXL [50] (used as discriminators in [38]). We use the most effective one - the encoders of pre-trained latent DMs as the backbone. We add a simple CNN-based prediction head on top of this backbone, together denoted as \mathcal{R}_ψ . We train it with a loss based on the Bradley-Terry model [3] in Eq. 10 w.r.t. the target reward signal r (potentially non-differentiable).

$$\mathcal{L}_{\text{surr}}(\psi; r) = -\mathbb{E}_{\mathbf{c}, \mathbf{z}^w, \mathbf{z}^l} [\log(y_\psi^{\mathbf{c}}(\mathbf{z}^w, \mathbf{z}^l))] \quad (10)$$

$$\mathbf{z}^w \succeq \mathbf{z}^l \text{ ranked according to reward } r$$

$$y_\psi^{\mathbf{c}}(\mathbf{z}^w, \mathbf{z}^l) = \frac{\exp(\mathcal{R}_\psi(\mathbf{z}^w, \mathbf{c}))}{\exp(\mathcal{R}_\psi(\mathbf{z}^l, \mathbf{c})) + \exp(\mathcal{R}_\psi(\mathbf{z}^w, \mathbf{c}))}$$

Namely, the surrogate reward \mathcal{R}_ψ produces $\mathcal{R}_\psi(\mathbf{z}, \mathbf{c}) \in \mathbb{R}$ given prompt \mathbf{c} and a latent code \mathbf{z} ; the winning and losing pair $(\mathbf{z}^w, \mathbf{z}^l)$ are selected from the samples (more on this shortly) according to the target reward r ; $\mathcal{L}_{\text{surr}}$ is a binary cross entropy loss that trains \mathcal{R}_ψ to predict a higher score for the winning sample \mathbf{z}^w , thus transforming the knowledge from r to \mathcal{R}_ψ . Additional benefits of this latent-space reward are the *improved computation & memory efficiency* (no need to back-prop through VAEs) and the potential to *capture image details better* (\mathcal{R}_ψ operates in the learned downsampled latent space; others [29, 80, 84] operates in the “brute-force” downsampled image space).

We target improvements in one and two-step image generation, a goal reflected in our design of *both* stages. In the pre-training stage with a pre-trained LCM f_θ , in each iteration, given a randomly drawn prompt \mathbf{c} , we sample N_s different 1st-step LCM outputs $\{\mathbf{z}_1 \triangleq f_\theta(\mathbf{z}_{\tau_0}, \tau_0, \mathbf{c})\}$ with N_s initial noise $\{\mathbf{z}_{\tau_0}\}$ (*off-policy* exploration, see Sec. 5.3) and then sample the 2nd LCM step to get $\{\mathbf{z}_2 \triangleq f_\theta(\mathbf{z}_1 + Z, \tau_{H/2}, \mathbf{c})\}$ with injected noise $\{Z\}$ as in Eq. 6. Presented in Alg. 1, in the pre-training stage, we train \mathcal{R}_ψ using $\{(\mathbf{z}_1^w, \mathbf{z}_1^l, \mathbf{c})\}$ selected from $\{\mathbf{z}_1\}$ and $\{(\mathbf{z}_2^w, \mathbf{z}_2^l, \mathbf{c})\}$

from $\{\mathbf{z}_2\}$, facilitating the downstream task - reward fine-tuning two-step DMs.

Algorithm 1 Pre-training Stage for Two-step DMs (e.g., LCMs)

Require: prompt set \mathcal{C} ; initial surrogate reward \mathcal{R}_ψ ; LCM f_θ ; learning rate η ; VAE decoder \mathcal{D} ; target reward (can be non-differentiable) r ; hyper-parameter N_s

- 1: **repeat**
- 2: Sample prompt $\mathbf{c} \sim \mathcal{C}$
- 3: Sample two-step LCM f_θ for N_s times to get $\{\mathbf{z}_1\}, \{\mathbf{z}_2\}$
- 4: Find a W/L pair $(\mathbf{z}_1^w, \mathbf{z}_1^l)$ from $\{\mathbf{z}_1\}$ acc. to $r(\mathcal{D}(\mathbf{z}_1), \mathbf{c})$
- 5: Find a W/L pair $(\mathbf{z}_2^w, \mathbf{z}_2^l)$ from $\{\mathbf{z}_2\}$ acc. to $r(\mathcal{D}(\mathbf{z}_2), \mathbf{c})$
- 6: Compute $\mathcal{L}_{\text{surr}}^1(\psi; r)$ from Eq. 10 given $(\mathbf{z}_1^w, \mathbf{z}_1^l, \mathbf{c})$
- 7: Compute $\mathcal{L}_{\text{surr}}^2(\psi; r)$ from Eq. 10 given $(\mathbf{z}_2^w, \mathbf{z}_2^l, \mathbf{c})$
- 8: $\psi \leftarrow \psi - \eta \nabla \psi [\mathcal{L}_{\text{surr}}^1(\psi; r) + \mathcal{L}_{\text{surr}}^2(\psi; r)]$
- 9: **until** convergence

5.2. Direct Reward Fine-tuning of Two-Step DMs

After pre-training, we enter the fine-tuning stage, which alternates between reward fine-tuning the DMs and online adapting the surrogate reward (which handles the distribution drift of the outputs of fine-tuned two-step DMs). We present this stage in Fig. 3 and in Alg. 2.

Algorithm 2 Fine-tuning Stage for Two-step DMs (e.g., LCMs)

Require: (besides $\mathcal{C}, f_\theta, \mathcal{D}, r, N_s$ from Alg. 1) pre-trained surrogate \mathcal{R}_ψ^* ; learning rates η_1, η_2 ; # steps N_1, N_2 ; reg. loss \mathcal{L}_{lcm} (Eq. 3); loss coeff. c, c_1, c_2 ; norm. and clipping function \mathcal{S}

```

1: repeat
2:   (Re-)initiate the replay buffer  $\mathcal{B} = \emptyset$ 
3:   repeat                                “Reward Fine-tuning”
4:     Sample prompt  $\mathbf{c} \sim \mathcal{C}$ 
5:     Sample  $\{\mathbf{z}_1\}, \{\mathbf{z}_2\}$  as results of a two-step LCM  $f_\theta$ 
6:     Compute  $\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_1, \mathbf{c})]$  and  $\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_2, \mathbf{c})]$ 
7:     Compute reg. loss with  $\mathcal{L}_{\text{lcm}}$  and  $\mathcal{L}_{\text{lasro}}$  from Eq. 11
8:      $\theta \leftarrow \theta - \eta_1 \mathcal{L}_{\text{lasro}}$ 
9:     Find  $\mathbf{z}_*^w, \mathbf{z}_*^l$  from  $\{\mathbf{z}_1\}, \{\mathbf{z}_2\}$  with  $r$ , similar to Alg. 1
10:    Insert  $(\mathbf{z}_1^w, \mathbf{z}_1^l, \mathbf{c})$  and  $(\mathbf{z}_2^w, \mathbf{z}_2^l, \mathbf{c})$  to  $\mathcal{B}$ 
11:  until  $N_1$  steps
12:  repeat                                “Online Adaptation”
13:    Sample  $(\mathbf{z}_*^w, \mathbf{z}_*^l, \mathbf{c}) \sim \mathcal{B}$  to compute  $\mathcal{L}_{\text{sur}}^*(\psi; r)$ 
14:    Update  $\psi$  similar to Alg. 1 with learning rate  $\eta_2$ 
15:  until  $N_2$  steps
16: until convergence

```

During reward fine-tuning, we use a normalized and clipped fine-tuning loss $\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_*, \mathbf{c})]$. By keeping track of the moving average and “maximum” value of the surrogate reward, \mathcal{S} essentially improves training stability (details in Appendix B). We use the original step-distillation loss for the distilled DMs (with a separate prompt and image set for distillation) as a regularization loss. For LCM, it is \mathcal{L}_{lcm} from Eq. 3. The total fine-tuning loss is:

$$\mathcal{L}_{\text{lasro}} = c\mathcal{L}_{\text{lcm}} + c_1\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_1, \mathbf{c})] + c_2\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_2, \mathbf{c})] \quad (11)$$

where c, c_1, c_2 are loss coefficients (hyper-parameters).

5.3. Theoretical Insight - Value-Based RL

The superior performance of LaSRO over popular RL methods can be partly explained by its connection to value-based RL. In contrast to pure policy-based RL that usually relies on on-policy samples to estimate policy gradients, value-based RL [47, 78] uses off-policy samples to learn action value functions $Q(s_t, a_t)$ that quantify how “desirable” a particular state and action is via temporal difference (TD) learning [71]. Particularly for short-horizon MDPs with immediate rewards, value-based RL proves advantageous [72] because (1) the learned value function can reduce gradient noise and is more *robust* than pure policy gradient methods and (2) a completely off-policy exploration strategy is much more *efficient* than the alternatives.

Value-based methods are largely *under-explored* for fine-tuning DMs. For instance, existing methods like DDPO [2] utilize the clipped RL objective in PPO yet use no learned value function. LaSRO shares several functional and structural similarities with value-based methods: **(i)** The TD loss for horizon $H = 2$ can be shown largely equivalent to the surrogate reward loss \mathcal{L}_{sur} from Eq. 10, with the surrogate \mathcal{R}_ψ largely equivalent to the value function Q . **(ii)** Optimizing LCMs f_θ is guided by \mathcal{R}_ψ , similar to how policy is guided by value function. **(iii)** Exploration is done in an efficient off-policy manner via also sampling the initial noise \mathbf{x}_{τ_0} for N_s times, rather than restricted to the on-policy distribution $p_\theta(\mathbf{x}_{\tau_H} | \mathbf{x}_{\tau_0}, \mathbf{c})$ with a fixed \mathbf{x}_{τ_0} . This connection gives us insights into the superiority of LaSRO. See more discussions in Appendix C.

6. Experiments

We demonstrate the superior performance of LaSRO for fine-tuning two-step DMs over three distinctive rewards under the non-differentiable (black-box) reward setup [75]. Some rewards (e.g., the text-alignment score that involves sampling) are truly non-differentiable. We further perform ablation studies to justify our design (training scheme, choice of architecture/backbone, etc.).

We choose LCMs [45] distilled from SSD-1B [17] as our major test bed since it is a strong baseline for 2-step image generation of 1024^2 resolution (SDXL-Turbo [61] focuses on 512^2 images and SDXL-Lightning [38] does not support 1-step generation). Our approach can also be applied to other step-distilled DMs, though (see Tab. 1).

Please refer to App. D for implementation details.

6.1. Case Study - Improving General Image Quality

We apply LaSRO to improve the general image quality of two-step DMs by optimizing Image Reward [84]. Image Reward is a popular metric for text-to-image evaluations by using a BLIP-based model (learned from labeled data) to

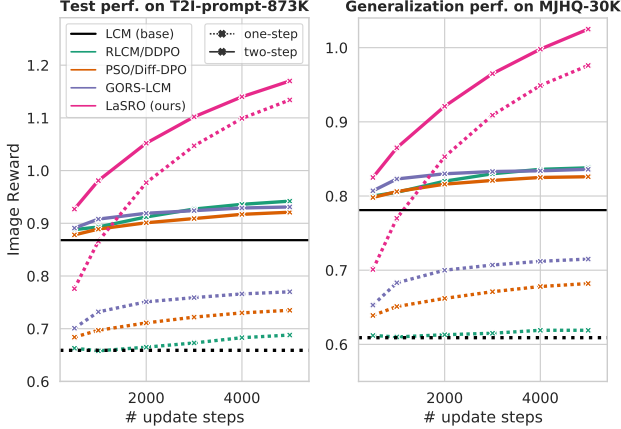


Figure 4. Comparison of LaSRO and other RL methods for fine-tuning two-step LCMs with Image Reward. Results on the test prompts of T2I-prompt-87K (**left**) and the (out-of-distribution) prompts of MJHQ-30K (**right**) show LaSRO effectively improve one-step (dashed lines) and two-step (solid) image generation.

rate images based on criteria such as text alignment and aesthetics. It outperforms other metrics [29, 62, 80] in terms of alignment with human preference. We consider it a black box (while it is originally differentiable) and *do not* use its gradient information for all compared methods.

Training Data We curate a set of 873k unique prompts, named T2I-prompt-873K, with text prompts from a combination of DiffusionDB [77] and open-prompts (link). We split it into 30k for the test split and the rest for training.

Evaluation Protocol Besides the test set performance on T2I-prompt-873K, we also compare the generalization performance (both measured in Image Reward) over the prompts of MJHQ-30K [34], a set curated to balance 10 common text-to-image categories (thus, a different prompt distribution). We tune all compared methods to mitigate reward overoptimization [15] with the help of qualitative evaluation. We report the improved Image Reward during training as curves and summarize each curve by selecting its best checkpoint according to the FID [18] measured with the reference image set of MJHQ-30K, which accounts for the image fidelity (see details in Appendix E).

Baselines and Results We compare LaSRO with strong RL fine-tuning baselines: GORS-LCM (an iterative generate-filter-distill method adapted to LCMs based on GORS [24]), PSO [46] (a method adapted for step-distilled DMs based on DM-DPO [76, 85]), and RLCM [48] (a variant of DDPO [2] for LCMs based on PPO [64]). Additionally, we apply LaSRO to fine-tuning SDXL-Turbo [61]. We report the results in Fig. 4, Tab. 1 and Fig. 1 & 6.

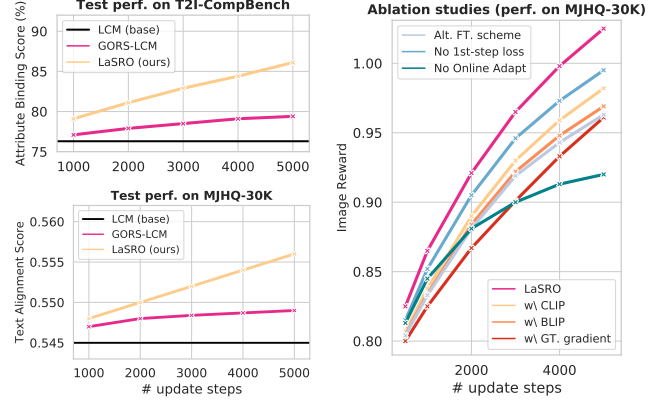


Figure 5. (**left**) LaSRO vs. GORS-LCM (other baselines fail) for fine-tuning LCMs with Attribute Binding Score and Text Alignment Score (all two-step perf. results). (**right**) Results from ablation studies that validate LaSRO’s design (see Sec. 6.3).

	# steps	resolution	Image Reward
SDXL-Turbo [61]	1	512 ²	0.817
SDXL-Turbo	2	512 ²	0.839
+ LaSRO (ours)	1	512 ²	0.904
+ LaSRO (ours)	2	512 ²	0.957
SDXL-Lightning [38]	2	1024 ²	0.634
SSD-1B-LCM [45]	1	1024 ²	0.609
SSD-1B-LCM	2	1024 ²	0.781
+RLCM/DDPO [2, 48]	2	1024 ²	0.830
+PSO/Diff-DPO [46, 76]	2	1024 ²	0.821
+ GORS-LCM [24]	2	1024 ²	0.836
+ LaSRO (ours)	1	1024 ²	0.946
+ LaSRO (ours)	2	1024 ²	0.998

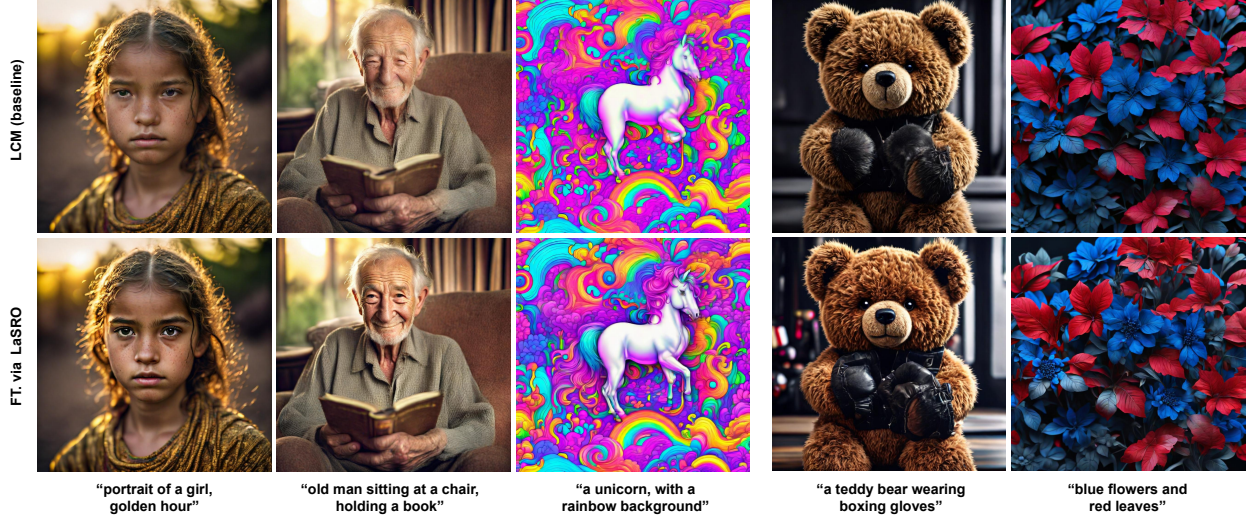
Table 1. Gen. perf. on MJHQ-30K of step-distilled DMs and the fine-tuned DMs with RL methods (see details in Appendix E). We find that LaSRO is agnostic to step-distillation techniques.

6.2. Additional Case Studies

Besides general image quality, we provide two case studies of applying LaSRO to optimize two-step DMs with *non-differentiable* rewards designed for tailored criteria.

Improving Attribute Binding we investigate using LaSRO to enhance attribute binding between the input prompts and the corresponding elements in generated images. Inspired by T2I-CompBench [24], we derive a reward that uses BLIP-VQA [35] to ask the generated image “if the subject exists?” and “if it has the correct attribute?” according to each noun phrase in the prompt (answering “yes” means score-1 and score-0 otherwise). The reward then computes the average score among all questions across all (prompt, image) pairs. Denoted as *Attribute Binding Score*, it is a non-differentiable reward reported as accuracy (%) and is more intuitive than the original one from [24] (which uses averaged raw output score for “yes”). It is also harder for RL fine-tuning as the reward is more “discrete”.

Figure 6. Two-step results (1024² resolution) of LCM baseline and of fine-tuned LCM via LaSRO (with Image Reward for column 1-3 and with Attribute Binding Score for 4-5). LaSRO significantly improves both reward signals. See additional visual results in Appendix F.



Improving Text Alignment Inspired by [2], we use a reward that first uses image captioning models (i.e., Florence-2 [83]) to capture the detailed text description of the generated image and then uses BERTScore [88] to compare the *Text Alignment Score* between this description and the original text prompt. This is a non-differentiable reward due to the image captioning procedure.

Training Data and Evaluation Protocol For the attribute binding task, we use the train split of the 3k attribute-binding prompts from T2I-CompBench for training and use the corresponding test split for evaluation and report the Attribute Binding Score. Regarding text alignment, we set aside 27k prompts (train split) from MJHQ-30K for training and the rest (3k prompts) for evaluation with Text Alignment Score as the reported metric.

Baselines and Results We compare LaSRO to the LCM baseline and GORS-LCM as we find other RL baselines suffer from training instability and are outperformed by GORS-LCM in both tasks. Please see results in Fig. 5 (left).

6.3. Ablation Studies

We conduct extensive ablation studies of LaSRO in improving the general image quality of two-step LCMs (by reporting the generalization perf. on MJHQ-30K). (a) We use CLIP or BLIP (instead of the UNet of latent DMs) as the surrogate reward’s backbone, denoted w_{CLIP} and w_{BLIP} . (b) We examine a variant *Alt.FT.scheme*, a reward fine-tuning scheme used in RG-LCM [36] and widely adopted in supervised reward fine-tuning of ordinary DMs

[7, 84]. Compared to LaSRO, this scheme does not directly target ≤ 2 -step generation. (c) We examine a variant *No 1st-step loss*, which set $c_1 = 0$ in Eq. 11 to disable also optimizing the 1st-step LCM results. (d) In *No Online Adapt*, we remove the surrogate reward online adaption substage in Alg. 2. (e) We include a non-RL baseline that directly uses Image Reward to replace the surrogate reward (gradient info. from Image Reward is used), denoted $w_{\text{GT.gradient}}$. We did not observe the high-frequency noise issue reported in [36] that partly motivates their approach since, contrary to reward-guided distillation [36], we study fine-tuning after distillation.

The numerical results in Fig. 5 (right) validate our design. Additionally, we find LaSRO relatively robust against manual noise perturbation of the GT. reward, partly explaining why it outperforms GT. reward gradients.

7. Conclusion

We address the challenges of fine-tuning 2-step image DMs with arbitrary reward signals. Our analysis identifies limitations in applying existing RL methods. In response, we propose LaSRO, a framework that learns differentiable surrogate rewards in the latent space with off-policy exploration to provide effective gradient guidance during DM fine-tuning. We show LaSRO’s superiority over existing RL methods for optimizing diverse reward signals. We validate LaSRO’s design through extensive ablation studies and further draw a connection between LaSRO and value-based RL to provide theoretical insights. We believe this work opens the door for both large-scale RL fine-tuning and customized model alignment (through flexible reward design) of ultra-few-step image diffusion models, with the potential to be applied in other data modalities.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3
- [2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 1, 2, 3, 4, 6, 7, 8, 5
- [3] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3-4):324–345, 1952. 5
- [4] Changgu Chen, Libing Yang, Xiaoyan Yang, Lianggangxu Chen, Gaoqi He, Changbo Wang, and Yang Li. Find: Fine-tuning initial noise distribution with policy optimization for diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6735–6744, 2024. 3
- [5] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 2
- [6] Joo Young Choi, Jaesung R Park, Inkyu Park, Jaewoong Cho, Albert No, and Ernest K Ryu. Simple drop-in lora conditioning on attention layers will improve your diffusion model. *arXiv preprint arXiv:2405.03958*, 2024. 4
- [7] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023. 3, 8
- [8] Fei Deng, Qifei Wang, Wei Wei, Tingbo Hou, and Matthias Grundmann. Prdp: Proximal reward difference prediction for large-scale reward finetuning of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7423–7433, 2024. 3, 5, 4
- [9] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023. 3
- [10] Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 3
- [11] Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023. 3
- [12] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.
- [13] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [14] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020. 5
- [15] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023. 7
- [16] Sean Gillen and Katie Byl. Leveraging reward gradients for reinforcement learning in differentiable physics simulations. *arXiv preprint arXiv:2203.02857*, 2022. 5
- [17] Yatharth Gupta, Vishnu V. Jaddipal, Harish Prabhala, Sayak Paul, and Patrick Von Platen. Progressive knowledge distillation of stable diffusion xl using layer level loss, 2024. 4, 6, 2, 3
- [18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7, 5
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 3
- [20] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey A. Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *ArXiv*, abs/2210.02303, 2022. 2
- [21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 2
- [22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4
- [23] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. DiffTaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019. 5
- [24] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 1, 3, 7, 4
- [25] Zhiao Huang, Yuanming Hu, Tao Du, Siyuan Zhou, Hao Su, Joshua B Tenenbaum, and Chuang Gan. Plasticinellab: A soft-body manipulation benchmark with differentiable physics. *arXiv preprint arXiv:2104.03311*, 2021. 5
- [26] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 2

- [27] Zhiwei Jia and Hao Su. Information-theoretic local minima characterization and regularization. In *International Conference on Machine Learning*, pages 4773–4783. PMLR, 2020. 5
- [28] Zhiwei Jia, Bodi Yuan, Kangkang Wang, Hong Wu, David Clifford, Zhiqiang Yuan, and Hao Su. Semantically robust unpaired image translation for data with unmatched semantics statistics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14273–14283, 2021. 5
- [29] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 36652–36663, 2023. 1, 3, 5, 7
- [30] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999. 5
- [31] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 2
- [32] Hojoon Lee, Hanseul Cho, Hyunseung Kim, Daehoon Gwak, Joonkee Kim, Jaegul Choo, Se-Young Yun, and Chulhee Yun. Plastic: Improving input and label plasticity for sample efficient reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024. 5
- [33] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023. 3
- [34] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2.5: Three insights towards enhancing aesthetic quality in text-to-image generation, 2024. 7
- [35] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 1, 3, 5, 7, 4
- [36] Jiachen Li, Weixi Feng, Wenhui Chen, and William Yang Wang. Reward guided latent consistency distillation. *arXiv preprint arXiv:2403.11027*, 13, 2024. 3, 5, 8, 4
- [37] Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Ji Li, and Liang Zheng. Step-aware preference optimization: Aligning preference with denoising performance at each step. *arXiv preprint arXiv:2406.04314*, 2 (5):7, 2024. 3
- [38] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 4, 5, 6, 7, 3
- [39] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1
- [40] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audiooldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023. 2
- [41] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 1
- [42] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. InstafLOW: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 3
- [43] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 1
- [44] Yujie Lu, Xianjun Yang, Xiujuan Li, Xin Eric Wang, and William Yang Wang. LlmScore: Unveiling the power of large language models in text-to-image synthesis evaluation. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3
- [45] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 2, 3, 6, 7
- [46] Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning timestep-distilled diffusion model using pairwise sample optimization. *arXiv preprint arXiv:2410.03190*, 2024. 2, 3, 4, 5, 7
- [47] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 6
- [48] Owen Oertel, Jonathan D Chang, Yiyi Zhang, Kianté Brantley, and Wen Sun. RL for consistency models: Faster reward guided text-to-image generation. *arXiv preprint arXiv:2404.03673*, 2024. 2, 3, 7, 4
- [49] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750, 2007. 3, 5
- [50] 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. *arXiv preprint arXiv:2307.01952*, 2023. 1, 3, 5
- [51] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, abs/2209.14988, 2022. 2
- [52] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. 3
- [53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 5, 4
- [54] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3, 5
- [55] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 2
- [56] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1, 2
- [57] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 1, 2
- [58] 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. 2
- [59] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 1, 2
- [60] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 1, 2
- [61] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 1, 3, 6, 7, 5
- [62] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 1, 7
- [63] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 2
- [64] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1, 4, 7, 2
- [65] Joar Skalse, Nikolaus Howe, Dmitrii Krashennnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022. 5
- [66] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1, 2, 3
- [67] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1
- [68] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 1, 2
- [69] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2
- [70] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1, 2, 3
- [71] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988. 6
- [72] Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018. 3, 6
- [73] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999. 4
- [74] Zhiyu Tan, Xiaomeng Yang, Luozheng Qin, Mengping Yang, Cheng Zhang, and Hao Li. Evalalign: Supervised fine-tuning multimodal llms with human-aligned data for evaluating text-to-image models. *CoRR*, 2024. 1, 3
- [75] 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. 6
- [76] 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. 1, 2, 3, 7, 4
- [77] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022. 7, 3
- [78] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992. 2, 3, 6
- [79] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Better aligning text-to-image models with human preference. *ArXiv*, abs/2303.14420, 2023. 3
- [80] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023. 1, 3, 5, 7
- [81] Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep reward supervisions for tuning text-to-image diffusion models. *arXiv preprint arXiv:2405.00760*, 2024. 3
- [82] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 3
- [83] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety

- of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024. [8](#)
- [84] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [8](#)
- [85] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihai Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024. [3](#), [7](#)
- [86] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *ArXiv*, abs/2210.06978, 2022. [2](#)
- [87] Jiacheng Zhang, Jie Wu, Yuxi Ren, Xin Xia, Huafeng Kuang, Pan Xie, Jiashi Li, Xuefeng Xiao, Min Zheng, Lean Fu, et al. Unifl: Improve stable diffusion via unified feedback learning. *arXiv preprint arXiv:2404.05595*, 2024. [3](#), [5](#)
- [88] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019. [8](#)

Appendix

A. Empirical Local Lipchitz of Two-step LCMs

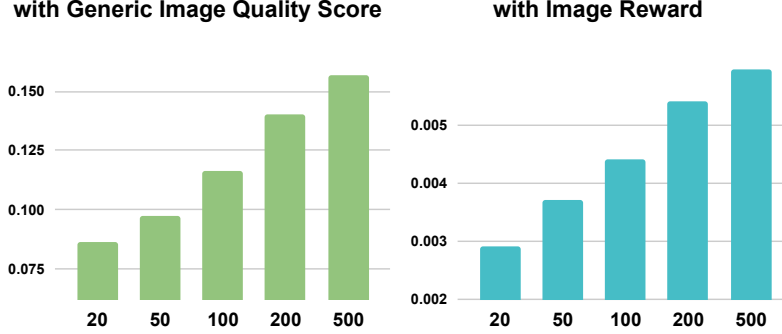


Figure 7. We show the growing empirical local Lipchitz of the mapping that underlies two-step LCM image generation. The x-axis is the input data’s noise level (i.e., timestep). The y-axis is the estimated local Lipchitz. The left and right figures incorporate the reconstruction-based image quality score and the Image Reward, respectively, in estimating the local Lipchitz.

In Sec. 4.3, we argue that the non-smoothness of the mapping that underlies two-step DMs makes it challenging to fine-tune them. Specifically, the mapping for which we measure the non-smoothness refers to the 2nd step of a distilled two-step diffusion sampler (e.g., an LCM f_θ) that maps a noisy image to a clean image. We present empirical evidence by estimating two different local Lipchitz constants characterizing this mapping.

Formally, given a prompt \mathbf{c} and initial noise \mathbf{z}_{τ_0} , the 1st LCM step (with re-injected noise) gives you $\mathbf{z}_1^t = f_\theta(\mathbf{z}_{\tau_0}, \tau_0, \mathbf{c}) + Z_t$, with notations of the denoised latent code \mathbf{z}_1 and the injected noise Z_t at noise level (i.e., timestep) t from Sec. 5.1. Note that, in a standard two-step LCM inference setup ($H = 2$), the noise Z is injected at noise level $t = \tau_{H/2} = \tau_1$ (the middle of the denoising trajectory), and so $\mathbf{z}_1^t = \mathbf{z}_1^{\tau_{H/2}} = \mathbf{z}_1^{\tau_1}$. Here, we use the notation that the injected noise level t might not be $\tau_{H/2}$. We then randomly sample another valid action $\mathbf{z}_1^t(\epsilon)$ ($\epsilon > 0$) in the neighborhood of \mathbf{z}_1^t , at the same noise level t , by $\mathbf{z}_1^t(\epsilon) = \sqrt{1 - \epsilon^2} \mathbf{z}_1^t + \epsilon \mathbf{z}'$, where $\mathbf{z}' \sim \mathcal{N}(0, \mathcal{I})$. We set $\epsilon = 0.01$ so that $\mathbf{z}_1^t(\epsilon)$ is close enough to \mathbf{z}_1^t for the Lipchitz estimation. Next, we obtain the 2nd LCM results $\mathbf{z}_2^t = f_\theta(\mathbf{z}_1^t, t, \mathbf{c})$ and $\mathbf{z}_2^t(\epsilon) = f_\theta(\mathbf{z}_1^t(\epsilon), t, \mathbf{c})$. To efficiently compute the local Lipchitz, we need a real-value function that further maps \mathbf{z}_2^t and $\mathbf{z}_2^t(\epsilon)$ to scalars. We use two functions, both of which help us to illustrate the non-smoothness property well.

We first propose a prompt-free generic image quality metric: the reconstruction error $\ell_{\text{rec}}(\mathbf{x}) = \|\mathcal{D}(\mathcal{E}(\mathbf{x})) - \mathbf{x}\|$ given a pre-trained VAE’s encoder \mathcal{E} and decoder \mathcal{D} (we use VAE from SDXL [50]), which estimates how well a generated image \mathbf{x} fit in the image manifold learned by the pre-trained VAE. We use the Image Reward [84], denoted $r(\mathbf{x}, \mathbf{c})$ as the second function. Namely, for ℓ_{rec} we have the following two terms:

$$\begin{aligned}\ell_{\text{rec}}(\mathbf{z}_2^t) &= \|\mathcal{D}(\mathcal{E}(\mathcal{D}(\mathbf{z}_2^t))) - \mathcal{D}(\mathbf{z}_2^t)\| \\ \ell_{\text{rec}}(\mathbf{z}_2^t(\epsilon)) &= \|\mathcal{D}(\mathcal{E}(\mathcal{D}(\mathbf{z}_2^t(\epsilon)))) - \mathcal{D}(\mathbf{z}_2^t(\epsilon))\|\end{aligned}$$

and for r we have:

$$\begin{aligned}r(\mathbf{z}_2^t, \mathbf{c}) &= r(\mathcal{D}(\mathbf{z}_2^t), \mathbf{c}) \\ r(\mathbf{z}_2^t(\epsilon), \mathbf{c}) &= r(\mathcal{D}(\mathbf{z}_2^t(\epsilon)), \mathbf{c})\end{aligned}$$

Combined, we can then compute the two empirical local Lipschitz constants given the input data’s noise level t as:

$$\begin{aligned}L_{\text{local}}(\ell_{\text{rec}}, t) &= \mathbb{E}_{(\mathbf{z}_{\tau_0}, \mathbf{c})} \left[\frac{|\ell_{\text{rec}}(\mathbf{z}_2^t) - \ell_{\text{rec}}(\mathbf{z}_2^t(\epsilon))|}{\|\mathbf{z}_1^t - \mathbf{z}_1^t(\epsilon)\|} \right] \\ L_{\text{local}}(r, t) &= \mathbb{E}_{(\mathbf{z}_{\tau_0}, \mathbf{c})} \left[\frac{|r(\mathbf{z}_2^t, \mathbf{c}) - r(\mathbf{z}_2^t(\epsilon), \mathbf{c})|}{\|\mathbf{z}_1^t - \mathbf{z}_1^t(\epsilon)\|} \right]\end{aligned}$$

Both empirical Lipchitz are estimated over $N = 1000$ sampled $(\mathbf{z}_{\tau_0}, \mathbf{c})$ pairs. To visualize the growing non-smoothness of the mapping when the input data has a larger noise level (i.e., the LCM covers a longer denoising trajectory in one single sampling step), we compute these two quantities with varying input noise level $t = 20, 50, 100, 200, 500$.

We illustrate the numerical values of these two local Lipchitz for a pre-trained LCM based on SSD-1B [17] in Fig. 7.

B. Normalization and Clipping Function

Similar to the advantage estimation [63] and the clipped RL objective proposed in PPO [64], we utilize a normalization and clipping function \mathcal{S} to improve the training stability of the reward fine-tuning stage of LaSRO. The terms $\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_*, \mathbf{c})]$ from Eq. 11 are defined as:

$$\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_1, \mathbf{c})] = \text{clip}\left(\frac{\mathcal{R}_\psi(\mathbf{z}_1, \mathbf{c}) - \bar{\mathcal{R}}_\psi(\mathbf{z}_1, \mathbf{c})}{\mathcal{R}_\psi^{90\%}(\mathbf{z}_1, \mathbf{c})}, -\inf, 1\right) \quad (12)$$

$$\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_2, \mathbf{c})] = \text{clip}\left(\frac{\mathcal{R}_\psi(\mathbf{z}_2, \mathbf{c}) - \bar{\mathcal{R}}_\psi(\mathbf{z}_2, \mathbf{c})}{\mathcal{R}_\psi^{90\%}(\mathbf{z}_2, \mathbf{c})}, -\inf, 1\right) \quad (13)$$

Namely, we keep track of the moving average of $\mathcal{R}_\psi(\mathbf{z}_*, \mathbf{c})$ as $\bar{\mathcal{R}}_\psi(\mathbf{z}_*, \mathbf{c})$, and we further keep track of the moving 90-percentile of $\mathcal{R}_\psi(\mathbf{z}_*, \mathbf{c}) - \bar{\mathcal{R}}_\psi(\mathbf{z}_*, \mathbf{c})$ as $\mathcal{R}_\psi^{90\%}(\mathbf{z}_*, \mathbf{c}) > 0$. After normalizing $\mathcal{R}_\psi(\mathbf{z}_*, \mathbf{c})$, we clip its maximum value to be 1 but not its minimum value. We find \mathcal{S} easy to implement, effectively stabilizing reward fine-tuning, and easing hyper-parameter tuning regarding the learning rates and the loss coefficients in Eq. 11.

C. Connection between LaSRO and Value-based RL

In Sec. 5.3, we argue that LaSRO shares several functional and structural similarities with value-based RL methods. Here we provide more details of this comparison.

Details of comparison (i) The TD learning loss \mathcal{L}_{TD} for value-based RL with an MDP of horizon $H = 2$ is largely equivalent to the surrogate reward loss \mathcal{L}_{sur} from Eq. 10. Specifically, when adapted to reward fine-tuning two-step DMs with $\mathbf{a}_t = \mathbf{x}_{\tau_{t+1}} = \mathbf{z}_{t+1} + Z$ (the MDP notation from Sec. 3.3 and the notation of \mathbf{z}_t from Sec. 5.1), we have

$$\mathcal{L}_{\text{TD}}(\psi) = \mathbb{E}_{(\mathbf{x}_{\tau_t}, \mathbf{a}_t, \mathbf{x}_{\tau_{t+1}}, \mathbf{c})} \left[\left(Q_\psi(\mathbf{x}_{\tau_t}, \mathbf{a}_t, \mathbf{c}) - \left(r + \gamma \max_{\mathbf{a}_{t+1}} Q_\psi(\mathbf{x}_{\tau_{t+1}}, \mathbf{a}_{t+1}, \mathbf{c}) \right) \right)^2 \right] \quad (14)$$

where r is an image-space reward signal. Since $Q_\psi(\mathbf{x}_{\tau_t}, \mathbf{a}_t, \mathbf{c}) = Q_\psi(\mathbf{z}_t, \mathbf{z}_{t+1} + Z, \mathbf{c})$ evaluates the “quality” of \mathbf{z}_{t+1} , it is functionally equivalent to $\mathcal{R}_\psi(\mathbf{z}_{t+1}, \mathbf{c})$. Consequently, we have

$$\mathcal{L}_{\text{TD}}(\psi) = \mathbb{E} \left[\left(\mathcal{R}_\psi(\mathbf{z}_{t+1}, \mathbf{c}) - \left(r(\mathcal{D}(\mathbf{z}_{t+1}), \mathbf{c}) + \gamma \max_{\mathbf{z}_{t+2}} \mathcal{R}_\psi(\mathbf{z}_{t+2}, \mathbf{c}) \right) \right)^2 \right] \quad (15)$$

Given that $H = 2$, and set the discount factor $\gamma = 0$ (we equally care about the individual reward of one-step and two-step LCM results), we further have

$$\mathcal{L}_{\text{TD}}(\psi) = \mathbb{E}_{(\mathbf{z}_1, \mathbf{c})} \left[\left(\mathcal{R}_\psi(\mathbf{z}_1, \mathbf{c}) - r(\mathcal{D}(\mathbf{z}_1), \mathbf{c}) \right)^2 \right] + \mathbb{E}_{(\mathbf{z}_2, \mathbf{c})} \left[\left(\mathcal{R}_\psi(\mathbf{z}_2, \mathbf{c}) - r(\mathcal{D}(\mathbf{z}_2), \mathbf{c}) \right)^2 \right] \quad (16)$$

which is essentially the surrogate reward loss \mathcal{L}_{sur} in an L2 regression form (instead of the binary classification form).

Details of comparison (iii) Similar to value-based methods which usually feature more efficient off-policy sampling, LaSRO explores the image (more precisely the latent) space in an off-policy manner. On-policy sampling refers to sampling data strictly according to the current RL policy, i.e. $p_\theta(\mathbf{x}_{\tau_H} | \mathbf{x}_{\tau_0}, \mathbf{c})$, for reward fine-tuning two-step DMs with $H = 2$. Exploration is limited in this case, since given the *fixed* initial noise and prompt $(\mathbf{x}_{\tau_0}, \mathbf{c})$, the stochasticity of the action distribution only comes from one step of noise injection. This limitation hinders most policy gradient methods, which require

on-policy samples due to the application of the policy gradient theorem. In contrast, value-based RL usually leverages off-policy samples from arbitrary probability distributions to learn value functions. In both training stages of LaSRO, we explore the latent space by sampling both the initial noise \mathbf{x}_{τ_0} and the injected noise Z , rather than restricted to a fixed initial noise (i.e., on-policy samples). This off-policy sampling helps LaSRO to achieve more effective exploration.

D. Implementation Details

T2I-prompt-873K We curate a set of 873k unique prompts from the text prompts of a combination of DiffusionDB [77] and open-prompts (link). Specifically, we first combine the two prompt sets to obtain a set of 2.9M prompts. After post-processing to de-duplication, we obtain 1.5M unique prompts. Then we adopt the active prompt selection from UniFL [87] to create a more diverse and balanced prompt subset where (almost) all prompt has its closest prompt at least greater than some thresh T for some distance measurement. We measure the distance between prompts as the normalized L2 distance of text features from this text encoder and use FAISS for nearest neighbor search. We finally ended up with 873k prompts.

Pre-trained LCM based on SSD-1B To perform latent consistency distillation using SSD-1B [17] as the teacher, we generate 700k (prompt, image) pairs as training data from 700k prompts sampled from T2I-prompt-873k. Each image is of resolution 1024^2 and generated by SDXL for 40 steps and SDXL-refiner (added noise of strength 0.2) for another 8 steps with a guidance scale of 7.0. We train this LCM with most hyper-parameters similar to the original LCM paper, with the exception that we use 100 sampling steps instead of 50 (i.e., a skipping step of 10 instead of 20). We use a fixed learning rate of $3e-6$ with a batch size of 128 for 150k steps, resulting in better ≤ 2 step image generation than in the original paper [45]. The inference time for this 2-step LCM for a 1024^2 image is around 0.15 seconds on an A100.

Architecture of Surrogate Reward Regarding the architecture of the surrogate reward, we use the UNet’s down- and mid-block of the pre-trained Segmind-Vega (a diffusion model distilled from SDXL) as the backbone and use stacks of (4 by 4 convolution with stride 2, group normalization [82] with 32 groups, SiLU activation [10]), similar to SDXL-Lightning [38], and a linear layer on top of them to produce the final scalar output of $\mathcal{R}_\psi(\mathbf{z}, \mathbf{c})$. During both the pre-training stage and the fine-tuning stage, we update all parameters involved in the surrogate model (no frozen weights).

Pre-training Stage of LaSRO The hyper-parameter N_s reflects how hard the exploration problem is for obtaining W/L samples of differed reward values (e.g., the Attribute Binding Score (%) is rather discrete). We set $N_s = 1$ when optimizing Image Reward and $N_s = 6$ otherwise. For both one-step and two-step results $\{\mathbf{z}_1\}$ and $\{\mathbf{z}_2\}$, we select the winning and losing (W/L) pair by ranking all z_* within each of the two groups and selecting the best and the worse according to the target reward signal r . This gives us $(\mathbf{z}_1^w, \mathbf{z}_1^l, \mathbf{c})$ and $(\mathbf{z}_2^w, \mathbf{z}_2^l, \mathbf{c})$ that have the maximum reward discrepancy between \mathbf{z}_*^w and \mathbf{z}_*^l , facilitating surrogate reward learning. The sampling strategy for prompts and injected noise is random sampling. We found it not a critical design choice as long as we vary the initial noise (off-policy exploration). In the pre-training stage for optimizing the Image Reward, we train the surrogate reward for 30k steps. In the pre-training stage for the Attribute Binding Score and the Text Alignment Score, we further fine-tune this trained version for an additional 2k and 10k steps, respectively. By leveraging the surrogate reward learned from Image Reward, we accelerate the pre-training stage for other reward signals as Image Reward is a rather general image metric. We use a fixed learning rate of $\eta = 1e-5$ for all three metrics. The training time of the pre-training stage of the surrogate reward for Image Reward is around 12 hours on a single node of 8 H100 GPUs.

Fine-tuning Stage of LaSRO In the fine-tuning stage, we use the same setup of N_s and W/L pair-finding mechanism in the pre-training stage. During the online adaptation of the surrogate reward, specifically in line 6 from Alg. 2, we randomly select one \mathbf{z}_1 out of N_s sampled $\{\mathbf{z}_1\}$ and similarly for \mathbf{z}_2 to compute $\mathcal{S}[\mathcal{R}_\psi(\mathbf{z}_*, \mathbf{c})]$, as we find it more efficient. We use the original loss \mathcal{L}_{lcm} used to train the LCM and the original training data (700k text-image pairs as introduced previously) as a regularizer during the fine-tuning stage. When optimizing the Image Reward, we alternate between updating the diffusion model for $N_1 = 1$ step and adapting the surrogate reward for $N_2 = 1$ step. When optimizing the Attribute Binding score, we set

$N_1 = 10$ and $N_2 = 20$. For the Text Alignment Score, it is $N_1 = 10$ and $N_2 = 50$. The normalization and clipping function \mathcal{S} applied to the reward optimization loss makes hyper-parameter tuning more robust. We set $c = 500$, $c_1 = 0.5$, $c_2 = 1.0$ and the learning rate $\eta_1 = 3e - 6$, $\eta_2 = 3e - 8$ when optimizing all three rewards. For RL methods compared including LaSRO, we report the results using checkpoints obtained with EMA update of a rate $\mu = 0.95$ to ensure more stable performance across iterations. The training time of the fine-tuning stage of LaSRO for Image Reward is around 48 hours on a single node of 8 H100 GPUs.

GORS-LCM GORS-LCM is an RL method based on GORS [24], which follows a simple generate-filter-distill paradigm and is adapted to fine-tuning LCMs. It shares a similar form as the original LCM learning process. In each iteration, it samples $N_s = 6$ times from the two-step LCM to obtain $\{z_1\}$ and $\{z_2\}$ given prompt c , selects the best z_1^{best} and z_2^{best} from them according to the target reward r , and then trains with the LCM loss \mathcal{L}_{lcm} to distill from SSD-1B using data (z_*^{best}, c) generated online instead of the offline distillation dataset (e.g., the 700k pairs described previously). While converging slowly due to no reward gradient guidance, this method is stable and can serve as an RL baseline for arbitrary rewards. We further apply regularization via \mathcal{L}_{lcm} with the 700k (text, image) pairs, similar to LaSRO.

RLCM/DDPO RLCM [48] is a policy-based RL method that adapts DDPO [2] to LCMs, essentially applying PPO [64] to perform reward fine-tuning. We adopt most of the hyper-parameters of RLCM (for its case of optimizing the aesthetics score), except that we use a batch size of 128 and a reduced learning rate of $1e - 5$ to ensure training stability. Moreover, we find that using LoRA updates [22] instead of full LCM weight updates helps to improve convergence as LoRA can be seen as a form of regularization through model parameter preservation [6]. To further mitigate the training instability caused by RLCM/DDPO, we carefully turn the hyper-parameters to apply the regularization loss \mathcal{L}_{lcm} with the 700k (text, image) data pairs in (similar to LaSRO).

PSO/Diff-DPO PSO [46], particularly its online version, is an online variant of Diffusion-DPO [76] adapted to step-distilled diffusion models, such as LCMs. We largely follow the hyper-parameters of Diffusion-DPO and the online PSO when reward fine-tuning the two-step LCMs (with a batch size of 128). Similar to our experiments of RLCM/DDPO, we find that updating LCM weights using LoRA serves as a regularization technique that leads to improvements in both the numerical and visual results (in contrast, LaSRO works well with or without LoRA). Similar to other RL baselines, we apply regularization via \mathcal{L}_{lcm} for a fair comparison to LaSRO. Note that, PRDP [8] can be seen as another online variant of Diffusion-DPO that applies reward-weighted regression in a reward-adapted manner. The difficulties encountered in reward fine-tuning ≤ 2 -step diffusion models using PSO/Diffusion-DPO are also relevant to PRDP.

Ablation Study (a) We replace our latent-space surrogate reward design based on pre-trained diffusion models with CLIP [53] or BLIP [35] to validate our choices. Specifically, we use the CLIP-ViT-L/14 from the original CLIP paper and a similar BLIP model fine-tuned for VQA tasks (ViT-L for the vision backbone). We train both in the latent space of SDXL as a fair comparison with our design in LaSRO. Both have a similar number of model parameters as the pre-trained UNet that we use. Both are trained in the pre-training and the reward fine-tuning (online adaptation) stages.

Ablation Study (b) In this ablation study, we examine the alternative reward fine-tuning scheme adopted by most existing fine-tuning methods with direct reward back-propagation, including RG-LCD [36]. This alternative does not target 2-step image generation. Instead, in each reward fine-tuning step, it randomly chooses a timestep t , which refers to the notation used in the consistency distillation loss \mathcal{L}_{lcm} from Eq. 3. Specifically, $t \in \{1, \dots, 50\}$ in the original LCM paper with a skipping step of 20 and $t \in \{1, \dots, 100\}$ in our LCM training setup with a skipping step of 10. For this timestep t , which could be anywhere in a denoising trajectory, the alternative scheme optimizes $\mathcal{R}_\psi(f_\theta(x_t, t, c), c)$ with x_t generated by adding Gaussian noise Z_t of level t to the samples from the 700k (text, image) training pairs. In comparison, LaSRO optimizes $\mathcal{R}_\psi(z_*, c)$, where z is generated completely online, either from the 1st-step LCM results given random initial noise, or from the 2nd-step LCM results given the 1st-step results. This leads to improved effectiveness in fine-tuning ≤ 2 -step image generation.

E. Additional Experiments Results and Implementation Details

Reward over-optimization Reward over-optimization [65] is a common issue for reward fine-tuning diffusion models, referring to optimizing overly on an imperfect reward so that the model performance measured by some unknown “true” objective gets dropped. Visually this usually means blurry images or abnormal image compositions and visual patterns where the fine-tuned model no longer preserves the high fidelity of its outputs (e.g., the Fig. 5 in the DDPO paper [2]). While no practical techniques are 100% exempt from reward over-optimization, we adopt several to mitigate this issue. Namely, for all compared methods, we use the latent consistency distillation loss \mathcal{L}_{lcm} as a regularizer on a large distillation dataset (T2I-prompt-873k and the corresponding images), which essentially penalizes large KL divergence between the fine-tuned model and the reference one (e.g., SSD-1B). When optimizing the Image Reward [84], we use a large number of diverse prompts from the same T2I-prompt-873k set, which is generated via the active prompt selection that helps mitigate reward over-optimization [87]. We carefully tune the learning rate (in most cases to be lowered) of all methods with qualitative evaluation to help monitor the image fidelity of the fine-tuned model’s outputs. While one can resort to FID [18] to quantitatively evaluate the image fidelity (whose results will be reported shortly), it requires a reference image set and the scores do not always align with human judgment.

Tradeoff - FID vs. Image Reward The inherent trade-off between preserving high fidelity and achieving target optimization is manifested with the tradeoff between FID vs. Image Reward. We use FID [18], a prompt-free metric, to measure the image fidelity of the results from fine-tuned diffusion models. We measure it between the generated images from prompts in MJHQ-30K and the human-curated reference image set provided by MJHQ-30K. Upon aggressively optimizing the Image Reward score using LaSRO, we report curves of FID vs. Image Reward in Fig. 8 of two-step LCMs for all compared methods in Fig. 4. Note that, in our experiments, we evaluate LaSRO on a well-trained distilled DM baseline to see if it can still provide meaningful improvement without losing visual qualities (reward hacking). The results further demonstrate the superior performance of LaSRO.

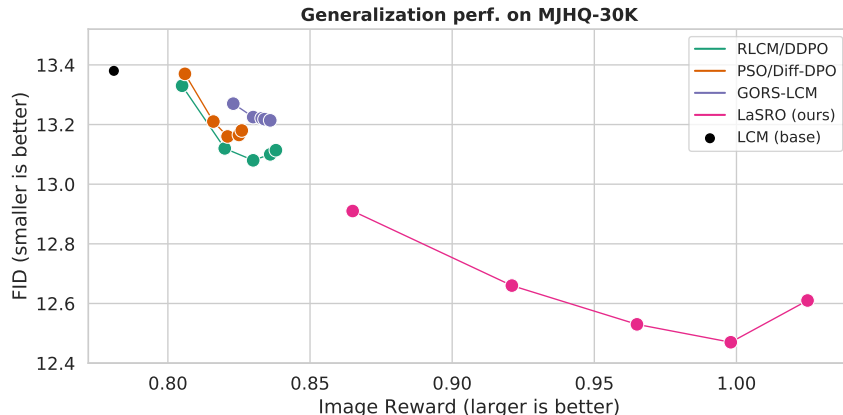


Figure 8. The curves that present the tradeoff between FID and Image Reward for the LCM baseline and the fine-tuned versions with all compared methods from Fig. 4. Each point is generated by one of the checkpoints with 1000, 2000, 3000, 4000, or 5000 reward fine-tuning update steps and measured on two-step generated images of 1024^2 pixels. The closer a point is to the lower right corner, the better the overall model performance is. LaSRO consistently outperforms other RL baselines.

Implementation Details for Tab. 1 Given FID as an auxiliary metric, we can summarize the Image Reward performance of each method (i.e., the curves shown in Fig. 4) as a single numerical number by reporting the result of the checkpoint with the lowest FID (the elbow rule), which is supposedly better than that of the LCM baseline. We report the numerical results in Tab. 1, where we also show the Image Reward of other step-distilled diffusion model baselines including SDXL-Lightning [38] (2-step) and SDXL-Turbo [61] (1- and 2-step), both based on adversarial distillation. Note that, SDXL-Turbo mainly focuses on image generation of 512^2 pixels and SDXL-Lightning only supports ≥ 2 step generation. In addition, we examine applying LaSRO toward improving SDXL-Turbo with Image Reward, whose result showcase that LaSRO is not specific to

LCMs. Namely, we do not use \mathcal{L}_{icm} as regularization (the pre-trained discriminator of SDXL-Turbo is not publicly released, and thus we can not use its original adversarial distillation loss for regularization) and reduce the learning rate $\eta_1 = 1e - 8$ in the fine-tuning stage to mitigate reward over-optimization.

F. Additional Visual Comparison (see next page)

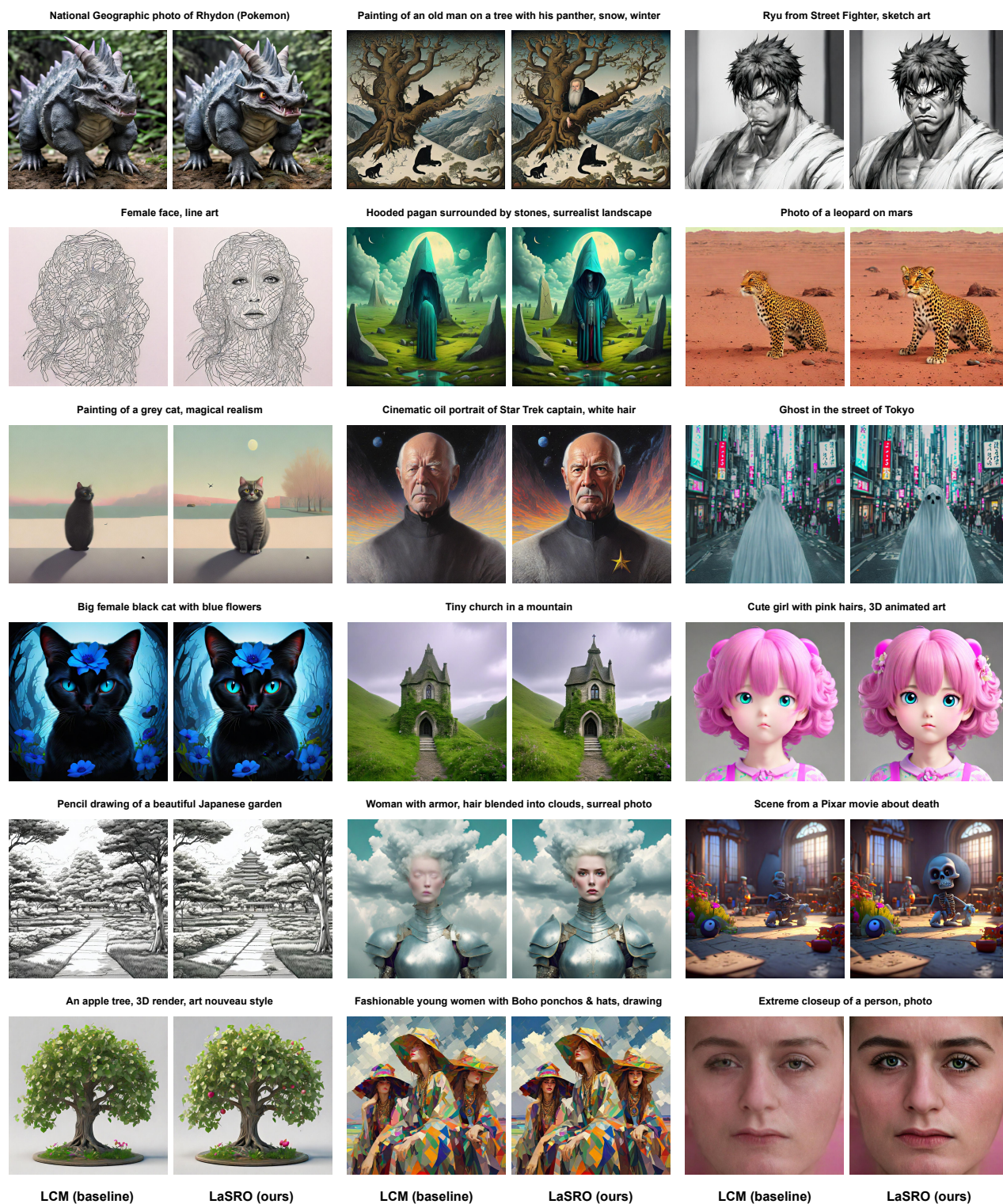


Figure 9. Additional visual comparison of two-step LCM results from the LCM baseline and the fine-tuned LCM via LaSRO with Image Reward. All images are of 1024^2 resolution with prompts randomly sampled from T2I-prompt-873K