

---

# David and Goliath: Small One-step Model Beats Large Diffusion with Score Post-training

---

Weijian Luo<sup>\*1</sup> Colin Zhang<sup>1</sup> Debing Zhang<sup>1</sup> Zhengyang Geng<sup>2</sup>

## Abstract

We propose *Diff-Instruct\*(DI\*)*, a data-efficient post-training approach for one-step text-to-image generative models to improve its human preferences without requiring image data. Our method frames alignment as online reinforcement learning from human feedback (RLHF), which optimizes the one-step model to maximize human reward functions while being regularized to be kept close to a reference diffusion process. Unlike traditional RLHF approaches, which rely on the Kullback-Leibler divergence as the regularization, we introduce a novel general score-based divergence regularization that substantially improves performance as well as post-training stability. Although the general score-based RLHF objective is intractable to optimize, we derive a strictly equivalent tractable loss function in theory that can efficiently compute its *gradient* for optimizations. We introduce *DI\*-SDXL-1step*, which is a 2.6B one-step text-to-image model at a resolution of  $1024 \times 1024$ , post-trained from DMD2 w.r.t SDXL. **Our 2.6B *DI\*-SDXL-1step* model outperforms the 50-step 12B FLUX-dev model** in ImageReward, PickScore, and CLIP score on the Parti prompts benchmark while using only 1.88% of the inference time. This result clearly shows that with proper post-training, the small one-step model is capable of beating huge multi-step diffusion models. Our model is open-sourced at this link: [https://github.com/pkulwj1994/diff\\_instruct\\_star](https://github.com/pkulwj1994/diff_instruct_star). We hope our findings can contribute to human-centric machine learning techniques.

---

<sup>1</sup>Hi Lab of Xiaohongshu Inc, Beijing, China. <sup>2</sup>School of Computer Science, Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Weijian Luo <luoweijian@xiaohongshu.com>.

## 1. Introductions

Generative models have made substantial progress in recent years, largely improving the creative content creation across various domains (Karras et al., 2020; Nichol & Dhariwal, 2021; Poole et al., 2022; Kim et al., 2022; Tashiro et al., 2021; Meng et al., 2021; Couairon et al., 2022; Ramesh et al., 2022; Esser et al., 2024; Oord et al., 2016; Ho et al., 2022; Brooks et al., 2024; Zhang et al., 2023a; Xue et al., 2023; Luo & Zhang, 2024; Luo et al., 2023b; Pokle et al., 2022; Zhang et al., 2023b; Feng et al., 2023; Deng et al., 2024; Luo et al., 2024d; Geng et al., 2024; Wang et al., 2024).

Among these advancements, two types of generative models have gained significant attention, diffusion models (DMs) and one-step generators. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020), or score-based generative models (Song et al., 2020), progressively corrupt data with diffusion processes and then train models to approximate the score functions of the noisy data distributions across varying noise levels. The learned score functions can generate high-quality samples by denoising the noisy samples through reversed stochastic differential equations. While DMs can produce good outputs, they often require a large number of model evaluations, which limits their efficiency in applications.

In contrast, one-step generators (Luo et al., 2024c;a; Zheng & Yang, 2024; Kang et al., 2023; Sauer et al., 2023a; Yin et al., 2024; Zhou et al., 2024a) have emerged as a highly efficient alternative to multi-step diffusion models. Unlike DMs, one-step generative models map latent noises directly to data samples in a single forward pass, making them highly efficient for real-time applications such as text-to-image and text-to-video generations on edge devices. Many existing works have demonstrated the leading performances of one-step text-to-image generators (Luo et al., 2024c; Zhou et al., 2024a; Yin et al., 2024) by employing diffusion distillation (Luo, 2023) as well as GAN techniques (Goodfellow et al., 2014; Brock et al., 2018; Karras et al., 2020; Sauer et al., 2023b).

- *Yet, these works only focus on matching the one-step model's distributions with pre-trained diffusion models or ground truth data, overlooking the critical*

*challenge of aligning one-step text-to-image models with human preferences, which is one of the central requirements of human-centric AI.*

To close this gap, we introduce *Diff-Instruct\** (*DI\**), a novel post-training approach that aligns one-step text-to-image generators with human preference while maintaining the ability to generate diverse and photo-realistic high-resolution images. We frame the human-preference alignment problem as reward maximization with score-based divergence constraints. This yields generated samples that not only have improved human reward but also adhere to user prompts. Unlike traditional RLHF methods (Christiano et al., 2017; Ouyang et al., 2022; Luo, 2024), which rely on Kullback-Leibler (KL) divergence for distribution regularization, we demonstrate that score-based regularization is important to preserve sample diversity and avoids reward hacking – the model generates weird, painting-like outputs with high rewards but low realism.

Our experiments demonstrate three key advantages: 1) **Efficiency**: The 2.6B *DI\*-SDXL-1step* model generates  $1024 \times 1024$  resolution images in a single step, using only 1.88% of the inference time and 29.3% of the GPU memory of the 50-step 12B FLUX-dev model. 2) **Superior Performance**: On Parti prompts, *DI\*-SDXL-1step* achieves higher PickScore, ImageReward, and CLIPScore than FLUX-dev-50step. It also sets a new state-of-the-art HPSv2.1 score of 31.19 among open-source models. 3) **Fidelity**: The model maintains diversity and prompt adherence, matching FLUX-dev and SD3.5-large on COCO benchmarks while being orders of magnitude faster.

We summarize our contributions as follows:

- We introduce *Diff-Instruct\** for preference alignment of one-step text-to-image generative models with a strict theoretical guarantee based on a new score-based PPO paradigm;
- We introduce two novel approaches to incorporate classifier-free guidance into human preference alignment through explicit-implicit reward decoupling.
- With extensive evaluations, our best open-sourced 2.6B *DI\*-SDXL-1step* text-to-image model, a leading text-to-image model with a resolution of  $1024 \times 1024$  that significantly outperforms the leading FLUX-dev-50step model with 1.88% inference time.

## 2. Preliminary

**Diffusion Models.** Here, we introduce preliminary knowledge and notations about diffusion models. Let  $q_0(\mathbf{x}) = q_d(\mathbf{x})$  be the data distribution. The goal of generative modeling is to train models to generate new samples  $\mathbf{x} \sim q_0(\mathbf{x})$ . Under mild conditions, the forward diffusion

process of a diffusion model can transform initial distribution  $q_0$  towards some simple noise distribution,

$$d\mathbf{x}_t = \mathbf{F}(\mathbf{x}_t, t)dt + G(t)d\mathbf{w}_t, \quad (2.1)$$

where  $\mathbf{F}$  is a pre-defined vector-valued drift function,  $G(t)$  is a pre-defined scalar-valued diffusion coefficient, and  $\mathbf{w}_t$  denotes an independent Wiener process. A continuous-indexed score network  $\mathbf{s}_\varphi(\mathbf{x}, t)$  is employed to approximate marginal score functions of the forward diffusion process (2.1). The learning of score networks is achieved by minimizing a weighted denoising score matching objective (Vincent, 2011; Song et al., 2020),

$$\begin{aligned} \mathcal{L}_{DSM}(\varphi) = & \int_{t=0}^T \lambda(t) \mathbb{E}_{\substack{\mathbf{x}_0 \sim q_0, \\ \mathbf{x}_t | \mathbf{x}_0 \sim p_t(\mathbf{x}_t | \mathbf{x}_0)}} \|\mathbf{s}_\varphi(\mathbf{x}_t, t) \\ & - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0)\|_2^2 dt. \end{aligned}$$

The weighting function  $\lambda(t)$  controls the importance of the learning at different time levels, and  $p_t(\mathbf{x}_t | \mathbf{x}_0)$  denotes the conditional transition of the forward diffusion (2.1). After training,  $\mathbf{s}_\varphi(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$  is a good approximation of the marginal scores of the diffused data distribution.

## 3. Score-based Post-training

**Problem Setup.** Our basic setting is that we have a known human reward function  $r(\mathbf{x}_0, \mathbf{c})$ , which encodes the human preference for an image  $\mathbf{x}_0$  and corresponding text description  $\mathbf{c}$ . Besides, we also have a pre-trained diffusion model which will later act as a reference distribution  $p_{ref}(\mathbf{x}_0) = q_0(\mathbf{x}_0)$ . The reference diffusion model is specified by the score function  $\mathbf{s}_{q_t}(\mathbf{x}_t) := \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$ , where  $q_t(\mathbf{x}_t)$ 's is the underlying distribution diffused at time  $t$  according to (2.1). The pre-trained diffusion model represents the ground-truth data distribution that is used to prevent the one-step model from seeking high human reward but losing the ability to generate realistic images.

Our goal is to train a human-preferred one-step text-to-image model  $g_\theta$  that generates images by directly mapping a random noise  $\mathbf{z} \sim p_z$  to obtain  $\mathbf{x}_0 = g_\theta(\mathbf{z}, \mathbf{c})$ , conditioned on the input text  $\mathbf{c} \sim \mathcal{C}$ . We want the one-step model's output distribution,  $p_\theta(\mathbf{x}_0 | \mathbf{c})$ , to maximize the expected human rewards while maintaining the ability to generate realistic images. We force  $p_\theta(\mathbf{x}_0 | \mathbf{c})$  to not diverge from  $p_{ref}(\cdot)$  by using a divergence regularization term. Let  $D(\cdot, \cdot)$  be a distribution divergence. For any fixed prompt  $\mathbf{c}$ , the training objective is defined as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0 | \mathbf{c})} \left\{ [-\alpha r(\mathbf{x}_0, \mathbf{c})] + D(p_\theta, p_{ref}) \right\} \quad (3.1)$$

Here  $\alpha$  is a coefficient that balances reward influences and  $D(\cdot)$  acts as a regularization term. When using Kullback-Leibler divergence as the regularization term, the objective (3.1) turns to the online Proximal Policy Gradient (PPO) algorithm (Schulman et al., 2017), which has much success



Figure 1: Generated  $1024 \times 1024$  images from one-step 2.6B DI\*-SDXL model. The DI\*-SDXL-1step model outperforms FLUX-dev-50step(Lab, 2024) and SD3.5-large-28step(AI, 2024) in human preference metrics on the Parti prompt benchmark and HPSv2.1 scores. Please zoom in to check the details.

in reinforcement learning from human feedback (RLHF) for large language models (Ouyang et al., 2022) and one-step text-to-image models alignment (Luo, 2024). Particularly, recently (Luo, 2024) has shown that traditional KL-PPO can fairly train human-preferred one-step text-to-image models. However, KL divergence is notorious for mode-seeking behavior, which makes Luo (2024) very easy to collapse to some distribution modes – such as

painting-like images – that have high human reward but lack diversity. To address this issue, we find that general score-based divergences have an appealing diversity-preserving property that has been proved in one-step diffusion distillation (Luo et al., 2024c; Zhou et al., 2024b). Based on this intuition, we propose Diff-Instruct\*, a novel post-training approach built upon a new Score-based online PPO paradigm that uses a general score-based divergence

to regularize the RLHF process.

### 3.1. Score-based Divergences

Inspired by Score-Implicit Matching (SIM (Luo et al., 2024c)) theory that revealed the advantages of general score-based divergences over the traditional KL divergence, we define the regularization term  $\mathbf{D}(p_\theta, p_{ref})$  in (3.1) via the following general score-based divergence. Assume  $\mathbf{d} : \mathbb{R}^d \rightarrow \mathbb{R}$  is a scalar-valued proper distance function (i.e., a non-negative function that satisfies  $\forall \mathbf{x}, \mathbf{d}(\mathbf{x}) \geq 0$  and  $\mathbf{d}(\mathbf{x}) = 0$  if and only if  $\mathbf{x} = \mathbf{0}$ ). Given a parameter-independent sampling distribution  $\pi_t$  that has large distribution support, we can formally define a time-integral score divergence as

$$\mathbf{D}^{[0,T]}(p_\theta, p_{ref}) := \int_0^T w(t) \mathbb{E}_{\pi_t} \mathbf{d}(s_{p_{\theta,t}} - s_{q_t}) dt. \quad (3.2)$$

where  $p_{\theta,t}$  and  $q_t$  denote the marginal densities of the diffusion process (2.1) at time  $t$  initialized with  $p_{\theta,0} = p_\theta$  and  $q_0 = p_{ref}$  respectively.  $w(t)$  is an integral weighting function. Clearly, we have  $\mathbf{D}^{[0,T]}(p_\theta, p_{ref}) = 0$  if and only if  $p_\theta(\mathbf{x}_0) = p_{ref}(\mathbf{x}_0)$ , a.s.  $\pi_0$ .

### 3.2. Theories for Tractable Objectives

Recall that  $g_\theta$  is a one-step model, therefore, samples from  $p_\theta$  can be implemented through a direct and differentiable mapping  $\mathbf{x}_0 = g_\theta(\mathbf{z}, \mathbf{c})$ . With the score-based regularization term (3.2), for each given text prompt  $\mathbf{c}$ , we can formally write down our training objective to minimize as:

$$\mathcal{L}_{Orig}(\theta) = \mathbb{E}_{\substack{\mathbf{z} \sim p_{\mathbf{z}}, \\ \mathbf{x}_0 = g_\theta(\mathbf{z}, \mathbf{c})}} [-\alpha r(\mathbf{x}_0, \mathbf{c})] + \mathbf{D}^{[0,T]}(p_\theta, p_{ref}) \quad (3.3)$$

Now we are ready to reveal the objective of Diff-Instruct\* that we use to train human-preferred one-step models  $g_\theta$ . Notice that directly minimizing objective (3.3) is intractable because we do not know the relationship between  $\theta$  and corresponding  $p_{\theta,t}$ . However, we show in Theorem 3.1 that an equivalent tractable loss (3.4) will have the same  $\theta$  gradient as the intractable loss function (3.3):

$$\begin{aligned} \mathcal{L}_{DI*}(\theta) = & \mathbb{E}_{\substack{\mathbf{z} \sim p_{\mathbf{z}}, \\ \mathbf{x}_0 = g_\theta(\mathbf{z}, \mathbf{c})}} \left[ -\alpha r(\mathbf{x}_0, \mathbf{c}) \right. \\ & + \int_{t=0}^T w(t) \mathbb{E}_{\substack{\mathbf{x}_t | \mathbf{x}_0 \\ \sim p_t(\mathbf{x}_t | \mathbf{x}_0)}} \left\{ -\mathbf{d}'(\mathbf{y}_t) \right\}^T \\ & \left. \left\{ s_{p_{sg[\theta],t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}_0) \right\} dt \right]. \end{aligned} \quad (3.4)$$

with  $\mathbf{y}_t := s_{p_{sg[\theta],t}}(\mathbf{x}_t) - s_{q_t}(\mathbf{x}_t)$ .

**Theorem 3.1.** Under mild assumptions, if we take the sampling distribution in (3.2) as  $\pi_t = p_{sg[\theta],t}$ , then the gradient of (3.3) w.r.t  $\theta$  is the same as (3.4):

$$\frac{\partial}{\partial \theta} \mathcal{L}_{Orig}(\theta) = \frac{\partial}{\partial \theta} \mathcal{L}_{DI*}(\theta).$$

In practice, we can use another assistant diffusion model  $s_\psi(\mathbf{x}_t, t)$  to approximate the one-step model’s score func-

tion  $s_{p_{sg[\theta],t}}(\mathbf{x}_t)$  pointwise, which was also done in the literature of diffusion distillations works such as Luo et al. (2024b;c); Zhou et al. (2024a;b); Yin et al. (2023; 2024). Therefore, we can alternate between 1) updating the assistant diffusion  $s_\psi(\mathbf{x}_t, t)$  using one-step model generated samples (which are efficient) and 2) updating the one-step model by minimizing the tractable objective (3.4). We name our training method that minimizes the objective  $\mathcal{L}_{DI*}(\theta)$  in (3.4) the Diff-Instruct\* because it is inspired by Diff-Instruct (Luo et al., 2024b) and Diff-Instruct++ (Luo, 2024) that involves an additional diffusion model and a reward model to train one-step text-to-image models.

### 3.3. Understanding Classifier-free Guidance

**Classifier-free Guidance Secretly Induced an Implicit Reward.** In previous sections, we have shown in theory that with explicitly available reward models, we can readily train the one-step models to align with human preference. In this section, we enhance the DI\* by incorporating the classifier-free reward that is implied by the classifier-free guidance of diffusion models.

The classifier-free guidance (Ho & Salimans, 2022) (CFG) uses a modified score function of the form

$$\tilde{s}_{ref}(\mathbf{x}_t, t | \mathbf{c}) := s_{ref}(\mathbf{x}_t, t | \emptyset) + \omega \{ s_{ref}(\mathbf{x}_t, t | \mathbf{c}) - s_{ref}(\mathbf{x}_t, t | \emptyset) \},$$

to replace the original conditional score function  $s_{ref}(\mathbf{x}_t, t | \mathbf{c})$ . Using CFG for diffusion models empirically leads to better sampling fidelity but with a cost of diversity.

As is first pointed out by Luo (2024), the classifier-free guidance is related to an implicit RLHF. In this part, we derive a tractable loss function that minimizes the so-called classifier-free reward, which we use together with the explicit reward  $r(\cdot, \cdot)$  in DI\*.

**Theorem 3.2.** Under mild conditions, if we set an implicit reward function as (3.6), the loss (3.5)

$$\begin{aligned} \mathcal{L}_{cfg}(\theta) = & \int_0^T \mathbb{E}_{\substack{\mathbf{z} \sim p_{\mathbf{z}}, \mathbf{x}_0 = g_\theta(\mathbf{z}, \mathbf{c}) \\ \mathbf{x}_t \sim p(\mathbf{x}_t | \mathbf{x}_0)}} w(t) [s_{ref}(sg[\mathbf{x}_t] | t, \mathbf{c}) \\ & - s_{ref}(sg[\mathbf{x}_t] | t, \emptyset)]^T \mathbf{x}_t dt. \end{aligned} \quad (3.5)$$

has the same gradient as the negative implicit reward function (3.6)

$$-r(\mathbf{x}_0, \mathbf{c}) = - \int_{t=0}^T \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} w(t) \log \frac{p_{ref}(\mathbf{x}_t | t, \mathbf{c})}{p_{ref}(\mathbf{x}_t | t)} dt. \quad (3.6)$$

The notation  $sg[\mathbf{x}_t]$  means detaching the  $\theta$  gradient on  $\mathbf{x}_t$ . Theorem 3.2 gives a tractable loss function (3.5) aiming to minimize the negative classifier-free reward function. Therefore, we can scale and add this loss  $\mathcal{L}_{cfg}(\theta)$  (3.5) to the DI\* loss (3.4) to balance the effects of explicit reward and implicit CFG reward.

**Choice 1: Incorporating CFG via Implicit CFG Reward.** Based on Theorem 3.2, we can incorporate



**Algorithm 1:** Diff-Instruct\* Pseudo Code.

---

**Input:** prompt dataset  $\mathcal{C}$ , generator  $g_\theta(\mathbf{x}_0|\mathbf{z}, \mathbf{c})$ , prior distribution  $p_z$ , reward model  $r(\mathbf{x}, \mathbf{c})$ , reward model scale  $\alpha_{rew}$ , CFG reward scale  $\alpha_{cfg}$ , reference diffusion model  $\mathbf{s}_{ref}(\mathbf{x}_t|\mathbf{c}, \mathbf{c})$ , assistant diffusion  $\mathbf{s}_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c})$ , forward diffusion  $p(\mathbf{x}_t|\mathbf{x}_0)$  (2.1), assistant diffusion updates rounds  $K_{TA}$ , time distribution  $\pi(t)$ , diffusion model weighting  $\lambda(t)$ , generator IKL loss weighting  $w(t)$ .

**while** not converge **do**

freeze  $\theta$ , update  $\psi$  for  $K_{TA}$  rounds by

1. sample prompt  $\mathbf{c} \sim \mathcal{C}$ ; sample time  $t \sim \pi(t)$ ; sample  $\mathbf{z} \sim p_z(\mathbf{z})$ ;
2. generate fake data:  $\mathbf{x}_0 = \text{sg}[g_\theta(\mathbf{z}, \mathbf{c})]$ ; sample noisy data:  $\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0)$ ;
3. update  $\psi$  by minimizing loss:  $\mathcal{L}(\psi) = \lambda(t)\|\mathbf{s}_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2$ ;

freeze  $\psi$ , update  $\theta$  using SGD:

1. sample prompt  $\mathbf{c} \sim \mathcal{C}$ ; sample time  $t \sim \pi(t)$ ; sample  $\mathbf{z} \sim p_z(\mathbf{z})$ ;
2. generate fake data:  $\mathbf{x}_0 = g_\theta(\mathbf{z}, \mathbf{c})$ ; sample noisy data:  $\mathbf{x}_t \sim p_t(\mathbf{x}_t|\mathbf{x}_0)$ ;
3. explicit reward:  $\mathcal{L}_{rew}(\theta) = -\alpha_{rew}r(\mathbf{x}_0, \mathbf{c})$ ;
4. CFG reward:  $\mathcal{L}_{cfg}(\theta) = \alpha_{cfg} \cdot w(t)\{\mathbf{s}_{ref}(\text{sg}[\mathbf{x}_t]|\mathbf{t}, \mathbf{c}) - \mathbf{s}_{ref}(\text{sg}[\mathbf{x}_t]|\mathbf{t}, \emptyset)\}^T \mathbf{x}_t$ ;
5. score-regularization:  $\mathcal{L}_{reg}(\theta) = -w(t)\{\mathbf{d}'(\mathbf{s}_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) - \mathbf{s}_{ref}(\mathbf{x}_t|\mathbf{t}, \mathbf{c}))\}^T \{\mathbf{s}_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\}$ ;
6. update  $\theta$  by minimizing DI\* loss:  $\mathcal{L}_{DI*}(\theta) = \mathcal{L}_{rew}(\theta) + \mathcal{L}_{cfg}(\theta) + \mathcal{L}_{reg}(\theta)$ ;

**end**

**return**  $\theta, \psi$ .

---

classifier-free guidance by adding the pseudo reward (3.6) to explicit human reward to get a mixed reward. The guidance scale  $\alpha_{cfg}$  and  $\alpha_{rew}$  balances the strength of explicit human reward and implicit CFG reward.

**Choice 2: Incorporate CFG via CFG-enhanced Reference Diffusion.** Besides the implicit CFG reward, we can also inject the CFG mechanism into Diff-Instruct\* by replacing the naive reference diffusion with a CFG-enhanced reference diffusion which writes  $\tilde{\mathbf{s}}_{ref}(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) := \mathbf{s}_{ref}(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) + \alpha_{cfg}(\mathbf{s}_{ref}(\mathbf{x}_t|\mathbf{t}, \emptyset))$ . This approach is straightforward. In practice, we find empirically that the **choice 2** is easier to tune and results in better performances. However, **choice 1** also leads to solid one-step models. Please check out ablation studies in Table 2 for details.

### 3.4. The Practical Algorithm

As Algorithm 1 shows, the DI\* involves three models, with one one-step model  $g_\theta$ , one reference diffusion model  $\mathbf{s}_{ref}$  and one assistant diffusion model  $\mathbf{s}_\psi$ . **The reference diffusion is frozen, while the one-step model and the assistant diffusion are updated alternately.** Two hyperparameters, the  $\alpha_{rew}$  and  $\alpha_{cfg}$ , control the strength of the explicit reward and the implicit CFG reward during training. The explicit rewards can either be off-the-shelf reward models,

such as PickScore (Kirstain et al., 2023), CLIP score (Radford et al., 2021), or rewards trained in-house with internal feedback data.

**Pseudo-Huber Distance Functions.** Various choices of distance function  $\mathbf{d}(\cdot)$  result in different training algorithms. For instance,  $\mathbf{d}(\mathbf{y}_t) = \|\mathbf{y}_t\|_2^2$  is a naive choice. This distance function has been studied in the pure diffusion distillation literature (Luo et al., 2024c; Zhou et al., 2024b;a). In this paper, we draw inspiration from (Luo et al., 2024c) and find that using the so-called pseudo-Huber distance leads to better performance. The distance writes  $\mathbf{d}(\mathbf{y}) := \sqrt{\|\mathbf{y}_t\|_2^2 + c^2} - c$ , and  $\mathbf{y}_t := \mathbf{s}_{p_{\text{sg}[\theta], t}}(\mathbf{x}_t) - \mathbf{s}_{q_t}(\mathbf{x}_t)$ , then

$$\begin{aligned} & \mathbf{D}^{[0, T]}(p_\theta, p_{ref}) \\ &= -\left\{ \frac{\mathbf{y}_t}{\sqrt{\|\mathbf{y}_t\|_2^2 + c^2}} \right\}^T \left\{ \mathbf{s}_\psi(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) \right\}. \end{aligned} \quad (3.7)$$

## 4. Related Works

**Diffusion Distillation Through Divergence Minimization.** Diff-Instruct\* is inspired by research on diffusion distillation (Luo, 2023), which aims to minimize distribution divergence to train one-step text-to-image models. Luo et al. (2024b) first studies the diffusion distillation by minimizing the Integral KL divergence. Yin et al. (2023) generalizes Diff-Instruct and adds a data regression loss for bet-

ter performance. Zhou et al. (2024b) study the distillation by minimizing the Fisher divergence. Luo et al. (2024c) study the distillation using the general score-based divergence. Many other works also introduced additional techniques and improved the performance (Geng et al., 2023; Kim et al., 2023; Song et al., 2023; Song & Dhariwal, 2024; Nguyen & Tran, 2023; Song et al., 2024; Yin et al., 2024; Zhou et al., 2024a; Heek et al., 2024; Xie et al., 2024; Salimans et al., 2024; Geng et al., 2024; Meng et al., 2022; Sauer et al., 2023b; Luo et al., 2023a; Liu et al., 2023; Gu et al., 2023; Xu et al., 2024; Ren et al., 2024; Lin et al., 2024; Zheng et al., 2024; Li et al., 2024; Berthelot et al., 2023; Zheng et al., 2022).

**Preference Alignment for Diffusion Models and One-step Models.** In recent years, many works have emerged trying to align diffusion models with human preferences. There are three main lines of alignment methods for diffusion models. 1) The first kind of method fine-tunes the diffusion model over a specifically curated image-prompt dataset (Dai et al., 2023; Podell et al., 2023). 2) The second line of methods tries to maximize some reward functions either through the multi-step diffusion generation output (Prabhudesai et al., 2023; Clark et al., 2023; Lee et al., 2023) or through policy gradient-based RL approaches (Fan et al., 2024; Black et al., 2023). For these methods, the backpropagation through the multi-step diffusion generation output is expensive and hard to scale. 3) The third line, such as Diffusion-DPO (Wallace et al., 2024), Diffusion-KTO (Yang et al., 2024), and MaPO (Hong et al., 2024) tries to directly improve the diffusion model’s human preference property with raw collected data instead of reward functions. Besides the human preference alignment of diffusion models, Diff-Instruct++(Luo, 2024) recently arose as the first attempt to improve human preferences for one-step text-to-image models. Though we get inspiration from Diff-Instruct++, our Diff-Instruct\* uses score-based divergences, which are technically different from the KL divergence used in DI++. Besides, as we show in Section 5.1, Diff-Instruct\* outperforms Diff-Instruct++.

## 5. Experiments

**Experiment Settings for SDXL Experiment.** We use the open-sourced SDXL of  $1024 \times 1024$  resolution as our reference diffusion in Algorithm 1. We construct the one-step model with the same architecture as the SDXL model and initialize the one-step model weights with the DMD2-SDXL-1step model (Yin et al., 2024), which is a good one-step text-to-image model distilled by using a combination of Diff-Instruct loss and GAN loss. We use the prompts of the LAION-AESTHETIC dataset with an aesthetic score larger than 6.25, resulting in a total of 2.2 million text prompts. We use the PickScore (Kirstain et al., 2023)(a

high-quality off-the-shelf reward model trained using human feedback data) as the explicit human reward. In Table 2, we do an ablation study that compares one-step DI\* models with different guidance strategies and reward scales and finds that using inner guidance with a scale of 7.5 and an explicit human reward with a scale of 100 results in the best performance.

**Quantitative Evaluations Metrics.** We compare the DI\*-SDXL-1step model with other leading open-sourced models of a resolution of  $1024 \times 1024$  on the Parti prompt benchmark, COCO-2014-30K benchmark, and also Human preference scores. These models include diffusion-based (or flow-matching) models such as FLUX-dev (Lab, 2024), Stable-Diffusion-3.5-Large(AI, 2024), and SDXL (Podell et al., 2023) with its DPO variant (Wallace et al., 2024); 1024 resolution one-step models such as DMD2(Yin et al., 2024), Diff-Instruct(Luo et al., 2024b), Score Implicit Matching (SIM) (Luo et al., 2024c), and also Diff-Instruct++ (Luo, 2024). On Parti and COCO prompt datasets, we compute four standard scores in Table 1: the Image Reward (Xu et al., 2023), the Aesthetic Score (Schuhmann, 2022), the PickScore (Kirstain et al., 2023), and the CLIP score (Radford et al., 2021). We also evaluate the 30K-FID (Heusel et al., 2017) on the COCO validation dataset. On Human Preference Score (Wu et al., 2023) benchmarks, we calculate HPSv2.1 scores.

### 5.1. Performances and Findings

**2.6B DI\*-SDXL-1step Model Shows Very Strong Performances.** The FLUX-dev-50step model (Lab, 2024) is recognized as a leading open-sourced diffusion model that is trained on millions of high-quality internal data samples. As shown in Table 1, our best 2.6B DI\*-SDXL-1step model outperforms the 12B FLUX-dev-50step diffusion model and 8B SD3.5-large-28step using only 1.88% inference time and 29.3% GPU memory costs on the Parti prompt dataset which consists of challenging prompts with rich concepts. The 2.6B DI\*-SDXL-1step is on par with FLUX and SD3.5 on COCO prompts in terms of preference scores and FIDs. As Table 3 shows, the DI\*-SDXL-1step model has a record-breaking HPSv2.1 score of 31.19, outperforming the rest of the text-to-image models of  $1024 \times 1024$  resolution. These empirical results confirm that proper post-training enables small one-step models to outperform huge multi-step models, analogous to the story of David and Goliath. See Figure 2 for visual comparisons with the 12B FLUX-Dev and the 8B Stable Diffusion 3.5-Large models.

**Ablation Study.** In Table 2, we conduct a comprehensive ablation study on the Parti prompt dataset to compare DI\* (RLHF using score-based divergence) with different guidance strategies and reward scales (which recover previous distillation approaches such as Diff-Instruct(Luo et al., 2024b) and Score Implicit Matching(Luo et al., 2024c)),

Table 1: Quantitative comparisons of  $1024 \times 1024$  resolution leading text-to-image models on **Parti**(Yu et al., 2022) Prompts (the upper part) and **MSCOCO-2014 validation** prompts (the under part). DI\*: short for Diff-Instruct\*. SIM: short for Score Implicit Matching. †: our implementation.

MODEL	STEPS↓	TYPE	PARAMS↓	IMAGE↑ REWARD	AES↑ SCORE	PICK↑ SCORE	CLIP↑ SCORE	INFER TIME↓ PER 10 IMAGES
PARTI PROMPTS								
SDXL-BASE(PODELL ET AL., 2023)	50	UNET	2.6B	0.887	5.72	0.2274	32.72	111 SEC
SDXL-DPO(WALLACE ET AL., 2024)	50	UNET	2.6B	1.102	5.77	0.2290	<b>33.03</b>	111 SEC
SD3.5-LARGE(AI, 2024)	28	DiT	8B	<b>1.133</b>	5.70	0.2306	32.70	66.23 SEC
FLUX-DEV(LAB, 2024)	50	DiT	12B	1.132	<b>5.90</b>	<b>0.2317</b>	31.70	118.64 SEC
DMD2-SDXL(YIN ET AL., 2024)	1	UNET	2.6B	0.930	5.51	0.2249	32.97	2.22 SEC
DIFF-INSTRUCT†(LUO ET AL., 2024B)	1	UNET	2.6B	1.058	5.60	0.2253	<b>33.02</b>	2.22 SEC
SIM†(LUO ET AL., 2024C)	1	UNET	2.6B	1.049	5.66	0.2273	32.93	2.22 SEC
DIFF-INSTRUCT++-SDXL†(LUO, 2024)	1	UNET	2.6B	1.061	5.58	0.2260	32.94	2.22 SEC
<b>DI*-SDXL (OURS)</b>	1	UNET	2.6B	1.067	5.74	0.2304	32.82	2.22 SEC
<b>DI*-SDXL (OURS, LONG TRAIN)</b>	1	UNET	2.6B	1.140	5.83	0.2331	32.75	2.22 SEC
<b>DI*-SDXL-DDPO (OURS, LONG TRAIN)</b>	1	UNET	2.6B	<b>1.210</b>	<b>5.90</b>	<b>0.2342</b>	32.91	2.22 SEC
COCO-2017-VAL 30K PROMPTS								COCO-FID↓
SDXL-BASE(PODELL ET AL., 2023)	50	UNET	2.6B	0.825	5.55	0.2281	31.86	<b>15.89</b>
SDXL-DPO(WALLACE ET AL., 2024)	50	UNET	2.6B	0.950	<b>5.69</b>	0.2295	<b>32.17</b>	21.48
SD3.5-LARGE(AI, 2024)	28	DiT	8B	1.048	5.46	0.2305	31.85	17.69
FLUX-DEV(LAB, 2024)	50	DiT	12B	<b>1.065</b>	5.64	<b>0.2333</b>	30.88	24.77
DMD2-SDXL(YIN ET AL., 2024)	1	UNET	2.6B	0.822	5.42	0.2251	31.89	<b>15.80</b>
DIFF-INSTRUCT-SDXL†(LUO ET AL., 2024B)	1	UNET	2.6B	0.907	5.48	0.2264	31.85	21.62
SIM-SDXL†(LUO ET AL., 2024C)	1	UNET	2.6B	0.925	5.54	0.2277	<b>31.90</b>	18.84
DIFF-INSTRUCT++-SDXL†(LUO, 2024)	1	UNET	2.6B	0.908	5.53	0.2265	31.85	21.42
<b>DI*-SDXL (OURS)</b>	1	UNET	2.6B	0.933	5.57	0.2305	31.75	19.09
<b>DI*-SDXL (OURS, LONGER TRAINING)</b>	1	UNET	2.6B	0.983	5.65	0.2335	31.57	21.15
<b>DI*-SDXL-DPO (OURS, LONG TRAIN)</b>	1	UNET	2.6B	<b>1.034</b>	<b>5.65</b>	<b>0.2342</b>	31.58	22.12

and DI++(Luo, 2024)(RLHF with KL divergence).

The results show that DI\* using score-based divergence outperforms the DI++ model under the same configurations and reward scales. Notably, with proper reward scales, DI\* models demonstrate superior human preference scores, indicating its simple yet effective contribution to aligning one-step text-to-image models with human feedback. Critically, DI++ with KL divergence tends to collapse to painting-like images with over-saturated color and lighting (Please refer to Figure 1 for visualizations), which lack diversity despite high rewards. Instead, score-based divergences empirically maintain diversity while steadily improving human preferences, suggesting that DI\* can improve images’ human preference scores regardless of their styles in paintings, photos, or animations.

In addition, we acknowledge the trade-off between preference scores and CLIP scores: increasing explicit human reward scales degrades CLIP scores. This phenomenon reveals a natural contradiction between human-preferred and objective data distributions. RLHF encourages the one-step model to generate human-preferred images, which inevitably shifts the underlying distribution from the ground-truth data distribution. This aligns with observations that FLUX, despite strong preference scores, underperforms in CLIPScore and FID compared to smaller models.

#### DI\* is compatible with Pre-aligned Diffusion Models

Beyond the ablation comparisons of different explicit reward strengths and CFG strengths, we also find that *DI\* is compatible with pre-aligned reference diffusion models*. As Table 1 and Table 3 show, if we replace the reference diffusion model from SDXL to open-sourced SDXL-DDPO (Stable Diffusion XL with Diffusion Direct Preference Optimization (Wallace et al., 2024)), the preference alignment effects of the resulting one-step model are measured by PickScore, ImageReward, AestheticScore, and HPSv2.1, and were further improved. This demonstrates the compatibility and robustness of Diff-Instruct\* with different reference diffusion models. In practice, we recommend that users use pre-aligned diffusion models that

**Qualitative Comparisons.** Figure 1 shows some  $1024 \times 1024$  images generated by DI\*-SDXL-1step models. Figure 2 shows a qualitative comparison of the DI\*-SDXL-1step model against FLUX and SD3.5 diffusion models. The DI\*-SDXL-1step model shows better aesthetic details, improved layouts, and reality lighting compared to FLUX and SD3.5, attributed to our score post-training. Figure 4 shows the visual comparison of DI\* (with different ablative reward scales) against our initial model (the DMD2-1step model(Yin et al., 2024)), other one-step models, and SDXL with and without DPO. While SIM (Luo et al., 2024c), DI++(Luo, 2024), and Diff-Instruct(Luo et al., 2024b) can output high-quality

Table 2: Ablation study on **Parti**(Yu et al., 2022) Prompts of 1024 resolution DI\*-SDXL-1step models with different reward scales. DI\*: short for Diff-Instruct\*. SIM: short for Score Implicit Matching. **DI\*-Out**: incorporating CFG with implicit CFG reward; **DI\*-In**: incorporating CFG with enhanced reference diffusion. †: our implementation.

MODEL	STEPS	PARAMS	IMAGE↑ REWARD	AES↑ SCORE	PICK↑ SCORE	CLIP↑ SCORE	$(\alpha_{rev}, \alpha_{cfg})$
DMD2-SDXL(INIT MODEL)	1	2.6B	0.938	5.51	0.2249	32.97	—
DI++-SDXL† (ALIGNED USING KL)	1	2.6B	0.846	5.50	0.2243	32.66	(0, 0)
DI++-SDXL† (EQU TO DIFF-INSTRUCT)	1	2.6B	1.058	5.60	0.2253	33.02	(0, 7.5)
DI++-SDXL† (ALIGNED USING KL)	1	2.6B	1.061	5.58	0.2260	32.94	(100, 7.5)
<b>DI*-Out-SDXL (OUT CFG)</b>	1	2.6B	1.082	5.63	0.2263	<b>33.03</b>	(100, 7.5)
<b>DI*-In-SDXL (BASELINE, NO REWARD)</b>	1	2.6B	0.782	5.74	0.2256	32.16	(0, 0)
<b>DI*-In-SDXL (EQU TO SIM, ONLY CFG)</b>	1	2.6B	1.049	5.66	0.2273	32.93	(0, 7.5)
<b>DI*-In-SDXL (HUMAN REWARD + CFG)</b>	1	2.6B	1.031	5.69	0.2274	32.87	(1, 7.5)
<b>DI*-In-SDXL</b>	1	2.6B	1.048	5.66	0.2278	32.91	(10, 7.5)
<b>DI*-In-SDXL</b>	1	2.6B	1.020	5.68	0.2278	32.82	(100, 4.5)
<b>DI*-In-SDXL</b>	1	2.6B	1.067	5.74	0.2304	32.82	(100, 7.5)
<b>DI*-In-SDXL (BEST, LONGER TRAINING)</b>	1	2.6B	<b>1.140</b>	<b>5.83</b>	<b>0.2331</b>	32.75	(100, 7.5)

Table 3: Quantitative evaluations of models on **HPSv2.1** scores. We compare open-sourced models regardless of their base model and architecture. † indicates our implementation.

MODEL	ANIMATION↑	CONCEPT-ART↑	PAINTING↑	PHOTO↑	AVERAGE↑
50STEP-SDXL-BASE(PODELL ET AL., 2023)	30.85	29.30	28.98	27.05	29.05
50STEP-SDXL-DPO(WALLACE ET AL., 2024)	32.01	30.75	30.70	28.24	30.42
28STEP-SD3.5-LARGE	31.89	30.19	30.39	28.01	30.12
50STEP-FLUX-DEV	32.09	30.44	31.17	29.09	30.70
1STEP-DMD2-SDXL(YIN ET AL., 2024)	29.72	27.96	27.64	26.55	27.97
1STEP-DIFF-INSTRUCT-SDXL(LUO ET AL., 2024B)	31.15	29.71	29.72	28.20	29.70
1STEP-SIM-SDXL(LUO ET AL., 2024C)	31.97	30.46	30.13	28.08	30.16
1STEP-DI++-SDXL(LUO, 2024)	31.19	29.88	29.61	28.21	29.72
<b>1STEP-DI*-SDXL(OURS)</b>	32.26	30.57	30.10	27.95	30.22
<b>1STEP-DI*-SDXL(OURS, LONGER TRAINING)</b>	33.22	31.67	31.25	28.62	31.19
<b>1STEP-DI*-SDXL-DDPO(OURS, LONGER TRAINING)</b>	<b>33.92</b>	<b>32.80</b>	<b>32.71</b>	<b>29.62</b>	<b>32.26</b>

samples, they suffer from oversaturation issues, meaning that the images have over-saturated colors and lighting that might make users feel uncomfortable. On the contrary, the DI\*-1step model shows very gentle lighting and aesthetic colors, which align better with human preferences. We empirically conclude that DI\* prevents the model from collapsing into high-reward painting-like images, in which DI++ collapsed. Besides, the SDXL and SDXL-DPO are easy to generate painting-like images instead of photo-realistic images.

## 6. Conclusion and Limitations

In this paper, we present Diff-Instruct\*, a novel approach for preference alignment of one-step text-to-image generative models. We introduce two novel approaches to incorporate traditional classifier-free guidance into alignment through the lens of explicit-implicit reward decoupling. Using DI\*, we introduce the 2.6B DI\*-SDXL-1step model that outperforms current leading models such as the 12B FLUX-dev-50step model with 1.88% inference cost.

Nonetheless, Diff-Instruct\* has its limitations. We empir-

ically found that flaws commonly observed in other one-step models also appeared in Diff-Instruct\*. For example, the one-step model sometimes generates inaccurate human faces and hands. We believe that these bad cases are caused by the limitations of the one-step generation approach, which poses challenges for the model in generating correct content in a single pass. We believe that generalizing Diff-Instruct\* to multi-step (few-step) models could help combine the advantages of both efficiency and generation correctness. Our results suggest that consistently improving the one-step model architecture, scaling its parameters, improving reference diffusions, and enhancing the reward models can lead to better models.

Finally, we identify directions for further research for score post-training. For instance, while DI\* needs a pre-trained reward model, methods like DPO (Rafailov et al., 2024; Wallace et al., 2024) have introduced aligning generative models directly using human feedback data. It would be valuable to explore efficient score post-training for one-step generative models by directly incorporating feedback data and eliminating the pre-trained reward models.



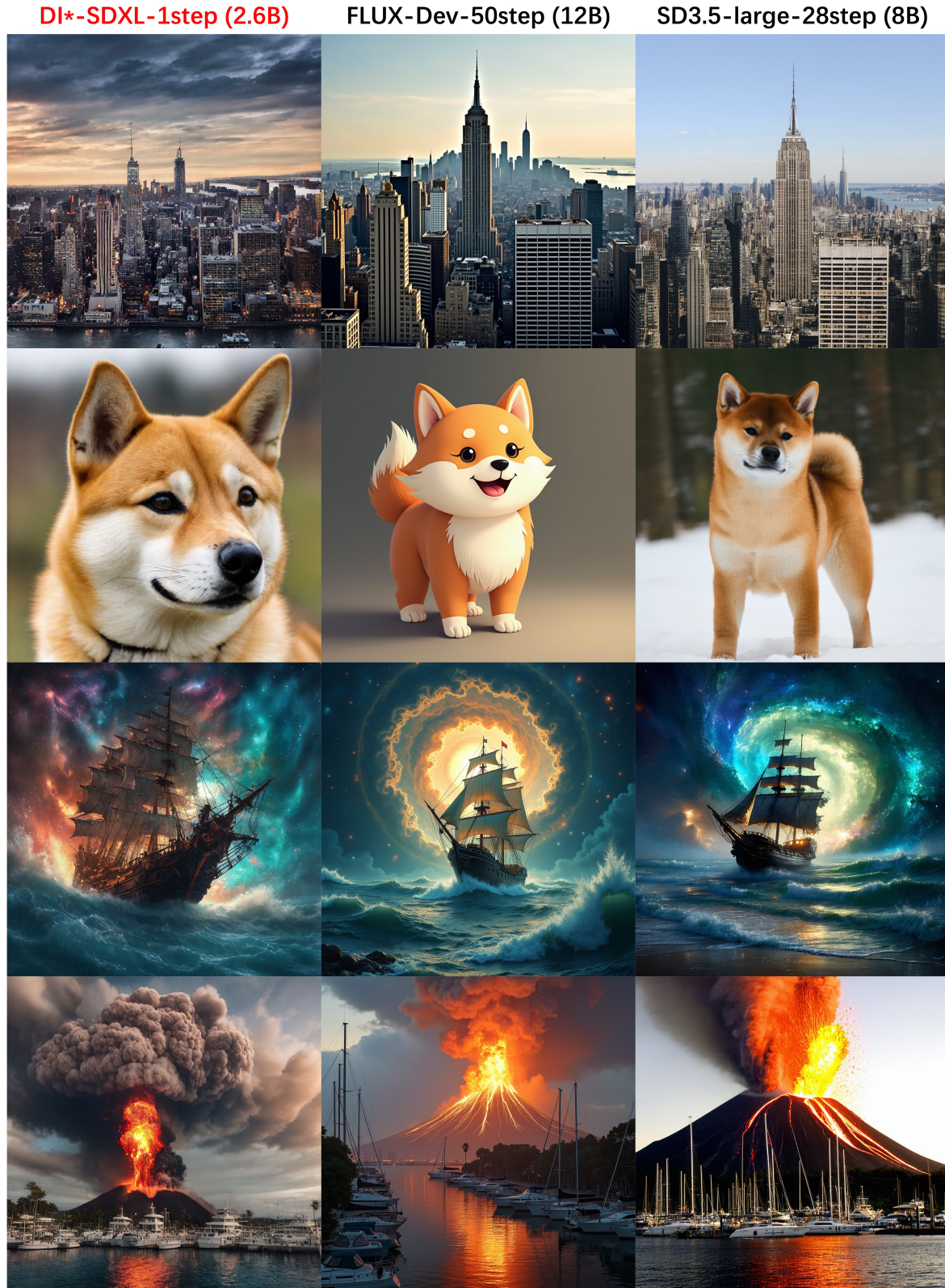


Figure 2: Comparison on Parti prompts of our **2.6B DI\*-SDXL-1step** model against 12B FLUX-dev-50step and 8B SD3.5-large-28step diffusion models. Prompts used are listed in the appendix.

## Acknowledgement

We acknowledge constructive suggestions and feedback from reviewers and ACs/SACs/PCs of NeurIPS 2024. We thank the authors of DMD2 for their open-weight DMD2 models, which we use as the base model before post-training.

## Impact Statement

We propose Diff-Instruct\* (DI\*) that can effectively post-train one-step text-to-image generative models to be aligned with human preferences. The method and corresponding aligned models lie in the domain of Reinforcement Learning using Human Feedback, which involves the participation of humans for training reward models. Though the Diff-Instruct\* algorithm itself does not generate harms to society, any misuse of the algorithm could potentially lead to misleading, fake, or biased generated content. We conduct experiments on academic benchmarks, whose resulting models are less likely to be misused. Further experiments are needed to better understand these consistency model limitations and propose solutions to address them

## References

- AI, S. Stable diffusion 3.5 large. <https://huggingface.co/stabilityai/stable-diffusion-3.5-large>, 2024.
- Berthelot, D., Autef, A., Lin, J., Yap, D. A., Zhai, S., Hu, S., Zheng, D., Talbott, W., and Gu, E. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.
- Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.
- Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., and Ramesh, A. Video generation models as world simulators. 2024.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.
- Couairon, G., Verbeek, J., Schwenk, H., and Cord, M. Diffedit: Diffusion-based semantic image editing with mask guidance. *ArXiv*, abs/2210.11427, 2022.
- Dai, X., Hou, J., Ma, C.-Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023.
- Deng, W., Luo, W., Tan, Y., Biloš, M., Chen, Y., Nevmyvaka, Y., and Chen, R. T. Variational schrödinger diffusion models. *arXiv preprint arXiv:2405.04795*, 2024.
- Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.
- Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Lee, K., and Lee, K. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Feng, Y., Luo, W., Huang, Y., and Wang, T. A lipschitz bandits approach for continuous hyperparameter optimization. *arXiv preprint arXiv:2302.01539*, 2023.
- Geng, Z., Pokle, A., and Kolter, J. Z. One-step diffusion distillation via deep equilibrium models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=b6XvK2de99>.
- Geng, Z., Pokle, A., Luo, W., Lin, J., and Kolter, J. Z. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- Gu, J., Zhai, S., Zhang, Y., Liu, L., and Susskind, J. Boot: Data-free distillation of denoising diffusion models with bootstrapping. *arXiv preprint arXiv:2306.05544*, 2023.
- Heek, J., Hoogetboom, E., and Salimans, T. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

- Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022.
- Hong, J., Paul, S., Lee, N., Rasul, K., Thorne, J., and Jeong, J. Margin-aware preference optimization for aligning diffusion models without reference. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2024.
- Kang, M., Zhu, J.-Y., Zhang, R., Park, J., Shechtman, E., Paris, S., and Park, T. Scaling up gans for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.
- Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.
- Kim, H., Kim, S., and Yoon, S. Guided-tts: A diffusion model for text-to-speech via classifier guidance. In *International Conference on Machine Learning*, pp. 11119–11133. PMLR, 2022.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., and Levy, O. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *arXiv preprint arXiv:2305.01569*, 2023.
- Lab, B. F. Flux. <https://github.com/black-forest-labs/flux>, 2024.
- Lee, K., Liu, H., Ryu, M., Watkins, O., Du, Y., Boutilier, C., Abbeel, P., Ghavamzadeh, M., and Gu, S. S. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.
- Li, J., Feng, W., Chen, W., and Wang, W. Y. Reward guided latent consistency distillation. *arXiv preprint arXiv:2403.11027*, 2024.
- Lin, S., Wang, A., and Yang, X. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.
- Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2014. URL <http://arxiv.org/abs/1405.0312>. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.
- Liu, X., Zhang, X., Ma, J., Peng, J., 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.
- Luo, S., Tan, Y., Huang, L., Li, J., and Zhao, H. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023a.
- Luo, W. A comprehensive survey on knowledge distillation of diffusion models. *arXiv preprint arXiv:2304.04262*, 2023.
- Luo, W. Diff-instruct++: Training one-step text-to-image generator model to align with human preferences. *arXiv preprint arXiv:2410.18881*, 2024.
- Luo, W. and Zhang, Z. Data prediction denoising models: The pupil outdoes the master, 2024. URL <https://openreview.net/forum?id=wYmcfur889>.
- Luo, W., Jiang, H., Hu, T., Sun, J., Li, Z., and Zhang, Z. Training energy-based models with diffusion contrastive divergences. *arXiv preprint arXiv:2307.01668*, 2023b.
- Luo, W., Hu, T., Zhang, S., Sun, J., Li, Z., and Zhang, Z. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024a.
- Luo, W., Hu, T., Zhang, S., Sun, J., Li, Z., and Zhang, Z. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024b.
- Luo, W., Huang, Z., Geng, Z., Kolter, J. Z., and Qi, G.-J. One-step diffusion distillation through score implicit matching. *arXiv preprint arXiv:2410.16794*, 2024c.
- Luo, W., Zhang, B., and Zhang, Z. Entropy-based training methods for scalable neural implicit samplers. *Advances in Neural Information Processing Systems*, 36, 2024d.
- Meng, C., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. Sdedit: Image synthesis and editing with stochastic

- differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. *arXiv preprint arXiv:2210.03142*, 2022.
- Nguyen, T. H. and Tran, A. Swiftbrush: One-step text-to-image diffusion model with variational score distillation. *arXiv preprint arXiv:2312.05239*, 2023.
- Nichol, A. and Dhariwal, P. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.
- Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.
- Pokle, A., Geng, Z., and Kolter, J. Z. Deep equilibrium approaches to diffusion models. *Advances in Neural Information Processing Systems*, 35:37975–37990, 2022.
- Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- Prabhudesai, M., Goyal, A., Pathak, D., and Fragkiadaki, K. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- Ren, Y., Xia, X., Lu, Y., Zhang, J., Wu, J., Xie, P., Wang, X., and Xiao, X. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. *arXiv preprint arXiv:2404.13686*, 2024.
- Salimans, T., Mensink, T., Heek, J., and Hoogeboom, E. Multistep distillation of diffusion models via moment matching. *arXiv preprint arXiv:2406.04103*, 2024.
- Sauer, A., Karras, T., Laine, S., Geiger, A., and Aila, T. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. In *International conference on machine learning*, pp. 30105–30118. PMLR, 2023a.
- Sauer, A., Lorenz, D., Blattmann, A., and Rombach, R. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023b.
- Schuhmann, C. Laion-aesthetics. <https://laion.ai/blog/laion-aesthetics/>, 2022. Accessed: 2023 - 11- 10.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.
- Song, Y. and Dhariwal, P. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.
- Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.
- Song, Y., Sun, Z., and Yin, X. Sdxs: Real-time one-step latent diffusion models with image conditions. *arXiv preprint arXiv:2403.16627*, 2024.
- Tashiro, Y., Song, J., Song, Y., and Ermon, S. Csd: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.



- Vincent, P. A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7): 1661–1674, 2011.
- Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Rushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.
- Wang, Y., Bai, W., Luo, W., Chen, W., and Sun, H. Integrating amortized inference with diffusion models for learning clean distribution from corrupted images. *arXiv preprint arXiv:2407.11162*, 2024.
- Wu, X., Hao, Y., Sun, K., Chen, Y., Zhu, F., Zhao, R., and Li, H. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.
- Xie, S., Xiao, Z., Kingma, D. P., Hou, T., Wu, Y. N., Murphy, K. P., Salimans, T., Poole, B., and Gao, R. Em distillation for one-step diffusion models. *arXiv preprint arXiv:2405.16852*, 2024.
- Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.
- Xu, Y., Zhao, Y., Xiao, Z., and Hou, T. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8196–8206, 2024.
- Xue, S., Yi, M., Luo, W., Zhang, S., Sun, J., Li, Z., and Ma, Z.-M. SA-solver: Stochastic adams solver for fast sampling of diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=f6a9XVfYIo>.
- Yang, K., Tao, J., Lyu, J., Ge, C., Chen, J., Shen, W., Zhu, X., and Li, X. 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*, pp. 8941–8951, 2024.
- Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W. T., and Park, T. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2023.
- Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., and Freeman, W. T. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024.
- Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.
- Zhang, B., Luo, W., and Zhang, Z. Enhancing adversarial robustness via score-based optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL <https://openreview.net/forum?id=MOAHXRzHhm>.
- Zhang, B., Luo, W., and Zhang, Z. Purify++: Improving diffusion-purification with advanced diffusion models and control of randomness. *arXiv preprint arXiv:2310.18762*, 2023b.
- Zheng, B. and Yang, T. Diffusion models are innate one-step generators. *arXiv preprint arXiv:2405.20750*, 2024.
- Zheng, H., Nie, W., Vahdat, A., Azizzadenesheli, K., and Anandkumar, A. Fast sampling of diffusion models via operator learning. *arXiv preprint arXiv:2211.13449*, 2022.
- Zheng, J., Hu, M., Fan, Z., Wang, C., Ding, C., Tao, D., and Cham, T.-J. Trajectory consistency distillation. *arXiv preprint arXiv:2402.19159*, 2024.
- Zhou, M., Wang, Z., Zheng, H., and Huang, H. Long and short guidance in score identity distillation for one-step text-to-image generation. *arXiv preprint arXiv:2406.01561*, 2024a.
- Zhou, M., Zheng, H., Wang, Z., Yin, M., and Huang, H. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. *arXiv preprint arXiv:2404.04057*, 2024b.

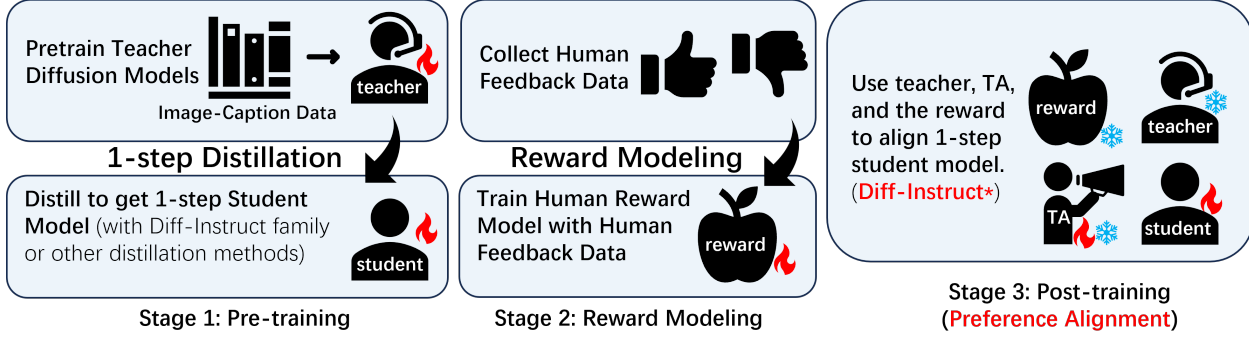


Figure 3: Workflow of developing one-step text-to-image generative models using Diff-Instruct\*. The workflow includes stages of pre-training, reward modeling, and preference alignment.

## A. Theoretical Results

### A.1. Proof of Theorem 3.1

*Proof.* Recall that  $p_\theta(\cdot)$  is induced by the one-step model  $g_\theta(\cdot)$ , therefore the sample is obtained by  $\mathbf{x}_0 = g_\theta(\mathbf{z}|\mathbf{c})$ ,  $\mathbf{z} \sim p_z$ . The term  $\mathbf{x}$  contains parameter through  $\mathbf{x}_0 = g_\theta(\mathbf{z}|\mathbf{c})$ ,  $\mathbf{z} \sim p_z$ . To demonstrate the parameter dependence, we use the notation  $p_\theta(\cdot)$ . Note that  $p_{ref}(\cdot)$  is the reference distribution. The alignment objective writes

$$\mathcal{L}_{Orig}(\theta) = \mathbb{E}_{\substack{\mathbf{z} \sim p_z, \\ \mathbf{x}_0 = g_\theta(\mathbf{z}|\mathbf{c})}} [-\alpha r(\mathbf{x}_0, \mathbf{c})] + \mathbf{D}^{[0,T]}(p_\theta, p_{ref}) \quad (\text{A.1})$$

The first loss term of (A.1)  $\alpha r(\mathbf{x}_0, \mathbf{c})$  is easy to compute by directly pushing the generated sample  $\mathbf{x}_0$  and the text prompt  $\mathbf{c}$  into the reward model  $r(\cdot, \cdot)$ . However, the second loss term (A.2) is intractable because we do not explicitly know the relation between  $\theta$  and  $p_{\theta,t}(\cdot)$ .

$$\mathbf{D}^{[0,T]}(p_\theta, p_{ref}) := \int_{t=0}^T w(t) \mathbb{E}_{\mathbf{x}_t \sim \pi_t} \left\{ \mathbf{d}(s_{p_{\theta,t}}(\mathbf{x}_t) - s_{q_t}(\mathbf{x}_t)) \right\} dt, \quad (\text{A.2})$$

We turn to derive the equivalent loss for  $\mathbf{D}^{[0,T]}(p_\theta, p_{ref})$ . First we take the  $\theta$  gradient of (A.2), show

$$\frac{\partial}{\partial \theta} \mathbf{D}^{[0,T]}(p_\theta, p_{ref}) = \frac{\partial}{\partial \theta} \int_{t=0}^T w(t) \mathbb{E}_{\mathbf{x}_t \sim \pi_t} \left\{ \mathbf{d}(s_{p_{\theta,t}}(\mathbf{x}_t) - s_{q_t}(\mathbf{x}_t)) \right\} dt \quad (\text{A.3})$$

$$= \mathbb{E}_{t, \mathbf{x}_t \sim \pi_t} w(t) \left\{ \mathbf{d}'(\mathbf{y}_t) \right\}^T \frac{\partial}{\partial \theta} s_{p_{\theta,t}}(\mathbf{x}_t) \quad (\text{A.4})$$

Notice that  $p_{\theta,t}(\cdot)$  is induced by first generating samples with one-step one-step model then adding noise with diffusion process (2.1), we do not know the term  $\frac{\partial}{\partial \theta} s_{p_{\theta,t}}(\mathbf{x}_t)$ . Therefore the gradient formula (A.4) is intractable. However, we will show that a tractable loss function can recover the intractable gradient (A.4), and therefore can be used for minimizing (A.2). Our proof is inspired by the theory from Vincent (2011), Zhou et al. (2024b) and Luo et al. (2024c).

We first present a so-called Score-projection identity (Theorem A.1), which has been studied in Zhou et al. (2024b) and Vincent (2011):

**Theorem A.1.** Let  $\mathbf{u}(\cdot)$  be a  $\theta$ -free vector-valued function under mild conditions, the identity holds:

$$\mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t | \mathbf{x}_0 \sim q_t(\mathbf{x}_t | \mathbf{x}_0)}} \mathbf{u}(\mathbf{x}_t)^T \left\{ s_{p_{\theta,t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t | \mathbf{x}_0) \right\} = 0, \quad \forall \theta. \quad (\text{A.5})$$

We give a short proof of Theorem A.1 as a clarification. Readers can also refer to Vincent (2011) or Zhou et al. (2024b) as a reference.

# One-step Model Beats Large Diffusion with Score Post-training

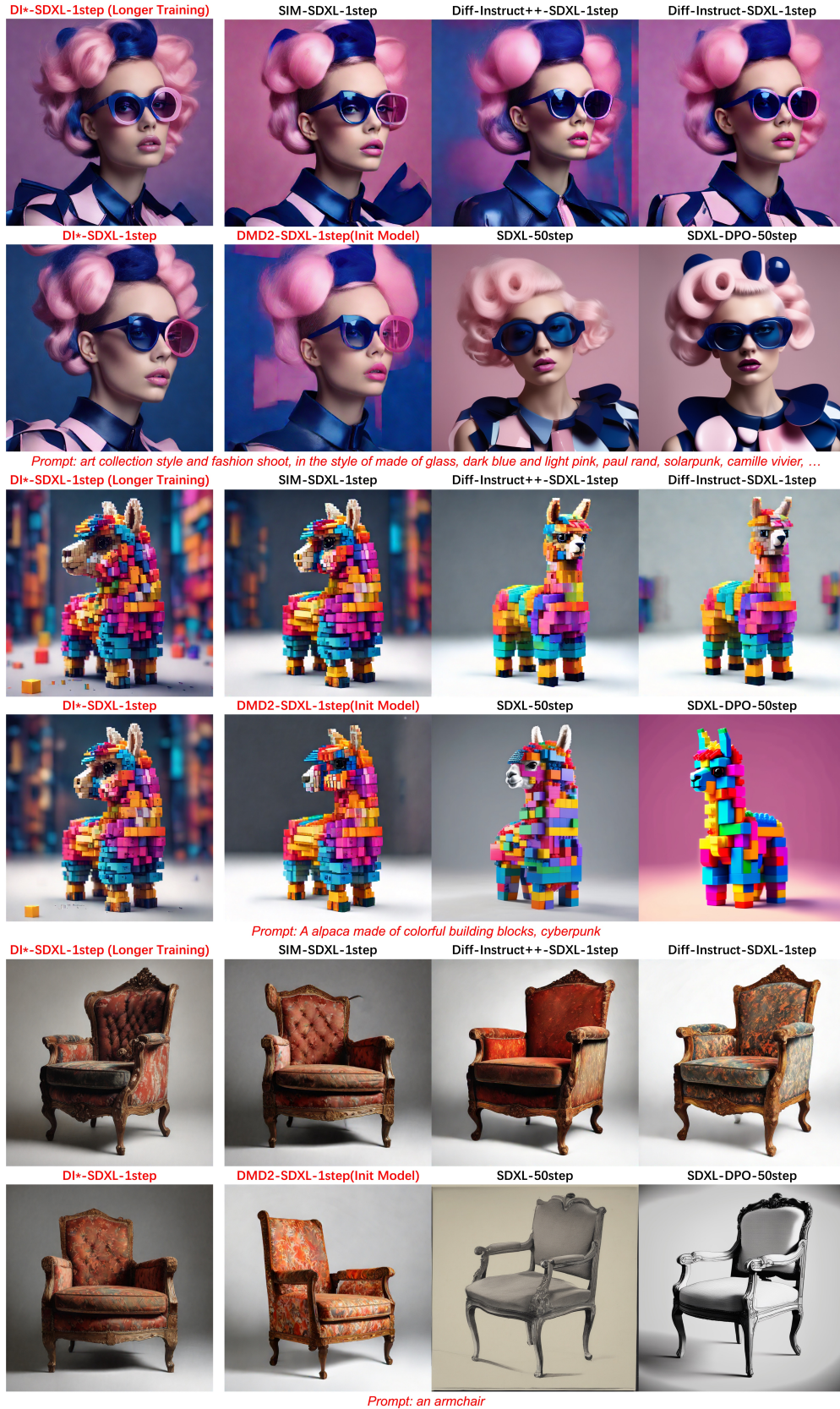


Figure 4: Ablation comparisons of DI\*-1step models and other SDXL-based models in Table 2.

Recall the relation between  $s_{p_{\theta,t}}(\mathbf{x}_t)$  and  $\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)$ , we know

$$\begin{aligned} s_{p_{\theta,t}}(\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log \int p_{\theta,0}(\mathbf{x}_0) q_t(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 \\ &= \frac{\int p_{\theta,t}(\mathbf{x}_0) \nabla_{\mathbf{x}_t} q_t(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0}{p_{\theta,t}(\mathbf{x}_t)} \\ &= \int \frac{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,t}(\mathbf{x}_0) q_t(\mathbf{x}_t|\mathbf{x}_0)}{p_{\theta,t}(\mathbf{x}_t)} d\mathbf{x}_0 \end{aligned}$$

We have

$$\begin{aligned} \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t|\mathbf{x}_0 \sim q_t(\mathbf{x}_t|\mathbf{x}_0)}} \mathbf{u}(\mathbf{x}_t)^T s_{p_{\theta,t}}(\mathbf{x}_t) &= \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \mathbf{u}(\mathbf{x}_t)^T s_{p_{\theta,t}}(\mathbf{x}_t) \\ &= \int p_{\theta,t}(\mathbf{x}_t) \mathbf{u}(\mathbf{x}_t)^T s_{p_{\theta,t}}(\mathbf{x}_t) d\mathbf{x}_t \\ &= \int p_{\theta,t}(\mathbf{x}_t) \mathbf{u}(\mathbf{x}_t)^T \int \frac{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,t}(\mathbf{x}_0) q_t(\mathbf{x}_t|\mathbf{x}_0)}{p_{\theta,t}(\mathbf{x}_t)} d\mathbf{x}_0 d\mathbf{x}_t \\ &= \int \mathbf{u}(\mathbf{x}_t)^T \int \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,t}(\mathbf{x}_0) q_t(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 d\mathbf{x}_t \\ &= \int \int \mathbf{u}(\mathbf{x}_t)^T \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) p_{\theta,t}(\mathbf{x}_0) q_t(\mathbf{x}_t|\mathbf{x}_0) d\mathbf{x}_0 d\mathbf{x}_t \\ &= \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t|\mathbf{x}_0 \sim q_t(\mathbf{x}_t|\mathbf{x}_0)}} \mathbf{u}(\mathbf{x}_t)^T \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) \end{aligned}$$

If we take the  $\theta$  gradient on both sides of (A.5), we have

$$\begin{aligned} 0 &= \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t|\mathbf{x}_0 \sim q_t(\mathbf{x}_t|\mathbf{x}_0)}} \left\{ \frac{\partial}{\partial \mathbf{x}_t} \left[ \mathbf{u}(\mathbf{x}_t)^T \{s_{p_{\theta,t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)\} \right] \frac{\partial \mathbf{x}_t}{\partial \theta} \right. \\ &\quad \left. - \mathbf{u}(\mathbf{x}_t)^T \frac{\partial}{\partial \mathbf{x}_0} \left[ \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) \right] \frac{\partial \mathbf{x}_0}{\partial \theta} \right\} + \mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \mathbf{u}(\mathbf{x}_t)^T \frac{\partial}{\partial \theta} \left\{ s_{p_{\theta,t}}(\mathbf{x}_t) \right\} \end{aligned} \quad (\text{A.6})$$

So we have an identity

$$\mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \mathbf{u}(\mathbf{x}_t)^T \frac{\partial}{\partial \theta} \left\{ s_{p_{\theta,t}}(\mathbf{x}_t) \right\} = - \frac{\partial}{\partial \theta} \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t|\mathbf{x}_0 \sim q_t(\mathbf{x}_t|\mathbf{x}_0)}} \left\{ \mathbf{u}(\mathbf{x}_t) \{s_{p_{\text{sg}[\theta],t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)\} \right\}$$

Notice that the left-hand side of equation (A.7) can be interpreted as the gradient of the loss function when the parameter dependency of the sampling distribution is cut off, i.e.

$$\mathbb{E}_{\mathbf{x}_t \sim p_{\theta,t}} \mathbf{u}(\mathbf{x}_t)^T \frac{\partial}{\partial \theta} \left\{ s_{p_{\theta,t}}(\mathbf{x}_t) \right\} = \frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{x}_t \sim p_{\text{sg}[\theta],t}} \left\{ \mathbf{u}(\mathbf{x}_t)^T s_{p_{\theta,t}}(\mathbf{x}_t) \right\} \quad (\text{A.7})$$

Therefore we have the final equation

$$\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{x}_t \sim p_{\text{sg}[\theta],t}} \left\{ \mathbf{u}(\mathbf{x}_t)^T s_{p_{\theta,t}}(\mathbf{x}_t) \right\} = - \frac{\partial}{\partial \theta} \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t|\mathbf{x}_0 \sim q_t(\mathbf{x}_t|\mathbf{x}_0)}} \left\{ \mathbf{u}(\mathbf{x}_t) \{s_{p_{\text{sg}[\theta],t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)\} \right\} \quad (\text{A.8})$$

which holds for arbitrary function  $\mathbf{u}(\cdot)$  and parameter  $\theta$ . If we set

$$\begin{aligned} \mathbf{u}(\mathbf{x}_t) &= \mathbf{d}'(\mathbf{y}_t) \\ \mathbf{y}_t &= s_{p_{\text{sg}[\theta],t}}(\mathbf{x}_t) - s_{q_t}(\mathbf{x}_t) \end{aligned}$$

Then we formally have

$$\begin{aligned} &\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{x}_t \sim p_{\text{sg}[\theta],t}} \left\{ \mathbf{d}'(\mathbf{y}_t) \right\}^T \left\{ s_{p_{\theta,t}}(\mathbf{x}_t) \right\} \\ &= \frac{\partial}{\partial \theta} \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_{\theta,0}, \\ \mathbf{x}_t|\mathbf{x}_0 \sim q_t(\mathbf{x}_t|\mathbf{x}_0)}} \left\{ -\mathbf{d}'(\mathbf{y}_t) \right\}^T \left\{ s_{p_{\theta,t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0) \right\} \end{aligned} \quad (\text{A.9})$$



This means that we can use the  $\theta$  gradient of a tractable loss:

$$\mathbb{E}_{\substack{t, \mathbf{x}_0 \sim p_{\theta, 0}, \\ \mathbf{x}_t | \mathbf{x}_0 \sim q_t(\mathbf{x}_t | \mathbf{x}_0)}} w(t) \left\{ -\mathbf{d}'(\mathbf{y}_t) \right\}^T \left\{ \mathbf{s}_{p_{\theta, t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t | \mathbf{x}_0) \right\} \quad (\text{A.10})$$

to replace the wanted  $\theta$  gradient (A.3), which can minimize the regularization loss (A.2).

Combining  $r(\mathbf{x}_0, \mathbf{c})$  and (A.10), we have the practical loss

$$\begin{aligned} \mathcal{L}_{DI*}(\theta) = & \mathbb{E}_{\substack{\mathbf{z} \sim p_{\mathbf{z}}, \\ \mathbf{x}_0 = g_{\theta}(\mathbf{z})}} \left[ -\alpha r(\mathbf{x}_0, \mathbf{c}) \right. \\ & \left. + \mathbb{E}_{\substack{t, \mathbf{x}_t | \mathbf{x}_0 \\ \sim q_t(\mathbf{x}_t | \mathbf{x}_0)}} w(t) \left\{ -\mathbf{d}'(\mathbf{y}_t) \right\}^T \left\{ \mathbf{s}_{p_{\text{sg}[\theta], t}}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t | \mathbf{x}_0) \right\} dt \right] \end{aligned} \quad (\text{A.11})$$

□

**Remark A.2.** In practice, most commonly used forward diffusion processes can be expressed as a form of scale and noise addition:

$$\mathbf{x}_t = \alpha(t)\mathbf{x}_0 + \beta(t)\epsilon, \quad \epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I}). \quad (\text{A.12})$$

So the term  $\mathbf{x}_t$  in equation (A.11) can be instantiated as  $\mathbf{z} \sim p_{\mathbf{z}}$ ,  $\epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})$ ,  $\mathbf{x}_t = \alpha(t)\mathbf{x}_0 + \beta(t)\epsilon$ .

## A.2. Proof of Theorem 3.2

*Proof.* Recall the definition of the classifier-free reward (3.6). The negative reward writes

$$-r(\mathbf{x}_0, \mathbf{c}) = -\mathbb{E}_{t, \mathbf{x}_t \sim p_{\theta, t}} w(t) \log \frac{p_{ref}(\mathbf{x}_t | t, \mathbf{c})}{p_{ref}(\mathbf{x}_t | t)}$$

This reward will put a higher reward on those samples that have higher class-conditional probability than unconditional probability, therefore encouraging class-conditional sampling. It is clear that

$$\begin{aligned} \frac{\partial}{\partial \theta} \left\{ -r(\mathbf{x}_0, \mathbf{c}) \right\} &= -\mathbb{E}_{t, \mathbf{x}_t \sim p_{\theta, t}} w(t) \left\{ \nabla_{\mathbf{x}_t} \log p_{ref}(\mathbf{x}_t | t, \mathbf{c}) - \nabla_{\mathbf{x}_t} \log p_{ref}(\mathbf{x}_t | t) \right\} \frac{\partial \mathbf{x}_t}{\partial \theta} \\ &= -\mathbb{E}_{t, \mathbf{x}_t \sim p_{\theta, t}} w(t) \left\{ \mathbf{s}_{ref}(\text{sg}[\mathbf{x}_t] | t, \mathbf{c}) - \mathbf{s}_{ref}(\text{sg}[\mathbf{x}_t] | t, \emptyset) \right\} \frac{\partial \mathbf{x}_t}{\partial \theta} \end{aligned} \quad (\text{A.13})$$

Therefore, we can see that the equivalent loss

$$\mathcal{L}_{cfg}(\theta) = \mathbb{E}_{\substack{t, \mathbf{z} \sim p_{\mathbf{z}}, \mathbf{x}_0 = g_{\theta}(\mathbf{z} | \mathbf{c}) \\ \mathbf{x}_t | \mathbf{x}_0 \sim p(\mathbf{x}_t | \mathbf{x}_0)}} w(t) \left\{ \mathbf{s}_{ref}(\text{sg}[\mathbf{x}_t] | t, \mathbf{c}) - \mathbf{s}_{ref}(\text{sg}[\mathbf{x}_t] | t, \emptyset) \right\}^T \mathbf{x}_t \quad (\text{A.14})$$

recovers the gradient formula (A.13). □

## B. Additional Discussions

### B.1. Meanings of Hyper-parameters.

**Meanings of Hyper-parameters.** As in Algorithm 1, the overall algorithms consist of two alternative updating steps. The first step is to update  $\psi$  of the assistant diffusion model by fine-tuning it with student-generated data. Therefore the assistant diffusion  $\mathbf{s}_{\psi}(\mathbf{x}_t | t, \mathbf{c})$  can approximate the score function of student one-step model distribution. This step means that the assistant diffusion needs to communicate with the student to know the student's status. The second step updates the one-step by minimizing the tractable loss (C.1) using SGD-based optimization algorithms such as Adam (Kingma & Ba, 2014). This step means that the teacher and the assistant diffusion discuss and incorporate the student's interests to instruct the student one-step model.

As we can see in Algorithm 1 (as well as Algorithm 2). Each hyperparameter has its intuitive meaning. The reward scale parameter  $\alpha_{rew}$  controls the strength of human preference alignment. The larger the  $\alpha_{rew}$  is, the stronger the one-step

model is aligned with human preferences. However, the drawback for a too large  $\alpha_{rew}$  might be the loss of diversity and reality. Besides, we empirically find that larger  $\alpha_{rew}$  leads to richer generation details and better generation layouts. But a very large  $\alpha_{rew}$  results in unrealistic and painting-like images.

The CFG reward scale controls the strength of using CFG rewards when training. We empirically find that the best CFG scale for Diff-Instruct\* is the same as the best CFG scale for sampling from the reference diffusion model. However,  $\alpha_{cfg}$  may conflicts with  $\alpha_{rew}$ . In the Stable Diffusion 1.5 experiment, we find that using a large CFG reward scale leads to worse human preferences. Therefore, the proper combination of  $(\alpha_{rew}, \alpha_{cfg})$  asks for careful tuning.

The diffusion model weighting  $\lambda(t)$  and the one-step model loss weighting  $w(t)$  controls the strengths put on each time level of updating assistant diffusion and the student generator. We empirically find that it is decent to set  $\lambda(t)$  to be the same as the default training weighting function for the reference diffusion. And it is decent to set the  $w(t) = 1$  for all time levels in practice. In the following section, we give more discussions on Diff-Instruct\*.

## B.2. Discussions on Diff-Instruct\*

**Flexible Choices of Divergences.** Various choices of distance function  $\mathbf{d}(\cdot)$  result in different training algorithms. In this part, we discuss two instances. The first choice distance function is a simple squared distance, i.e.  $\mathbf{d}(\mathbf{y}_t) = \|\mathbf{y}_t\|_2^2$ . The corresponding derivative term writes  $\mathbf{d}'(\mathbf{y}_t) = 2\mathbf{y}_t$ . In fact, such a distance function recovers the practical diffusion distillation loss studied in Zhou et al. (2024b;a). The second distance is the pseudo-Huber distance, which shows more robust performances than the simple squared distance. The pseudo-Huber distance is defined with  $\mathbf{d}(\mathbf{y}) := \sqrt{\|\mathbf{y}\|_2^2 + c^2} - c$ , where  $c$  is a pre-defined positive constant. The corresponding regularization loss (3.4) writes

$$\mathbf{D}^{[0,T]}(p_\theta, p_{ref}) = - \left\{ \frac{\mathbf{y}_t}{\sqrt{\|\mathbf{y}_t\|_2^2 + c^2}} \right\}^T \left\{ \mathbf{s}_\psi(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t | \mathbf{x}_0) \right\}. \quad (\text{B.1})$$

Here  $\mathbf{y}_t := \mathbf{s}_{p_{\text{sg}}[\theta], t}(\mathbf{x}_t) - \mathbf{s}_{q_t}(\mathbf{x}_t)$ .

**DI\* Does Not Need Image Data for Training.** One appealing advantage of DI\* is the image data-free property, which means that DI\* requires neither the image datasets nor synthetic images that are generated by reference diffusion models. This strength distinguishes DI\* from previous fine-tuning methods such as generative adversarial training (Goodfellow et al., 2014) which require training additional neural classifiers over image data, as well as those fine-tuning methods over large-scale synthetic or curated datasets.

## C. Qualitative Results

**Evaluation Details** To quantitatively evaluate the performances of the one-step model trained with different alignment settings, we compare our generators elaborately with other  $1024 \times 1024$  resolution open-sourced models that are either based on SDXL diffusion models or larger models such as FLUX-dev and SD 3.5 large. For all models, we compute four standard scores: the Image Reward (Xu et al., 2023), the Aesthetic Score (Schuhmann, 2022), the PickScore (Kirstain et al., 2023), and the CLIP score (Radford et al., 2021). Since most existing literature tests the human preference scores with different prompts which are possibly not available, to make a fair comparison, in our experiment, we use 30k prompts from the COCO-2014 (Lin et al., 2014) validation dataset and intensively evaluate a wide range of existing open-sourced models. All models are tested with the same prompts and the same computing devices.

Besides the COCO prompts, we also evaluate one-step models with open-sourced models with Human Preference Score v2.0 (HPSv2.0) (Wu et al., 2023) over their benchmark prompts. The HPS is a widely used standard benchmark that evaluates models' capability of generating images of 4 styles: Animation, concept art, Painting, and Photo. The score reflects the prompt following and the human preference strength of text-to-image models. We use the HPSv2's <sup>1</sup> default protocols for evaluations.

<sup>1</sup><https://github.com/tgxs002/HPSv2>

**Algorithm 2:** Diff-Instruct\*.

**Input:** prompt dataset  $\mathcal{C}$ , one-step model  $g_\theta(\mathbf{x}_0|\mathbf{z}, \mathbf{c})$ , prior distribution  $p_z$ , reward model  $r(\mathbf{x}, \mathbf{c})$ , reward model scale  $\alpha_{rew}$ , CFG reward scale  $\alpha_{cfg}$ , reference diffusion model  $s_{ref}(\mathbf{x}_t|\mathbf{c}, \mathbf{c})$ , assistant diffusion  $s_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c})$ , forward diffusion  $p_t(\mathbf{x}_t|\mathbf{x}_0)$  (2.1), assistant diffusion updates rounds  $K_{TA}$ , time distribution  $\pi(t)$ , diffusion model weighting  $\lambda(t)$ , student loss time weighting  $w(t)$ .

**while not converge do**

freeze  $\theta$ , update  $\psi$  for  $K_{TA}$  rounds using SGD by minimizing

$$\mathcal{L}(\psi) = \mathbb{E}_{\substack{\mathbf{c} \sim \mathcal{C}, \mathbf{z} \sim p_z, t \sim \pi(t) \\ \mathbf{x}_0 = g_\theta(\mathbf{z}|\mathbf{c}), \mathbf{x}_t|\mathbf{x}_0 \sim p_t(\mathbf{x}_t|\mathbf{x}_0)}} \lambda(t) \|s_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0)\|_2^2 dt.$$

freeze  $\psi$ , update  $\theta$  using SGD by minimizing loss

$$\begin{aligned} \mathcal{L}_{DI*}(\theta) = & \mathbb{E}_{\substack{\mathbf{c} \sim \mathcal{C}, \mathbf{z} \sim p_z, \\ \mathbf{x}_0 = g_\theta(\mathbf{z}|\mathbf{c})}} \left\{ -\alpha_{rew} \cdot r(\mathbf{x}_0, \mathbf{c}) + \mathbb{E}_{\substack{t \sim \pi(t), \\ \mathbf{x}_t|\mathbf{x}_0 \sim p_t(\mathbf{x}_t|\mathbf{x}_0)}} \left[ \right. \right. \\ & -w(t) \{ \mathbf{d}'(s_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) - s_{ref}(\mathbf{x}_t|\mathbf{t}, \mathbf{c})) \}^T \{ s_\psi(\mathbf{x}_t|\mathbf{t}, \mathbf{c}) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{x}_0) \} \\ & \left. \left. + \alpha_{cfg} \cdot w(t) \{ s_{ref}(\text{sg}[\mathbf{x}_t]|\mathbf{t}, \mathbf{c}) - s_{ref}(\text{sg}[\mathbf{x}_t]|\mathbf{t}, \emptyset) \}^T \mathbf{x}_t \right] \right\} \end{aligned} \quad (\text{C.1})$$

**end**

**return**  $\theta, \psi$ .

**C.1. Pytorch style pseudo-code of Score Implicit Matching**

In this section, we provide a PyTorch-style pseudo-code for the algorithm below.

```
import torch
import torch.nn as nn
import torch.optim as optim
import copy

use_cfg = True
use_reward = True

# Initialize generator G
G = Generator()

## load teacher DM
Drf = DiffusionModel().load('/path_to_ckpt').eval().requires_grad_(False)
Dta = copy.deepcopy(Drf) ## initialize online DM with teacher DM
r = RewardModel() if use_reward else None

# Define optimizers
opt_G = optim.Adam(G.parameters(), lr=0.001, betas=(0.0, 0.999))
opt_Sta = optim.Adam(Dta.parameters(), lr=0.001, betas=(0.0, 0.999))

# Training loop
while True:
    ## update Dta
    Dta.train().requires_grad_(True)
    G.eval().requires_grad_(False)

    ## update assistant diffusion
    prompt = batch['prompt']
    z = torch.randn((1024, 4, 64, 64), device=G.device)
    with torch.no_grad():
        fake_x0 = G(z, prompt)

    sigma = torch.exp(2.0*torch.randn([1,1,1,1], device=fake_x0.device) - 2.0)
    noise = torch.randn_like(fake_x0)
```

```

fake_xt = fake_x0 + sigma*noise
pred_x0 = Dta(fake_xt, sigma, prompt)

weight = compute_diffusion_weight(sigma)

batch_loss = weight * (pred_x0 - fake_x0)**2
batch_loss = batch_loss.sum([1,2,3]).mean()

optimizer_Dta.zero_grad()
batch_loss.backward()
optimizer_Dta.step()

## update G
Dta.eval().requires_grad_(False)
G.train().requires_grad_(True)

prompt = batch['prompt']
z = torch.randn((1024, 4, 64, 64), device=G.device)
fake_x0 = G(z, prompt)

sigma = torch.exp(2.0*torch.randn([1,1,1,1], device=fake_x0.device) - 2.0)
noise = torch.randn_like(fake_x0)
fake_xt = fake_x0 + sigma*noise

with torch.no_grad():
    if use_cfg:
        cfg_vector = (Drf(fake_xt, sigma, prompt) - Drf(fake_xt, sigma, None))
    else:
        cfg_vector = None

    pred_x0_rf = Drf(fake_xt, sigma, prompt)
    pred_x0_ta = Dta(fake_xt, sigma, prompt)

denoise_diff = pred_x0_ta - pred_x0_rf
adp_wgt = torch.sqrt(denoise_diff.square().sum([1,2,3], keepdims=True) + phuber_c**2)
weight = compute_G_weight(sigma, denoise_diff)

# compute score regularization loss
batch_loss = weight * denoise_diff * (fake_D_yn - D_yn)/adp_wgt

# compute explicit reward loss if needed
if use_reward:
    reward_loss = -reward_scale * r(fake_x0, prompt)
    batch_loss += reward_loss

# compute cfg reward loss if needed
if use_cfg:
    cfg_reward_loss = cfg_scale * cfg_vector*fake_x0
    batch_loss += cfg_reward_loss

batch_loss = batch_loss.sum([1,2,3]).mean()

optimizer_G.zero_grad()
batch_loss.backward()
optimizer_G.step()

```

Listing 1: Pytorch Style Pseudo-code of Diff-Instruct\*