
Reward-Instruct: A Reward-Centric Approach to Fast Photo-Realistic Image Generation

Yihong Luo^{1,*}, Tianyang Hu^{2,*}, Weijian Luo³, Kenji Kawaguchi⁴, Jing Tang^{5,1†}
¹HKUST ²CUHK(SZ) ³Xiaohongshu Inc ⁴NUS ⁵HKUST(GZ)

Abstract

This paper addresses the challenge of achieving high-quality and fast image generation that aligns with complex human preferences. While recent advancements in diffusion models and distillation have enabled rapid generation, the effective integration of reward feedback for improved abilities like controllability and preference alignment remains a key open problem. Existing reward-guided post-training approaches targeting accelerated few-step generation often deem diffusion distillation losses indispensable. However, in this paper, we identify an interesting yet fundamental paradigm shift: as conditions become more specific, well-designed reward functions emerge as the primary driving force in training strong, few-step image generative models. Motivated by this insight, we introduce **Reward-Instruct**, a novel and surprisingly simple reward-centric approach for converting pre-trained base diffusion models into reward-enhanced few-step generators. Unlike existing methods, Reward-Instruct does not rely on expensive yet tricky diffusion distillation losses. Instead, it iteratively updates the few-step generator’s parameters by directly sampling from a reward-tilted parameter distribution. Such a training approach entirely bypasses the need for expensive diffusion distillation losses, making it favorable to scale in high image resolutions. Despite its simplicity, Reward-Instruct yields surprisingly strong performance. Our extensive experiments on text-to-image generation have demonstrated that Reward-Instruct achieves state-of-the-art results in visual quality and quantitative metrics compared to distillation-reliant methods, while also exhibiting greater robustness to the choice of reward function.

1 Introduction

High-quality, controllable, and fast image generation stands as a paramount goal in the field of Artificial Intelligence Generated Content (AIGC). Recent advancements in diffusion models [57, 18, 53] and, particularly, diffusion distillation [56, 28] have yielded impressive few-step image generators capable of rapid synthesis of photo-and-movie-realistic images and videos [47, 48, 68, 30, 32]. While these advancements have significantly improved generation speed and visual fidelity, how to achieve improved **controllability** and **alignment with complex human preferences** remains challenging.

Inspired by the success of reinforcement learning (RL) and more general reward-driven methodologies in large-language models [73, 36, 1], the image generation community has made considerable strides in developing effective *reward functions* for images. These rewards, broadly include any discriminative model capable of evaluating image quality or adherence to specific criteria, offer a promising avenue for guiding generative models towards desired attributes, such as human preferences, instruction following, as well as safety. Yet, the optimal strategies for effectively integrating and leveraging these reward signals in image generation workflows are still actively being investigated.

*Equal contribution.

†Corresponding author: Jing Tang.



Figure 1: 4-step samples at 1024 resolution generated by **Reward-Instruct**. The **Reward-Instruct** here is trained from SD3-medium purely by reward maximization.

Current standard practices for incorporating reward signals into image generation pipelines predominantly occur as a post-training stage. These methods generally fall into two main categories: (1). integrating reward optimization directly into a pre-trained diffusion model, often followed by a distillation process for efficient sampling, as exemplified by SDE control and other reward-based post-training techniques [13, 8]; (2). Developing methods that post-train some already distilled, fast sampling models by applying reward fine-tuning methods together with expensive diffusion distillation losses. Typical works on this line are Diff-Instruct++ (DI++) [27]. However, both lines of approaches are considered computationally demanding and not inherently designed for directly converting pre-trained base diffusion models into few-step, reward-enhanced generators. Furthermore, despite the seemingly good metrics on benchmarks, *close examination of state-of-the-art reward-enhanced few-step generators reveals evidence of reward hacking* — certain artifacts or repeated objects in the background, as illustrated in Fig. 2.

Achieving reward-enhanced fast image generation hinges on two key driving forces: the knowledge embedded within the base pre-trained generative model and the knowledge derived from reward signals. While language modeling has demonstrated the primacy of the latter, reward signals, such as reward-centric approaches remain underexplored in image modeling, as diffusion distillation losses are often considered indispensable in current works. With these observations, we are strongly motivated by an important scientific problem:

- *Can we develop a reward-centric training approach, that can result in fast generation speed without the need of tricky yet expensive diffusion distillation losses?*

In this paper, we give a positive answer to this question and introduce a reward-centric method for training few-step generative models. Our journey starts with an analysis in Section 3, which shows that the post-training of DI++ is largely driven by reward signals, rather than the diffusion distillation objective for preserving knowledge of the image distribution from pretrained base models. This points towards a *phase transition* to modern conditional generation tasks — the increasing specificity and diversity of desired conditions or reward signals are causing a fundamental shift in the generation process, moving away from primarily modeling the conditional distribution and towards regularized reward maximization.

Taking this perspective to its logical conclusion, we propose a novel reward-centric approach termed **Reward-Instruct (RI)**, which is capable of efficiently training few-step generative models using reward. Fundamentally differs from existing approaches [27, 23], our methods directly operate on the base diffusion model and performs a direct reward maximization with simple yet effective regularization without the explicit requirement of a separate distillation loss or training images. Specifically, we start from a few-step sampler from pretrained diffusion models with enhanced stochasticity (via random η sampling in Section 3.2.1) at each step. Our subsequent training process can be conceptually understood as directly sampling from a reward-tilted distribution within the



Figure 2: Samples are taken from the corresponding papers of DI++ and RG-LCM. It can be observed that certain artifacts exist in samples, e.g., repeated text/objects in the background. We hypothesize this comes from *reward hacking*.

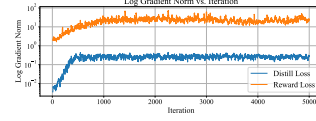


Figure 3: Log Gradient norm curve of DI++ through training process. We use the best configuration reported in DI++ [27].

generator’s parameter space. Such a simple reward-centric design significantly improves the training efficiency, training stability, resulting in state-of-the-art results in few-step text-to-image generation with high visual quality (Fig. 1). Particularly, our method outperforms previous methods that combine diffusion distillation and reward learning regarding visual quality and machine metrics (Fig. 5 and Table 1), while being more robust to the reward choice (Fig. 6).

2 Preliminary

Diffusion Models. Diffusion models (DMs) [53, 18, 57] define a forward diffusion process that $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$, where $\mathbf{x}$ is sampled from the data distribution, and σ_t specifies the noise schedule. By training a denoising function to predict the added noise, the model implicitly learns the score of the data distribution, enabling the generation of new samples by simulating a reverse stochastic differential equation or ordinary differential equation [57, 54, 25, 70, 65]. Conditional generation is often achieved through techniques like classifier-free guidance (CFG), which modulates the denoising process based on the desired conditions [57, 54, 25, 70, 65, 31]. As illustrated in [27], CFG can be seen as an implicit reward on condition alignment.

Preference Alignment with Diffusion Distillation. Currently several methods [27, 29, 23] have been proposed for developing preference align few-step text-to-image models. Their approach can be summarized as a combination of distillation loss and reward loss: $\min_{\theta} L(\theta) = L_{\text{distill}}(\mathbf{x}_{\theta}) - R(\mathbf{x}_{\theta}, c)$, where $\mathbf{x}_{\theta}$ denotes the model samples and $R(\mathbf{x}_{\theta}, c)$ denotes the reward measure the alignment between $\mathbf{x}_{\theta}$ and condition c . The distillation loss can be consistency distillation [23] or reverse-KL divergence [27]. The previous method either requires real data for training [23], or requires training an extra online score model [27]. These components increase the complexity of the post-training and serve solely as an overly complicated regularization.

3 Reward-Instruct: A Reward-Centric Approach to Image Generation

3.1 Discriminative vs Generative: The Phase Transition

The concerning phenomenon of *reward hacking*, as visually evidenced in Fig. 2, strongly suggests a potential imbalance where the generation process becomes excessively driven by reward models, potentially at the expense of image quality and generalization. To further verify this, our examination of the gradient norms of the reward loss and the distillation loss in our re-implementation of DI++ [27], shown in Figure 3, clearly indicates that updates in DI++ are overwhelmingly dominated by the reward term, relegating the diffusion distillation objective to a secondary role.

Extending this observation, we observe similar phenomena when considering a generalized concept of rewards that encompasses any discriminative model capable of judging the goodness of generated samples. In modern text-to-image generation models, various guidance modules for aligning conditions [19, 2, 33] are disproportionately amplified compared with the seemingly more important diffusion generation component. For instance, large CFG coefficients are indispensable (7.5 by default in Stable Diffusion [46] and 100 in DreamFusion [42]). When dealing with strong conditions, such as generating an image of “a red cube on top of a blue sphere behind a green pyramid”, a higher CFG value leads to more semantically compliant images.

These findings prompt a fundamental rethinking of conditional generation tasks with strong conditions, such as aligning to various reward functions in text-to-image generation. For these tasks, we usually model the conditional density $p(\mathbf{x}|\mathbf{y})$ through the decomposition $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x}) \cdot p(\mathbf{y}|\mathbf{x})$, where the terms correspond to the marginal density and discriminative model respectively. In diffusion

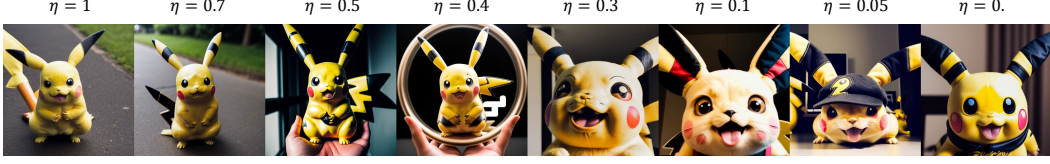


Figure 4: Samples with 100 NFE and 7.5 CFG by varying the η in sampling. The samples are generated from the same initial noise.

models, learning the marginal density $p(\mathbf{x})$ is usually the main focus while the condition part is usually handled by CFG or external guidance in diffusion models [2, 33]. However, as conditions get stronger and more complicated, the conditional distribution may be *ill-conditioned* to estimate, since the sample size for each condition is oftentimes only 1.

This leads us to the key insight: for tasks with strong conditions, there is a fundamental shift in the nature of the problem — moving away from directly estimating the conditional distribution towards what is more accurately described as **regularized rewards maximization**, where the discriminative part $p(y|\mathbf{x})$ captured by various rewards becomes the primary driving force, and the base generative model acts as a regularizer. In the following section, we will formally reformulate the problem of conditional generation with rewards.

3.2 Regularized Reward Maximization for Image Generation

Let $R(\mathbf{x})$ denote a reward function mapping $\mathbb{R}^d \rightarrow \mathbb{R}$. We can characterize the reward-centric generation task as searching in the *image domain* for samples that maximize rewards, which is naturally an optimization problem. To characterize the image domain constraint, consider having access to the image likelihood $p(\mathbf{x})$ for natural images. If $p(\mathbf{x})$ is below a certain threshold, $\mathbf{x}$ can be deemed outside the image domain. Therefore, we can rethink the generation as a regularized reward maximization problem with the following objective:

$$\max_{\mathbf{x} \in \mathbb{R}^d} R(\mathbf{x}) \text{ s.t. } p(\mathbf{x}) \leq c. \quad (3.1)$$

where $c > 0$ is some threshold. With this formulation, there are two important design choices to consider: (1) How to parameterize images? (2) How to effectively optimize the generator while preventing reward hacking? We will demonstrate in later sections that with a well-parameterized generator and proper regularization, we can achieve state-of-the-art text-to-image generation with only a few reward functions, without relying on diffusion distillation.

3.2.1 Parameterizing Images with Generator

To achieve fast image generation, a natural strategy is to utilize GAN-style generators that directly transform random noise into images. With this formulation, the optimization problem shifts from the image space to the generator’s parameter space. Ideally, the generator should have large-enough capacity and a reasonable initial grasp of the image distribution. We address this by proposing the use of an unrolled few-step sampler directly from pre-trained diffusion models. Conveniently, the number of unrolled steps allows us to control both the generator’s capacity and its initial knowledge.

Specifically, we parameterize the generator $g_{\theta, \eta}$ to accept K noisy levels of a diffusion model as inputs, creating a network that progresses from noise to obtain clean samples in K steps. The specific parameterization is as follows:

$$g_{\theta, \eta}(\mathbf{z}) = g_{\theta, \sigma_1, \eta} \circ g_{\theta, \sigma_2, \eta} \dots \circ g_{\theta, \sigma_K, \eta}(\mathbf{z}), \quad g_{\theta, \sigma_k, \eta}(\mathbf{x}_k) = \sqrt{1 - \sigma_{k-1}^2} \frac{\mathbf{x}_k - \sigma_k \epsilon_{\theta}(\mathbf{x}_k)}{\sqrt{1 - \sigma_k^2}} + \sigma_{k-1} \hat{\epsilon}_{\eta} \quad (3.2)$$

where $\hat{\epsilon}_{\eta} = \eta \epsilon_{\theta}(\mathbf{x}_k) + \sqrt{1 - \eta^2} \epsilon$, $\sigma_K := 1$ and $\mathbf{z}, \epsilon \sim \mathcal{N}(0, I)$. We can use a pre-trained score net as the initialization for enough model capacity and better initial images. Our optimization target is θ , which can be the full score network parameters or low-rank adaptation of them (LoRA) [20].

Random η -Sampling. The η plays an important role in our parameterized generator $g_{\theta, \eta}$. In particular, when $\eta = 1$, the generator is parameterized into the discrete format of DDIM sampler [54]. In practice, we find that varying η in diffusion sampling results in significant differences in the style

and layout of the generated images, as shown in Fig. 4. Motivated by this, we propose to parameterize our generator by inputting random η at each step for augmenting the generator distribution that allows the generator explore more diverse area:

$$g_{\theta, \eta}(z) = g_{\theta, \sigma_1, \eta_1} \circ g_{\theta, \sigma_2, \eta_2} \dots \circ g_{\theta, \sigma_K, \eta_K}(z), \quad (3.3)$$

where $\eta_i \sim U[0, 1]$ and $i = 1, 2, \dots, K$. After training, we can fix a $\eta \in [0, 1]$ in sampling. The design can effectively augment the generator distribution by randomly adding stochastic, serving as an effective regularization to enhance performance.

3.2.2 Optimization as Sampling from Reward-tilted Distribution

The generator $g_{\theta, \eta}(z)$ initialized from the base diffusion model provides a non-trivial starting point for reward-centric training. We aim to adjust the parameter θ such that generated images are more preferred by the rewards. Formally, we can define the target θ^* to be distributed following a reward-informed posterior distribution

$$p^*(\theta) \propto p_0(\theta) \exp\{\bar{R}(\theta)\},$$

where $\bar{R}(\theta) = \mathbb{E}_{z, \eta}[R(g_{\theta, \eta}(z))]$ and $p_0(\theta)$ is the initial distribution from the base diffusion models, which can be viewed as the likelihood to specify the image domain likelihood. For simplicity, we choose $p_0(\theta)$ to be a Gaussian distribution centered in the pretrained θ_0 with variance $\sigma^2 I$. More advanced prior distributions will definitely lead to better performance.

Remark 3.1. The above formulation is similar to existing works in reward-driven approaches [40, 39, 51, 44, 8]. Given a base generative model with base distribution $p_0(x)$, the reward-tilted target is usually defined as $p^*(x) \propto p_0(x) \exp\{R(x)\}$. The key difference is that our formulation is on the generator parameter space.

With the target distribution p^* specified, a straight-forward method to obtain $\theta^* \sim p^*(\theta)$ using stochastic Langevin dynamics. Concretely, starting from $\theta^{(0)} = \theta_0$, $\theta^{(t+1)}$ can be iteratively calculated via the stochastic gradient Langevin dynamics (SGLD) [59]

$$\begin{aligned} \theta^{(t+1)} - \theta^{(t)} &= \lambda \nabla \log \left(p_0(\theta^{(t)}) \exp\{\bar{R}(\theta^{(t)})\} \right) + \sqrt{2\lambda} \epsilon_t \\ &= \underbrace{\lambda \nabla \bar{R}(\theta^{(t)})}_{\text{Rewards maximization}} - \underbrace{\frac{\lambda}{2\sigma^2} \nabla \|\theta^{(t)} - \theta_0\|_2^2}_{\text{Regularization}} + \sqrt{2\lambda} \epsilon_t, \end{aligned} \quad (3.4)$$

where λ is the learning rate and $\epsilon_t \sim N(0, 1)$ is a random noise. As can be seen above, each update constitutes of reward maximization and regularization. The regularization is also two-fold, one being an l_2 -penalty or weight decay, the other one being random noise perturbation.

3.3 Elucidating the Design Space of Reward-Instruct

By incorporating all aforementioned designs, we have developed a surprisingly simple yet effective reward-centric approach to fast image generation. Rooted in regularized reward maximization, we call our method Reward-Instruct and it is a framework that directly converts pre-trained base diffusion models into reward-enhanced few-step generators. Detailed algorithms are summarized in Algorithm 1 in the Appendix.

Such a simple reward-centric design significantly improves the training efficiency, training stability, resulting in state-of-the-art results in few-step text-to-image generation. Fig. 5 illustrates the qualitative comparisons among other methods. Detailed comparison for computation cost are deferred to Appendix B.1. Next, We proceed to explore the various facets of its design space.

3.3.1 Form of Regularization

The primary role of regularization is to constrain the generator’s distribution to the vicinity of the image manifold, thus avoiding reward hacking. In our formulation, the specific form of the θ -regularization is dependent on the form of $p_0(\theta)$. Choosing Gaussian distribution will give rise to l_2 regularization. This is very similar to the KL penalty in RL to control the update to be not too large [36]. The random noises introduced in the SGLD sampling algorithm also provide a regularization effect against reward hacking. Its effect is ablated in Appendix B.2.

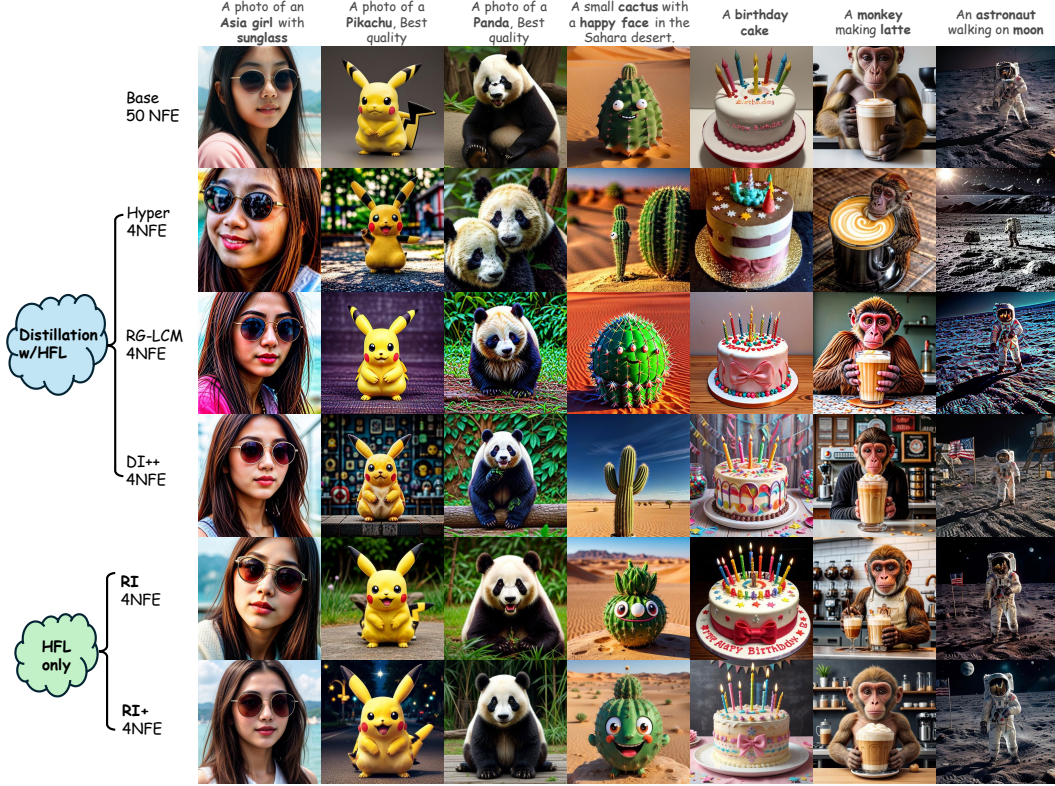


Figure 5: Qualitative comparisons of Reward-Instruct against distillation based and diffusion based models on SD-v1.5 backbone. All images are generated by the same initial noise. We surprisingly observe that our proposed Reward-Instruct has better image quality and text-image alignment compared to prior distillation-based reward maximization methods in 4-step text-to-image generation.

Diffusion distillation as regularization. More closely related to our generator parameterization, existing works such as RG-LCM [23] and DI++ [27] introduce reward maximization into diffusion distillation. These methods typically require training an additional score model for the distilled generator to ensure its closeness to the original model, which is memory and computation-intensive. Surprisingly, we found that the reward gradient dominates throughout the training process, turning diffusion distillation into a sort of costly regularization role (Fig. 3). Specifically, the RG-LCM and DI++ employ HPS v2.0 [61] or Image Reward [62] as the reward loss in their original approach, but we discovered that if the reward is set to HPS v2.1, it would cause the generators of RG-LCM and DI++ to collapse into undesirable distributions, as shown in Fig. 6.

Built upon the above formulation and regularization techniques, we found that our method integrated with HPS v2.1 can generate high-quality images, suffering less from artifacts compared to RG-LCM and DI++ (Fig. 7), where each regularization effectively improves image quality. This indicates the importance of proper regularization in reward maximization. However, we found that optimization with a single explicit reward still suffers from the over-saturation issue. To address this, we suggest maximizing multiple explicit rewards. This indicates the importance of proper regularization in reward maximization.

3.3.2 Power of Multiple Rewards

Thanks to a plethora of preference data and powerful RL methods, we have access to a diverse collection of learned reward models. Although learning from a zoo of pre-trained models has long been studied in various vision tasks [60, 9, 4, 10], utilizing multiple reward models is relatively underexplored. Denote $R_1(x), \dots, R_m(x)$ as the reward functions, each with its own set of modes, some genuinely good and some corresponding to artifacts. Our assumption is that the good ones are associated with the ground truth, while the bad ones are random and not shared with other reward

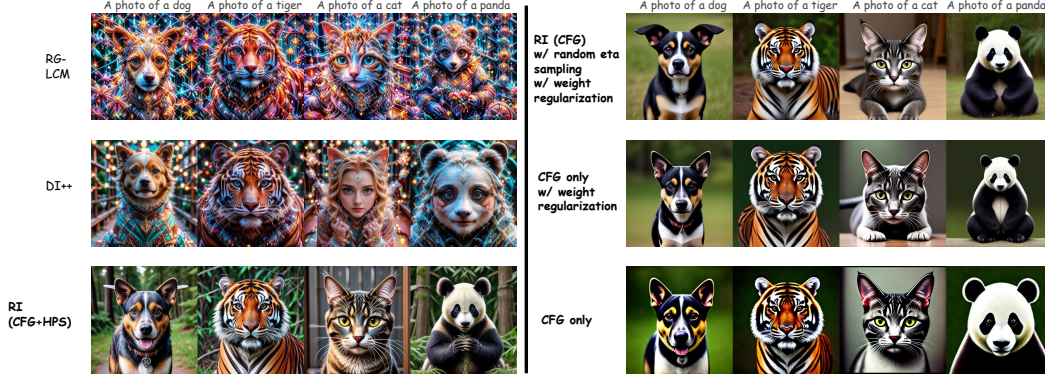


Figure 6: The prior distillation-based reward maximization methods collapse when the reward is chosen to be HPS v2.1. In contrast, our Reward-Instruct still works well, benefiting from the proposed effective regularization technique.

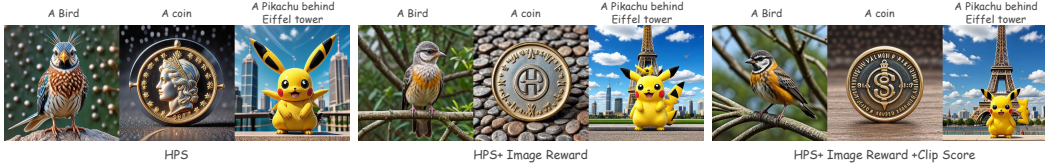


Figure 7: Four-step samples generated by Reward-Instruct. We observed that the quality of the image monotonically increases with the gradual increment of the reward count.

functions. Therefore, with a diverse collection of different reward functions focusing on different aspects of the data, images preferred by all of them tend to be good.

Therefore, the goal of utilizing multiple rewards is to find common data modes that have high rewards from different perspectives. This design can effectively prevent the model from hacking each individual reward, therefore significantly stabilizing the training process. We can extend (3.1) to be

$$\max_{\mathbf{x} \in \mathbb{R}^d} \sum_{i=1}^m \omega_i R_i(\mathbf{x}) \text{ s.t. } p(\mathbf{x}) \leq c, \quad (3.5)$$

where ω_i 's are positive weights for balancing the rewards. However, all we have is a gradient pointing towards the direction of the steepest climb. How to find the most effective direction is of critical importance. In practice, we found that optimizing multiple rewards with a naive *weighted combination* may fail to maximize all rewards in the training process. As shown in Fig. 9, the clip score does not converge to a high value.

To balance the learning of different rewards, we suggest *gradient normalization*, normalizing the gradients from each reward and forming the average direction. Therefore, we can choose the weights ω_i in (3.1) as

$$\omega_i = \hat{\omega}_i / \text{sg}(\|\frac{\partial R_i(g(\mathbf{z}))}{\partial g(\mathbf{z})}\|_2). \quad (3.6)$$

This is equal to setting a dynamic weighting. Note that the normalizing operation is also performed for the implicit reward (CFG). Fig. 9 shows that after applying the gradient normalization, we can maximize multiple rewards well. Fig. 7 demonstrates that the image quality and image-text alignment become significantly better when we maximize multiple rewards.

The Complementary Effect of Rewards. Since the common modes of multiple rewards tend to be more well-behaved than those from signal rewards, the combination of multiple rewards in (3.6) also serves as a kind of implicit regularization for the generator. To verify this, we maximize HPS, image reward, and clip score separately, without using random- η sampling. We find that individual rewards perform poorly, but when combined, they can generate images of reasonable quality as shown in Fig. 8, which highly emphasizes the complementarity between rewards and their effectiveness as implicit regularization.

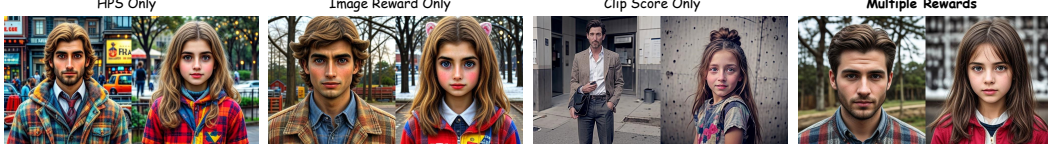


Figure 8: The **complementary effect** of different reward. We do not use random η sampling and set small weight regularization in training here to highlight the complementary effect between rewards.

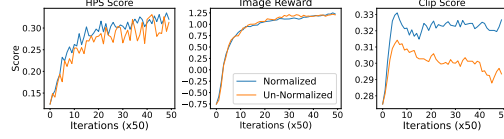


Figure 9: Training progress of various metrics over iterations. It can be seen that the normalized gradient shows better performance. This is evaluated on 1k prompts from HPS benchmark.

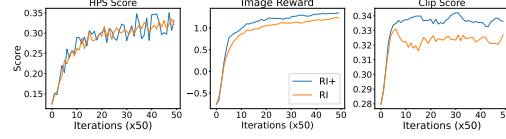


Figure 10: Comparison on the convergence speed between RI and RI+. We note that each training iteration of RI+ takes only 65% of the time required by RI.

3.4 Reward-Instruct+: Additional Supervision on Intermediate Steps

The reward supervision in Reward-Instruct is only at the final generated samples in an end-to-end fashion, which is similar to the DeepSeek-R1 [7]. To further enhance performance, we can incorporate extra supervision into intermediate generation steps to form Reward-Instruct+.

Although Reward-Instruct is already capable of generating high-quality images in a setting where only reward maximization is considered, the reward signal is only provided at the end. This leads to two issues: on one hand, the efficiency of reward maximization is low because there is no direct reward signal feedback during the process; on the other hand, the gradient needs to be backpropagated through the entire generator, resulting in high memory usage and significant computational costs.

To address the above issues, we propose learning the generator with intermediate supervision as well, forming Reward-Instruct+. We rewrite the generator as follows:

$$\begin{aligned} \mathbf{x}_{k+1} &= \text{sg}(g_{\theta, \sigma_{k+2}, \eta_{k+2}} \circ g_{\theta, \sigma_{k+3}, \eta_{k+3}} \cdots \circ g_{\theta, \sigma_K, \eta_K}(\mathbf{z})), \\ \mathbf{x}_k &= \sqrt{1 - \sigma_k^2} \frac{\mathbf{x}_{k+1} - \sigma_{k+1} \epsilon_{\theta}(\mathbf{x}_{k+1})}{\sqrt{1 - \sigma_{k+1}^2}} + \sigma_k \hat{\epsilon}_{\eta_k}, \quad \mathbf{x}_0^{(k)} = \frac{\mathbf{x}_{k+1} - \sigma_{k+1} \epsilon_{\theta}(\mathbf{x}_{k+1})}{\sqrt{1 - \sigma_{k+1}^2}} \end{aligned} \quad (3.7)$$

where $k = 1, 2, \dots, K$, $\eta_i \sim U[0, 1]$, $\hat{\epsilon}_{\eta_k} = \eta_k \epsilon_{\theta}(\mathbf{x}_{k+1}) + \sqrt{1 - \eta_k^2} \epsilon$, $\sigma_K := 1$ and $\mathbf{z}, \epsilon \sim \mathcal{N}(0, I)$. For computing CFG, we diffuse samples from $\mathbf{x}_k$. The k is randomly sampled during training. The update of θ has the same form as Eq. (3.4), differing in how to obtain samples.

By doing so, we can effectively supervise the intermediate process in the generation process. Fig. 10 shows that after applying the intermediate supervision, the convergence speed of Reward-Instruct+ has significantly improved compared to Reward-Instruct. Moreover, training RI+ is more efficient than RI, since each training iteration of RI+ takes only 65% of the time required by RI.

Comparison on Generation Path Between Reward-Instruct and Reward-Instruct+. Reward-Instruct and Reward-Instruct+ are significantly different and it is interesting to explore the generation paths corresponding to these two models. In Fig. 16, we can observe that Reward-Instruct+ generates much clearer images during the early process compared to Reward-Instruct. Although Reward-Instruct+ struggles with severe artifacts in the early stages of generation, surprisingly, these artifacts are gradually removed rather than accumulating throughout the process. In contrast, Reward-Instruct's path progresses from blurry to clear, with less affected by artifacts during the process.

4 Evaluations

To verify the effectiveness of the Reward-Instruct and Reward-Instruct+, we compare them with previous distillation-based reward maximization methods. We put the experiment details in the Appendix.

Table 1: Comparison of machine metrics on text-to-image generation across SOTA methods. We **highlight** the best among fast sampling methods. The FID is measured based on COCO-5k dataset.

Model	Type	Backbone	NFE	HPS↑	Aes↑	CS↑	FID↓	Image Reward↑
Base Model (Realistic-vision)		SD-v1.5	50	30.19	5.87	34.28	29.11	0.81
Hyper-SD [45]	Distillation + Reward	SD-v1.5	4	30.24	5.78	31.49	30.32	0.90
RG-LCM [23]		SD-v1.5	4	31.44	6.12	29.14	52.01	0.67
DI++ [27]		SD-v1.5	4	31.83	6.09	29.22	55.52	0.72
ReFL [62]	Reward Centric	SD-v1.5	50	31.82	5.97	31.78	39.38	1.16
DRaFT [5]		SD-v1.5	50	33.10	6.18	30.70	37.10	0.85
Reward-Instruct (Ours)	Reward Centric	SD-v1.5	4	33.70	6.11	32.13	33.79	1.22
Reward-Instruct+ (Ours)		SD-v1.5	4	34.37	6.20	32.97	37.53	1.27
Base Model	Reward Centric	SD3-Medium	56	31.37	5.84	34.13	28.72	1.07
Reward-Instruct (Ours)		SD3-Medium	4	34.04	6.27	33.89	31.97	1.13



Figure 11: Qualitative comparison against competing methods and applications in downstream tasks.

Table 3: Ablation study on the proposed components in our Reward-Instruct.

Model	Backbone	Steps	HPS↑	Aes↑	CS↑	FID↓	Image Reward↑
Reward-Instruct (Ours)	SD-v1.5	4	33.70	6.11	32.13	33.79	1.22
Reward-Instruct w/ single reward	SD-v1.5	4	32.08	5.80	31.01	38.21	0.89
Reward-Instruct w/o random- η -sampling	SD-v1.5	4	33.41	6.12	32.11	36.90	1.18
Reward-Instruct w/o weight regularization	SD-v1.5	4	34.26	6.12	32.45	39.25	1.27

Metric. We assess image quality using the Aesthetic Score (AeS) [50], image-text alignment and human preference with the Human Preference Score (HPS) v2.1 [61] and Image Reward, and image-text alignment with the CLIP score (CS) [16]. Additionally, we use zero-shot FID on the COCO-5k dataset for a more comprehensive evaluation.

Qualitative Results. We present the qualitative results in Fig. 5. It can be observed that our proposed Reward-Instruct and Reward-Instruct+, without using a distillation loss, demonstrate better image quality and text-image alignment compared to existing distillation-based reward maximization methods and RL-based finetuning methods. We also include additional visualization regarding complex prompts in Fig. 15.

User Study. To further verify the effectiveness of our proposed method without concern for reward-hacking, we conduct a user study on SD-v1.5 backbone. We refer to Section B.2 for details of the user study. The results in Table 2 show that our method outperforms the base diffusion model.

Table 2: User Preference Study.

Model	NFE	User Preference↑
Base Model	50	41.7%
RI (Ours)	4	58.3%

Quantitative Results. We present the quantitative results in Table 1. Our proposed Reward-Instruct and Reward-Instruct+ achieve state-of-the-art (SOTA) performance across various text-image alignment and human preference metrics. Notably, our model also achieves a zero-shot COCO FID comparable to the original model, which demonstrates that our method does not suffer from artifacts.

Ablation Study. We provide quantitative results about removing one technique at a time to show the effectiveness. The results in Table 3 demonstrate the critical role of each component: 1) Multiple rewards: Improves both rewards and FID, mitigating artifacts and reward hacking. 2) Random-sampling: Maintains similar reward performance but significantly improves FID, aiding to find better mode with fewer artifacts. 3) Weight regularization: Trades slight reward gains for better FID, ensuring the generator stays within the image manifold.

Additional Application. We show our Reward-Instruct’s capabilities in various tasks: **1) Image-to-Image Editing:** As shown in Fig. 11, Reward-Instruct performs high-quality image editing [35] in

just four steps; **2) Compatibility with ControlNet and Base Models:** Illustrated in Fig. 11, Reward-Instruct-LoRA is compatible with ControlNet [69] and works seamlessly with various fine-tuned base models (e.g., Dreamshaper from SD 1.5), preserving their unique styles.

5 Discussion

Our work targets a reward-centric approach to photo-realistic image generation. Our proposed Reward-Instruct demonstrates that via regularized reward maximization, we can convert pretrained base diffusion models to reward-aligned few-step generators, without diffusion distillation losses or training images. Our method enables 4-step 1024px generation (Fig. 1), matching or exceeding both the inference speed and sample quality of previous approaches. In Appendix C, we further explore more application scenarios and extensions where our method can shine.

Our approach is inspiring but have its limitations. For instance, our method relies on trained differentiable reward functions. It’s an interesting future work to extend such a reward-centric approach to include black-box reward functions. It would be exciting to explore DPO-style [44] pipelines that directly fine-tune generators with raw preference data, and extending these principles to other generative domains like video and 3D content. Moreover, our Reward-Instruct framework can potentially work with any few-step generator and doesn’t have to relate to diffusion models but this is not explored in the current work. Making this investigating is an important extension.

Acknowledgments

Jing Tang’s work is partially supported by National Key R&D Program of China under Grant No. 2024YFA1012700 and No. 2023YFF0725100, by the National Natural Science Foundation of China (NSFC) under Grant No. 62402410 and No. U22B2060, by Guangdong Provincial Project (No. 2023QN10X025), by Guangdong Basic and Applied Basic Research Foundation under Grant No. 2023A1515110131, by Guangzhou Municipal Science and Technology Bureau under Grant No. 2024A04J4454, by Guangzhou Municipal Education Bureau (No. 2024312263), and by Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628) and Guangzhou Municipal Key Laboratory of Financial Technology Cutting-Edge Research (No. 2024A03J0630).

References

- [1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- [2] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.
- [3] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.
- [4] Yimeng Chen, Tianyang Hu, Fengwei Zhou, Zhenguo Li, and Zhi-Ming Ma. Explore and exploit the diverse knowledge in model zoo for domain generalization. In *International Conference on Machine Learning*, pages 4623–4640. PMLR, 2023.
- [5] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.
- [6] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023.
- [7] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

- [8] Carles Domingo-Enrich, Michal Drozdal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.
- [9] Qishi Dong, Awais Muhammad, Fengwei Zhou, Chuanlong Xie, Tianyang Hu, Yongxin Yang, Sung-Ho Bae, and Zhenguo Li. Zood: Exploiting model zoo for out-of-distribution generalization. *Advances in Neural Information Processing Systems Volume 35*, 2022.
- [10] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pages 8489–8510. PMLR, 2023.
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.
- [12] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization, 2024.
- [13] Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.
- [14] 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.
- [15] Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.
- [16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.
- [17] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [19] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [21] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.
- [22] 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.
- [23] Jiachen Li, Weixi Feng, Wenhui Chen, and William Yang Wang. Reward guided latent consistency distillation. *arXiv preprint arXiv:2403.11027*, 2024.
- [24] Xuanton Liu, Tianyang Hu, Wenjia Wang, Kenji Kawaguchi, and Yuan Yao. Referee can play: An alternative approach to conditional generation via model inversion. *arXiv preprint arXiv:2402.16305*, 2024.

- [25] 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. *arXiv preprint arXiv:2206.00927*, 2022.
- [26] Weijian Luo. A comprehensive survey on knowledge distillation of diffusion models. *arXiv preprint arXiv:2304.04262*, 2023.
- [27] Weijian Luo. Diff-instruct++: Training one-step text-to-image generator model to align with human preferences. *arXiv preprint arXiv:2410.18881*, 2024.
- [28] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36, 2023.
- [29] Weijian Luo, Colin Zhang, Debing Zhang, and Zhengyang Geng. Diff-instruct*: Towards human-preferred one-step text-to-image generative models. *arXiv preprint arXiv:2410.20898*, 2024.
- [30] Yihong Luo, Xiaolong Chen, Xinghua Qu, Tianyang Hu, and Jing Tang. You only sample once: Taming one-step text-to-image synthesis by self-cooperative diffusion gans, 2024.
- [31] Yihong Luo, Tianyang Hu, Yifan Song, Jiacheng Sun, Zhenguo Li, and Jing Tang. Adding additional control to one-step diffusion with joint distribution matching. *arXiv preprint arXiv:2503.06652*, 2025.
- [32] Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. *arXiv preprint arXiv:2503.06674*, 2025.
- [33] Jiajun Ma, Tianyang Hu, Wenjia Wang, and Jiacheng Sun. Elucidating the design space of classifier-guided diffusion generation. *arXiv preprint arXiv:2310.11311*, 2023.
- [34] Jiajun Ma, Shuchen Xue, Tianyang Hu, Wenjia Wang, Zhaoqiang Liu, Zhenguo Li, Zhi-Ming Ma, and Kenji Kawaguchi. The surprising effectiveness of skip-tuning in diffusion sampling. *arXiv preprint arXiv:2402.15170*, 2024.
- [35] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- [36] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [37] Junting Pan, Keqiang Sun, Yuying Ge, Hao Li, Haodong Duan, Xiaoshi Wu, Renrui Zhang, Aojun Zhou, Zipeng Qin, Yi Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Journeymb: A benchmark for generative image understanding, 2023.
- [38] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.
- [39] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- [40] 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.
- [41] 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.
- [42] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

- [43] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.
- [44] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024.
- [45] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis, 2024.
- [46] 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.
- [47] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.
- [48] Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *arXiv preprint arXiv:2406.04103*, 2024.
- [49] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.
- [50] 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.
- [51] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [52] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2024.
- [53] 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.
- [54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [55] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*.
- [56] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.
- [57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.
- [58] 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.
- [59] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

- [60] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.
- [61] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.
- [62] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 15903–15935, 2023.
- [63] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. 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*, pages 8196–8206, 2024.
- [64] Shuchen Xue, Zhaoqiang Liu, Fei Chen, Shifeng Zhang, Tianyang Hu, Enze Xie, and Zhenguo Li. Accelerating diffusion sampling with optimized time steps. *arXiv preprint arXiv:2402.17376*, 2024.
- [65] Shuchen Xue, Mingyang Yi, Weijian Luo, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhi-Ming Ma. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:77632–77674, 2023.
- [66] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihang 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.
- [67] Zilyu Ye, Zhiyang Chen, Tiancheng Li, Zemin Huang, Weijian Luo, and Guo-Jun Qi. Schedule on the fly: Diffusion time prediction for faster and better image generation. *arXiv preprint arXiv:2412.01243*, 2024.
- [68] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2023.
- [69] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.
- [70] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.
- [71] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.
- [72] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *International Conference on Machine Learning*, 2024.
- [73] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

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

Answer: [Yes]

Justification: The introduction part outlines the paper’s structure. The abstract summarizes the main body of this work.

Guidelines:

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

2. Limitations

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

Answer: [Yes]

Justification: We have discussed our limitations in the Discussion section, and those limitations leads to future work.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [\[Yes\]](#)

Justification: This work doesn't contain any theorem. However, for math derivations, necessary details are included.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [\[Yes\]](#)

Justification: In the main part, we discussed the algorithm and experiment setup. More details for reproducibility is included in the appendix. Furthermore, we will release codes if got accepted.

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: We will provide code and data for reproduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

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

Answer: [\[Yes\]](#)

Justification: The experiment settings and details are included in the appendix.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [\[No\]](#)

Justification: Most experiments are very expensive to run and do not have multiple runs.

Guidelines:

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

8. Experiments compute resources

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

Answer: [\[Yes\]](#)

Justification: Information on the computer resources is discussed in detail in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.

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

9. Code of ethics

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

Answer: [Yes]

Justification: The research conducted in this paper conform with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

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

Answer: [Yes]

Justification: We discuss it in the appendix.

Guidelines:

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

11. Safeguards

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

Answer: [Yes]

Justification: We discussed it in the appendix.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [\[Yes\]](#)

Justification: All existing assets used in this paper have been properly credited.

Guidelines:

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

13. New assets

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

Answer: [\[NA\]](#)

Justification: We do not release new assets in the submission phase.

Guidelines:

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

14. Crowdsourcing and research with human subjects

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

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

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

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

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

16. Declaration of LLM usage

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Additional Related Works

Few-Step Diffusion Sampling. Despite significant advancements in training-free accelerated sampling of DMs [25, 71, 64, 52, 34], diffusion distillation [26] is dispensable for satisfactory few-step Sampling. Typically, the distilled sampler involves a single or multiple transformations from random noise to images. Among various approaches, trajectory matching [56, 21, 55, 15, 48] and distribution matching [68, 28, 72, 49, 63, 30] are the most popular methods for diffusion distillation in few-step diffusion sampling. Very recently, trajectory distribution matching [32] has shown its promising performance in distillation.

Preference Alignment for Text-to-Image Models. In recent years, significant efforts have been made to align diffusion models with human preferences. These approaches can be broadly categorized into three main lines of work: 1) fine-tuning DMs on carefully curated image-prompt datasets [6, 41]; 2) maximizing explicit reward functions, either through multi-step diffusion generation outputs [43, 5, 22, 17] or policy gradient-based reinforcement learning (RL) methods [14, 3, 67]. 3) implicit reward maximization, exemplified by Diffusion-DPO [58] and Diffusion-KTO [66], directly utilizes raw preference data without the need for explicit reward functions.

Reward driven image generation. There is an active line of research investigating using various rewards in the post-training process for improved image alignment. For instance, [13] proposed to fine-tune DDPM samplers via policy gradient, achieving good few-step sampling performance. [8] considered reward fine-tuning diffusion models or flow models via stochastic optimal control (SOC), and proposed Adjoint Matching which outperforms existing SOC algorithms. ReNO [12] optimizes the initial noise by maximizing multiple rewards for enhanced performance given a frozen one-step generator. Our Reward-Instruct significantly differs from existing works in that it is the first reward-driven few-step image generation method that directly converts pretrained diffusion models to reward-enhanced few-step generators, without relying on complicated diffusion distillation or training data.

Another closely related work is “Referee can play” [24], where the authors emphasized the importance of discriminative models in conditional generation and presented a text-to-image generation pipeline by inverting Vision Language Models (VLMs). Specifically, they utilized the decoder from stable diffusion and optimized the latent for maximizing the alignment score given by VLMs. On a high-level, the VLMs employed in [24] can also be viewed as reward models, providing matching scores for text-image pairs. Even though [24] provides an interesting proof-of-concept demonstration, it lacks a systematic formulation and its generating process has major downsides in real-world scenarios. (1) The optimization process is highly sensitive to initialization and hyperparameter choices, which is not robust. (2) To generate a new image, they have to do hundreds of function evaluations (NFEs), which is computationally intensive. Visual comparisons are illustrated in Figure 12.



Figure 12: Comparison with **Referee can play** [24]. The baseline samples are taken from their paper. It can be seen that our Reward-Instruct has significantly better visual quality.

Algorithm 1 Reward-Instruct

Require: Generator f_θ , Pre-trained score f_ψ , Reward models $\{r_i\}$, desired sampling steps K , total iterations N , learning rate λ .

Ensure: optimized generator f_θ .

```
1: Initialize weights  $\theta$  by  $\psi$ ;
2: for  $i \leftarrow 1$  to  $N$  do
3:   Sample noise  $\epsilon$  from standard normal distribution;
4:   Sample  $\mathbf{x}$  with initialized noise  $\epsilon$  from generator  $f_\theta$  by  $K$  steps via random- $\eta$  sampling.
5:   # Compute regularization loss
6:    $\mathcal{L}_{reg} \leftarrow \|\theta - \psi\|_2^2$ 
7:   # Compute Rewards
8:    $\mathcal{L}_{reward} \leftarrow -\sum_i \frac{\hat{\omega}_i}{\text{sg}(\|\frac{\partial R_i(\mathbf{x})}{\partial \mathbf{x}}\|_2)} R_i(\mathbf{x}, y)$ 
9:   # Compute CFG guidance loss
10:  Sample  $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x})$ 
11:   $\text{cfg\_grad} \leftarrow \nabla_{\mathbf{x}_t} \log p_\psi(c|\mathbf{x}_t)$ 
12:   $\mathcal{L}_{cfg} \leftarrow \|\mathbf{x}_t - \text{sg}(\mathbf{x}_t + \text{cfg\_grad})\|_2^2$ 
13:  # Compute Total Loss and Update
14:   $\mathcal{L}_{total} \leftarrow \omega_{reg}\mathcal{L}_{reg} + \mathcal{L}_{reward} + \omega_{cfg}\mathcal{L}_{cfg}$ 
15:  Update  $\theta$  using SGLD step with temperature  $\tau$ :  $\theta \leftarrow \theta - \frac{\lambda}{\tau} \nabla_\theta \mathcal{L}_{total} + \sqrt{2\lambda}\epsilon$ .
16: end for
```

Listing 1: Torch-style pseudo code of SGLD step with temperature τ .

```
1 loss.backward()
2 noise_scale = (2 * learning_rate * tau) ** 0.5 / learning_rate
3 for param in model.parameters():
4     with torch.no_grad():
5         if param.grad is not None:
6             noise_para = torch.randn_like(param.grad.data) *
              noise_scale
7             param.grad.data.add_(noise_para)
8 optimizer.step()
9 optimizer.zero_grad()
```

B Experiment Details

Baseline Models. We perform experiments on SD-v1.5 [46] and SD3-medium [11], including both UNet and MM-DiT [11, 38] architectures, indicating the broad applicability of our approach.

Experiment Setting. Training is performed on the JourneyDB dataset [37] using prompts, without requiring images, as our method is image-free. We primarily compare with previous distillation-based reward maximization methods. For Hyper-SD, we use the public checkpoint, while for RG-LCM and DI++, we reproduce their methods. We adopt the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and the learning rate of $2e - 5$. We use a batch size of 256.

Details of User Study. We randomly selected 20 prompts for image generation. Around 20 user responses are collected on 20 pairs in total. We note that multiple users review the same pair of images, which potentially reduces evaluation bias.

B.1 Training Efficiency of Reward-Instruct

Our proposed Reward-Instruct is efficient to train, since it does not require training images and does not require online auxiliary models during training. And its overall training cost is also not expensive compared to multi-step RL-based methods and distillation-based RL methods, as shown in the Table 4.

Method	Few-step	Image-Free	Single Trainable Network	Training Cost
DRaFT	✗	✓	✓	72 Hours
ReFL	✗	✓	✓	64 Hours
RG-LCM	✓	✗	✓	20 Hours
DI++	✓	✓	✗	36 Hours
Reward-Instruct (Ours)	✓	✓	✓	28 Hours

Table 4: Comparison on the training efficiency. Our method is efficient for training and deploying in various aspects. The training cost is measured by GPU hours on RTX-4090. We use the same batch size and iterations for each method.



Figure 13: The effect of regularization loss. Images are from the same initial noise.



Figure 14: The effect of SGLD update. Images are from the same initial noise.

Table 5: Comparison between Reward-Instruct using different rewards.

Model	Backbone	Steps	HPS↑	Aes↑	CS↑	FID↓	Image Reward↑
Reward-Instruct (Ours)	SD-v1.5	4	33.70	6.11	32.13	33.79	1.22
Reward-Instruct w/ HPS + Clip Score + AeS (Ours)	SD-v1.5	4	32.35	6.21	32.43	31.26	1.01

B.2 Additional Ablation

The Indispensability of Weight Regularization. The regularization loss is the key to learning a good generator. Without weight regularization, even if we employ multiple rewards, the generator can easily converge to an undesirable distribution, which underscores the importance of explicitly regularizing the generator, as shown in Fig. 13.

The Indispensability of SGLD’s randomness. The SGLD randomness also serves as a key to regularize the generator. We observe that by introducing SGLD’s randomness in learning, the model is more robust to avoid artifacts. Under larger reward weightings, without SGLD’s randomness, the model may generate some repeated objects in the background related to the prompts, as shown in Fig. 14.

Different Rewards for Training Reward-Instruct. We provided additional experiments incorporating Aesthetic Score (AeS) in Table 5. The results demonstrate that incorporating AeS — a reward focused solely on image aesthetics — maintains strong performance and even improves the zero-shot FID (from 33.79 to 31.26). This improvement may stem from AeS’s complementary effect, as it emphasizes visual aesthetics differently from HPS and CLIP, potentially enhancing the regularization effect. These findings suggest that other rewards can indeed be effective to train our Reward-Instruct.

Verify the Complementarity of Rewards. We can assess the complementarity of the rewards to some extent by examining their correlations on a validation set. Specifically, we computed the Pearson correlation among the rewards used during training, utilizing the COCO5k dataset. The

	HPS	Image Reward	Clip Score
HPS	1	0.51	0.46
Image Reward	0.51	1	0.41
Clip Score	0.46	0.41	1

Table 6: Pearson correlation between rewards computed on the COCO5k dataset



Figure 15: Samples generated by Reward-Instruct (4 NFE) with long and complex prompts. The Reward-Instruct here is post-trained from SD3-medium.

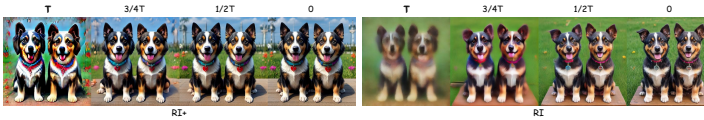


Figure 16: Path comparison between Reward-Instruct and Reward-Instruct+. The prompt is “Two dogs, best quality”.



Figure 17: The effect of high-resolution guidance loss. Images are from the same noise.

results in Table 6 indicate that correlations among the different rewards are positive and relatively strong.

C More Applications for Reward-Instruct

C.1 Super-resolution

Tackling High-Resolution Generation. Existing reward functions for text-to-image generation are mostly trained on low-resolution inputs (e.g., 224×224 pixels). This creates a fundamental limitation: directly optimizing such rewards during high-resolution (e.g., 1024×1024 pixels) synthesis struggles to preserve fine-grained details. To address this, we propose training a high-resolution classifier as a complementary guidance signal. This classifier explicitly prioritizes perceptual quality in high-resolution outputs.

Implicit High-Resolution Classifier. We can form a high-resolution classifier via the Bayesian rule:

$$\log p(\text{HighRes}|\mathbf{x}_t) = \log \frac{p(\mathbf{x}_t|\text{HighRes})p(\text{HighRes})}{p(\mathbf{x}_t|\text{LowRes})}, \quad (\text{C.1})$$

where $\mathbf{x}_t$ denotes the noisy samples at timesteps t . Its gradient can be obtained as follows:

$$\nabla_{\mathbf{x}_t} \log p(\text{HighRes}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\text{HighRes}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\text{LowRes}). \quad (\text{C.2})$$

The $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\text{HighRes})$ can be directly replaced by the original diffusion model, since it has the capability to generate high-resolution images. For obtaining $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\text{LowRes})$, we can finetune the pre-trained diffusion over low-resolution data. By doing so, we can obtain a powerful Implicit classifier for high-resolution guidance. Fig. 17 shows that after applying the implicit high-resolution guidance proposed by us, the generated images are significantly clearer.

D Limitations

Our model, akin to most text-to-image diffusion models, may not always perform perfectly regarding fairness and the accurate depiction of specific details. We intend to investigate these outstanding challenges within the generation field in our subsequent research. The goal of this future work will be to improve the model’s proficiency in text generation, ensure fairer outcomes, and provide finer control over generated details.

E Broader Impacts

This work introduces Reward-Instruct, a reward-centric approach for developing efficient text-to-image models. From a positive perspective, although this is an academic study, we believe that the proposed Reward-Instruct could be widely adopted in industry. Conversely, if these swift generation models are exploited by individuals with malicious intent, they could also streamline and expedite the production of detrimental content. While our work is centered on scientific inquiry, we are committed to implementing measures to mitigate the spread of harmful information, such as by removing inappropriate content in the dataset.

F Safeguards

The Reward-Instruct is trained on the prompt of JourneyDB dataset [37], which has undergone rigorous human and machine-based filtering to ensure that there are no harmful or violent prompts in the dataset.