This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PromptLoop: Plug-and-Play Prompt Refinement via Latent Feedback for Diffusion Model Alignment

Suhyeon Lee Jong Chul Ye
Kim Jaechul Graduate School of AI, KAIST
{suhyeon.lee, jong.ye}@kaist.ac.kr

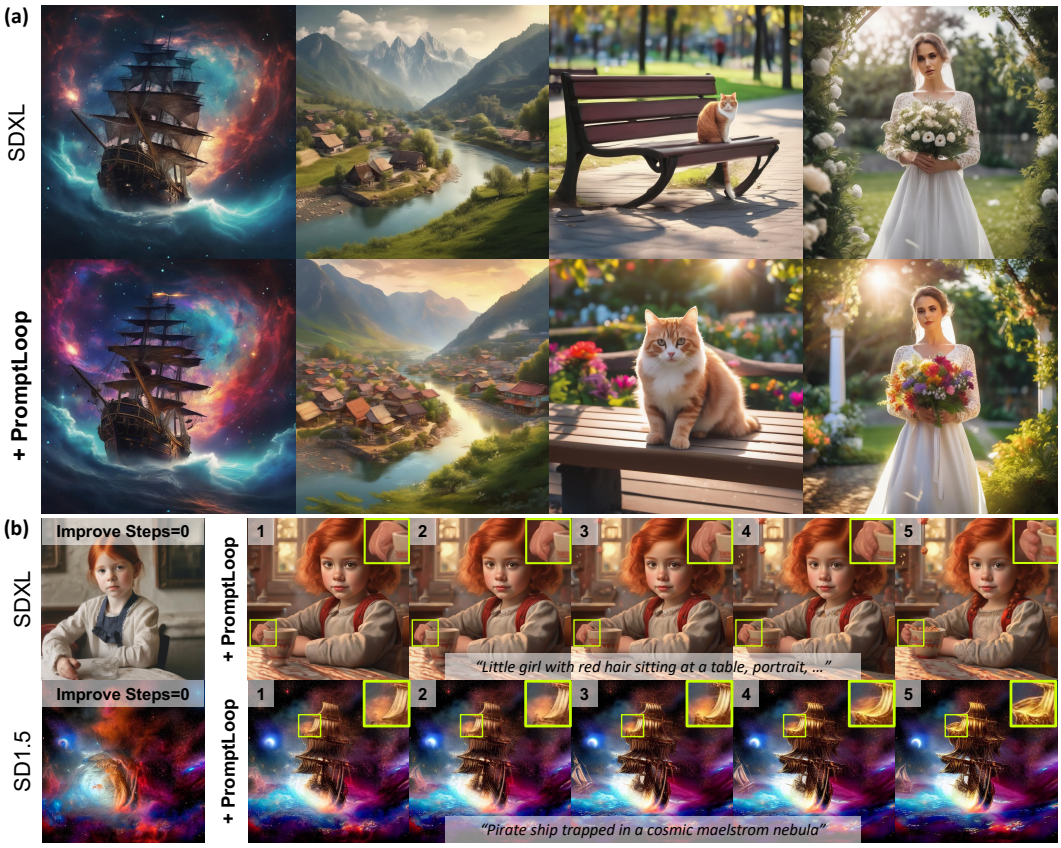


Figure 1. (a) PromptLoop uses latent feedback for stepwise prompt refinement, achieving functional equivalence to diffusion model RL and effective reward alignment (shown with ImageReward). (b) Multiple timestep-aware prompt updates during a single sampling yield stronger alignment.

Abstract

Despite recent progress, reinforcement learning (RL)-based fine-tuning of diffusion models often struggles with generalization, composability, and robustness against reward hacking. Recent studies have explored prompt refinement as a modular alternative, but most adopt a feed-forward approach that applies a single refined prompt throughout the entire sampling trajectory, thereby failing to fully leverage the sequential nature of reinforcement learning. To address this, we introduce PromptLoop*, a plug-and-play RL framework that incorporates latent feedback into step-wise prompt refinement. Rather than modifying diffusion model weights, a multimodal large language model (MLLM) is trained with RL to iteratively update prompts based on intermediate latent states of diffusion models. This design achieves a structural analogy to the Diffusion RL approach, while retaining the flexibility and generality of prompt-based alignment. Extensive experiments across diverse reward functions and diffusion backbones demonstrate that PromptLoop (i) achieves effective reward optimization, (ii) generalizes seamlessly to unseen models, (iii) composes or-*

thogonally with existing alignment methods, and (iv) mitigates over-optimization and reward hacking while introducing only a practically negligible inference overhead.

1. Introduction

Diffusion models [14, 39, 47] have now become the state of the art for image generation. Recently, increasing attention has been directed toward reinforcement learning (RL)-based approaches [48] that align these models with user preferences through explicit reward optimization. Algorithms such as PPO [42] and DPO [38] have been applied directly to fine-tune diffusion model parameters [2, 51]. With reward functions defined over aesthetic quality, safety, human preference, or prompt alignment, these methods successfully steer model behavior without requiring new training data. However, direct RL fine-tuning remains limited: improvements often fail to generalize across models, additional enhancements are not easily composable once fine-tuning is complete, and pathological behaviors such as reward hacking or over-optimization can arise [22].

In parallel, the rapid development of large language models (LLMs) [3, 10, 11] and multimodal large language models (MLLMs) [29, 53, 54] has inspired a new research direction: refining the input prompts rather than the diffusion model itself. These prompt-alignment methods either guide an LLM to improve a user's prompt or adopt iterative feedback loops for prompt refinement [19, 21, 32, 62]. Building further, [12, 52, 57] propose to fine-tune LLMs with RL, enabling them to generate goal-directed prompt modifications more effectively. Compared to weight-level tuning, prompt refinement is attractive because prompts are shared across all text-to-image (T2I) models, inherently supporting generalization and orthogonal composability. Moreover, prompts, being abstract and discrete, may act as a buffer against reward hacking by decoupling reward optimization from direct parameter updates [8, 27, 59]. For a detailed discussion of related works, see Appendix A. Nevertheless, prompt-based strategies remain structurally distinct from weight-level approaches. In diffusion models, parameters interact directly with intermediate latent variables $\boldsymbol{x}_t$ in a feedback loop, where each denoising step conditions on $\boldsymbol{x}_t$ to produce $\boldsymbol{x}_{t-1}$. By contrast, existing RL-based prompt refinement methods typically operate in a feed-forward manner, producing a refined prompt once and applying it uniformly across all timesteps, without leveraging the evolving latent trajectory.

To bridge this gap, we propose a generalized RL-based reward alignment framework called *PromptLoop* that achieves structural analogy to weight-level fine-tuning while preserving the modularity of prompt refinement (Fig. 2). Specifically, our method introduces a plug-and-play prompt refinement module as a policy. This module leverages an MLLM to process feedback from the intermediate latent $\boldsymbol{x}_t$ as one of the states, analogous to diffusion RL formulations, and then refines the prompt $\boldsymbol{c}_t$ as the action injected into subsequent denoising steps. Thus, the sampling dynamics are adaptively adjusted without direct fine-tuning of the diffusion model itself. Unlike approaches that either delay feedback until after sampling or confine it to external loops, our method adopts a diffusion RL–style closed-loop design that embeds refinement directly within a single diffusion pass, ultimately enabling fine-grained adaptive control and improved efficiency. Despite the introduction of iterative MLLM-based policy inference, our implementation keeps the inference-time overhead at a practically manageable level compared to standard diffusion model sampling.

Extensive experiments across diverse diffusion models and reward functions demonstrate that our approach not only achieves effective reward optimization, but also generalizes robustly to unseen models, composes orthogonally with existing alignment methods, and mitigates over-optimization and reward hacking. These results establish PromptLoop as a practical and versatile approach to reward alignment for diffusion models. Our contributions are summarized as follows:

- PromptLoop incorporates step-wise latent feedback into prompt refinement, achieving structural analogy to parameter-level tuning without modifying model weights.
- We demonstrate broad generalization, effective reward optimization, and mitigation of reward hacking across diverse models and reward functions while incurring only minimal practical inference overhead.

2. Preliminaries

Diffusion Models. Diffusion models [14, 44, 46] are a class of latent variable generative models that approximate the data distribution $\boldsymbol{x}_0 \sim p_{\text{data}}$ through a hierarchical latent process. The generative distribution is formulated as

$$p_\phi(\boldsymbol{x}_0) = \int p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\phi^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\, d\boldsymbol{x}_{1:T}, \qquad (1)$$

where the prior $p(\boldsymbol{x}_T)$ is typically a standard Gaussian distribution. The latent sequence $\{\boldsymbol{x}_t\}_{t=1}^T$ is obtained via a forward noising process, which follows a Markov chain with a variance schedule $\{\beta_t\}_{t=1}^T$:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t \mid \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)I), \qquad (2)$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t \mid \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)I), \qquad (3)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Training is carried out by learning to predict the injected Gaussian noise $\boldsymbol{\epsilon}$ using a neural network $\hat{\boldsymbol{\epsilon}}_\phi$, which is often conditioned by $\boldsymbol{c}$, known as ϵ-*matching*. This is equivalent to denoising score

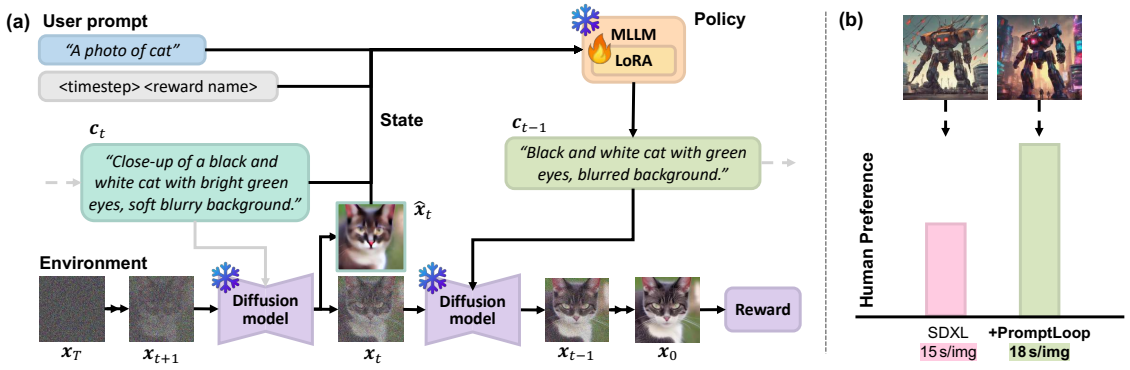


Figure 2. **(a) Closed-loop prompt refinement framework with RL. (b) The proposed framework enhances human preference in a plug-and-play manner with minimal additional inference cost.** At each denoising step, the policy MLLM takes the current state—denoised estimates, the user query, and prior refinements—and generates an action, a refined prompt. The diffusion model then updates the state, and this loop continues until the final image is produced and scored by the reward model.

matching (DSM) [46, 50], which estimates the score function $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$:

$$\mathcal{L}_{\epsilon-\text{matching}} = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\|\hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, \boldsymbol{c}) - \boldsymbol{\epsilon}\|_2^2\right], \quad (4)$$

where $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. Once trained, the model iteratively reverses the noising process as follows:

$$\begin{aligned}\boldsymbol{x}_{t-1} &= f(\boldsymbol{x}_t, \boldsymbol{z}_t, \boldsymbol{c}, t) \\ &:= \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, \boldsymbol{c})\right) + \sigma_t \boldsymbol{z}_t, \quad (5)\end{aligned}$$

where $\boldsymbol{z}_t \sim \mathcal{N}(0, I)$ and $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. This corresponds to the canonical DDPM sampler [14]. In general, $f(\cdot)$ can be replaced by a variety of alternative samplers such as DDIM [45], PNDM [30], Euler [18], DPM-solver [31].

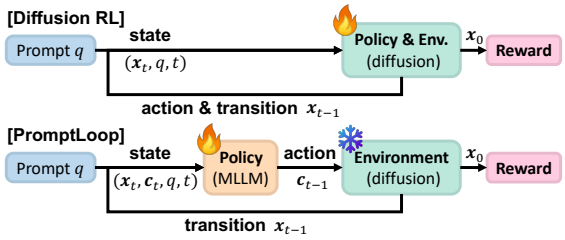
3. PromptLoop

3.1. MDP Formulation

In PromptLoop, as shown in Fig. 2, we aim to generate a refined text prompt $\boldsymbol{c}_{t-1}$ conditioned on user input q and interpret intermediate visual states $\boldsymbol{x}_t$ arising during the reverse diffusion process. The refined text prompt is then used to generate the next visual sample $\boldsymbol{x}_{t-1}$. To this end, we adopt a multimodal language model (MLLM) [29, 53, 54] that accepts multimodal inputs and outputs refined prompts at each timestep. Then, our RL framework aims to train the MLLM to maximize the reward at the final visual state $\boldsymbol{x}_0$. Formally, our Markov decision process (MDP) is defined as the T-step reverse process with state s_t and actions a_t:

$$s_t = (\boldsymbol{x}_t, \boldsymbol{c}_t, q, t), \quad a_t = \boldsymbol{c}_{t-1}, \quad (6)$$

which are conditioned on an initial user prompt q, a previously updated prompt $\boldsymbol{c}_t$, and a visual state $\boldsymbol{x}_t$. Then, an



	Diffusion RL	PromptLoop (Ours)
State s_t	$(\boldsymbol{x}_t, q, t)$	$(\boldsymbol{x}_t, \boldsymbol{c}_t, q, t)$
Policy	Diffusion model p_ϕ	MLLM π_θ
Action a_t	$\boldsymbol{x}_{t-1} \sim p_\phi(\cdot\|s_t)$	$\boldsymbol{c}_{t-1} \sim \pi_\theta(\cdot\|s_t)$
Transition	–	$\boldsymbol{x}_{t-1} = f(\boldsymbol{x}_t, \boldsymbol{z}_t, \boldsymbol{c}_{t-1}, t)$
Reward R	$r(\boldsymbol{x}_0, q)$	$r(\boldsymbol{x}_0, q)$

Figure 3. **Structural analogy and key differences in MDP formulation between Diffusion RL and proposed framework.** Latent feedback establishes a functional correspondence with Diffusion RL, while PromptLoop diverges by adjusting the diffusion dynamics through timestep-aware prompts as actions.

action is sampled from the *MLLM policy* as $a_t \sim \pi_\theta(\cdot \mid s_t)$ and the visual state $\boldsymbol{x}_{t-1}$ is updated using the frozen diffusion model with the updated prompt $\boldsymbol{c}_{t-1}$. A terminal reward $r(\boldsymbol{x}_0, q)$ is assigned at the final step.

This is in contrast to directly training the diffusion model's parameters [2, 51] using RL, where the MDP is defined with the state and action:

$$s_t = (\boldsymbol{x}_t, q, t), \quad a_t = \boldsymbol{x}_{t-1} \quad (7)$$

where an action is sampled from the *diffusion* policy $\boldsymbol{x}_{t-1} \sim p_\phi(\cdot|s_t)$. The difference between the original Diffusion-RL and our RL framework is detailed in Fig. 3.

Note that our MDP formulation provides a structural correspondence between diffusion-model-based RL and the prompt refinement framework, enabled by a timestep-aware closed-loop latent feedback mechanism. On the other hand,

in direct fine-tuning of diffusion models using RL, the diffusion model should be trained as the optimization target. This direct RL fine-tuning remains limited: improvements often fail to generalize across models, additional enhancements are not easily composable once fine-tuning is complete, and pathological behaviors such as reward hacking or over-optimization can arise. In our framework, the timestep-aware prompt-level actions can approximate the functional role of weight-level control, while retaining plug-and-play modularity, generalization, composability, and robustness against reward hacking.

Furthermore, our approach has fundamental advantages over other prompt fine-tuning approaches. Specifically, prior prompt-tuning approaches either lack an intrinsic feedback loop [12, 52, 57] or deliver feedback only after full sampling [19, 21, 32], making them fundamentally different from our MDP formulation.

3.2. Optimization

At the end of each episode (*i.e.*, $\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, \ldots, \boldsymbol{x}_0$), the fully generated image $\boldsymbol{x}_0$ is evaluated by a reward function r to produce a reward $R = r(\boldsymbol{x}_0, q)$. This can encode diverse criteria such as aesthetic quality [41], safety [26], prompt alignment [37], or human preference [58, 60]. The diffusion model and the reward model are both treated as black-box components: no gradient flows through them, and the policy is updated solely based on observed rewards.

Policy gradient methods [49, 56] optimize this objective by estimating gradients with respect to θ. A widely used algorithm is Proximal Policy Optimization (PPO) [42], which improves stability by constraining policy updates through a clipped surrogate objective:

$$\begin{aligned}\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_t \left[\min\Big(\rho_t(\theta)\,\hat{A}_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t\Big)\right] \\ -\beta\,\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t) \parallel \pi_\theta(\cdot \mid s_t)]\,, \\ \text{where} \quad \rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}.\end{aligned} \tag{8}$$

Here, β is a hyperparameter controlling the strength of the KL penalty, and the advantage $\hat{A}_t$ measures how much better an action is than the expected value under the current policy. Specifically, Group Relative Policy Optimization (GRPO) [11] replaces the advantage estimator with a group-normalized reward to stabilize training and reduce variance:

$$A_i = \frac{r_i - \text{mean}(\{r_j(\cdot)\}_{j=1}^G)}{\text{std}(\{r_j(\cdot)\}_{j=1}^G)}, \tag{9}$$

where $\{r_j(\cdot)\}_{j=1}^G$ are the rewards of G sampled outputs for the same prompt. Therefore, we employ the standard token-level GRPO [11]. Each training episode is initialized with user prompts drawn from a prompt-only dataset and proceeds via an online, on-policy reinforcement learning procedure.

3.3. Implementation

As part of our implementation, we design the MLLM's input to be denoised latent representations rather than raw noisy states $\boldsymbol{x}_t$. Specifically, we convert the noisy visual latent state $\boldsymbol{x}_t$ into its denoised estimate $\hat{\boldsymbol{x}}_t$, which lies closer to the data manifold and thus provides a more semantically meaningful input to the policy model:

$$\hat{\boldsymbol{x}}_t = \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\boldsymbol{x}_{t+1} - \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_{t+1}, \boldsymbol{c}_t, t)\big). \tag{10}$$

Notably, many studies adopt these estimates for the final sample estimation [4, 63], even for evaluation with vision-language models [20, 43]. We further provide empirical evidence to support this design choice (Appendix E.2).

While our framework achieves structural equivalence, it introduces additional computational overhead: the policy model must be invoked during every denoising step of the diffusion process. This requirement also significantly increases memory costs, as both the diffusion model and the policy MLLM must be co-resident on the accelerator (*e.g.* VRAM), or alternatively, incur large transfer times under offloading. Such constraints not only limit practical applicability but also complicate seamless integration of our approach into existing user-level diffusion-based image generation pipelines.

To mitigate these issues, we adopt a sparse refinement strategy, where *prompt refinement steps* are defined as a set of timesteps $\mathcal{R} \subseteq \{1, \ldots, T\}$ with $|\mathcal{R}| = N_R$. The policy model is applied only at these steps rather than at every denoising step. For example, if the policy refines the prompt at timestep t_1 and the next refinement occurs at t_2 with $t_1 > t_2$, then $\boldsymbol{c}_{t_1-1:t_2} = \pi_\theta(\cdot \mid s_{t_1})$ and remains fixed until the next refinement step. During training, $\mathcal{R}$ is sampled uniformly at random, while during inference it is deterministically set at even intervals. This design allows the policy to generalize to an arbitrary number of refinement steps during sampling, while introducing only minimal inference overhead (see Sec. Runtime Overhead).

We empirically observe that visual feedback from intermediate denoised states—though essential during training—is not strictly necessary at inference. Once the policy has learned the transition dynamics of the environment (*i.e.*, the diffusion process coupled with the reward model), it can generate effective refinements without explicit access to intermediate visual signals. Consequently, refined prompts for all timesteps can be generated *a priori*, allowing the diffusion process to proceed without interruptions during inference. This design yields substantial generalization capability and efficiency gains while remaining fully compatible with existing diffusion model ecosystems. It requires no modification to the generation loop and offers the ease of integration seen in feed-forward prompt optimization methods, yet uniquely retains the advantages of closed-loop RL

Table 1. Quantitative evaluation on single-reward alignment with SD1.5 and SDXL, showing comparison with baselines and demonstrating orthogonality and generalizability.

Training Setup	Method	Image Reward	HPSv2	Aesthetics	MLLM Score
SDXL & Image Reward	SDXL	0.7244	0.2805	6.073	0.735
	+ ReFL [60]	1.0119	0.2740	6.286	0.715
	+ Qwen2.5-VL-3B [1]	0.5114	0.2739	6.279	0.741
	+ GPT-5	0.6299	0.2794	6.231	0.740
	+ RePrompt [57]	1.0148	0.2796	6.518	0.763
	+ PromptLoop (ours)	**1.0948**	**0.2807**	**6.583**	**0.764**
	SDXL + Diffusion-DPO [51]	0.9921	**0.2868**	6.015	0.731
	+ PromptLoop (ours)	**1.2898**	0.2862	**6.491**	**0.763**
	SDXL + NPNet [65]	0.7357	0.2805	6.059	0.733
	+ PromptLoop (ours)	**1.1213**	**0.2811**	**6.561**	**0.762**
	SD1.5 [39]	0.0816	0.2678	5.458	0.675
	+ PromptLoop (ours)	**0.4546**	**0.2688**	**5.813**	**0.723**
SD1.5 & Image Reward	SD1.5	0.0816	0.2678	5.458	0.675
	+ DDPO [2]	0.6051	0.2726	5.562	0.693
	+ ReFL [60]	0.6248	**0.2748**	5.577	0.691
	+ Qwen2.5-VL-3B [1]	-0.1720	0.2628	5.668	0.693
	+ GPT-5 [33]	-0.0950	0.2647	5.726	0.700
	+ RePrompt [57]	0.4344	0.2684	5.850	0.722
	+ PromptLoop (ours)	**0.6320**	0.2701	**5.853**	**0.725**
	SD1.5 + DDPO [2]	0.6051	0.2726	5.562	0.693
	+ PromptLoop (ours)	**0.9842**	**0.2742**	**5.926**	**0.726**
	SD1.5 + DanceGRPO [61]	0.6156	0.2795	5.662	0.704
	+ PromptLoop (ours)	**0.8950**	**0.2799**	**5.990**	**0.728**
	SD1.5 + Diffusion-DPO [51]	0.3012	0.2717	5.568	0.687
	+ PromptLoop (ours)	**0.7920**	**0.2739**	**5.968**	**0.734**
	SD1.5 + ReFL [60]	0.6248	0.2748	5.577	0.691
	+ PromptLoop (ours)	**0.9271**	**0.2751**	**5.877**	**0.724**
	SDXL [36]	0.7244	0.2805	6.073	0.735
	+ PromptLoop (ours)	**1.0859**	**0.2807**	**6.535**	**0.763**

Table 2. Quantitative evaluation on composite-reward alignment with SDXL-turbo, showing comparison with baselines and demonstrating orthogonality and generalizability.

Training Setup	Method	GenEval	Image Reward	HPSv2
SDXL-turbo & RePrompt	SDXL-turbo [40]	0.5445	0.7769	0.2915
	+ Qwen2.5-VL-3B [1]	0.5212	0.6417	0.2893
	+ GPT-5 [33]	0.5456	0.7593	0.2915
	+ RePrompt [57]	0.5101	0.7876	0.2912
	+ PromptLoop (ours)	**0.5483**	**0.8516**	**0.2938**
	SDXL [36]	0.5431	0.5518	0.2886
	+ PromptLoop (ours)	**0.5505**	**0.7420**	**0.2906**
	SD1.5 [39]	0.4206	-0.1315	0.2783
	+ PromptLoop (ours)	**0.4399**	**-0.0375**	**0.2793**

fine-tuning.

4. Experimental Results

4.1. Methods

Tasks. To evaluate our framework as a general black-box reward alignment system, we consider two categories of reward models: *single reward* and *composite reward*.

Table 3. Ablation study results showing the effectiveness of each proposed component.

Components	ImageReward	HPSv2	MLLM Score
SD1.5	0.0816	0.2678	0.675
+ policy model	-0.2315	0.2617	0.681
+ GRPO training	0.4344	0.2684	0.722
+ multiple improvements	0.4912	0.2690	0.724
+ visual feedback	**0.6320**	**0.2701**	**0.725**

For the single reward setting, we adopt ImageReward [60], a widely used neural network–based reward function for human preference and prompt alignment, along with incompressibility, compressibility, and aesthetic score models [2, 41]. These rewards are applied to train Stable Diffusion v1.5 [39] (SD1.5) and Stable Diffusion XL [36] (SDXL) using prompts from the Pick-a-Pic v2 dataset [23]. For the composite reward setting, we follow a RePrompt-style design [57], which combines ImageReward, MLLM-reward [34], and additional task-specific signals such as format and length reward. This composite reward style is intended to better capture human preference and object-focused alignment. Compared to the single reward setting, the composite reward is more complex and difficult to optimize, since it requires balancing multiple heterogeneous objectives simultaneously. We use it to train Stable Diffusion XL Turbo [40] (SDXL-turbo), a distillation model designed for few-step generation, with the prompt dataset introduced by Wu et al. [57].

Evaluations. In evaluation, we validate our model's capability along three aspects: performance, orthogonality, and generalizability. For performance evaluation, we compare against baseline reward alignment methods, including DDPO [2], ReFL [60], Qwen2.5-VL-3B [1], GPT-5 [33], and RePrompt [57]. For orthogonality, we apply our trained policy model to other diffusion models that are fine-tuned or augmented with additional modules for human preference alignment, demonstrating that our method can be applied orthogonally to existing preference alignment techniques. Specifically, we evaluate on DDPO [2], DanceGRPO [61], Diffusion-DPO [51], ReFL, and NPNet [65]. These experiments demonstrate that our method can be applied orthogonally to diverse alignment techniques without requiring retraining. For generalizability, we evaluate our trained policy model on different versions of text-to-image diffusion models that were not seen during training. It is important to note that for both orthogonality and generalizability, the policy model was only trained on the vanilla diffusion model environment, which differs from the sampling variants.

4.2. Results

Single Reward. After aligning SD1.5 and SDXL models with the ImageReward reward function, we conducted quantitative evaluations (Tab. 1). The results demonstrate

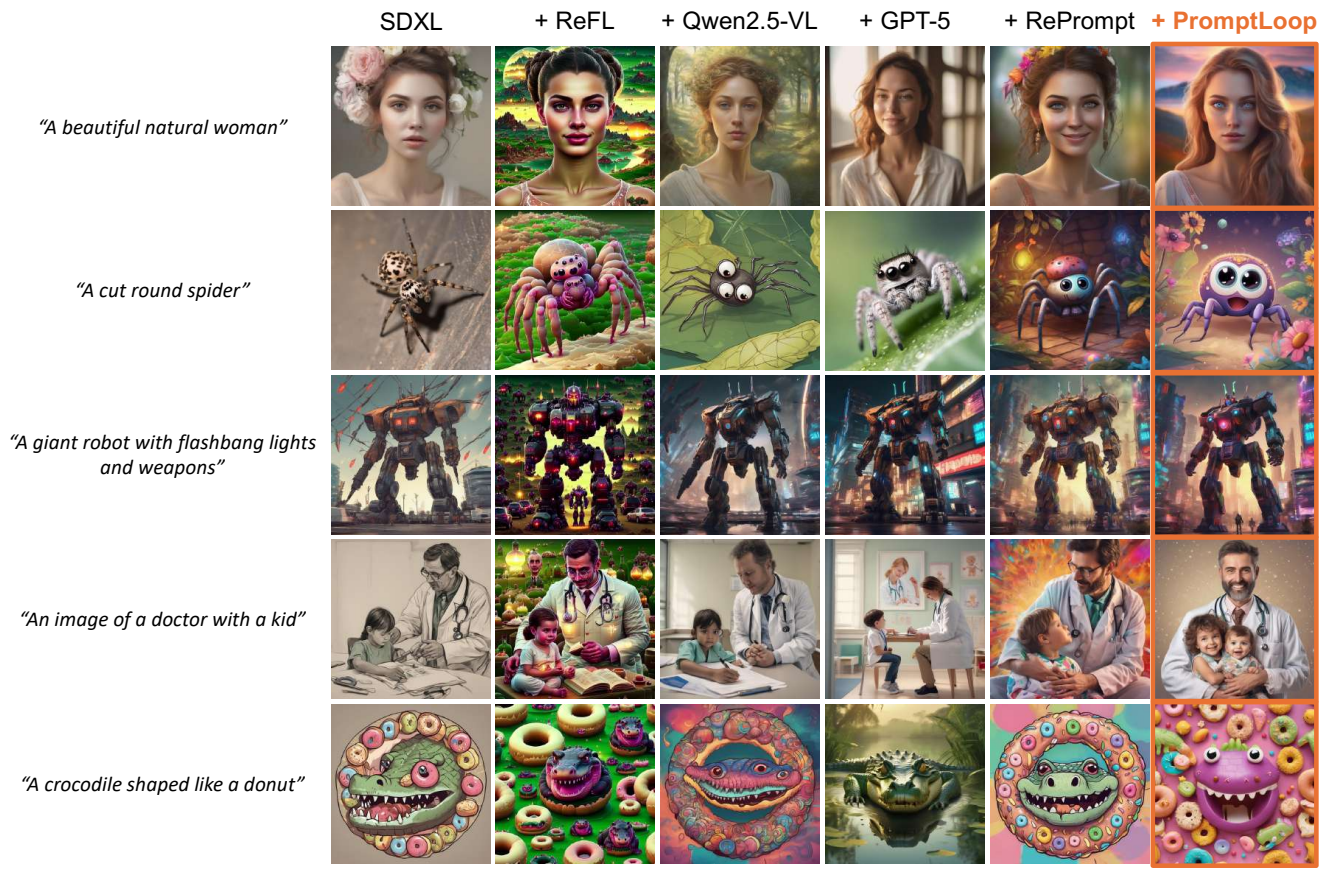


Figure 4. Qualitative comparison of single-reward alignment, illustrating improvements over baseline methods. (SDXL & ImageReward)

that our proposed methodology consistently outperforms baselines not only with respect to the target reward but also across most evaluation metrics. Crucially, our method is orthogonal, demonstrating broad compatibility with a variety of human preference alignment strategies—including noise optimization, reinforcement learning, and gradient-based optimization—regardless of their internal mechanisms. Consequently, establishing superiority over full alignment frameworks is not the sole objective; rather, one of the key values of our approach is its ability to seamlessly enhance and extend these methods, not replace them.

The qualitative comparisons in Fig. 4, 5, which present SDXL results, highlight effective alignment to the reward signal, composability of our method, and robustness against over-optimization, an aspect not always captured by quantitative metrics. For instance, ReFL optimized the ImageReward signal through strategies resembling reward hacking from a human perspective. However, this degradation was not clearly reflected in commonly used metrics such as HPS or aesthetic scores. Thus, the qualitative evaluation further underscores the value of our approach in revealing such vulnerabilities.

Composite Reward. As one of the evaluation tasks, we consider RePrompt-style multi-reward alignment, which imposes challenging conditions such as a few-step distillation model and object-centric prompt alignment benchmarks (Tab. 2). Our framework achieves strong qualitative and quantitative results under these settings, showing consistently high performance across an object-centric prompt alignment benchmark and multiple human-preference benchmarks. This indicates that our method effectively avoids over-optimization while achieving robust alignment. Moreover, we observe similar generalization to diffusion models unseen during training.

Ablation Studies. We conducted a series of ablation studies to validate the contributions of our proposed components and to analyze the effects of key hyperparameters. All experiments were performed on a single reward task (ImageReward) using the SD1.5 model. Tab. 3 summarizes the results, where each major component was added incrementally to highlight its individual effect. First, simply applying the policy model to improve prompts without training (+ policy model) degraded performance, as the model could not fully capture the task despite the use of a system prompt.

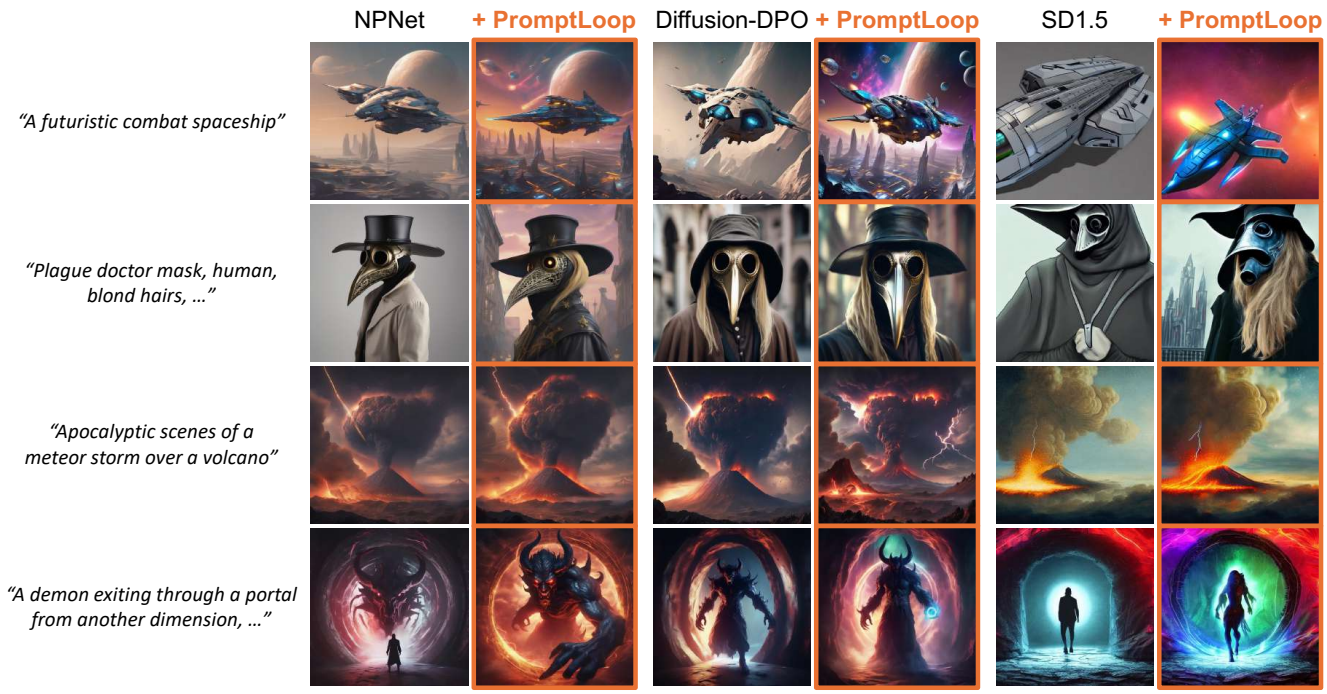


Figure 5. Qualitative results showing the orthogonality and generalizability achieved by applying our method to unseen reward-alignment baselines (SDXL & ImageReward).

Training the policy model with GRPO (+ GRPO training) led to significant improvements across all metrics. Incorporating multiple prompt refinements within a single diffusion trajectory (+ multiple improvements, 5 steps) further boosted performance. Finally, introducing training-time visual feedback substantially increased the target reward without reducing other metrics, suggesting that it helps mitigate reward hacking (+ visual feedback).

We also investigated the impact of the number of prompt refinement steps (Fig. 6). Increasing the number of refinement steps improved not only the reward metric but also other evaluation metrics. Importantly, increasing the number of refinement steps does not increase the number of diffusion sampling steps. When trained without visual feedback, these improvements were much smaller or absent. These findings highlight that visual feedback and iterative prompt refinement are indispensable components of our equivalence MDP formulation. Together, they establish the closed-loop structure that mirrors direct RL on diffusion models, and the ablation results confirm that this formulation is not only structurally well-founded but also empirically effective.

For further analyses, including timestep-wise prompt evolution analysis and additional qualitative results, please refer to Appendix E and D.

Runtime Overhead. To evaluate the runtime overhead introduced by the integrated MLLM policy model, we measure the inference time as a function of the number of prompt improvement steps. Tab. 4 shows that PromptLoop increases the total inference time only marginally, from 15s to 18s in our default configuration with five prompt refinement steps. This result indicates that our method keeps the inference-time overhead at a manageable level, demonstrating the user-level practicality of our framework.

Table 4. Inference time analysis showing minimal runtime overhead of the proposed method (A100×1, batch size = 8).

Model	Improvement Steps	Inference Time	
		s/img	relative
SDXL	0	15.00	1.0
+ PromptLoop	1	15.73	1.05
	2	16.29	1.09
	3	16.95	1.13
	4	17.55	1.17
	5	18.43	1.23

Importantly, the MLLM policy model remains fixed in size and requires no retraining when applied to new diffusion backbones. As diffusion models continue to scale, the relative inference cost introduced by our policy model naturally decreases. This plug-and-play design thus offers broad architectural compatibility and long-term scalability, making the framework increasingly efficient and readily applicable to next-generation diffusion models without additional training cost.



Figure 6. Ablation study demonstrating that incorporating visual feedback and increasing the number of refinement steps consistently enhances reward alignment. (Left: SDXL, Right: SD1.5; reward: ImageReward)

5. Conclusion

In this work, we introduced PromptLoop, a plug-and-play framework for reward alignment of diffusion models via step-wise prompt refinement with latent feedback. By leveraging a multimodal policy model trained with reinforcement learning, our method attains structural equivalence to parameter-level fine-tuning while retaining the flexibility, generality, and modularity of prompt-based alignment. Experiments demonstrate that PromptLoop achieves effective reward optimization, generalizes seamlessly to unseen diffusion backbones, composes orthogonally with existing alignment techniques, and mitigates over-optimization and reward hacking. These results position PromptLoop not only as a structurally sound but also as a practically robust complement to weight-level tuning. Overall, PromptLoop provides a simple yet effective path toward more reliable and adaptable generative models, while its plug-and-play nature facilitates integration into user-facing applications, underscoring strong potential for real-world deployment.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation) and by the National Research Foundation of Korea under Grant RS-2024-00336454.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 1

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 5, 1, 9

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[4] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 4, 7

[5] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. 2022. 7

[6] Carles Domingo-Enrich, Michal Drozdzal, 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. 1

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 7

[8] Tim Genewein, Kevin Wenliang Li, Jordi Grau-Moya, Anian Ruoss, Laurent Orseau, and Marcus Hutter. Understanding prompt tuning and in-context learning via meta-learning. *arXiv preprint arXiv:2505.17010*, 2025. 2

[9] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023. 5

[10] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2

[11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 4

[12] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023. 2, 4, 1

[13] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. 7

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3

[15] 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. In *The Tenth International Conference on Learning Representations*, 2022. 1

[16] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024. 1, 5

[17] Wenxuan Huang, Shuang Chen, Zheyong Xie, Shaosheng Cao, Shixiang Tang, Yufan Shen, Qingyu Yin, Wenbo Hu, Xiaoman Wang, Yuntian Tang, et al. Interleaving reasoning for better text-to-image generation. *arXiv preprint arXiv:2509.06945*, 2025. 1

[18] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 3

[19] Mohammad Abdul Hafeez Khan, Yash Jain, Siddhartha Bhattacharyya, and Vibhav Vineet. Test-time prompt refinement for text-to-image models. *arXiv preprint arXiv:2507.22076*, 2025. 2, 4, 1

[20] Jaemin Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Free2guide: Training-free text-to-video alignment using image lvlm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17920–17929, 2025. 4, 7

[21] Semin Kim, Yeonwoo Cha, Jaehoon Yoo, and Seunghoon Hong. Reward-agnostic prompt optimization for text-to-image diffusion models. *arXiv preprint arXiv:2506.16853*, 2025. 2, 4, 1

[22] Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[23] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023. 5

[24] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024. 5

[25] Black Forest Labs. Flux.1-dev. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024. Text-to-image diffusion model. 7

[26] LAION-AI. Laion safety: CLIP-based nsfw detection. GitHub repository, 2023. Available at https://github.com/LAION-AI/LAION-SAFETY. 4

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. 2

[28] Yaqi Li, Peng Chen, Mingyang Han, Bu Pi, Haoxiang Shi, Runzhou Zhao, Yang Yao, Xuan Zhang, and Jun Song. Visual-cog: Stage-aware reinforcement learning with chain of guidance for text-to-image generation. *arXiv preprint arXiv:2508.18032*, 2025. 1

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 2, 3

[30] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *The Tenth International Conference on Learning Representations*, 2022. 3

[31] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022. 3

[32] Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. Improving text-

to-image consistency via automatic prompt optimization. *arXiv preprint arXiv:2403.17804*, 2024. 2, 4, 1

[33] OpenAI. Openai platform: gpt-5. https://platform.openai.com/docs/models/gpt-5, 2025. Accessed on 2025-11-13. 5

[34] OpenAI. Openai platform: gpt-5-mini. https://platform.openai.com/docs/models/gpt-5-mini, 2025. Accessed on 2025-09-14. 5, 3

[35] Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. *arXiv preprint arXiv:2509.16131*, 2025. 5

[36] 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. 5, 1

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4, 6

[38] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. 2

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 5, 1

[40] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024. 5, 1

[41] Christoph Schuhmann. CLIP+MLP Aesthetic Score Predictor. GitHub repository, 2025. https://github.com/christophschuhmann/improved-aesthetic-predictor. 4, 5, 3, 9

[42] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2, 4

[43] Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025. 4, 7

[44] 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. 2

[45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3, 1

[46] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2, 3

[47] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[48] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998. 2

[49] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999. 4

[50] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011. 3

[51] 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. 2, 3, 5, 1

[52] Linqing Wang, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Jiale Tao, Qixun Wang, Ruihuang Li, Xin Li, Mingrui Wu, Xinchi Deng, et al. Promptenhancer: A simple approach to enhance text-to-image models via chain-of-thought prompt rewriting. *arXiv preprint arXiv:2509.04545*, 2025. 2, 4, 1

[53] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2, 3, 4

[54] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 2, 3

[55] Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernández Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *arXiv preprint arXiv:2404.13040*, 2024. 5

[56] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. 4

[57] Mingrui Wu, Lu Wang, Pu Zhao, Fangkai Yang, Jianjin Zhang, Jianfeng Liu, Yuefeng Zhan, Weihao Han, Hao Sun, Jiayi Ji, et al. Reprompt: Reasoning-augmented reprompting for text-to-image generation via reinforcement learning. *arXiv preprint arXiv:2505.17540*, 2025. 2, 4, 5, 1

[58] 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. 4, 3

[59] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations*, 2022. 2

[60] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023. 4, 5, 1, 3

[61] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025. 5, 1

[62] Zhengyuan Yang, Jianfeng Wang, Linjie Li, Kevin Lin, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. Idea2img: Iterative self-refinement with gpt-4v for automatic image design and generation. In *European Conference on Computer Vision*, pages 167–184. Springer, 2024. 2, 1

[63] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023. 4, 5, 7, 8

[64] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. 7

[65] Zikai Zhou, Shitong Shao, Lichen Bai, Shufei Zhang, Zhiqiang Xu, Bo Han, and Zeke Xie. Golden noise for diffusion models: A learning framework. In *International Conference on Computer Vision*, 2025. 5