Appendix

Supplementary Materials

A. Related Works

Aligning Diffusion Models. Following the success of RLHF for LLMs, there has been growing interest in aligning diffusion models with human preferences or arbitrary reward functions. Methods such as DDPO [2], Diffusion-DPO [51], and Dance-GRPO [61] treat the diffusion sampling process as a Markov decision process (MDP), and train the diffusion model using RL algorithms. In contrast to RL-based approaches that rely on black-box rewards, other methods directly exploit the gradient of the reward or objective function. For example, ReFL [60] optimizes sampling trajectories via reward gradients, applying the reward to intermediate denoised estimates to avoid full backpropagation. ELLA [16] introduces a timestep-aware connector module that maps encoded prompt embeddings before they are fed into the diffusion model. More recently, Adjoint Matching [6] casts reward fine-tuning as a stochastic optimal control (SOC) problem, optimizing with reward gradients.

Prompt-based Improvements for Diffusion Models. In text-to-image generation, prompts serve as a powerful control signal and have been widely leveraged as a means of alignment. Prior work such as OPT2I [32], Idea2Img [62], RATTPO [21], and TIR [19] explores LLM-based prompt refinement without fine-tuning, relying on feedback from evaluations of fully generated images to suggest improved prompts. To align LLM-based prompt refinement more closely with reward, Promptist [12], RePrompt [57], and PromptEnhancer [52] fine-tune LLMs with reinforcement learning, treating the diffusion model simply as a black-box reward model in a feedforward manner. RL-based alignment has also been extended beyond diffusion models to autoregressive (AR) multimodal models, where methods such as Visual-CoG [28] and IRGL [17] adopt CoT-style approaches that iteratively generate prompts and images through self-feedback to achieve reward alignment.

B. Detailed Algorithm

We summarize the procedure of PromptLoop in two parts. Algorithm 1 presents the training process, while Algorithm 2 details the sampling procedure.

C. Implementation Details

C.1. Framework and Training

We use Qwen2.5-VL-3B-Instruct [1] as the policy model, and Stable Diffusion 1.5 [39] (SD1.5), XL [36] (SDXL), and XL-Turbo [40] (SDXL-turbo) as the text-to-image diffusion backbones, with the specific model chosen according to the task setting. Generation resolution, classifier-free guidance (CFG) scale, inference steps, and sampler were set to each model's default configuration, except that we used the DDIM sampler [45] for SD1.5 and 5 sampling steps for SDXL-turbo.

For GRPO training, we build on the TRL library[1] and implement our framework on top of it. Training is performed with the GRPO algorithm using a learning rate of 5×10^{-6}, batch size 8, group size 8, and β (the KL-regularization coefficient) set to 0.005 for single-reward training and 0 for composite-reward training, without PPO clipping (num-iterations $= 1$). We further apply parameter-efficient fine-tuning (LoRA) [15] using the PEFT library[2], with rank $r = 16$, scaling factor $\alpha = 64$, dropout 0.05, and updates applied to all linear projection layers in the transformer blocks. All experiments are conducted in `bf16` precision on four NVIDIA A100 80GB GPUs, and each training run takes approximately three days to complete.

To optimize our framework, we use 2 training-prompt improvement steps and 5 sampling-prompt improvement steps. Visual feedback is resized to 256×256 from the original denoised estimates obtained during the sampling process and provided to the policy model. During sampling, we insert the built-in token `<|image_pad|>` as a placeholder to replace the visual feedback.

C.2. Prompting Policy Models

The policy models used for prompt refinement are guided by the instruction shown in Fig. 7, 8. As described earlier, the policy model is conditioned on the raw user input, the previously applied improved prompt, and the current timestep. In addition, we provide auxiliary information such as the total number of timesteps and the name of the target reward function. The model is then required to output an improved prompt that is suitable for the current denoising step. For the reward specification, we only provide the name of the reward (*e.g.*, `ImageReward`, `HPSv2`), without detailed definitions. This design leaves

[1] https://github.com/huggingface/trl
[2] https://github.com/huggingface/peft

Algorithm 1: Training PromptLoop

Input: Policy π_θ, diffusion denoiser $\hat{\boldsymbol{\epsilon}}_\phi$, sampler f, prompts p_{data}, reward R, # refinement steps N_R, GRPO group size G, total steps T

Output: Reward-aligned plug-and-play policy π_θ

repeat
  Sample $q \sim p_{\text{data}}$
  Sample $\mathcal{R} \sim \text{Unif}(\{\, R \subseteq \{1, \ldots, T\} \,:\, |R| = N_R \,\})$
  for $g \in \{1, \ldots, G\}$ **do**
    $\boldsymbol{c} \leftarrow q$ `// init text prompt`
    $\tau^g \leftarrow [\,]$ `// trajectory: (state, action) pairs`
    Sample $\boldsymbol{x}_T \sim \mathcal{N}(0, I)$
    for $t = T, T-1, \ldots, 1$ **do**
      if $t \in \mathcal{R}$ **then**
        $s_t \leftarrow (\hat{\boldsymbol{x}}_t, \boldsymbol{c}, q, t)$
        Sample $\boldsymbol{c} \sim \pi_\theta(\cdot \mid s_t)$ `// prompt refinement`
        τ^g.append(s_t); τ^g.append$(\boldsymbol{c})$
      end
      `// perform one sampler step`
      Sample $\boldsymbol{z}_t \sim \mathcal{N}(0, I)$
      $\boldsymbol{x}_{t-1} \leftarrow f(\boldsymbol{x}_t, \boldsymbol{z}_t, \boldsymbol{c}, t)$
      $\hat{\boldsymbol{x}}_{t-1} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, \boldsymbol{c})\big)$
    end
    $r^g \leftarrow R(\boldsymbol{x}_0, q)$ `// reward calculation`
  end
  Update π_θ with GRPO using $\{(\tau^g, r^g)\}_{g=1}^G$
until *optimization complete*

Algorithm 2: Sampling with PromptLoop

Input: Policy π_θ, diffusion denoiser $\hat{\boldsymbol{\epsilon}}_\phi$, sampler f, input prompt q, refinement steps $\mathcal{R} \subseteq \{1, \ldots, T\}$

Output: Reward-aligned sample $\boldsymbol{x}_0$

Sample $\boldsymbol{x}_T \sim \mathcal{N}(0, I)$
$\boldsymbol{c} \leftarrow q$
for $t = T, T-1, \ldots, 1$ **do**
  if $t \in \mathcal{R}$ **then**
    $s_t \leftarrow (\hat{\boldsymbol{x}}_t, \boldsymbol{c}, q, t)$
    Sample $\boldsymbol{c} \sim \pi_\theta(\cdot \mid s_t)$ `// prompt refinement`
  end
  Sample $\boldsymbol{z}_t \sim \mathcal{N}(0, I)$
  $\boldsymbol{x}_{t-1} \leftarrow f(\boldsymbol{x}_t, \boldsymbol{z}_t, \boldsymbol{c}, t)$
  $\hat{\boldsymbol{x}}_{t-1} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\big(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\,\hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, \boldsymbol{c})\big)$
end

open the possibility of using the reward identifier as a mechanism for multi-reward alignment in future work. For composite rewards, the increased complexity results in longer prompts, which can hinder the diffusion model's responsiveness. To address this, we employ a dedicated prompt design that explicitly accounts for this issue.

Policy Model Prompt (Single Reward)

User Prompt:
You are helping to refine a prompt for an image generation diffusion model. At each timestep, you are given the input prompt, lastly improved prompt with timestep, current timestep, total timesteps, a target reward function, and the partially generated image at the current diffusion timestep. Your task is to suggest an improved prompt that better aligns with the goal. Do not attempt to correct blurriness, as the partially generated image is expected to be unclear during diffusion.

Respond *only* with a valid JSON object in the following format without any other text:

```
{
  "improved_prompt": "<your improved prompt string>"
}
```

Input:

```
{
  "input_prompt": {input_prompt},
  "last_prompt": {applied_prompt},
  "target_reward": {target_reward},
  "current_timestep": {current_timestep},
  "total_timesteps": {total_timesteps},
}
```

Figure 7. Prompt provided to the policy model for refinement. The instruction specifies the available context (user input, last improved prompt, timestep information, and reward name), and the model must output an improved prompt in JSON format.

C.3. Reward Models

In the single-reward setting, we used ImageReward [60], incompressibility [2], compressibility [2], and aesthetic score models [41] without any modification from their official implementations and checkpoints. For the composite reward in the RePrompt-style setting, we adopted the same components—visual reasoning, length, and structure rewards. The visual reasoning reward consists of ImageReward and an MLLM-based reward, weighted equally, where the latter is implemented with `gpt-5-mini-2025-08-07` [34]. The evaluation prompt for the MLLM reward is shown in Fig. 9. This design complements ImageReward by preventing reward hacking related to weak text alignment and aesthetic biases. The length reward follows the original formulation without change, while the structure reward is adapted to match our output format (JSON). Across all reward components, the scoring ranges and configurations remain unchanged.

C.4. Evaluations

Baselines. We use the official public PyTorch implementations of DDPO[3] and ReFL[4], training them on the same dataset and reward model as PromptLoop. For ReFL on SD1.5, we perform full model fine-tuning, whereas for DDPO and ReFL on SDXL we adopt LoRA-based training. Reported performance values correspond to checkpoints where evaluation rewards match those of PromptLoop. Qwen2.5-VL-3B and GPT-5 (`gpt-5-2025-08-07`) are incorporated without GRPO training, relying solely on prompting (including visual feedback and multi-turn refinement), while maintaining the overall framework. RePrompt is implemented by removing visual feedback and multi-turn refinement from PromptLoop; reasoning is also omitted to ensure fair comparison under equivalent conditions. For Diffusion-DPO[5] and NPNet[6], we directly used their officially released checkpoints and inference code without modification. For DanceGRPO,[7] we reproduce its results using the official training code and dataset, training for 50 epochs with the HPSv2 [58] reward model.
Metrics. For the single-reward setting, we evaluate models using ImageReward [60], HPSv2 [58], and an aesthetic scoring

[3] https://github.com/kvablack/ddpo-pytorch
[4] https://github.com/zai-org/ImageReward
[5] https://github.com/SalesforceAIResearch/DiffusionDPO
[6] https://github.com/xie-lab-ml/Golden-Noise-for-Diffusion-Models
[7] https://github.com/XueZeyue/DanceGRPO

Policy Model Prompt (Composite Reward)

User Prompt:
You are helping to refine a prompt for an image generation diffusion model.

[IMPORTANT] However, you must make *minimal changes* to the original user's input and *keep the prompt as simple as possible*. I *strongly* recommend *not modifying* the input prompt if possible. [IMPORTANT]

Respond *only* with a valid JSON object in the following format without any other text:

```
{
  "improved_prompt": "<your improved prompt string>"
}
```

Input:

```
{
  "input_prompt": {input_prompt},
  "last_prompt": {applied_prompt},
  "target_reward": {target_reward},
  "current_timestep": {current_timestep},
  "total_timesteps": {total_timesteps},
}
```

Figure 8. Prompt provided to the policy model for refinement. The instruction specifies the available context (user input, last improved prompt, timestep information, and reward name), and the model must output an improved prompt in JSON format.

MLLM Reward Model Prompt

User Prompt: You are an expert evaluator of text-to-image alignment. Your primary goal is to check whether the image faithfully matches the input prompt. Pay special attention to object identity, count, attributes (such as color, size, shape), and spatial relationships.
Penalize any elements that are not requested in the prompt — unnecessary decorations, background additions, or irrelevant visual noise. Missing or incorrect objects should also lower the score.
The best images are object-centric: focused on the entities and relationships specified in the prompt, while also being visually coherent and pleasant.

Please rate this image on a scale of 0-10 (10 being perfect) and explain your reasoning. Please put your score in <score> score </score>. Prompt: {p}

Figure 9. Prompt template for the MLLM reward in the RePrompt-style composite setting, guiding fine-grained alignment checks and producing a structured score.

model [41]. These metrics assess prompt alignment, consistency with human preference, and robustness to over-optimization. We follow the standard evaluation protocols provided in the public implementations without any modifications.

In addition, we compute MLLM scores using a pretrained multimodal large language model, Qwen2.5-VL-3B-Instruct [53]. The evaluation is performed locally with carefully designed prompts that balance human-preference alignment and aesthetic quality. Input images are resized to 512×512 before being fed into the model. The evaluator is instructed to provide a score between 0 and 10, with 10 indicating perfect quality. Scores are subsequently normalized to the range $[0, 1]$ during post-processing. The full evaluation prompt is shown in Fig. 10.

For all these metrics, the evaluation prompts are drawn from the validation split of the Pick-a-Pic v2 dataset.

MLLM Score Metric Prompt

User Prompt:
You are an expert image evaluator. Your task is to judge an image based on two equally weighted aspects:

1. *Faithfulness to Prompt*: Does the image accurately reflect the user's input prompt in terms of objects, attributes, style, and composition?
2. *Aesthetic Quality*: Is the image visually appealing, well-composed, and artistically pleasant from a human perspective?

Please rate this image on a scale of 0-10 (10 being perfect) and explain your reasoning. Please put your score in <score> score </score>. Prompt: {prompt}

Figure 10. Evaluation prompt used for computing MLLM scores. The scoring model jointly considers prompt faithfulness and aesthetic quality, and outputs a rating from 0 to 10, which is subsequently normalized to the range [0, 1] in a post-processing step.

In the composite-reward setting, we additionally evaluate on the GenEval benchmark [9], which emphasizes object-centric aspects of text-to-image generation. We directly adopt the prompts and evaluation procedures provided by the GenEval benchmark without modification. When measuring ImageReward and HPSv2, we also use the prompts and the sample counts from GenEval.

D. Prompt Evolution Analysis

D.1. Quantitative Analysis of Prompt Evolution

Since our method controls the sampling dynamics of the diffusion model through textual prompts, the evolution trajectory over diffusion timesteps optimized via reinforcement learning remains interpretable, unlike Hu et al. [16]. To analyze this, we examine the outputs of a policy model trained on SDXL with ImageReward as a single reward signal. Tab. 5 illustrates how the optimized prompts evolve as the diffusion timesteps progress.

Not every case follows the exact same trajectory, but a consistent overall pattern emerges across examples. At early timesteps, prompts typically emphasize meta-level descriptors highlighting quality, style, and realism (*e.g.*, "photorealistic," "vivid colors"), establishing a broad atmospheric framing. As inference advances to intermediate timesteps, these high-level descriptors give way to more concrete and fine-grained details, such as object properties, environmental elements, or specific lighting conditions, resulting in richer and more grounded descriptions. Toward later timesteps, we observe two dominant tendencies: in some cases, prompts continue to preserve the specificity around salient elements of the scene, while in others they collapse back into prototypical atmospheric cues (*e.g.*, "warm glow," "serene atmosphere"). This overall progression—from evaluative abstraction, to concrete specificity, and finally toward either preserved details or prototypical generalities—highlights how reinforcement-learned prompt evolution balances descriptive richness with compact, high-level guidance throughout the diffusion trajectory.

Interestingly, the RL-optimized prompt evolvement trajectory aligns with well-known scheduling strategies of classifier-free guidance (CFG). In diffusion models, it is established that the early steps focus on generating coarse global structures, while later steps refine finer details [63]. Consistent with this, prior studies have demonstrated that applying a strong CFG too early can be harmful, leading to a variety of scheduling strategies. Two dominant families of approaches exist: those that monotonically increase CFG strength throughout the sampling process and those that increase CFG up to intermediate timesteps before decreasing it again toward the final steps [24, 35, 55]. Since stronger CFG effectively enforces sharper and more detailed conditioning, our results suggest that the RL-trained policy implicitly learns both types of dynamics at the textual level, adapting prompt specificity in ways that mirror optimal CFG schedules. This emergent behavior, despite not being explicitly instructed, is intriguing.

Table 5. Comparative analysis of prompt evolvement at different timesteps. Early prompts emphasize broad atmospheric qualities, intermediate prompts expand into concrete details, and later prompts either preserve these specifics or revert to prototypical descriptors.

	Initial ($t = 981.0$)	**Middle** ($t = 581.0$)	**Final** ($t = 181.0$)
Corgi Dog	...corgi wearing a hat and sunglasses, sitting on a beach chair, with a **picturesque beach and ocean in the background**.	...corgi puppy wearing a multicolored bucket hat and sunglasses, sitting on a **plush beach chair** with its paws on the cushion, set against a background of a **vibrant sandy beach, choppy waves, and lush tropical scenery...**	...corgi wearing a colorful straw hat and large sunglasses, sitting on a sunlit beach chair with a **tropical beach landscape, including palm trees and the ocean waves in the background**.
City Night Scene	...lively city street at night with bright lights, towering skyscrapers, and people walking, with **vibrant colors and realistic lighting effects**, in the background there are **numerous illuminated signs and decorations**.	...bustling city street at night with bright lights, tall buildings, and people walking, **realistic-looking photo with vibrant colors and detailed textures**.	...lively city street at night with bright lights, tall buildings with illuminated signs, bustling crowds, and **vibrant city lights surrounding it**, **realistic photo-like scene with warm and inviting glow**.
Mountain View	...stunning mountain landscape with snow-capped peaks, vibrant pine trees, and a clear blue sky, with **stunning lighting and vibrant colors**.	...stunning mountain landscape with snow-capped peaks, vibrant pine trees, a clear blue sky **with fluffy clouds**, realistic photo, **warm sunset lighting**, **beautiful natural scenery**.	...stunning mountain landscape with snow-capped peaks, vibrant pine trees, and a clear blue sky in the background, with **colorful lighting effects** and a **fluffy cloud in the sky**.

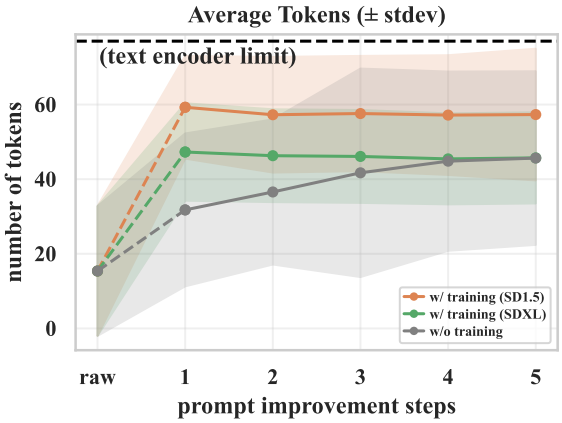


Figure 11. Token counts of raw and refined prompts across improvement steps. Prompt length increases gradually without catastrophic growth or exceeding the text encoder limit (ImageReward task).

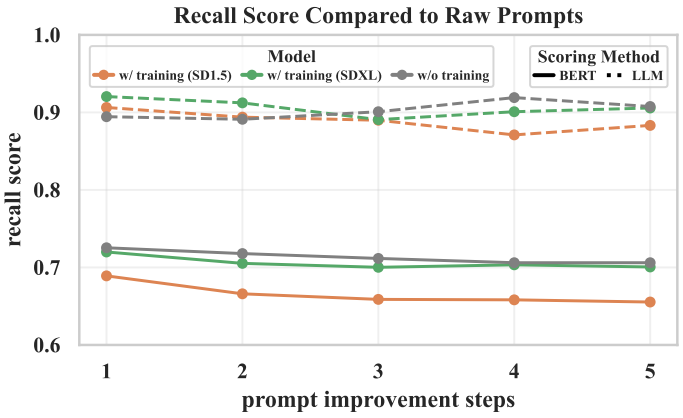


Figure 12. Quantitative analysis of semantic drift during prompt refinement. Overall semantic similarity (BERTScore-recall) shows minor changes, while core user intent (LLM-based recall) is well preserved (1.0 indicates identical semantics; ImageReward task).

D.2. Prompt Length Growth

We further analyze how prompt length evolves under iterative refinement. In our framework, the prompt serves as the sole control signal for the environment (*i.e.*, the diffusion model). Without explicit regularization on prompt length, the policy-refined prompts may continuously grow or exceed the text encoder limit (77 tokens in our setting [37]). Fig. 11 illustrates the change in prompt length (measured in the number of tokens) across prompt improvement steps under a single-reward setting, where no explicit constraint on prompt length is imposed. As expected, the prompt length increases over time since it is the primary control signal. However, reinforcement learning does not lead to catastrophic length inflation nor does it exceed the text encoder limit. Furthermore, we observe that more text-aligned environments (*e.g.*, SDXL) require fewer tokens for precise control, whereas less aligned environments (*e.g.*, SD1.5) tend to rely on longer prompts. This suggests that the growth in prompt length is driven by the capability of the environment rather than instability in the reinforcement learning process.

D.3. Semantic Drift in Iterative Prompt Refinement

Iterative prompt refinement may introduce semantic drift, potentially causing the refined prompt to deviate from the original user intent. However, this effect is mitigated by two key design choices in our framework. First, the policy model has access

Training Setup	Method	Image Reward	HPSv2	Aesthetics	MLLM Score
SDXL & Image Reward	FLUX.1-dev	1.001	**0.286**	6.202	0.741
	+ ours	**1.246**	**0.286**	**6.570**	**0.757**
	SD3.5-large	1.079	**0.288**	5.957	0.729
	+ ours	**1.254**	**0.288**	**6.197**	**0.744**
SD1.5 & Image Reward	FLUX.1-dev	1.001	**0.286**	6.202	0.741
	+ ours	**1.258**	**0.286**	**6.542**	**0.758**
	SD3.5-large	1.079	**0.288**	5.957	0.729
	+ ours	**1.242**	0.287	**6.229**	**0.744**

Table 6. Quantitative results demonstrating zero-shot generalization to recent flow-matching models.

not only to the prompt from the previous step but also to the original prompt at every refinement step, enabling it to preserve the initial intent. Second, the entire refinement process is treated as a single episode in reinforcement learning, where only the final reward is provided. As a result, any deviation from the original intent leads to negative feedback during training, discouraging semantic drift.

To quantitatively analyze this effect, Fig. 12 compares refined prompts with the original prompts using two recall-based similarity metrics: BERTScore-recall [64] (deberta-xlarge-mnli [13]) for overall semantic similarity, and an LLM-based recall metric[8] for core user intent preservation. The results indicate limited semantic change and no significant loss of core intent compared to non-RL prompt refinement, suggesting that reinforcement learning does not exacerbate semantic drift. Consistent with prior observations, more text-aligned environments (*e.g.*, SDXL) exhibit smaller variations, whereas less aligned models (*e.g.*, SD1.5) require larger token-level modifications.

E. Additional Results

E.1. Generalization to Recent Flow-matching Models

To evaluate generalization beyond diffusion models, we assess our trained policy on recent flow-matching text-to-image models that were not seen during training. The policy is trained in a diffusion-model environment (SD1.5 or SDXL with ImageReward) and directly applied to flow-matching model environments without additional fine-tuning. Specifically, we evaluate on SD3.5-large (8B) [7] and FLUX.1-dev (12B) [25]. Quantitative results in Tab. 6 demonstrate that our method generalizes effectively to these unseen environments, despite substantial differences in model scale, architecture, training paradigm, and release date.

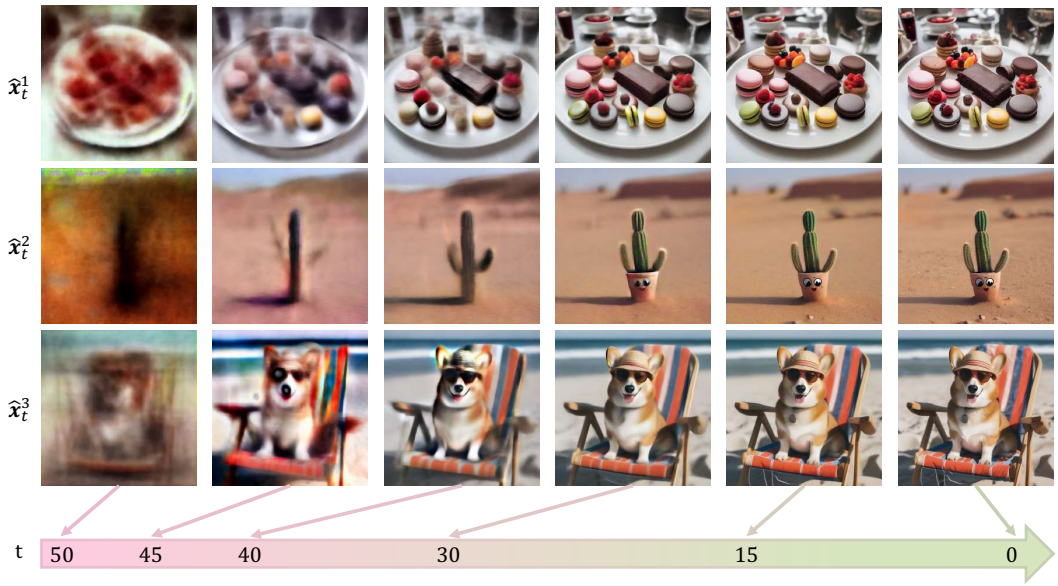
Since flow-matching models employ a different noise parameterization and scheduler from diffusion models, we align their timesteps by matching the noise-to-signal ratio (SNR). Concretely, each flow-matching timestep is converted into a simulated diffusion timestep with an equivalent SNR, which is then provided as input to the policy model. For sampling, we use the default hyperparameters of each flow-matching model. All PromptLoop sampling hyperparameters are kept identical to those in the main experiments, including the sparse refinement strategy applied at evenly spaced intervals in the original timestep domain of the flow-matching models.

E.2. Denoised Estimate as an MLLM input

We feed the visual state $\boldsymbol{x}_t$ to the MLLM policy model in the form of its denoised estimate $\hat{\boldsymbol{x}}_t$. Using $\hat{\boldsymbol{x}}_t$ as an approximation of the final sample $\boldsymbol{x}_0$ is a well-established practice in the diffusion literature [4, 20, 43, 63], regardless of the specific guidance mechanism, energy function formulation, or evaluation function used. For example, denoised estimates are commonly used as inputs to a wide range of energy functions, including measurement-consistency objectives [4, 5], neural network–based energy models [63], and MLLM reward models [20, 43].

As shown in Fig. 13, the denoised estimate $\hat{\boldsymbol{x}}_t$ contains semantically meaningful information even at early diffusion timesteps, despite the inherent blur introduced by its expectation-based formulation. Importantly, these blurred estimates remain interpretable to humans, as coarse object shapes, colors, and global layouts are still recognizable even though fine details are missing. The descriptions generated by the MLLM policy model exhibit a similar behavior. The MLLM can analyze and explain these early denoised estimates in much the same way humans do when presented with a blurred image.

[8] An LLM (Qwen2.5-VL-3B-Instruct) extracts core user-intent concepts from the raw prompt, and we measure the ratio of concepts retained in the refined prompt.



Sample	t	MLLM Generated Description
0	50	The image is blurry and shows **a plate with red berries**, possibly raspberries or strawberries, on **a white plate**.
	45	The image is blurry and shows **a plate with various food items**, including what appears to be fruits and possibly some vegetables or grains.
	40	The image is a blurry photograph of **a plate with various pastries and desserts**, including **macarons and other confections**.
1	50	The image appears to be a blurred, abstract representation of a **landscape with warm colors**, possibly featuring a body of water and some vegetation or terrain.
	45	The image appears to be a blurred photograph of a person's face, with the focus on the nose and mouth area. The background is indistinct, but it seems to be **an outdoor setting** with some greenery visible.
	40	The image depicts **a cactus in a desert landscape** with **reddish-brown sand** and a **clear sky**. The cactus has **a long, thin trunk** with **two small branches** extending from it. The overall scene is arid and typical of a desert environment.
2	50	The image appears to be a blurred photograph of a person **wearing a hat** and a light-colored top, **possibly outdoors** with greenery in the background.
	45	The image depicts **a dog** holding a stack of books, with a **colorful background.** The dog appears to be **wearing sunglasses** and is standing on a table or surface.
	40	The image shows **a cartoon dog wearing sunglasses and a hat**, **sitting on a striped beach chair by the ocean**. The dog appears relaxed and is enjoying a sunny day at the beach.

Figure 13. **Top: Visualization of the denoised estimates along the diffusion sampling trajectory. Bottom: Descriptions generated by the MLLM policy model conditioned on these denoised estimates.** Denoised estimates provide identifiable visual states even at early diffusion timesteps, for both humans and the MLLM policy model.

This empirical evidence indicates that denoised estimates provide identifiable visual states for both humans and the MLLM policy model, which enables them to generate meaningful guidance throughout the diffusion sampling trajectory. In addition, prior work shows that the early phase of diffusion sampling primarily captures the low-frequency structure of the image, such as global shapes and coarse layout [63]. At this stage, high-level textual guidance is especially relevant, and the blurred $\hat{x}_t$ is sufficiently informative to support such guidance.

E.3. Training Dynamics and Stability

Despite employing RL, our training remains stable, as shown in Fig. 14. This stability primarily stems from three factors: (i) initialization from a strong MLLM-based policy, (ii) parameter-efficient updates via LoRA, and (iii) a fixed diffusion environment that interacts with the policy only through prompts, reducing non-stationarity. Combined with GRPO, these

Figure 14. Training dynamics of proposed framework, showing stable optimization. Stronger policy initialization and improved environment lead to faster convergence and more effective reward alignment (ImageReward).

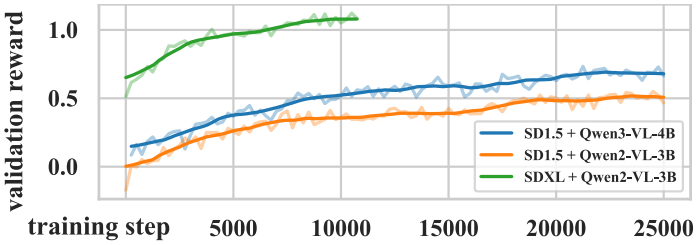


Table 7. Ablation results showing that stronger policy initialization improves reward alignment (SD1.5 & ImageReward).

Policy Model	Image Reward	HPS v2	Aesthetics	MLLM Score
Qwen2.5-VL-3B	0.6320	0.2701	5.853	0.725
Qwen3-VL-4B	**0.6922**	**0.2705**	**6.024**	**0.730**

design choices effectively mitigate common RL instabilities such as oscillatory behavior.

Fig. 14 further reveals that both convergence speed and final reward are largely determined by the quality of the policy initialization and the environment, rather than the intrinsic instability of RL itself. In particular, stronger initial policies lead to faster and more stable optimization trajectories. This trend is quantitatively supported in Tab. 7, where replacing the policy with a more capable model consistently improves all evaluation metrics. These results highlight that policy initialization is a critical factor for achieving reliable and efficient preference alignment.

E.4. More Qualitative Samples

We present qualitative samples corresponding to the quantitative evaluation of single-reward alignment on SD1.5 and composite-reward alignment on SDXL-turbo reported in Tab. 1 and Tab. 2, which could not be included in the main text due to space constraints. Specifically, Fig. 15, 17 illustrates comparisons against baseline reward alignment methods, Fig. 16, 18 highlights the orthogonality of our approach to other reward alignment techniques. The results, consistent with the quantitative findings, demonstrate clear advantages in prompt alignment and human preference, while also highlighting the orthogonality and generalization capability of our approach. In addition to ImageReward as a single-reward task, we also trained models using aesthetic quality [41], compressibility, and incompressibility rewards [2], as shown in Fig. 19. These experiments further demonstrate that our proposed framework can be generally applied across diverse reward types.

F. LLM Usage

Large Language Models (LLMs) were used solely as an editorial aid to improve the clarity and readability of the manuscript. Specifically, LLMs assisted in polishing grammar, refining sentence structure, and ensuring consistency in style. They were not used in any aspect of research ideation, experimental design, data analysis, or in the generation of substantive scientific content. All ideas, results, and interpretations presented in this paper are the responsibility of the authors.

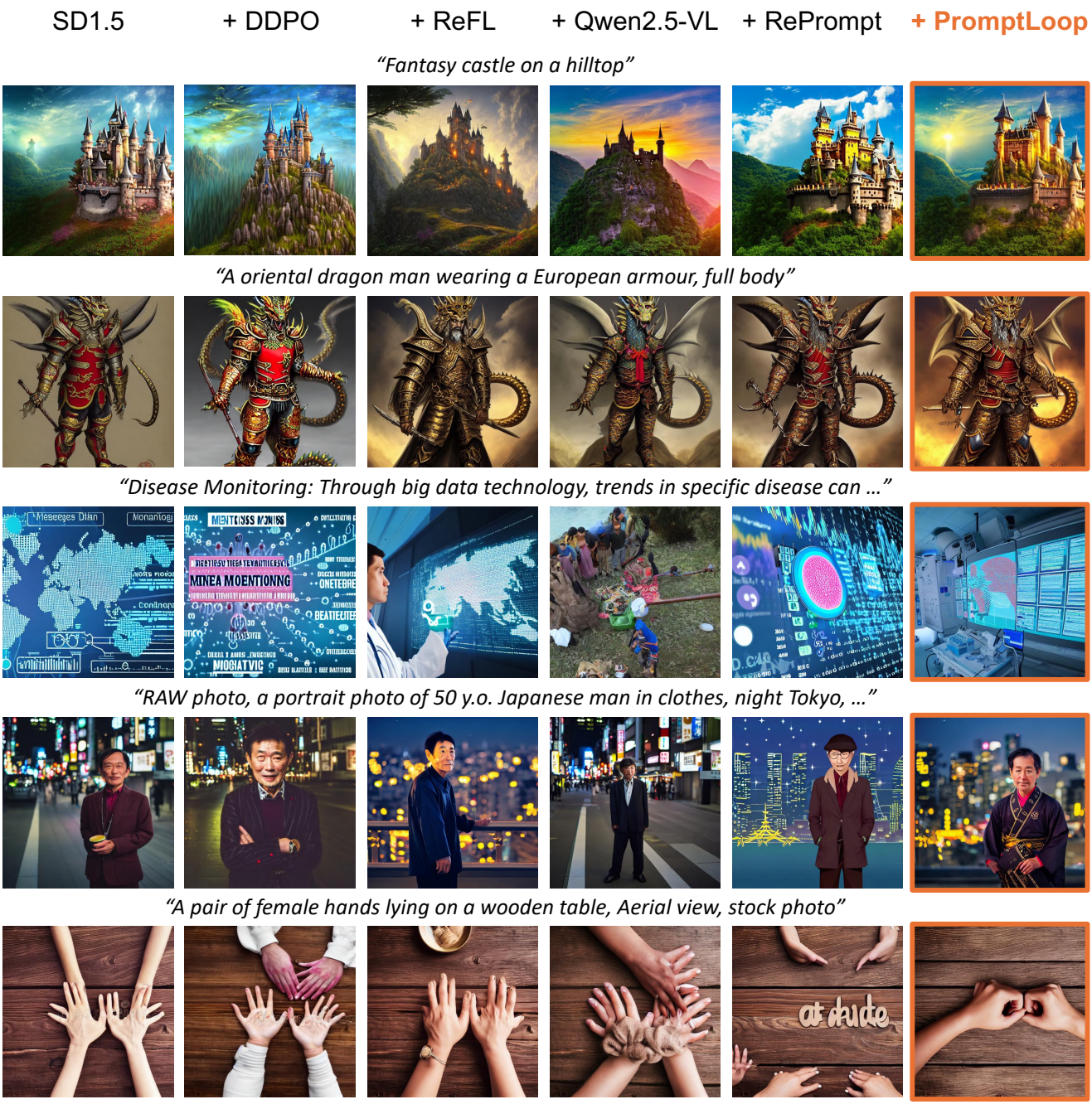


Figure 15. Qualitative comparison of baseline methods (SD1.5 & ImageReward).

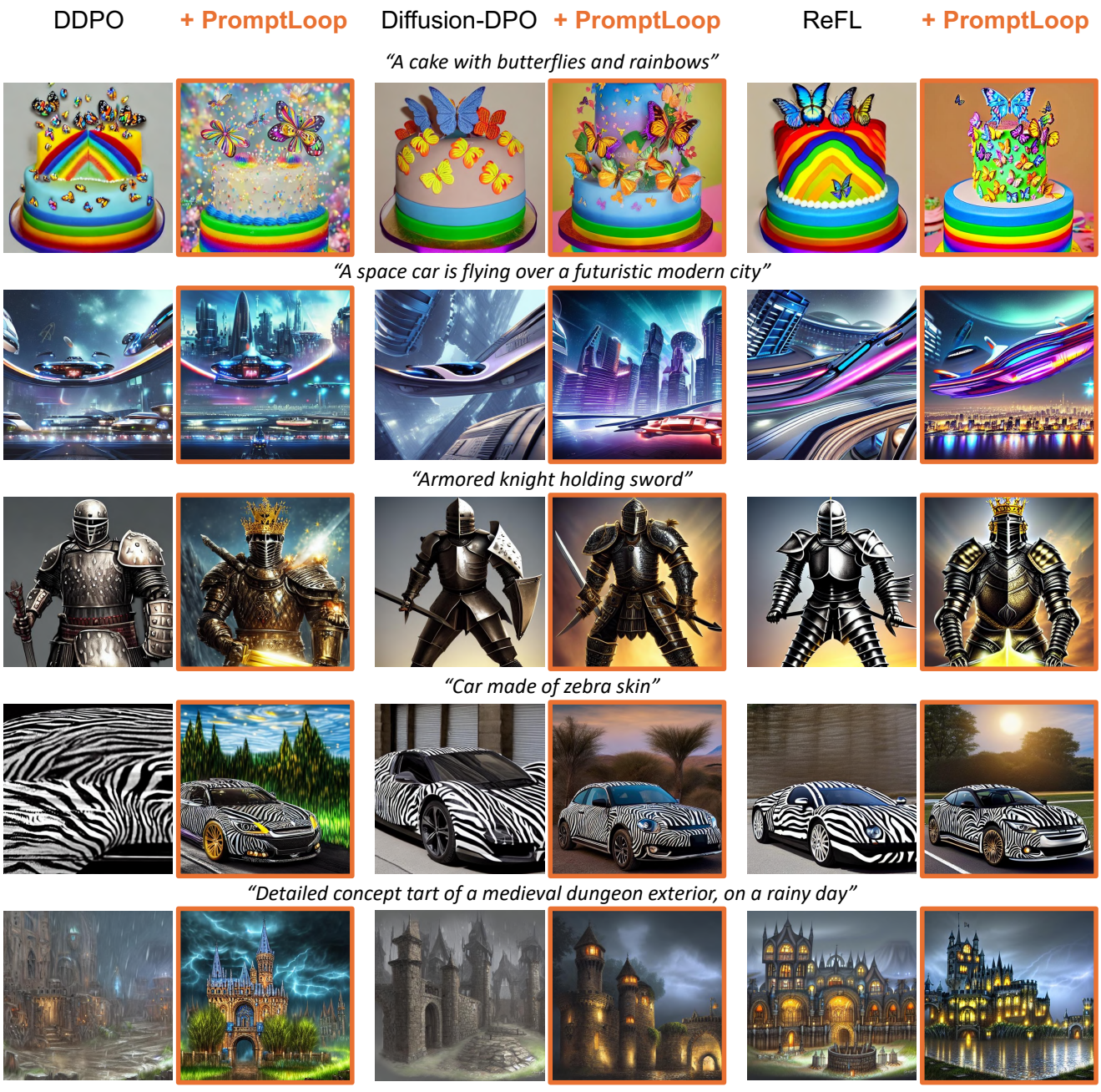


Figure 16. Qualitative results demonstrating the orthogonality of our method compared with reward-aligned baselines (SD1.5 & ImageReward).

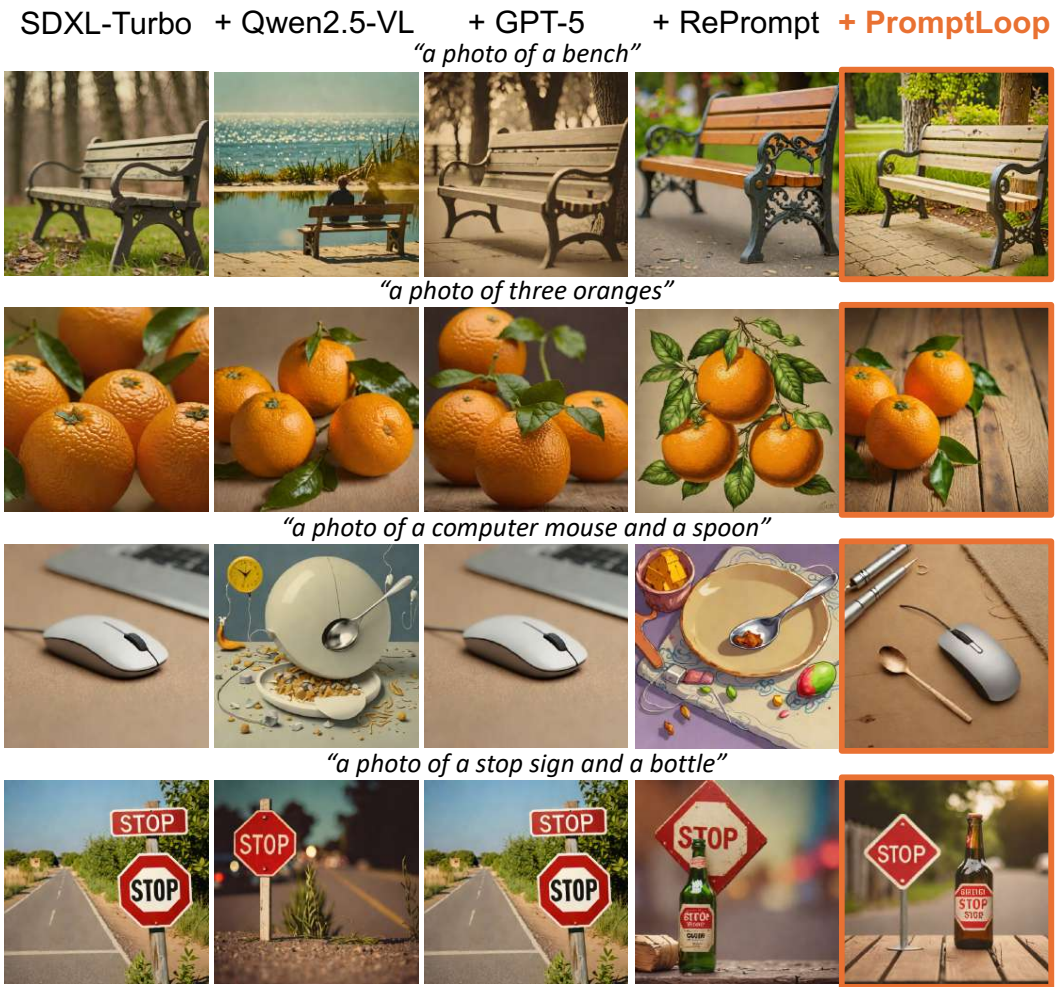


Figure 17. Qualitative comparison of composite-reward alignment, illustrating improvements over baseline methods. (SDXL-turbo & RePrompt)

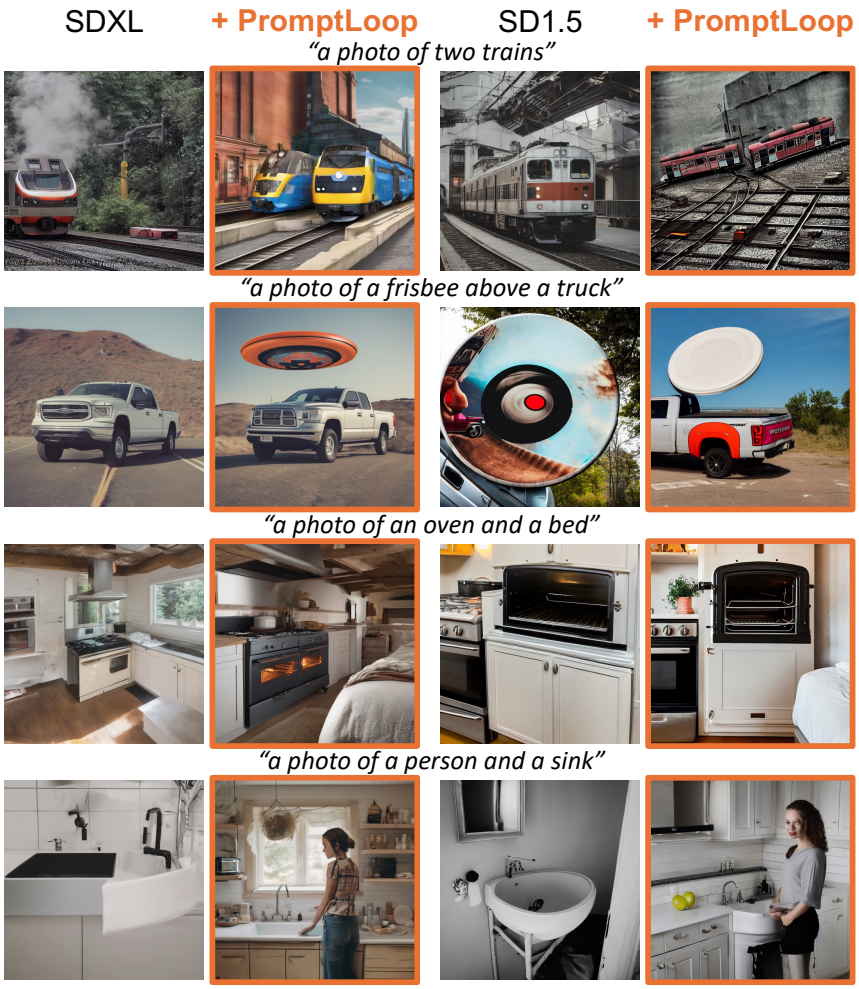


Figure 18. Qualitative results showing the orthogonality and generalizability achieved by applying our method to unseen reward-alignment baselines (SDXL-turbo & RePrompt).

Figure 19. Qualitative results demonstrating the applicability of our framework to diverse reward signals.