# Fine-Tuning Diffusion Generative Models via Rich Preference Optimization

Hanyang Zhao* [1] hz2684@columbia.edu
Haoxian Chen* [2] † haoxianc@amazon.com
Yucheng Guo* [3] yg7348@princeton.edu
Genta Indra Winata[4] genta.winata@capitalone.com
Tingting Ou[1] to2372@columbia.edu
Ziyu Huang[1] zh2532@columbia.edu
David D. Yao[1] ddy1@columbia.edu
Wenpin Tang[1] wt2319@columbia.edu

## Abstract

*We introduce Rich Preference Optimization (RPO), a novel pipeline that leverages rich feedback signals to improve the curation of preference pairs for fine-tuning text-to-image diffusion models. Traditional methods, like Diffusion-DPO, often rely solely on reward model labeling, which can be opaque, offer limited insights into the rationale behind preferences, and are prone to issues such as reward hacking or overfitting. In contrast, our approach begins with generating detailed critiques of synthesized images to extract reliable and actionable image editing instructions. By implementing these instructions, we create refined images, resulting in synthetic, informative preference pairs that serve as enhanced tuning datasets. We demonstrate the effectiveness of our pipeline and the resulting datasets in fine-tuning state-of-the-art diffusion models.*

## 1. Introduction

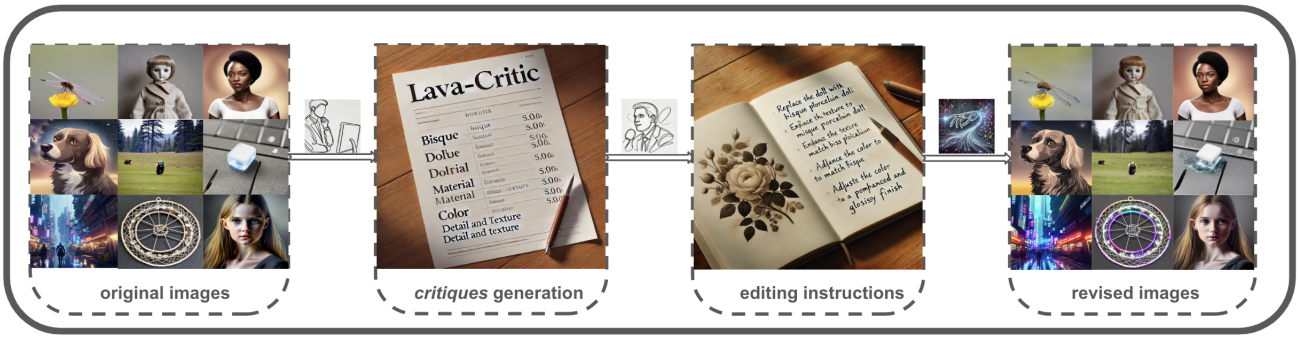
Learning from feedback and critiques is essential for enhancing the performance of a model by guiding the model to rectify unsatisfactory outputs. Improvements arise not only from distinguishing right from wrong, but from receiving thoughtful feedback that offers clear direction for enhancement. For instance, in the natural language tasks, feedback has proved to be useful in code debugging [5], games [11, 25], and agents [29, 32, 38]. Feedback is also found useful in vision tasks, such as visual commonsense reasoning [4, 6, 19].

To effectively leverage feedback in model training, it is crucial to ensure that the feedback is detailed, informative, and nuanced. Simply relying on numerical scores, as reward models often do, falls short in identifying specific areas for model enhancement. Comprehensive feedback provides insights that extend beyond numerical evaluations, allowing for more targeted and substantial model improvements. In this context, exploring the use of critic models can be highly beneficial, as they offer deeper insights into the model's intricacies, and contribute to a more robust understanding of improvement opportunities.

To bridge the gap in preference learning, we draw inspiration from the way students learn from their teachers. In this paper, we introduce Rich Preference Optimization (RPO), a novel approach designed to enhance preference learning for images by leveraging vision-language models (VLMs). These models provide detailed critiques, offering rich feedback that mirrors the comprehensive guidance students receive in modern educational systems. Rather than merely receiving a final score on assignments or exams, students are given specific feedback that identifies logical errors, misunderstandings, or calculation mistakes. This feedback enables students to iteratively refine their initial responses, fostering deeper learning through a process of continuous improvement. Similarly, in RPO, we extract actionable editing instructions from VLMs and employ instruction-driven image-editing models for refinement. This scalable method generates informative preference pairs that are crucial for effective preference learning. The process of receiving detailed feedback and making refinements is akin to learning from true preference pairs, reflecting the natural and effective way in which humans learn. The contribution of this paper is summarized as follows:

- We introduce RPO, a novel approach for generating preference datasets for images by leveraging VLMs to provide detailed critiques as rich feedback. We extract actionable editing instructions from VLMs, and employ instruction-driven image-editing models for refinement. This scalable approach yields informative prefer-

*Equal Contribution. [1]Columbia University. [2]Amazon. [3]Princeton University. [4]Capital One (work done outside Capital One). †Work done during PhD at Columbia.



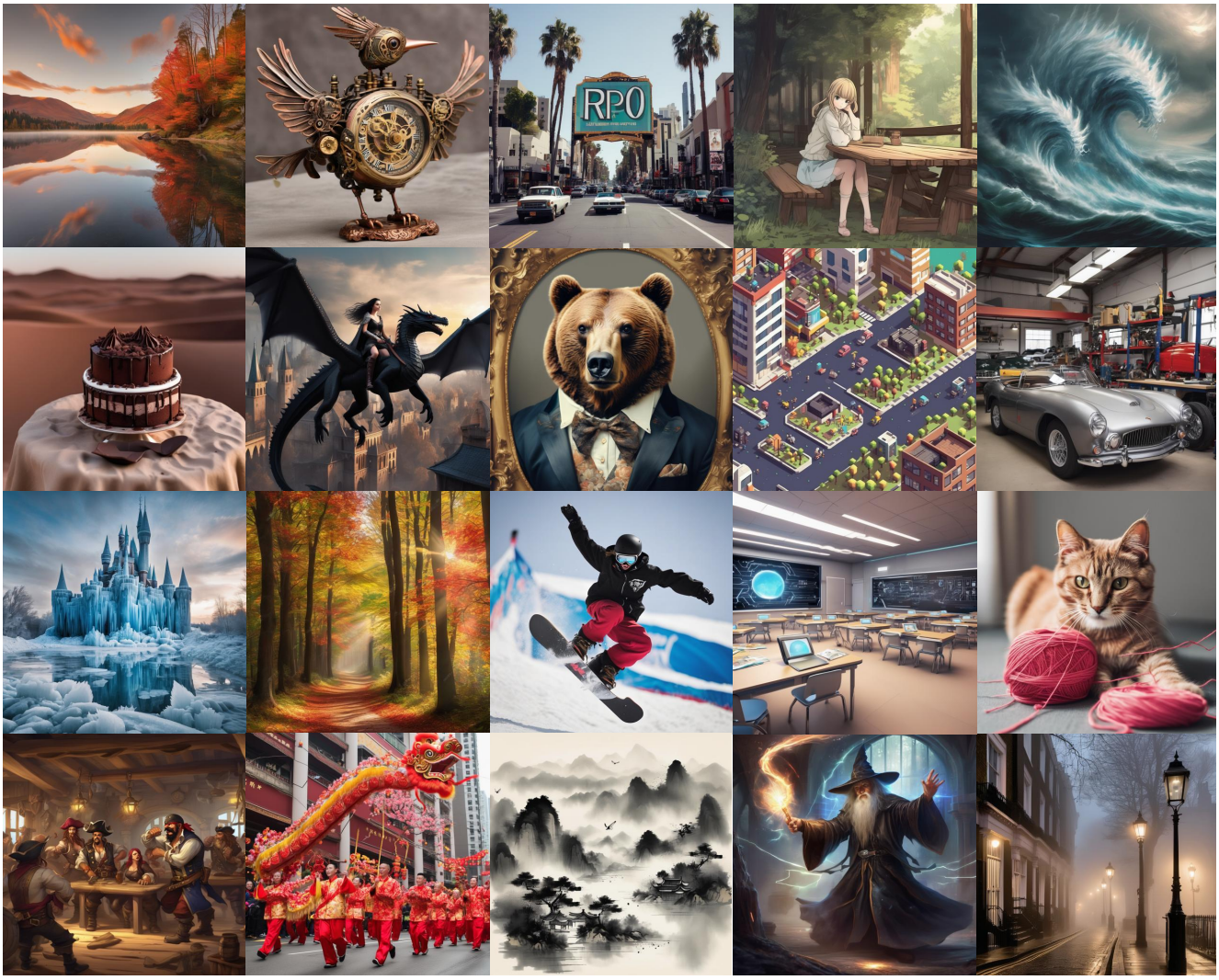




Figure 1. **(Top)** We develop Rich Preference Optimization (RPO), a novel pipeline for curating informative preference pairs from images generated from the base diffusion models, and further aligning the diffusion models from these synthetic preferences. RPO pipeline is composed of 1) Rich Feedback/Critic generation by a Vision Language Model (for which we choose LLaVa-Critic-7B), 2) Actionable editing instruction generation based on the critiques by another VLM (for which we chose Qwen2.5-VL-8B-Instruct), 3) Instruction-following image editing from the generated editing instructions (for which we choose ControlNet), 4) Diffusion DPO training using our further reward model filtered synthetic preference pairs. **(Bottom)** Sample images generated by RPO fine-tuned Stable Diffusion XL.

ence pairs, which enhances preference learning.

- We show that the intermediate critiques can improve the quality of editing instructions than direct generation, which is reminiscent to the Chain-of-Thought concept in mathematical reasoning for LLMs. We also propose to use ControlNet for image-editing by using the original image as the conditional one, which ensures fine-grained control over the image to be revised, while ensuring most of the image to be unchanged.
- We showcase that our synthetic preferences are more efficient than the offline preferences when using Diffusion DPO for preference learning. We also showcase that the synthetic preference datasets that we curate can significantly improve the performance of Diffusion DPO (trained on the original offline dataset) by further learning from rich preferences on the improved images.

Figure 1 provides an illustration of our proposed rich feedback pipeline and the generation quality of our fine-tuned SDXL model.

## 2. Preliminaries

**Diffusion Models**. Diffusion models are a class of generative models $p_\theta(x_0)$, whose goal is to turn noisy/non-informative initial distribution $p_{\text{noise}}(x_T)$ to a desired target distribution $p_{\text{tar}}(x_0)$ through a well-designed denoising process [10, 33, 34]. Here we adopt the discrete-time formulation of diffusion models.

Given noise scheduling functions $\alpha_t$ and $\sigma_t$ (as defined in [30]), the forward process is specified by $q(x_t \,|\, x_s) = \mathcal{N}\left(\frac{\alpha_t}{\alpha_s} x_s,\, \left(\sigma_t^2 - \frac{\alpha_t^2}{\alpha_s^2}\sigma_s^2\right) I\right)$ for $s < t$. Its time-reversed process is a Markov chain parameterized by $p_\theta(x_{0:T}) = \prod_{t=1}^{T} p_\theta(x_{t-1} \,|\, x_t)\, p_{\text{noise}}(x_T)$, where

$$p_\theta(x_{t-1} \,|\, x_t) = \mathcal{N}\left(\mu_\theta(t, x_t),\, \Sigma_t I\right), \tag{1}$$

with $\mu_\theta(t,x) := \frac{\alpha_{t-1}}{\alpha_t} x_t - \left(\frac{\alpha_{t-1}}{\alpha_t} - \frac{\alpha_t \sigma_{t-1}^2}{\alpha_{t-1}\sigma_t^2}\right) \sigma_t \epsilon_\theta(t, x_t)$ by the reparametrization in [10], and $\Sigma_t = \sigma_{t-1}^2 - \frac{\alpha_t^2 \sigma_{t-1}^4}{\alpha_{t-1}^2 \sigma_t^2}$.

The model (1) is trained by minimizing the evidence lower bound (ELBO):

$$\min_\theta \mathcal{L}(\theta) := \mathbb{E}_{t,\epsilon,x_0,x_t}\left[\omega\left(\lambda_t\right) ||\epsilon - \epsilon_\theta(t, x_t)||_2^2\right], \tag{2}$$

where $t \sim \mathcal{U}(0,T)$, $\epsilon \sim \mathcal{N}(0,I)$, $x_0 \sim p_{\text{tar}}(x_0)$, $x_t \sim q(x_t \,|\, x_0) = \mathcal{N}(\alpha_t x_0,\, \sigma_t^2 I)$, $\lambda_t := \frac{\alpha_t^2}{\sigma_t^2}$ is the signal-to-noise ratio, and $\omega : \mathbb{R}_+ \to \mathbb{R}_+$ is some weight function. The training process (2) is also known as denoising score matching (see [36, Section 4.3]). It is expected that for $\theta_*$ solving the optimization problem (2), the model's output distribution $p_{\theta_*}(x_0) \approx p_{\text{tar}}(x_0)$, see [16, 17] for the theory.

**Rich Feedback**. As mentioned, a good critic allows the recipient to learn and improve from the feedback. In T2I generation, Rich Feedback [20] is a VLM, which aims to identify misalignment in a multimodal instruction (i.e., an image-prompt pair), and hence, enriches the feedback. A by-product is the Rich Human Feedback dataset (RichHF-18k), consisting of fine-grained scores, and misalignment image regions and text descriptions on 18K Pick-a-Pic images [13]. However, the Rich Feedback model has not been released.

As an alternative to Rich Feedback, LLaVa-Critic [43] is an open-source VLM that is primarily developed to give evaluation of multimodal tasks. e.g., VLM-as-a-judge and preference learning. It shows a high correlation and comparable performance to proprietary GPT models (GPT-4V/4o). In our approach, we use LLaVa-Critic as an VLM-as-a-judge: the input is a text-image pair, and the output is a critic to image-prompt misalignment. Following the Chain-of-Thought concept [40], such obtained critic will be subsequently passed to an open-source LLM to provide an editing instruction. Our experiment shows that the proposed open-source critic + editing pipeline yields more reliable improvements than directly querying a VLM, e.g., GPT4o, for editing instruction. See Section 3.2 for examples of critic + editing instruction.

**Instruction-following Image Editing Models**. The image-editing part in our pipeline is related to the literature of instruction-following image-editing models. We focus on diffusion-based models in this paper due to both their advantage over autoregressive models, and that our base models for fine-tuning are also diffusion-based. The pioneer models include InstructPix2Pix (IP2P) [3], which first enables editing from instructions that inform the model which action to perform. Follow-up works include Magic Brush [45], Emu Edit [31], UltraEdit [48] and HQEdit [12] by introducing additional datasets for further fine-tuning based on IP2P and enhancing the performance. Existing work like HIVE [47] has also considered to align IP2P with human feedback to enhance generation capability.

As previously mentioned, we edit images based on ControlNet [46] to ensure a better coherence to the original image. ControlNet has been widely used for controlling image diffusion models by conditioning the model with an additional input image. Further applications include implementations for the state-of-the-art proprietary models like Stable Diffusion 3, Stable Diffusion XL [8] and FLUX [15], multi-image support extensions like ControlNet++ [18]. We will stick to the original ControlNet implementation in this paper (because the offline dataset is generated by the similar model scaled by Stable Diffusion v1.5), and leave the usage of more advanced ControlNet variants in future work.

**Diffusion-DPO**. Direct Preference Optimization (DPO) [28] has been an effective approach for learning from human preference for language models. [37] proposed Diffusion-DPO, a method to align diffusion models to human preferences by directly optimizing on human comparison data. Here we follow the definition from [37].

The idea of Diffusion-DPO is to generate $x_{0:T}$ given the conditional input $c$ (the prompt), and a win-loss pair $(x_0^w, x_0^l)$. Let $\beta > 0$ be the temperature parameter, and $p_{\text{ref}}(x_{0:T}|c)$ be a (pretrained) reference model. The Diffusion-DPO objective is:

$$L_{\text{DPO}}(\theta) = -\mathbb{E}_{c,x_0^w,x_0^l} \log \sigma \Bigg( \beta \mathbb{E}_{\substack{x_{1:T}^w \sim p_\theta(x_{1:T}^w|x_0^w,c) \\ x_{1:T}^l \sim p_\theta(x_{1:T}^l|x_0^l,c)}} \left[ \log \frac{p_\theta(x_{0:T}^w|c)}{p_{\text{ref}}(x_{0:T}^w|c)} - \log \frac{p_\theta(x_{0:T}^l|c)}{p_{\text{ref}}(x_{0:T}^l|c)} \right] \Bigg).$$

However, this objective function is not easy to train. By Jensen's inequality,

$$L_{\text{DPO}}(\theta) \leq -\mathbb{E}_{\substack{x_0^w,x_0^l,t \\ x_{t-1,t}^w \sim p_\theta(x_{t-1,t}^w|x_0^w,c) \\ x_{t-1,t}^l \sim p_\theta(x_{t-1,t}^l|x_0^l,c)}} \log \sigma \Bigg( \beta T \left[ \log \frac{p_\theta(x_{t-1}^w|x_t^w,c)}{p_{\text{ref}}(x_{t-1}^w|x_t^w,c)} - \log \frac{p_\theta(x_{t-1}^l|x_t^l,c)}{p_{\text{ref}}(x_{t-1}^l|x_t^l,c)} \right] \Bigg),$$

the right hand side of which allows for efficient training via stochastic gradient descent. However, as [37] pointed out, sampling the reverse process $p_\theta(x_{1:T}|x_0, c)$ is still intractable. So the idea is to approximate $p_\theta(x_{1:T}|x_0, c)$ by the forward process $q(x_{1:T}|x_0)$, which yields the following loss function (which is used as the loss for Diffusion-DPO training):

$$\begin{aligned} L(\theta) = -\mathbb{E}_{x_0^w,x_0^l,t,x_t^w \sim q(x_t^w|x_0^w),x_t^l \sim q(x_t^l|x_0^l)} \log \sigma \Bigg( \beta T \omega(\lambda_t) \\ \Big( -\|\epsilon^w - \epsilon_\theta(x_t^w,t)\|_2^2 + \|\epsilon^w - \epsilon_{\text{ref}}(x_t^w,t)\|_2^2 \\ + (\|\epsilon^l - \epsilon_\theta(x_t^l,t)\|_2^2 - \|\epsilon^l - \epsilon_{\text{ref}}(x_t^l,t)\|_2^2) \Big) \Bigg), \end{aligned} \tag{3}$$

where $x_t^{w,l} \sim q(x_t^{w,l} \mid x_0^{w,l}) = \mathcal{N}(\alpha_t x_0^{w,l} \sigma_t^2 I)$, and $\epsilon^{w,l} \in \mathcal{N}(0, I)$. More details are discussed in [37] and its supplementary materials.

# 3. Curating Preference Pairs with Rich Feedback Signals

In this section, we present the concrete components in our pipeline for creating preference pairs. We utilize 1.6k rows of prompts and images from the test set of RichHF dataset provided by [20] as the validation set for the ablation study. We first discuss instruction-following editing (despite it being the last part before preference tuning in our pipeline), and then use the best instruction-following image-editing model that we find for the ablation study on the first two components: utilizing multimodal models/VLMs for generating feedback information, and providing concise and actionable editing instructions.

## 3.1. Instruction-Following Image Editing

Despite the numerous proposed image-editing methods, either based on diffusion models or not in the literature, we find that they struggle to perform fine-grained control to follow specific editing instructions, which is crucial to generate images that are direct improvements over the existing ones. Existing methods usually change the image to another one which may yield higher score but looks fundamentally different.

To tackle the above issue, we propose a ControlNet-based image-editing approach by adopting ControlNet-based models. Concretely, we exploit the image2image (pipeline) of the ControlNet by setting the conditional image to be the same as the original one, which guarantees that the changes adhere to the original image. In addition, we concatenate the prompt that generates the image with our generated editing instructions, which we find will greatly enhance the quality of the edited image generation. This is illustrated in Figure 2.

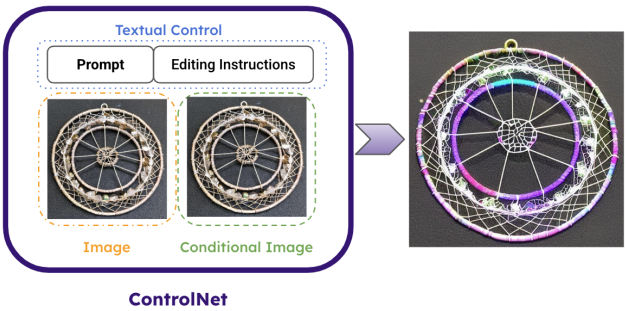



Figure 2. We utilize ControlNet by using the same input image as the conditional image, and concatenating the prompt with the editing instruction as the textual control.

We qualitatively showcase the results in Figure 3, featuring two images (with their associated prompts) and two sets of editing instructions. Additionally, we quantitatively assess the instruction-following capabilities of various diffusion-based image-editing models. We also examine the impact of incorporating prompts before editing instructions, using GPT4o for pairwise comparisons on both a standard instruction-following dataset and a test set. Our findings indicate that ControlNet-based editing is particularly effective. However, we highlight that the choice of image-editing model is flexible and can be updated to include more advanced options, such as ControlNet SDXL

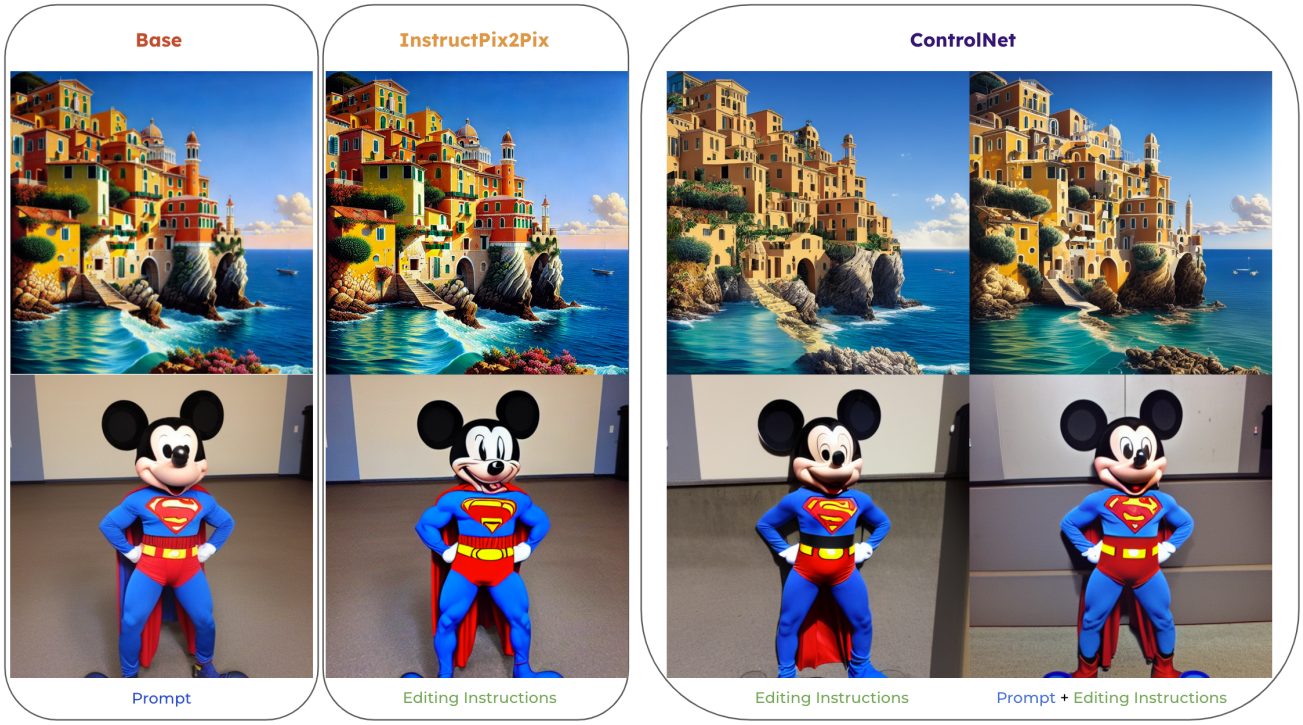



Figure 3. We compare ControlNet [46] with InstructPix2Pix and also ablate the necessity of concatenating the prompt with the editing instructions, which are generated by ChatGPT 4o. **(Top)**: The prompt is *"Italian coastline, buildings, ocean, architecture, surrealism by Michiel Schrijver."* The editing instruction is *"Incorporate iconic Italian elements like olive trees or Vespa scooters. Enhance the coastline with more distinct Mediterranean features. Add intricate architectural details typical of Italian structures. Intensify the ocean's depth with gradient blues, and ensure the surrealism reflects Michiel Schrijver's style by blending dreamlike elements."* In this case, InstructPix2Pix struggles to make any fine-grained modifications. **(Bottom)**: The prompt is *"Mickey Mouse in a Superman outfit bodybuilding."* The editing instruction is *"Adjust the character to have Mickey Mouse's face, including distinct ears, in a Superman outfit. Include bodybuilding elements such as visible muscles or weights. Ensure the outfit is accurate with the Superman logo prominently displayed."* In this case, InstructPix2Pix distorts the image. In both cases, adding the image prompt to the editing instruction for the final instruction yields better results in terms of following the instructions while keeping most of the original image unchanged.

and other variants. We plan to explore these options in future work.

### 3.2. Generating Rich Feedback Signals

Unlike RichFB [20] that requires to train a model to detect the heatmaps and misaligned words and to make the reward score more accurate, we leverage the textual feedback on the quality of generation, like a movie critic, as in Figure 5.

To compare and ablate about what type of feedback will be most useful for us to generate better editing instruction, we use the SOTA multimodal model GPT-4o to generate the instructions on how to improve the image and use ControlNet that we dicussed earlier to edit the image. GPT-4o generates editing instructions based on various types of feedback, including those from RichFB, Llava-Critic, and ChatGPT-4o itself. From the RichFB dataset, we use the image, prompt, misalignment information within the prompt, and the misalignment heatmap of the image as inputs. From Llava-Critic, we incorporate its textual feedback. For ChatGPT-4o, we first prompt it to generate textual feedback based on the image-prompt pair, which is then used to derive editing instructions. For comparison, we also generate editing instructions directly from the input image

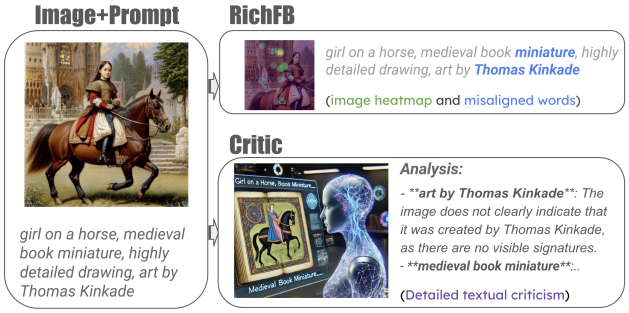
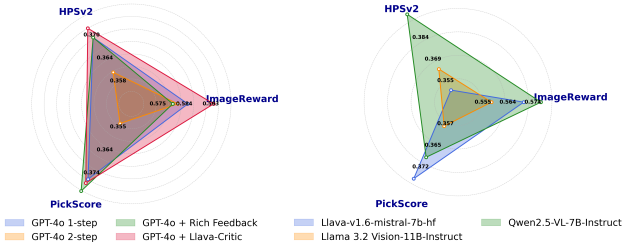



Figure 4. Comparison of RichFB generated informative feedback and our adopted textual criticism generate by carefully prompting a capable VLM.

and prompt, bypassing the intermediate step of generating feedback from the image-prompt pair.

Quite surprisingly, as in Figure 5a, we found that GPT-4o yield better editing instructions when generating directly than first reasoning or critique about the misalignment between prompt and image itself. However, GPT-4o generates much better editing instructions conditioning on the critiques generated by LLaVA-critique. So we stick to using LLaVA-critique for our pipeline.




(a) Comparison of RichFB generated informative feedback and our adopted textual criticism generated by carefully prompting a capable VLM.

(b) Comparison of open-source VLMs combined with Llava-Critic in image editing quality, showing that Qwen2.5-VL-7B-Instruct is the most capable VLM model.

Figure 5. Comparisons of feedback approaches and VLM performance for enhanced image editing, evaluated by ImageReward, HPSv2 and PickScore.

### 3.3. Generating Editing Instructions

We also compare the performance of Llama 3.2 Vision 11B Instruct [9], Llava-v1.6 Mistral 7B [21], and Qwen2.5 VL 7B Instruct [39] generated editing instructions, in combined with Llava-Critic and ControlNet based editing as we argued earlier. As shown in Figure 5b, Qwen2.5-VL-7B-Instruct yield the best results in leading to highest rewards in both HPSv2 and ImageReward after adopting the image-editing instructions. Thus we choose Qwen2.5-VL-7B-Instruct as the VLM for editing instruction generation in our RPO pipeline.

### 3.4. Reward Model Relabeling

After obtaining the pair of original and edited images, we rearrange them into preferences by further querying a reward model or an LLM-as-a-judge. On the test set of size 16K, we find that roughly 60% of our images yield a higher score than the original image under the ImageReward [44] metric. As a remark, it is possible to use RL fine-tuning methods to encourage the VLM to generate better editing instructions, which may get a higher score of edited images; we leave this for future work. Nevertheless, the edited images that fail to have higher scores than the original ones can still serve as the non-preferred images, and hence, yield the preference pair. To conclude, our RPO pipeline provides a generic and training-model-free way to generate preference pairs, because we do not need extra generations from the base model. We further use our curated dataset for fine-tuning large-scale SOTA diffusion models.

## 4. Experiments

In this section, we evaluate our RPO pipeline by first fine-tuning different baseline models on our synthetic preferences data using the Diffusion-DPO algorithm, and then utilizing a couple of reward models to score the images generated by the fine-tuned models.

### 4.1. Settings

**Baseline Models.** We use SD1.5 [30] and SDXL-1.0 [26] as our starting point. We create the following checkpoint models by Diffusion-DPO fine-tuning the two base models: (a) DPO-SD1.5-100k, which is fine-tuned from SD1.5 using the first 100k rows of the Pick-a-Pic v2 dataset; (b) DPO-SD1.5-200k, which is fine-tuned from SD1.5 using the first 200k rows of the Pick-a-Pic v2 dataset; (c) DPO-SD1.5-100k (ImageReward-Aligned), which is fine-tuned from SD1.5 using the first 100k rows of the Pick-a-Pic v2 dataset, with the preference modified by the relative order of ImageReward scores; (d) DPO-SDXL-100k, which is fine-tuned from SDXL-1.0 using the first 100k rows of the Pick-a-Pic v2 dataset.

**Produced Models.** To evaluate our curated dataset, we produce the following models by Diffusion-DPO fine-tuning the baseline models using our 100k synthetic preferences data: (a) DPO-SD1.5-100k+RPO100k, which is fine-tuned from model DPO-SD1.5-100k; (b) DPO-SD1.5-100k (ImageReward-Aligned)+RPO100k, which is fine-tuned from model DPO-SD1.5-100k (ImageReward-Aligned); (c) DPO-SDXL-100k + RPO100k, which is fine-tuned from model DPO-SD1.5-100k (ImageReward-Aligned).

**Diffusion-DPO Training** For Diffusion-DPO training, we follow the same setting and use the same hyperparameters in [37] when fine-tuning SD1.5-based models. More specifically, we use AdamW [22] as the optimizer; the effective batch size is set to 2048; we train at fixed square resolutions, and use a learning rate of $\frac{2000}{\beta} 2.048 \cdot 10^{-8}$ with $25\%$ linear warmup; for the divergence penalty parameter, we keep $\beta = 5000$. When fine-tuning SDXL-based models, we use the LoRA implementation provided by the Github project [35] to optimize training efficiency.

**Evaluation.** We generate images using the prompts from the Pick-a-Pic test set [14] (which contains 500 unique prompts) and evaluate the generation with reward models including PickScore [14], ImageReward [44], HPSv2 [42] and LAION-Aesthetic Predictor (a ViT-L/14 CLIP model

Table 1. Model performance of the Pick-a-Pic Test Set on four different metrics.

| Model | PickScore | ImageReward | Aesthetic | HPSv2 |
|---|---|---|---|---|
| SD1.5 | 20.33 | 0.1733 | 5.949 | 0.2622 |
| DPO-SD1.5-100k | 20.66 | 0.2784 | 6.044 | 0.2650 |
| DPO-SD1.5-200k | 20.74 | 0.3638 | 6.088 | 0.2657 |
| DPO-SD1.5-100k + RPO100k | **20.75** | **0.4395** | **6.113** | **0.2663** |
| DPO-SD1.5-100k (ImageReward-Aligned) | 20.45 | 0.3913 | 6.097 | 0.2645 |
| DPO-SD1.5-100k (ImageReward-Aligned) + RPO100k | **20.54** | **0.5252** | **6.105** | **0.2660** |
| SDXL1.0 | 21.74 | 0.8473 | 6.551 | 0.2692 |
| DPO-SDXL-100k | 21.92 | 0.9183 | 6.566 | 0.2706 |
| DPO-SDXL-100k + RPO100k | **21.89** | **0.9353** | **6.585** | **0.2707** |

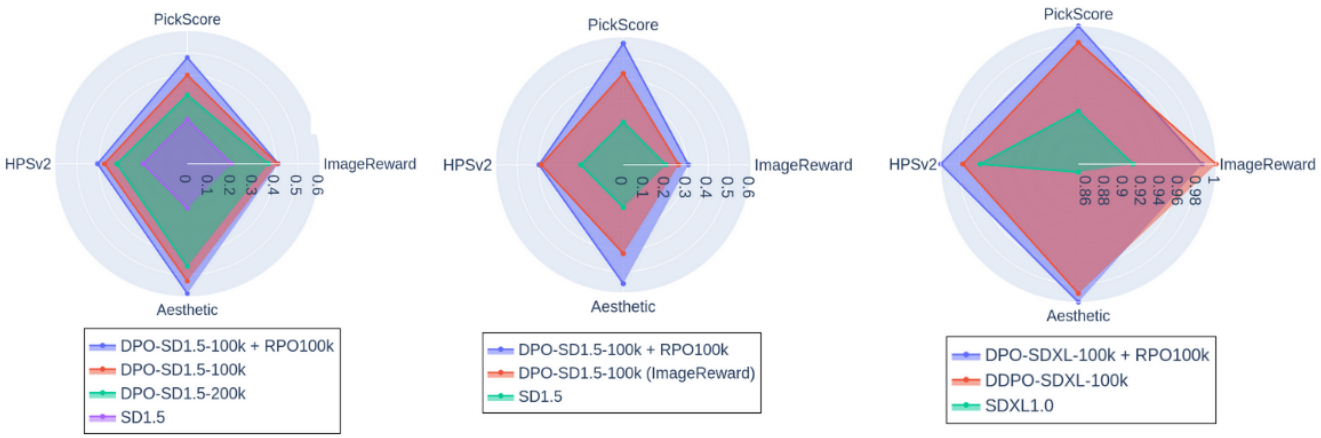



Figure 6. Model performance evaluation (normalized) on PickScore, ImageReward, Aesthetic, and HPSv2.

trained with SAC dataset [27]). The PickScore, ImageReward, HPSv2 reward models are used to evaluate human-preference alignment, and LAION-Aesthetic Predictor is expected to evaluate visual aesthetic appeal. For each model, we report the average scores over all prompts.

### 4.2. Results and Analysis

We present the performance of all the models, as measured by the reward models, in Table 1. Above all, models obtained by fine-tuning on our 100k synthetic preferences data achieve comparable levels of the baseline models by all metrics and outperform the corresponding baseline models in most cases, which confirms our hypothesis that the pipeline we designed for synthesizing preference pairs has the potential to yield better results when combined with preference-based training algorithms such as Diffusion-DPO.

Comparing the fine-tuned model DPO-SD1.5-100k+RPO100k with the baseline model DPO-SD1.5-200k, we note that our pipeline leads to a higher data efficiency for Diffusion-DPO training, in the sense that the former model utilizes a mixture of 100k human-labeled data and 100k synthetic data, while the latter uses 200k human-labeled data, which is naturally considered to be of higher quality. With the same reasoning, we do not directly compare models fine-tuned based on our synthetic preferences data with models fine-tuned based on human-labeled data, as our focus is on verifying the ability of the synthetic preferences data to augment given (high-quality) dataset and enhance the performance of fine-tuning.

We also present a qualitative comparison of the three SDXL-based models in Figure 7. Although the RPO fine-tuned model does not demonstrate, generally speaking, any significant improvement in terms of visual quality, we remark that it is better at connecting different elements in the prompts in a deep and profound way (see the first and the third columns). Understanding how rich feedback and guided revision help fine-tuning diffusion models will be left for future work.


Figure 7. A comparison of generations made by SDXL1.0, DPO-SDXL-100k, and DPO-SDXL-100k+RPO100k. Prompts (from left to right): "tiger wearing casual outfit", "an adventurer walking along a riverbank in a forest during the golden hour in autumn", "samurai pizza cat", "anime portrait of a beautiful vamire witch, sci fi suit, intricate, elegant, highly detailed, digital painting, artstation, concept art, smooth, sharp focus, illustration, art by grep rutkowski and" (truncated due to the limit of number of tokens). The prompts are from the Pick-a-pic dataset [14].

## 5. Discussions

In contemporary RLHF pipelines for LLMs, preference pairs are generated by sampling various responses and subsequently ranking them using either human evaluators or pretrained reward models, which serve as AI-based labels. This approach is widely utilized in both reinforcement learning (RL)-based techniques [2, 24] and offline methods, such as DPO [28], SimPO [23], and RainbowPO [49], for learning preferences in LLMs. Additionally, it is applied in Diffusion-DPO [37] for diffusion models. See [41] for a comprehensive review.

However, the resulting preferences often lack transparency because reward scores are typically black-box, and they are also not very informative, as they provide minimal insight into why a particular choice (whether an answer or a generated image) is preferred. Consequently, while the current preference curation pipeline is scalable, it is less efficient for aligning the model or agent based on these preferences. This inefficiency can even lead to issues like reward hacking, as noted in existing literature [1, 7]. Addressing the challenge of creating high-quality synthetic preferences for generative models after training is a crucial question that needs to be answered. Furthermore, although not explored in this paper, our pipeline can be readily adapted to online algorithms, such as iterative DPO or reinforcement learning-based methods. We consider this as a promising avenue for future research.

## 6. Conclusion

In this work, we present **Rich Preference Optimization (RPO)**, a method that utilizes rich feedback about the prompt image alignment to improve the curation of synthetic preference pairs for fine-tuning text-to-image diffusion models. After extracting actionable editing instructions from VLMs, we employ ControlNet to modify the images, thereby producing a diverse range of refined samples. The

edited images are then combined with the original versions and undergo a relabeling process using a reward model to create a curated set of preference data. By further fine-tuning checkpoint models on this synthetic dataset, we significantly enhance the performance of Diffusion-DPO training and achieve greater data efficiency. Moreover, we believe that our pipeline represents a promising direction and avenue for the data curation of synthetic vision language preference data, one that holds significant potential for future research advancements.

**Acknowledgement**: Haoxian Chen is supported by the Amazon CAIT fellowship. Tingting Ou is supported by NSF grants CNS-2138834 and EEC-2133516. Wenpin Tang is supported by NSF grant DMS-2206038, the Columbia Innovation Hub grant, and the Tang Family Assistant Professorship. The works of Haoxian Chen, Hanyang Zhao, and David Yao are part of a ColumbiaCityU/HK collaborative project that is supported by InnoHK Initiative, The Government of the HKSAR and the AIFT Lab.

## References

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. arXiv preprint arXiv:1606.06565, 2016. 8

[2] 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. 8

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In CVPR, pages 18392–18402, 2023. 3

[4] Jiali Chen, Xusen Hei, Yuqi Xue, Yuancheng Wei, Jiayuan Xie, Yi Cai, and Qing Li. Learning to correction: Explainable feedback generation for visual commonsense reasoning distractor. In MM, pages 8209–8218, 2024. 1

[5] Xinyun Chen, Maxwell Lin, Nathanael Schaerli, and Denny Zhou. Teaching large language models to self-debug. In ACL, 2023. 1

[6] Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. arXiv preprint arXiv:2411.00855, 2024. 1

[7] Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, and Jared Kaplan. Sycophancy to subterfuge: Investigating reward-tampering in large language models. arXiv preprint arXiv:2406.10162, 2024. 8

[8] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, and Frederic Boesel. Scaling rectified flow transformers for high-resolution image synthesis. In ICML, 2024. 3

[9] 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. 6

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Neurips, pages 6840–6851, 2020. 3

[11] Frederikus Hudi, Genta Indra Winata, Ruochen Zhang, and Alham Fikri Aji. Textgames: Learning to self-play text-based puzzle games via language model reasoning. arXiv preprint arXiv:2502.18431, 2025. 1

[12] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. arXiv preprint arXiv:2404.09990, 2024. 3

[13] 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. In Neurips, pages 36652–36663, 2023. 3

[14] 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. In Neurips, pages 36652–36663, 2023. 6, 8

[15] Black Forest Labs. Flux.1 [dev]: A 12 billion parameter rectified flow transformer for text-to-image generation, 2024. Accessed: 2025-03-05. 3

[16] Gen Li and Yuling Yan. $O(d/t)$ convergence theory for diffusion probabilistic models under minimal assumptions. In ICLR, 2025. 3

[17] Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. In ICLR, 2024. 3

[18] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. Controlnet++: Improving conditional controls with efficient consistency feedback: Project page: liming-ai. github. io/controlnet_plus_plus. In ECCV, pages 129–147, 2024. 3

[19] Zhiyuan Li, Dongnan Liu, Chaoyi Zhang, Heng Wang, Tengfei Xue, and Weidong Cai. Enhancing advanced visual reasoning ability of large language models. arXiv preprint arXiv:2409.13980, 2024. 1

[20] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, and Feng Yang. Rich human feedback for text-to-image generation. In CVPR, pages 19401–19411, 2024. 3, 4, 5

[21] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In CVPR, pages 26296–26306, 2024. 6

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In ICLR, 2019. 6

[23] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In Neurips, pages 124198–124235, 2025. 8

[24] 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. In Neurips, pages 27730–27744, 2022. 8

[25] Davide Paglieri, Bartłomiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wolczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, et al. Balrog: Benchmarking agentic llm and vlm reasoning on games. arXiv preprint arXiv:2411.13543, 2024. 1

[26] 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. In ICLR, 2024. 6

[27] John David Pressman, Katherine Crowson, and Simulacra Captions Contributors. Simulacra aesthetic captions. Technical Report Version 1.0, Stability AI, 2022. url https://github.com/JD-P/simulacra-aesthetic-captions . 7

[28] 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. In Neurips, pages 53728–53741, 2023. 4, 8

[29] Matthew Renze and Erhan Guven. Self-reflection in llm agents: Effects on problem-solving performance. arXiv preprint arXiv:2405.06682, 2024. 1

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In CVPR, pages 10684–10695, 2022. 3, 6

[31] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In CVPR, pages 8871–8879, 2024. 3

[32] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In Neurips, pages 8634–8652, 2023. 1

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In ICLR, 2021. 3

[34] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In ICLR, 2021. 3

[35] Suzukimain and Steven Liu. Github repository: Diffusion model alignment using direct preference optimization. Github, 2024. 6

[36] Wenpin Tang and Hanyang Zhao. Score-based diffusion models via stochastic differential equations–a technical tutorial. arXiv preprint arXiv:2402.07487, 2024. 3

[37] 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 CVPR, pages 8228–8238, 2024. 4, 6, 8

[38] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. arXiv preprint arXiv:2305.16291, 2023. 1

[39] 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. 6

[40] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Neurips, pages 24824–24837, 2022. 3

[41] Genta Indra Winata, Hanyang Zhao, Anirban Das, Wenpin Tang, David D Yao, Shi-Xiong Zhang, and Sambit Sahu. Preference tuning with human feedback on language, speech, and vision tasks: A survey. arXiv preprint arXiv:2409.11564, 2024. 8

[42] 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. 6

[43] Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. arXiv preprint arXiv:2410.02712, 2024. 3

[44] 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 Neurips, pages 15903–15935, 2023. 6

[45] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. In Neurips, pages 31428–31449, 2023. 3

[46] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In ICCV, pages 3836–3847, 2023. 3, 5

[47] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, and Stefano Ermon. Hive: Harnessing human feedback for instructional visual editing. In CVPR, pages 9026–9036, 2024. 3

[48] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. In Neurips, pages 3058–3093, 2025. 3

[49] Hanyang Zhao, Genta Indra Winata, Anirban Das, Shi-Xiong Zhang, David D Yao, Wenpin Tang, and Sambit Sahu. Rainbowpo: A unified framework for combining improvements in preference optimization. In ICLR, 2025. 8

## A. Input Prompt to ChatGPT-4o

**Input Prompt:**

- **Llava-critic feedback to editing instructions**:

> You are an AI assistant that provides 2-3 concise suggestions (separated by a semicolon) with each suggestion being no more than 8 words. Please make sure that each suggestion suggests concrete change, not just a high-level idea. Your goal is to improve images so they better align with the prompt: {prompt}.
> I want you to give short, concise editing instructions based on the following inputs regarding misalignment information. Some instructions are (1) Keep it concise: "Change the red dog to yellow" is better than "Please make the dog that is red in the image a bright yellow color". (2) Be specific: Avoid ambiguous instructions like Make it more colorful. Instead, use Change the red dog to yellow and make the background green. (3) Avoid redundancy: Don't repeat the same intent multiple times. The image is the generated image based on the prompt: prompt.
> Here, we have the feedback given by the llava critic model: fb. Please give short editing instructions for the image to solve the misalignment as a text string, where instructions are separated by a semicolon.

- **RLHF feedback to editing instructions:**

> You are an AI assistant that provides 2-3 concise suggestions (separated by a semicolon) with each suggestion being no more than 8 words. Please make sure that each suggestion suggests concrete change, not just a high-level idea. Your goal is to improve images so they better align with the prompt: {prompt}. The first image is the generated image that we want to improve. The second image is a heatmap highlighting areas that misalign with the prompt.
> I want you to give short, concise editing instructions based on the following inputs regarding misalignment information. Some instructions are (1) Keep it concise: "Change the red dog to yellow" is better than "Please make the dog that is red in the image a bright yellow color". (2) Be specific: Avoid ambiguous instructions like Make it more colorful. Instead, use Change the red dog to yellow and make the background green. (3) Avoid redundancy: Don't repeat the same intent multiple times. The image is the generated image based on the prompt: {prompt}.
> Here, we have the list of pairs where the first element is a word in the prompt, and the second element is 1 if there's misalignment for this word, and 0 otherwise.The list of pairs are: {misalignment-pairs}. If the pairs are None, this means that this info is unavailable, please use the image and the prompt to give advice. We also have a heatmap, which is the second image attached, highlighting the misalignment area in the original (first) generated image.
> Now please give short editing instructions for the image to solve the misalignment as a text string, where instructions are separated by a semicolon.

- **ChatGPT image-prompt to editing instructions**

> You are an AI assistant that provides 2-3 concise suggestions (separated by a semicolon) with each suggestion being no more than 8 words. Please make sure that each suggestion suggests concrete change, not just a high-level idea. Your goal is to improve images so they better align with the prompt: {prompt }.
> I want you to give short, concise editing instructions based on the following inputs regarding misalignment information. Some instructions are (1) Keep it concise: "Change the red dog to yellow" is better than "Please make the dog that is red in the image a bright yellow color". (2) Be specific: Avoid ambiguous instructions like Make it more colorful. Instead, use Change the red dog to yellow and make the background green. (3) Avoid redundancy: Don't repeat the same intent multiple times. The image is the generated image based on the prompt: {prompt}.
> Now please give short editing instructions for the image to solve the misalignment as a text string, where instructions are separated by a semicolon.

- **ChatGPT image-prompt to feedback and then editing instructions**

> You are an AI assistant that helps improve a text-to-image model. Your task is to first analyze and critique whether the image aligns with the given prompt (i.e., give some feedback) and then provide 2-3 concise suggestions (separated by a semicolon) with each suggestion being no more than 8 words. Please separate the feedback and the editing instructions with an asterisk. Please make sure that each suggestion suggests concrete change, not just a high-level idea. Your goal is to improve images so they better align with the prompt: {prompt}.
> I want you to first generate feedback based on the given input image (and the prompt) and then give

short, concise editing instructions based on the given image and the given prompt. For the feedback, it should be a couple of sentences. Some instructions for the editing instructions are ((1) Keep it concise: “Change the red dog to yellow” is better than “Please make the dog that is red in the image a bright yellow color”. (2) Be specific: Avoid ambiguous instructions like Make it more colorful. Instead, use Change the red dog to yellow and make the background green. (3) Avoid redundancy: Don’t repeat the same intent multiple times. The image is the generated image based on the prompt: {prompt}.
Now please give feedback and short editing instructions for the image to solve the misalignment as a text string, where feedback and instructions are separated by an asterisk and the instructions are separated by a semicolon.

## B. Input Prompt to VLMs for editing instruction generation

- **Generating editing instruction from Rich Feedback**

> You are an AI assistant providing exactly 2 to 3 concise, specific image editing suggestions (separated by semicolons), each no more than 8 words. Suggestions must describe only how to modify the *image itself* to better align with the prompt. Do not instruct changes to the text prompt.
> Formatting rules:
> 1. Output a single-line string with edits, separated by semicolons.
> 2. No explanations, bullet points, or extra details."
> 3. Do not repeat exact misaligned words; describe the needed visual change.
> 4. Avoid vague edits. Instead of 'Make it colorful,' say 'Turn the red dog bright yellow.'
> 5. Always generate a response unless no relevant objects exist.
> The image is generated from this prompt: `prompt`. Below is a list of (concept, flag) pairs. A flag of 0 means the image is misaligned; a flag of 1 means it is correct. For each concept flagged 0, provide one specific visual correction. List of pairs: `mis_pairs`. Output only the editing instructions in a single line. image: `base64_combined_image`.

- **Generating editing instruction from Llava-Critic**

> You are an AI assistant providing exactly 2 to 3 concise, specific image editing suggestions (separated by semicolons), each no more than 8 words. Suggestions must describe only how to modify the *image itself* to better align with the prompt. Do not instruct changes to the text prompt.
> Formatting rules:
> 1. Output a single-line string with edits, separated by semicolons.
> 2. No explanations, bullet points, or extra details."
> 3. Do not repeat exact misaligned words; describe the needed visual change.
> 4. Avoid vague edits. Instead of 'Make it colorful,' say 'Turn the red dog bright yellow.'
> 5. Always generate a response unless no relevant objects exist.
> The image is generated from this prompt: `prompt`. Below is an image critique highlighting deviations from the prompt. Identify the specific visual misalignments and suggest precise edits to correct them. Critique: `critique`.
> Output only the editing instructions in a single line. image: `base64_combined_image`.