

# Calibrated Multi-Preference Optimization for Aligning Diffusion Models

Kyungmin Lee<sup>1,2,†</sup> Xiaohang Li<sup>3</sup> Qifei Wang<sup>1</sup> Junfeng He<sup>4</sup> Junjie Ke<sup>1</sup>  
 Ming-Hsuan Yang<sup>1</sup> Irfan Essa<sup>1,5</sup> Jinwoo Shin<sup>2</sup> Feng Yang<sup>1,‡</sup> Yinxiao Li<sup>1,‡</sup>

<sup>1</sup>Google DeepMind <sup>2</sup>KAIST <sup>3</sup>Google <sup>4</sup>Google Research <sup>5</sup>Georgia Institute of Technology

## Abstract

Aligning text-to-image (T2I) diffusion models with preference optimization is valuable for human-annotated datasets, but the heavy cost of manual data collection limits scalability. Using reward models offers an alternative, however, current preference optimization methods fall short in exploiting the rich information, as they only consider pairwise preference distribution. Furthermore, they lack generalization to multi-preference scenarios and struggle to handle inconsistencies between rewards. To address this, we present Calibrated Preference Optimization (CaPO), a novel method to align T2I diffusion models by incorporating the general preference from multiple reward models without human annotated data. The core of our approach involves a reward calibration method to approximate the general preference by computing the expected win-rate against the samples generated by the pretrained models. Additionally, we propose a frontier-based pair selection method that effectively manages the multi-preference distribution by selecting pairs from Pareto frontiers. Finally, we use regression loss to fine-tune diffusion models to match the difference between calibrated rewards of a selected pair. Experimental results show that CaPO consistently outperforms prior methods, such as Direct Preference Optimization (DPO), in both single and multi-reward settings validated by evaluation on T2I benchmarks, including GenEval and T2I-Compbench.<sup>1</sup>

## 1. Introduction

Recent text-to-image (T2I) diffusion models [4, 11, 43, 54, 55] generate high-quality images from text prompts. While these models perform well, synthesizing images that closely match subtle human preferences is a challenging task. Following the success of reinforcement learning from human

feedback (RLHF) in language models [44], training a reward model to mimic human preference [28, 31, 66–68, 71], and fine-tuning diffusion models with RL algorithms shows promise [5, 10, 12, 32]. However, the computational expense of backpropagation through the diffusion trajectories limits the scalability to large-scale diffusion models. To address this problem, Diffusion-DPO [63] applies direct preference optimization (DPO) [48] to diffusion models, with good results for large-scale diffusion models [11]. Nonetheless, since Diffusion-DPO entails an expensive paired human preference dataset, it remains unclear how to leverage multiple reward models to align large-scale T2I diffusion models.

Building on this line of research, we explore an alternative approach to fine-tune large-scale T2I diffusion models without relying on human preference datasets. Instead, we generate training data using pretrained T2I diffusion models and simulate human preferences through multiple reward models. Unlike Diffusion-DPO [63], which relies on explicit pairwise preference data, our method fully leverages the rich knowledge embedded in reward signals. However, directly optimizing rewards risks overfitting and reward hacking if the reward values are not properly calibrated [15].

To address this issue, we propose *Calibrated Preference Optimization* (CaPO) to enhance preference optimization of T2I diffusion models by improving how reward signals are used. Instead of directly optimizing reward values, we introduce the concept of general preference [3], defined as the expected win-rate against a pretrained model. We approximate this by averaging pairwise win-rates among multiple samples, providing a robust and calibrated signal. Our fine-tuning objective uses the regression loss to match the difference of calibrated rewards with the difference of implicit reward from diffusion models, which is simple and effective that enhances the performance. In addition, we introduce a novel Frontier-based rejection sampling method to enhance the multi-reward preference optimization. This approach addresses the limitations of combining rewards with linear weights [7, 10] by selecting training pairs from the upper and lower Pareto frontiers using a non-dominated sorting

<sup>†</sup>Work done during an internship at Google DeepMind.

<sup>‡</sup>Corresponding Authors. <sup>‡</sup>Equal advising.

<sup>1</sup>Project page: <https://kyungmnlee.github.io/capo.github.io/>

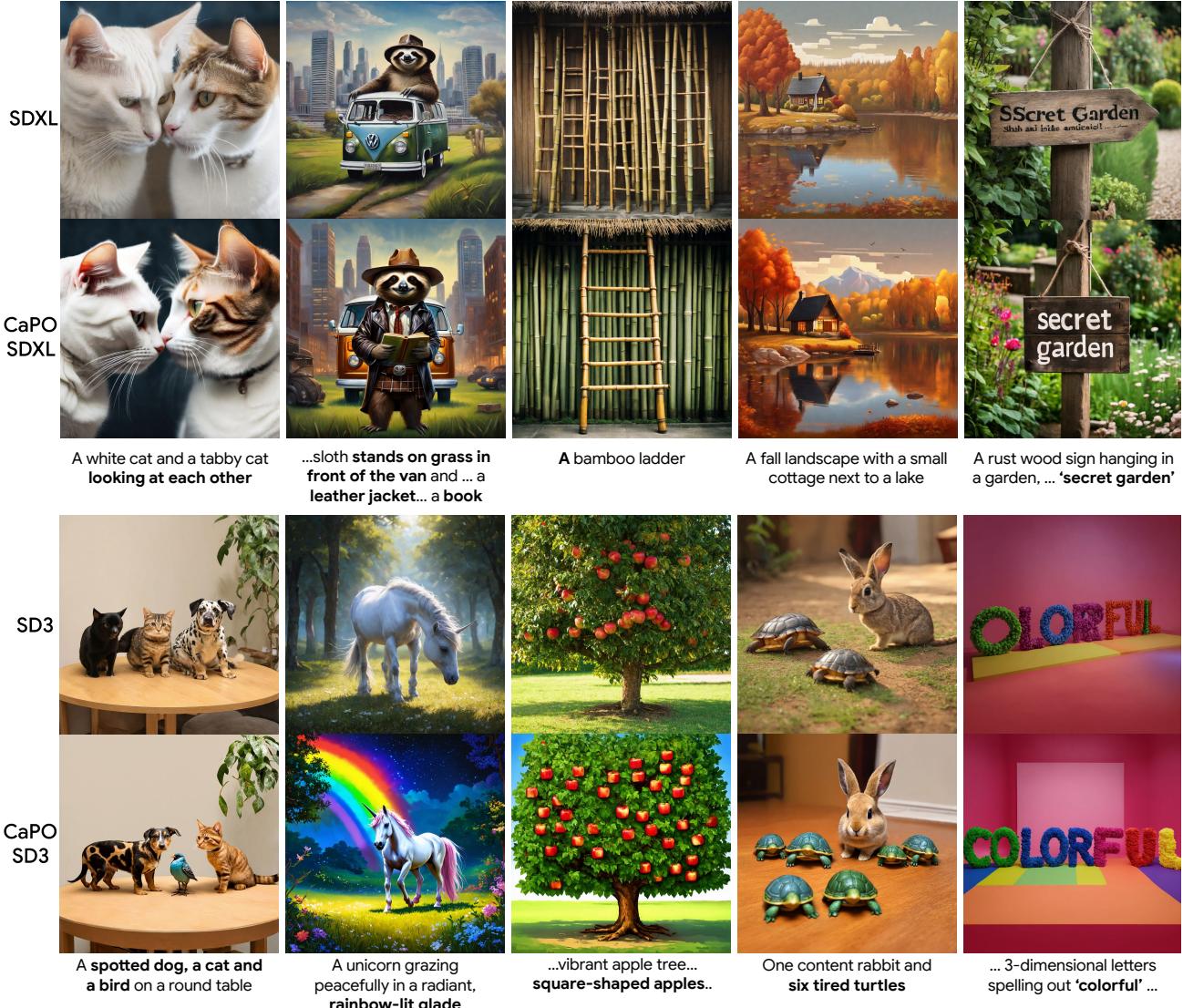


Figure 1. Calibrated Preference Optimization (CaPO) improves the performance of diffusion models by optimizing the model with diverse reward signals. The top and bottom groups are using SDXL and SD3-medium, respectively. For each group, the first row is from base model and the second row is applying CaPO to the base model. CaPO tends to generate images of higher quality (*e.g.*, image aestheticism, text rendering), and better prompt alignment (*e.g.*, compositional generation), without using any human preference dataset.

algorithm. Jointly optimizing diverse reward signals enables the model to achieve balanced response to multiple rewards and mitigate the over-optimization problem when using a single reward. Lastly, we propose an effective loss weighting scheme to improve the diffusion preference optimization.

Through extensive experiments, we show that CaPO consistently outperforms other fine-tuning methods including DPO [63], achieving better alignment with human preferences across different benchmarks. Our contributions are:

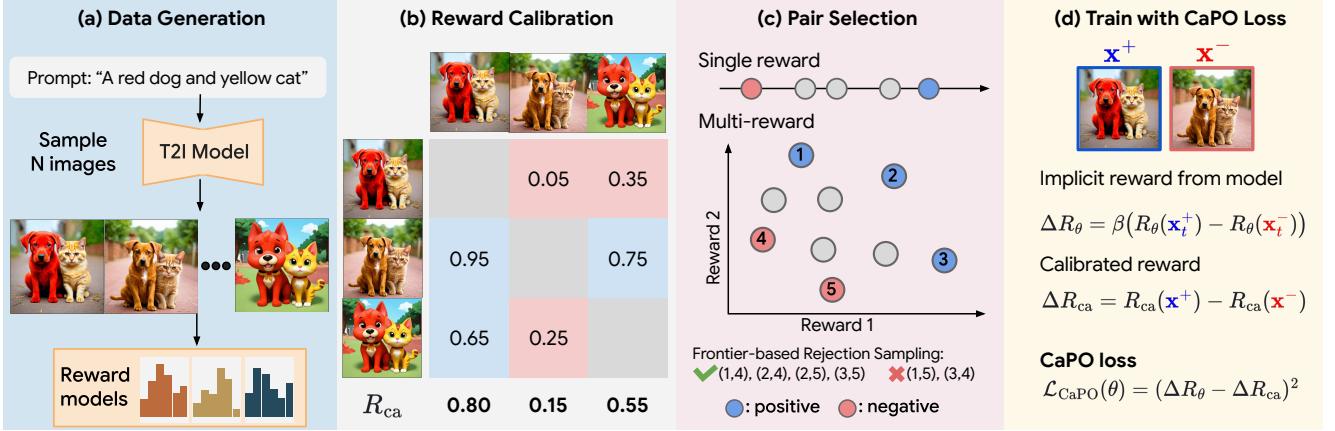
- We propose CaPO, which leverages a novel reward calibration method by incorporating approximated win-rates to fine-tune diffusion models and mitigate reward hacking;
- We expand the applicability of CaPO to multi-reward fine-

tuning problems by introducing frontier-based rejection sampling to jointly optimize with diverse reward signals;

- We demonstrate the effectiveness of CaPO with favorable visual generation quality against state-of-the-art models on benchmark datasets.

## 2. Related Work

**Modeling human preference for visual generation.** Motivated by the recent success in incorporating human preference modeling to convert language models into advanced chatbot models, numerous methods have been developed to transfer the success into visual generation. Lee et al. [31] demonstrates the capability of training reward mod-



**Figure 2. Overview.** (a) We generate  $N$  images using pretrained T2I diffusion model using the prompt dataset, and infer the scores from reward models. (b) Then, we calibrate the rewards by making pairwise comparison between images. For each image, we compute the win-rates between other  $N - 1$  images using Eq. (2), and average them to obtain calibrated reward  $R_{\text{ca}}$  (see Sec. 4.2). (c) We select pair by choosing the best-of- $N$  and worst-of- $N$  when using single reward. For multi-reward, we use non-dominated sorting algorithm to select upper Pareto set as positives, and lower Pareto set as negatives. The accepted and rejected pairs are also listed using proposed rejection sampling method. (d) Lastly, during training, we select a pair from (c), and compute CaPO loss (*i.e.*, Eq. (8)), which perform regression task to match the difference in calibrated rewards, *i.e.*,  $\Delta R_{\text{ca}}$  by the difference of implicit reward model  $\Delta R_\theta$ .

els according to human preference using a small dataset. Subsequently, numerous reward models [28, 66–68, 71] for text-to-image generation have been proposed by fine-tuning a vision-language model (*e.g.*, CLIP [46] or BLIP [34]) with Bradley-Terry model based on a paired human preference dataset. Alternatively, Lin et al. [37] uses a multi-modal large language models to exploit the knowledge of LLMs by performing visual question answering task to measure the alignment between texts and images. While existing reward models can operate as a proxy for ground-truth reward models, the inherent noise within the data due to finite capacity and coverage, inevitably affect the performance negatively when used for fine-tuning. Our work addresses the above-mentioned issues by introducing a calibration method that approximates the win-rate, rather than using the rewards.

**Fine-tuning diffusion models with rewards.** Numerous methods have been developed for fine-tuning T2I diffusion models with reward models [5, 7, 10, 12, 31, 32, 63]. By formulating discrete diffusion sampling process as a reinforcement learning problem, Black et al. [5] and Fan et al. [12] develop fine-tuning diffusion models with policy gradient algorithms. However, those methods are computationally expensive and the training processes are usually unstable. While Deng et al. [10] presents a scheme to scale RL fine-tuning for large-scale prompt dataset, it is not clear whether this approach can be applied to large-scale diffusion models. Instead of using RL, [7] proposes to directly fine-tune diffusion models by using the gradients from reward models. Yet, those approaches can only be applied to differentiable reward models, and extending to large-scale reward models (*e.g.*, LLMs) is computationally prohibitive in practice. Inspired

by the success of DPO [48], Wallace et al. [63] introduce Diffusion-DPO, which can effectively alleviate the computational loads and can be applied to large-scale diffusion models [11]. Consequent works [23, 30, 35, 36, 70] built upon Diffusion DPO to enhance preference optimization or customization of large-scale T2I diffusion models.

### 3. Preliminaries

We first describe the preliminaries on preference optimization for diffusion models before presenting our method. More details can be found in Appendix A.

**Diffusion models.** The denoising diffusion model [17, 57, 59, 60] consists of forward processes, which gradually add noise to the data, and reverse processes, which generate data from noise. The forward process of a data  $\mathbf{x}$  at time  $t \in [0, 1]$  forms a distribution  $q(\mathbf{x}_t | \mathbf{x})$ , given by  $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ , where  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ , and  $\alpha_t, \sigma_t$  are noise schedules. Let  $\lambda_t = \log(\alpha_t^2 / \sigma_t^2)$  be log signal-to-noise ratio (SNR), then we express the diffusion training objective as a weighted  $\epsilon$ -prediction loss as in [27]:

$$\mathcal{L}_{\text{DM}}(\mathbf{x}) = \mathbb{E}_{t \sim \mathcal{U}(0,1), \epsilon} \left[ -w_t \lambda'_t \| \epsilon_\theta(\mathbf{x}_t; t) - \epsilon \|^2_2 \right], \quad (1)$$

where  $w_t$  is a weighting function, and  $\lambda'_t = d\lambda/dt$ . Note that most of diffusion [22] and flow matching [38] training objectives can be expressed in Eq. (1) by choosing  $w_t$  and  $\lambda_t$ . The reverse process generates data by solving time-discretized SDE [60] or ODE [22, 58], which gradually denoises Gaussian noise into data by using trained diffusion model. Text-to-image diffusion models [43, 52–55] are conditional diffusion models that use text embeddings  $\mathbf{c}$  from text encoders [46, 49] as condition to generate image from

text input. In this work, we denote  $\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t)$  as T2I diffusion model, and  $p_\theta(\cdot | \mathbf{c})$  as distribution of the generated data given prompt  $\mathbf{c}$ .

**Reward models.** Given an image  $\mathbf{x}$  and a condition  $\mathbf{c}$ , a reward model  $R(\mathbf{x}, \mathbf{c})$  is a function that measures an utility of the input. The common approach is Bradley-Terry (BT) model [6, 44], which defines the preference distribution for a triplet  $(\mathbf{c}, \mathbf{x}, \mathbf{x}')$ :

$$\mathbb{P}(\mathbf{x} \succ \mathbf{x}' | \mathbf{c}) := \sigma(R(\mathbf{x}, \mathbf{c}) - R(\mathbf{x}', \mathbf{c})), \quad (2)$$

where  $\sigma(u) = (1 + \exp(-u))^{-1}$  is a sigmoid function.

**Diffusion preference optimization.** The goal of reward fine-tuning is to optimize the model  $p_\theta$  that maximizes the expected reward of generated output, which comes with KL regularization to prevent over optimization:

$$\max_{\theta} \mathbb{E}_{\mathbf{c}, \mathbf{x} \sim p_\theta(\mathbf{x} | \mathbf{c})} [R(\mathbf{x}, \mathbf{c})] - \beta D_{\text{KL}}(p_\theta(\cdot | \mathbf{c}) \| p_{\text{ref}}(\cdot | \mathbf{c})), \quad (3)$$

where  $\beta$  is a hyperparameter that controls the divergence. To solve Eq. (3), direct alignment methods, *e.g.*, DPO [48], have been applied to diffusion models [63]. At its core, it uses the closed-form solution of Eq. (3), which is given by  $p^*(\mathbf{x} | \mathbf{c}) \propto p_{\text{ref}}(\mathbf{x} | \mathbf{c}) \exp(\frac{1}{\beta} R(\mathbf{x}, \mathbf{c}))$ . By replacing  $p^*$  with  $p_\theta$  and rearranging for  $r$ , applying Eq. (2) for a ranked data pair  $(\mathbf{x}^+, \mathbf{x}^-)$  gives us following general preference optimization objective [61]:

$$\ell(\theta) = g \left( \beta \log \frac{p_\theta(\mathbf{x}^+ | \mathbf{c})}{p_{\text{ref}}(\mathbf{x}^+ | \mathbf{c})} - \beta \log \frac{p_\theta(\mathbf{x}^- | \mathbf{c})}{p_{\text{ref}}(\mathbf{x}^- | \mathbf{c})} \right), \quad (4)$$

where  $g$  is any convex loss function, *e.g.*,  $g(u) = -\log \sigma(u)$  gives us DPO [48] objective, and  $g(u) = (1 - u)^2$  gives us identity preference optimization (IPO) [3] objective.

However, directly applying Eq. (4) to diffusion models is not straightforward as the log-likelihoods of diffusion models are intractable. Wallace et al. [63] propose a method to compute log-ratio and derive DPO loss for diffusion models by marginalizing the log-ratio through forward process  $q(\mathbf{x}_{0:1} | \mathbf{x})$  to compute the log-ratio with  $\epsilon$ -prediction losses:

$$\mathbb{E}_{q(\mathbf{x}_{0:1} | \mathbf{x})} \left[ \log \frac{p_\theta(\mathbf{x}_{0:1} | \mathbf{c})}{p_{\text{ref}}(\mathbf{x}_{0:1} | \mathbf{c})} \right] = \mathbb{E}_{t, \epsilon} [R_\theta(\mathbf{x}_t, \mathbf{c}, t)],$$

where  $R_\theta(\mathbf{x}_t, \mathbf{c}, t) = \lambda'_t (\|\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2 - \|\epsilon_{\text{ref}}(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2)$  for  $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$  with  $t \sim \mathcal{U}(0, 1)$  and  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . By applying this to Eq. (4) and taking the expectation out of  $g$  yields diffusion preference optimization objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \epsilon^+, \epsilon^-} [g(\beta R_\theta(\mathbf{x}_t^+, \mathbf{c}, t) - \beta R_\theta(\mathbf{x}_t^-, \mathbf{c}, t))], \quad (5)$$

where  $\mathbf{x}_t^+ = \alpha_t \mathbf{x}^+ + \sigma_t \epsilon^+$ ,  $\mathbf{x}_t^- = \alpha_t \mathbf{x}^- + \sigma_t \epsilon^-$  for  $t \sim \mathcal{U}(0, 1)$  and  $(\epsilon^+, \epsilon^-) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \times \mathcal{N}(\mathbf{0}, \mathbf{I})$ .

## 4. Proposed Method

In this section, we introduce our method for calibrated preference optimization. We refer to Fig. 2 for the overview.

### 4.1. Motivation

The challenges in multi-reward optimization is in achieving the Pareto optimality among reward signals, especially even when they conflict. For example, when optimizing models for image aesthetics, it often results in reduced image-text alignment as aesthetic reward models do not consider textual information (*e.g.*, see Tab. 1). One common practice is to use weighted sum of rewards as a proxy for the total reward function [7]. However, those rigid formulations cannot effectively consider all aspects of utilities, which might lead to suboptimal performance, *e.g.*, biased towards certain reward signals. Another approach is using the rewarded soups [50], which merges the independently reward fine-tuned model with model soup [65]. Nevertheless, optimizing for a single reward is prone to reward over-optimization [15, 47] and result in significant performance loss.

Our core assumption is that the difficulties in multi-reward optimization lie in the inconsistency between the black-box distribution of rewards. To address this challenge, we propose calibrated preference optimization to minimize inconsistencies by fine-tuning with general and unified metrics. In the following, we provide details of our method.

### 4.2. CaPO

Although we consider reward models as a proxy to represent the utility of a sample, directly using the reward values can lead to unsatisfactory results if they are not properly calibrated. Specifically, when using Bradley-Terry model [6], the reward value often does not measure the goodness of a sample, even though the model exhibits high prediction accuracy in classifying the human preference. Furthermore, the varying range of reward becomes problematic when using multiple reward signals, making it difficult to obtain balanced updates.

**Calibrated rewards.** To address these issues, we propose to use expected win-rate as a unified measure for maximization target. Formally, let  $\mathbb{P}(\mathbf{x} \succ \mathbf{x}' | \mathbf{c})$  be a win-rate of data  $\mathbf{x}$  over  $\mathbf{x}'$  with prompt  $\mathbf{c}$ . We define the win-rate of data  $\mathbf{x}$  over a distribution  $(\cdot | \mathbf{c})$ :

$$\mathbb{P}(\mathbf{x} \succ p | \mathbf{c}) := \mathbb{E}_{\mathbf{x}' \sim p(\cdot | \mathbf{c})} [\mathbb{P}(\mathbf{x} \succ \mathbf{x}' | \mathbf{c})]. \quad (6)$$

As our goal is to improve over reference model  $p_{\text{ref}}$ , we consider  $\mathbb{P}(\mathbf{x} \succ p_{\text{ref}} | \mathbf{c})$  as our target of interest. By using expected win-rate over reference model, we directly seek for improvement over a pretrained model, which quantifies the general goodness of a data. Furthermore, the bounded range makes it more favorable for multi-reward optimization. Since the expected win-rate is not available in general,

we approximate it through averaging the pairwise win-rate computed by a reward model. Suppose we generate  $N$  batch of samples  $\{\mathbf{x}_i\}_{i=1}^N$  from  $p_{\text{ref}}(\cdot|\mathbf{c})$ , then we define *calibrated reward*  $R_{\text{ca}}(\mathbf{x}_i, \mathbf{c})$  for each sample  $i$ :

$$R_{\text{ca}}(\mathbf{x}_i, \mathbf{c}) = \frac{1}{N-1} \sum_{j \neq i} \sigma(R(\mathbf{x}_i, \mathbf{c}) - R(\mathbf{x}_j, \mathbf{c})), \quad (7)$$

where we have  $R_{\text{ca}}(\mathbf{x}, \mathbf{c}) \approx \mathbb{P}(\mathbf{x} \succ p_{\text{ref}}|\mathbf{c})$  for large  $N$ .

**CaPO loss.** We replace  $R(\mathbf{x}, \mathbf{c})$  in Eq. (3) with  $R_{\text{ca}}(\mathbf{x}, \mathbf{c})$ , and introduce calibrated preference optimization objective that fine-tunes the model to maximize the calibrated reward. Similar to Eq. (5), we define CaPO loss by matching the difference of the calibrated rewards with regression loss [10, 13], which also guarantees the optimality condition. Specifically, given data pair  $(\mathbf{x}^+, \mathbf{x}^-)$ , we define CaPO objective:

$$\begin{aligned} \mathcal{L}_{\text{CaPO}}(\theta) = & \mathbb{E}_{t, \epsilon, \epsilon'} \left[ \left( R_{\text{ca}}(\mathbf{x}^+, \mathbf{c}) - R_{\text{ca}}(\mathbf{x}^-, \mathbf{c}) \right. \right. \\ & \left. \left. - \beta(R_\theta(\mathbf{x}_t^+, \mathbf{c}, t) - R_\theta(\mathbf{x}_t^-, \mathbf{c}, t)) \right)^2 \right], \end{aligned} \quad (8)$$

where  $\mathbf{x}_t^+ = \alpha_t \mathbf{x}^+ + \sigma_t \epsilon^+$ ,  $\mathbf{x}_t^- = \alpha_t \mathbf{x}^- + \sigma_t \epsilon^-$ , for  $t \sim \mathcal{U}(0, 1)$  and  $(\epsilon^+, \epsilon^-) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \times \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Note that CaPO is a special case of Eq. (5) with  $g(u) = (\Delta R - u)^2$ , where  $\Delta R = R^+ - R^-$  is a difference between calibrated rewards. Thus, CaPO is a generalization of IPO [3], which strictly assign  $\Delta R = 1$  for all pairs. Compared to IPO, CaPO assigns a dynamic target for the preference learning, which helps maximizing the gain without reward over-optimization.

### 4.3. Preference Pair Selection

The best-of- $N$  sampling [8, 42] or rejection sampling [62] methods that select samples with highest reward from  $N$  generation are commonly used in RLHF. For a single reward, it is straightforward to choose the sample  $\mathbf{x}^+$  with highest reward, and  $\mathbf{x}^-$  that has lowest reward to maximize the margin between the pair. For multi-reward optimization, the naïve approach is to use weighted sum as the total proxy reward model, and perform rejection sampling with it. However, choosing the weights often relies on heuristics, and the optimal weights might be dynamic depending on the input, which can lead to suboptimal performance.

In order to achieve the Pareto optimal solution, we propose *frontier-based rejection sampling* (FRS), which selects the set of positive samples  $X^+(\mathbf{c})$  and negative samples  $X^-(\mathbf{c})$  for each prompt  $\mathbf{c}$  by finding Pareto optimal set. Specifically, we use a non-dominated sorting algorithm [9] to find the upper and lower Pareto frontier. The goal of FRS is to push apart from the lower Pareto frontier and pull towards the upper Pareto frontier, which helps to achieve Pareto optimality. Given  $L$  reward models, let  $R_{\text{ca}}^{(j)}$  be  $j$ -th calibrated rewards for  $j = 1, \dots, L$ , then we define  $\mathbf{x}$

	MPS	VQA	VILA	MPS	VQA	VILA	MPS	VQA	VILA
DPO	58.5	49.3	61.7	53.1	50.6	55.9	52.6	46.4	81.8
IPO	56.8	<b>50.1</b>	64.1	53.1	51.9	53.8	53.3	48.5	76.1
CaPO	<b>61.1</b>	49.7	<b>64.9</b>	<b>55.5</b>	<b>53.2</b>	<b>58.7</b>	<b>54.1</b>	<b>49.6</b>	<b>83.1</b>

(a) Base model SDXL

	MPS	VQA	VILA	MPS	VQA	VILA	MPS	VQA	VILA
DPO	55.2	53.2	54.4	52.1	53.2	52.9	53.1	48.7	70.1
IPO	51.1	52.1	48.3	52.8	51.9	51.1	<b>58.3</b>	50.2	70.8
CaPO	<b>58.1</b>	<b>53.3</b>	<b>63.4</b>	<b>54.4</b>	<b>55.4</b>	<b>59.4</b>	57.4	<b>50.8</b>	<b>74.0</b>

(b) Base model SD3-M

Table 1. **Single reward results.** We report the win-rate (%) over base model by using automatic evaluation with each reward model. We use Parti prompts [69] and DPG-bench prompts [20] to generate images for each SDXL, and SD3-M models, respectively. We highlight the column to indicate the rewards used for fine-tuning.

dominates  $\mathbf{x}'$  if and only if  $R_{\text{ca}}^{(j)}(\mathbf{x}, \mathbf{c}) \geq R_{\text{ca}}^{(j)}(\mathbf{x}', \mathbf{c})$  for all  $j = 1, \dots, L$ . Then finding a set of non-dominated data points is referred as finding Pareto set, which forms an upper frontier. Conversely, one can define a set of dominated data that forms a lower frontier. After removing the potential duplicates of non-dominated and dominated sets, we take  $X^+(\mathbf{c})$  by filtered non-dominated sets and  $X^-(\mathbf{c})$  by set of dominated set. Given positive set  $X^+(\mathbf{c})$  and  $X^-(\mathbf{c})$ , we sample a positive sample  $\mathbf{x}^+ \sim X^+(\mathbf{c})$  and  $\mathbf{x}^- \sim X^-(\mathbf{c})$  to construct a pair. We use CaPO loss to update the model with ensemble of calibrated rewards for optimization target:

$$R_{\text{ca}}(\mathbf{x}, \mathbf{c}) = \frac{1}{L} \sum_{j=1}^L R_{\text{ca}}^{(j)}(\mathbf{x}, \mathbf{c}),$$

and use CaPO loss in Eq. (8) for the update.

### 4.4. Loss weighting

The choice of log-SNR  $\lambda_t$  and weighting function  $w_t$  has large impact on the generation quality and convergence of diffusion model pretraining. Intuitively, when  $\lambda_t$  is large, *i.e.*, small amount of noise is added, the denoising task becomes easier, and conversely the task becomes harder as  $\lambda_t$  becomes smaller, thus weighting function as a monotonically decreasing weighting function of  $\lambda_t$  seems a reasonable choice. In [27], those monotonic weighting are theoretically shown to be the weighted evidence lower bound (ELBO), and demonstrated better quality than the non-monotonic counterpart. In this work, we also propose to use monotonic loss weighting to our CaPO loss, which is equivalent to regularizing with weighted ELBO instead of KL divergence in Eq. (3). Specifically, we apply sigmoid weighting with bias, *i.e.*,  $w_t = w(\lambda_t) = \sigma(-\lambda_t + b)$ , where  $b$  is a bias hyperparameter [19, 27]. See supplementary for details.

## 5. Experiments

**Models and datasets.** We use Stable Diffusion XL (SDXL) [45] and Stable Diffusion 3 medium (SD3-M) [11] as our

Objective	Method	MPS		VQAscore		VILA	
		Win (%)	Score	Win (%)	Score	Win (%)	Score
SDXL	-	-	11.30	-	0.826	-	5.953
DPO	SUM	57.2	11.48	52.1	0.829	71.9	6.193
	SOUP	56.5	11.46	52.2	0.830	74.3	6.227
	FRS	58.1	11.54	52.9	0.834	78.6	6.294
IPO	SUM	57.4	11.49	51.1	0.828	66.8	6.111
	SOUP	55.4	11.44	52.0	0.830	70.3	6.154
	FRS	57.8	11.52	52.0	0.830	74.4	6.238
CaPO	SUM	<b>61.2</b>	11.62	52.5	0.834	75.0	6.258
	SOUP	59.4	11.44	52.8	0.835	77.6	6.259
	FRS	<b>61.2</b>	<b>11.66</b>	<b>54.6</b>	<b>0.839</b>	<b>79.2</b>	<b>6.340</b>

(a) Base model SDXL

Objective	Method	MPS		VQAscore		VILA	
		Win (%)	Score	Win (%)	Score	Win (%)	Score
SD3-M	-	-	13.39	-	0.908	-	5.793
DPO	SUM	55.3	13.50	52.8	0.910	55.0	5.832
	SOUP	56.1	13.39	54.7	0.908	63.4	5.875
	FRS	56.7	13.55	53.2	0.909	68.7	5.922
IPO	SUM	54.1	13.47	53.9	0.912	58.9	5.847
	SOUP	55.6	13.39	53.5	0.910	60.4	5.848
	FRS	55.5	13.55	54.6	0.913	64.7	5.913
CaPO	SUM	57.8	13.56	54.3	0.912	57.0	5.833
	SOUP	<b>59.4</b>	<b>13.60</b>	54.9	0.911	67.6	5.896
	FRS	59.0	13.58	<b>55.7</b>	<b>0.914</b>	<b>69.3</b>	<b>5.943</b>

(b) Base model SD3-M

Table 2. **Multi-reward results.** We report the average reward scores (Score) and win-rate (%) over base model by using automatic evaluation with each reward model (Win). We compare preference objectives DPO [63], IPO [3], and CaPO and combination with different pair selection methods, *e.g.*, using sum of rewards to conduct top-1 and worst-1 sampling (SUM), and using frontier-based rejection sampling (FRS). Furthermore, we compare our method with rewarded soup [50], by merging single reward optimized models (SOUP).



Figure 3. **Qualitative comparison.** We present qualitative comparison of using multiple rewards using our frontier rejection sampling method when fine-tuning with CaPO objective.

base text-to-image diffusion models in all experiments. To collect the training dataset, we use 100K prompts from DiffusionDB [64], and generate  $N = 16$  images per prompt. We also experiment with using 8, or 32 images per prompt, and select 16 images per prompt, which provides a good trade-off between computational cost and performance improvement. To generate images, we use DDIM [58] sampler with guidance scale 7.5 for 50 steps, and flow DPM-solver [39] with guidance scale 5.0 for 50 steps for each SDXL and SD3-M, respectively. Note that we only use the images generated by the same diffusion model for experiments. We refer to supplementary for detailed experimental setup.

**Reward models.** We consider three reward models that cover diverse aspects of the T2I generation. For general human preference (*i.e.*, overall quality), we use MPS score [71], which is a state-of-the-art reward model for human preference. For image-text alignment, we use VQAscore [37], which uses a vision-language model (CLIP-FlanT5-XXL) to compute scores by performing visual question answering tasks. Specifically, VQAscore measures the probability  $P(\text{Yes}|\mathbf{x}, \mathbf{Q}(\mathbf{c}))$  by using the output logits of the model, where  $\mathbf{Q}$  is a template for the question. We also use VILA score [24] pretrained on AVA [41] dataset to evaluate image aesthetics. While VQAscore and VILA score are not trained

Model	GenEval						T2I-Compbench						
	Single	Two	Counting	Colors	Position	Color Attribution	Overall	Color	Shape	Texture	Complex	Spatial	Non Spatial
FLUX-dev	0.98	0.84	0.67	0.76	0.23	0.46	0.66	0.740	0.486	0.650	0.477	0.220	0.306
FLUX-schnell	0.99	0.81	0.58	0.63	0.25	0.35	0.60	0.642	0.509	0.646	0.424	0.185	0.304
SD3.5-L	0.99	0.89	0.62	0.82	0.26	0.53	0.69	0.763	0.602	0.766	0.520	0.219	0.314
SDXL	0.98	0.74	0.39	0.85	0.15	0.23	0.55	0.592	0.500	0.608	0.465	0.159	0.312
CaPO+SDXL	0.99	0.79	0.48	0.86	0.15	0.28	0.59	0.646	0.537	0.633	0.491	0.172	0.312
SD3-M	0.99	0.84	0.56	0.84	0.32	0.52	0.68	0.775	0.546	0.712	0.505	0.221	0.309
CaPO+SD3-M	0.99	0.87	0.63	0.86	0.31	0.59	0.71	0.788	0.572	0.731	0.509	0.230	0.313

Table 3. **T2I Benchmarks evaluation.** We compare the benchmark results of CaPO-SDXL and CaPO-SD3-M on text-to-image benchmarks, *e.g.*, GenEval [16] and T2I-Compbench [21], with various open-source state-of-the-art models (*e.g.*, Flux-dev [29], and Flux-schnell [29], and SE3.5-L [11]). We observe that with CaPO, the majority of evaluation metrics for SDXL and SD3-M show improvement. For comparison, we also include the most recent three image generation models, which are 3× larger compared to SDXL and SD3-M.

with BT model, we adjust to approximate with BT model (see supplementary for detail).

## 5.1. Single reward experiments

**Experimental setups.** We evaluate CaPO against state-of-the-art preference learning objectives such as DPO [63] and IPO [3] for diffusion models. For each method, we train with three reward models (MPS, VQAscore, and VILA) by selecting the top-1 and worst-1 pair. For evaluation, we use Parti prompts [69] to generate images for SDXL fine-tuned model and DPG-Bench [20] prompts to generate images for SD3-M fine-tuned model. We report the win-rate against the base model using each reward model.

**Results.** Tab. 1 shows that CaPO achieves the highest win-rate for each reward model used for fine-tuning, as well as other reward models. Especially, when using the VILA model for training, DPO shows significant drop in VQAscore, while showing comparable performance with CaPO in VILA score. On the other hand, IPO shows better robustness than DPO in reward hacking, but the gain of the performance is lower than DPO and CaPO in general. We notice that even though we optimized for a reward, other rewards also increase at some cases. This is partially due to the inherent correlation residing in reward models, *e.g.*, increasing MPS score results in increase in VILA score, as image aesthetics is an important factor in overall quality.

## 5.2. Multi-reward experiments

**Experimental setups.** We consider MPS, VQAscore, and VILA scores for multi-reward experiments. For baselines, we evaluate CaPO against DPO and IPO as in Sec. 5.1. Furthermore, we conduct experiments on different methods in adapting for multiple rewards. Specifically, we compare frontier-based rejection sampling (FRS) (*i.e.*, Sec. 4.3) with sum-of-rewards (Sum), and merging the models fine-tuned with single reward (*i.e.*, model soup [65]). For sum-of-

rewards, we directly add the calibrated rewards, and perform top-1 and worst-1 pair selection for training data. For model soup, we re-use fine-tuned models from Sec. 5.1, and use spherical linear interpolation [51, 56] to merge models, which performs slightly better than linear interpolation in our experiments. We use uniform weights (*i.e.*, 1/3 each) for both sum-of-rewards and model soup. For evaluation, we generate images using Parti prompt dataset [69] and DPG-bench prompt dataset for SDXL and SD3-M, respectively. For evaluation, we report the average reward scores, and the win-rate against the base model by using each reward model.

**Quantitative results.** Tab. 2 shows the results. First, joint training of multiple rewards by using frontier-based rejection sampling consistently outperforms pair selection with sum of rewards on all preference optimization objectives. While model merging (Soup) shows comparable performance to FRS when using DPO and IPO objectives for training, using CaPO objective with FRS outperforms model soup of CaPO fine-tuned models with single reward. When comparing CaPO, DPO, and IPO, CaPO with FRS shows higher win-rates and average rewards compared to DPO and IPO with FRS, which is consistent with the results of Tab. 1.

**Qualitative results.** In Fig. 3, we provide qualitative comparison of our method on SDXL and SD3-M, compared to DPO trained with multi-reward frontier-based rejection sampling. While both DPO and CaPO shows improved image aesthetics such as contrast or color compared to base SDXL, we see that CaPO demonstrates better image-text alignment and aesthetic quality compared to DPO, following the quantitative results in Tab. 2. We refer to supplementary for additional examples.

**Benchmark results.** For quantitative analysis, we evaluate our models on various T2I benchmarks; GenEval [16] which evaluates object-focused generation, and T2I-Compbench [21] for compositional generation. We compare our method with the base models SDXL and SD3-M, as well

	Pickscore	MPS	VQA	VILA
Diffusion-DPO [63]	22.71	11.59	0.834	6.049
DPO-Syn	22.74	11.59	0.825	6.074
CaPO	<b>22.83</b>	<b>11.71</b>	<b>0.838</b>	<b>6.141</b>

Table 4. **Comparison with Diffusion-DPO [63].** We compare CaPO with Diffusion-DPO, which is trained on human annotated preference dataset Pick-a-pic [28]. For fair comparison, we train CaPO with same prompts from Pick-a-pic, but trained with generated images from SDXL. Also, we use Pickscore [28], which is trained on Pick-a-pic dataset. We also train DPO for our synthetic dataset, denoted as DPO-Syn. We report Pickscore, MPS, VQAscore, and VILA score by generating images from Parti prompts.



Figure 4. **Qualitative comparison with Diffusion-DPO [63].** We show qualitative comparison between SDXL, Diffusion-DPO, and CaPO using Pickscore [28].

as open-source state-of-the-art T2I diffusion models such as FLUX-dev [29], FLUX-schnell [29], and Stable Diffusion 3.5-large (SD3.5-L) [11]. Tab. 3 shows that CaPO improves the performance of the base model, *e.g.*, 0.55→0.59 for SDXL, 0.68→0.71 on GenEval overall score, and on almost every metrics in T2I-Combench.

### 5.3. Ablation Studies

**Comparison with Diffusion-DPO [63].** We compare our method with Diffusion-DPO [63], which fine-tunes SDXL on the human preference dataset Pick-a-pic [28]. For fair comparisons, we use the same 58K prompts in Pick-a-pic v2 dataset, and use Pickscore [28], which is a reward model trained on Pick-a-pic dataset, as our reward signal. Here, we generate  $N = 16$  images for each prompt, and select a pair by choosing highest and lowest reward, following Sec. 5.1. We also train DPO on our synthetic data (DPO-Syn), to show the effect of synthetic data for fine-tuning. For evaluation, we generate images using Parti prompts [69], and compare Pickscore, MPS, VQAscore, and VILA scores. Tab. 4 shows the results. Note that while DPO-Syn scores higher than

	MPS	VQA	VILA
Constant weighting	56.5	51.8	70.8
Sigmoid weighting ( $b = 1.0$ )	59.1	54.5	73.3
Sigmoid weighting ( $b = 1.5$ )	61.2	54.6	79.2
Sigmoid weighting ( $b = 2.0$ )	58.6	52.6	75.2

Table 5. **Ablation on loss weighting.** We show the results of CaPO multi-reward fine-tuning SDXL with constant weighting (*i.e.*,  $-w_t \lambda'_t = 1$ ), and sigmoid weighting by varying bias  $b = 1.0, 1.5, 2.0$ . Using sigmoid weighting shows better results than constant weighting, and  $b = 1.5$  performs the best.

Diffusion-DPO on Pickscore and VILA score, Diffusion-DPO outperforms on VQAscore. On the other hand, CaPO strictly shows better performance than Diffusion-DPO. In Fig 4, we show visual comparison between SDXL, Diffusion-DPO, and CaPO, which shows consistent trends as in Tab. 4.

**Effect of loss weighting.** We demonstrate the effect of loss weighting that we proposed in Sec. 4.4. Specifically, we compare the performance of CaPO when using sigmoid loss weighting, and without loss weights (*i.e.*,  $w_t \lambda'_t = -1$ ). We vary the bias of loss weight by  $b = 1.0, 1.5, 2.0$ . Tab. 5 shows the results of SDXL CaPO models trained with multi-reward experimental setup. We notice using loss weighting significantly improves the performance, while the best  $b$  achieves at  $b = 1.5$ . Note that the trend of bias differs for SD3-M, which we refer to supplementary for details.

### 5.4. Discussions

While our method can improve the quality of T2I generation, some of the improvements (*e.g.*, improving the text rendering for SDXL) is difficult, which is bounded by the performance of original model. However, for more powerful diffusion models (*e.g.*, SD3), we show that our method can improve the text rendering as well. Furthermore, our approach is built upon offline data generation, which often suffers from slow convergence. Extending CaPO to online learning problems is a promising direction and we leave it for future work.

## 6. Conclusion

In this paper, we present calibrated preference optimization, a robust preference learning objective that fine-tunes the diffusion models to align with human preference by using multiple reward models. Specifically, we propose a simple, yet effective method to calibrate the rewards to approximate the win-rate against the base model. We then propose a diffusion preference optimization objective that regresses the difference between the calibrated rewards, which effectively learns from the reward without over-optimization. Furthermore, we extend our approach to a multi-reward problem by providing a frontier-based rejection sampling method that enables joint optimization of various reward signals. Extensive experimental results demonstrate that our approach is efficient and can boost the model performance without using

any human-collected preference dataset.

## Acknowledgements

Kyungmin Lee acknowledges the partial support from Artificial Intelligence Graduate School Program (KAIST) (RS-2019-II190075) and Institute of Information & communications Technology Planning & Evaluation (IITP) (No.RS-2021-II212068, Artificial Intelligence Innovation Hub).

## References

- [1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 12
- [2] Amazon. Amazon mechanical turk. <https://www.mturk.com/>, 2005. 17
- [3] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, 2024. 1, 4, 5, 6, 7
- [4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. <https://cdn.openai.com/papers/dalle-3.pdf>, 2023. 1
- [5] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024. 1, 3
- [6] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 1952. 4
- [7] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024. 1, 3, 4
- [8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 5
- [9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 2002. 5
- [10] Fei Deng, Qifei Wang, Wei Wei, Tingbo Hou, and Matthias Grundmann. Prdp: Proximal reward difference prediction for large-scale reward finetuning of diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 3, 5
- [11] 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 *International Conference on Machine Learning*, 2024. 1, 3, 5, 7, 8, 12, 14, 15, 16
- [12] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, 2023. 1, 3
- [13] Adam Fisch, Jacob Eisenstein, Vicky Zayats, Alekh Agarwal, Ahmad Beirami, Chirag Nagpal, Pete Shaw, and Jonathan Berant. Robust preference optimization through reward model distillation. *arXiv preprint arXiv:2405.19316*, 2024. 5
- [14] Roy Frostig, Matthew James Johnson, and Chris Leary. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 2018. 15
- [15] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2023. 1, 4
- [16] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, 2024. 7, 16
- [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 3, 12, 14
- [18] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, 2023. 17
- [19] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion (sid2): 1.5 fid on imagenet512 with pixel-space diffusion. *arXiv preprint arXiv:2410.19324*, 2024. 5, 17
- [20] 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. 5, 7, 16
- [21] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. 7, 16
- [22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022. 3, 12, 14
- [23] Shyamgopal Karthik, Huseyin Coskun, Zeynep Akata, Sergey Tulyakov, Jian Ren, and Anil Kag. Scalable ranked preference optimization for text-to-image generation. *arXiv preprint arXiv:2410.18013*, 2024. 3
- [24] Junjie Ke, Keren Ye, Jiahui Yu, Yonghui Wu, Peyman Milanfar, and Feng Yang. Vila: Learning image aesthetics from user comments with vision-language pretraining. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 6, 15
- [25] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, 2021. 12

- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 15
- [27] Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *Advances in Neural Information Processing Systems*, 2023. 3, 5, 12, 13, 14
- [28] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Maitana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. 1, 3, 8, 14
- [29] Black Forest Labs. Flux. <https://blackforestlabs.ai/>, 2024. 7, 8, 16
- [30] Kyungmin Lee, Sangkyung Kwak, Kihyuk Sohn, and Jinwoo Shin. Direct consistency optimization for robust customization of text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, 2024. 3
- [31] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. In *International Conference on Machine Learning*, 2024. 1, 2, 3
- [32] Seung Hyun Lee, Yinxiao Li, Junjie Ke, Innfarm Yoo, Han Zhang, Jiahui Yu, Qifei Wang, Fei Deng, Glenn Entis, Junfeng He, et al. Parrot: Pareto-optimal multi-reward reinforcement learning framework for text-to-image generation. In *European Conference on Computer Vision*, 2024. 1, 3
- [33] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Tiffany Ling, Xide Xia, Pengchuan Zhang, Graham Neubig, et al. Genai-bench: Evaluating and improving compositional text-to-visual generation. *arXiv preprint arXiv:2406.13743*, 2024. 17
- [34] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022. 3, 16
- [35] Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *arXiv preprint arXiv:2404.04465*, 2024. 3
- [36] Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Ji Li, and Liang Zheng. Step-aware preference optimization: Aligning preference with denoising performance at each step. *arXiv preprint arXiv:2406.04314*, 2024. 3
- [37] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, 2024. 3, 6, 14
- [38] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3, 12, 14
- [39] 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. In *Advances in Neural Information Processing Systems*, 2022. 6, 15
- [40] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. 12
- [41] Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6
- [42] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 5
- [43] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 1, 3
- [44] 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 *Advances in Neural Information Processing Systems*, 2022. 1, 4
- [45] 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 *International Conference on Learning Representations*, 2024. 5
- [46] 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*, 2021. 3, 16
- [47] Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *arXiv preprint arXiv:2406.02900*, 2024. 4
- [48] 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 *Advances in Neural Information Processing Systems*, 2024. 1, 3, 4
- [49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020. 3
- [50] Alexandre Ramé, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, 2024. 4, 6
- [51] Alexandre Ramé, Johan Ferret, Nino Vieillard, Robert Dadashi, Léonard Hussonot, Pierre-Louis Cedoz,

- Pier Giuseppe Sessa, Sertan Girgin, Arthur Douillard, and Olivier Bachem. Warp: On the benefits of weight averaged rewarded policies. *arXiv preprint arXiv:2406.16768*, 2024. [7](#), [15](#)
- [52] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, 2021. [3](#)
- [53] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. [1](#)
- [55] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. [1](#), [3](#)
- [56] Ken Shoemake. Animating rotation with quaternion curves. In *ACM SIGGRAPH conference proceedings*, 1985. [7](#)
- [57] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015. [3](#)
- [58] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. [3](#), [6](#), [12](#), [15](#)
- [59] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. [3](#)
- [60] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. [3](#), [12](#)
- [61] Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024. [4](#)
- [62] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. [5](#)
- [63] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#), [13](#), [17](#), [19](#)
- [64] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022. [6](#), [15](#)
- [65] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022. [4](#), [7](#)
- [66] 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. [1](#), [3](#)
- [67] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *IEEE International Conference on Computer Vision*, 2023.
- [68] 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 *Advances in Neural Information Processing Systems*, 2023. [1](#), [3](#)
- [69] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunnar Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. [5](#), [7](#), [8](#), [16](#), [17](#)
- [70] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *arXiv preprint arXiv:2402.10210*, 2024. [3](#)
- [71] Sixian Zhang, Bohan Wang, Junqiang Wu, Yan Li, Tingting Gao, Di Zhang, and Zhongyuan Wang. Learning multi-dimensional human preference for text-to-image generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. [1](#), [3](#), [6](#), [14](#)
- [72] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. [16](#)

# Calibrated Multi-Preference Optimization for Aligning Diffusion Models

## Supplementary Materials

### A. Additional description

In this section, we provide additional details to Sec. 3 and Sec. 4 of the main manuscript. Specifically, we review the preliminaries on diffusion models and flow-based models (Sec. A.1), preference optimization for diffusion models (Sec. A.2), and provide details on loss weighting scheme (Sec. A.3).

#### A.1. Background on diffusion and flow-based models

**Diffusion models.** Let  $q(\mathbf{x})$  be the density of data distribution of a sample  $\mathbf{x}$  and  $p_\theta(\mathbf{x})$  be a generative model parameterized by  $\theta$  that approximates  $q$ . Given  $\mathbf{x} \sim q(\mathbf{x})$ , the diffusion model considers a series of latent variables  $\mathbf{x}_t$  at time  $t \in [0, 1]$ . Specifically, the forward process forms a conditional distribution  $q(\mathbf{x}_t | \mathbf{x})$ , where the marginal distribution is given by

$$\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \quad (9)$$

where  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , and  $\alpha_t, \sigma_t$  are noise scheduling functions such that satisfies  $\alpha_0 \approx 1$ ,  $\alpha_1 \approx 0$ , and  $\sigma_0 \approx 0$ ,  $\sigma_1 \approx 1$ . Let us denote  $\lambda_t = \log(\alpha_t^2 / \sigma_t^2)$  log signal-to-noise ratio (log-SNR), then  $\lambda_t$  is a decreasing function of  $t$ . Here,  $\alpha_t$  and  $\sigma_t$  (or equivalently  $\lambda_t$ ) is chosen to satisfy that  $\mathbf{x}_1$  is indiscernible from Gaussian noise (*i.e.*,  $p(\mathbf{x}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$ ), and conversely,  $\mathbf{x}_0$  matches the data density  $q(\mathbf{x})$ . Then the reverse generative process gradually denoises the random Gaussian noise  $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  to recover  $\mathbf{x}_0$ . Specifically, the sampling process is governed by solving time-discretized SDE [17, 60] or probability flow ODE [22, 58], by using the score function  $\nabla \log q(\mathbf{x}_t)$ . Training diffusion model then optimizes the neural network to approximate the score function by  $s_\theta(\mathbf{x}_t; t)$ . Especially, using the noise-prediction model [17] is a common practice, where the training objective can be written as following weighted loss objective [27]:

$$\mathcal{L}_{\text{DM}}(\theta; \mathbf{x}) = \mathbb{E}_{t \sim \mathcal{U}(0, 1), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[ -\frac{1}{2} w_t \lambda'_t \|\boldsymbol{\epsilon}_\theta(\mathbf{x}_t; t) - \boldsymbol{\epsilon}\|_2^2 \right], \quad (10)$$

where  $w_t$  is a weighting function and  $\lambda'_t$  is a time-derivative of  $\lambda_t$ . Note that when  $w_t = 1$  for all  $t \in (0, 1)$ , it becomes the variational lower bound (vlb) of KL divergence [25], and the original DDPM uses  $w_t \lambda'_t = -1$ .

**Flow models.** Alternatively, flow-based models or stochastic interpolants [1, 38, 40] consider approximating the velocity field  $\mathbf{v}(\mathbf{x}_t, t)$  on  $\mathbf{x}$  at time  $t \in (0, 1)$ , and solve following probability flow ODE to transport noise to data distribution:

$$\mathbf{x}'_t = \mathbf{v}(\mathbf{x}_t, t), \quad (11)$$

where the marginal distribution of the solution of ODE matches the distribution  $q_t(\mathbf{x}_t)$ . Given  $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$  for some  $t \in (0, 1)$  and  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , the velocity field satisfies following:

$$\mathbf{v}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{x}'_t | \mathbf{X}_t = \mathbf{x}_t] = \alpha'_t \mathbb{E}[\mathbf{x} | \mathbf{X}_t = \mathbf{x}_t] + \sigma'_t \mathbb{E}[\boldsymbol{\epsilon} | \mathbf{X}_t = \mathbf{x}_t], \quad (12)$$

and training objective for flow matching model is given as follows:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0, 1), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\alpha'_t \mathbf{x} + \sigma'_t \boldsymbol{\epsilon})\|_2^2]. \quad (13)$$

Note that Eq. (13) is a special case of Eq. (10), when  $w_t = -\frac{1}{2} \lambda'_t \sigma_t^2$  [11, 27]. In case of Rectified Flow [38], we set  $\alpha_t = 1 - t$ ,  $\sigma_t = t$ , and  $\lambda_t = 2 \log(\frac{1-t}{t})$ , and the training objective of rectified flow model is given as follows:

$$\mathcal{L}_{\text{RF}}(\theta) = \mathbb{E}_{t, \boldsymbol{\epsilon}} [\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\boldsymbol{\epsilon} - \mathbf{x})\|_2^2]. \quad (14)$$

For SD3-M [11], we use Eq. (14) to compute the loss.

#### A.2. Diffusion preference optimization

For preference optimization with diffusion models, we consider following relaxation of original RLHF objective:

$$\max_{\theta} \bar{R}(\mathbf{x}_{0:1}, \mathbf{c}) - \beta D_{\text{KL}}(p_\theta(\mathbf{x}_{0:1} | \mathbf{c}) \| p_{\text{ref}}(\mathbf{x}_{0:1} | \mathbf{c})), \quad (15)$$

where  $\bar{R}(\mathbf{x}_{0:1}, \mathbf{c})$  satisfies following:

$$R(\mathbf{x}, \mathbf{c}) = \mathbb{E}_{q(\mathbf{x}_{0:1}|\mathbf{x})} [\bar{R}(\mathbf{x}_{0:1}, \mathbf{c})]. \quad (16)$$

Then by rearranging the equation derived from the closed solution of Eq. (15), we have following:

$$\bar{R}(\mathbf{x}_{0:1}, \mathbf{c}) = \beta \log \frac{p_\theta(\mathbf{x}_{0:1}|\mathbf{c})}{p_{\text{ref}}(\mathbf{x}_{0:1}|\mathbf{c})} - \beta \log Z(\mathbf{c}), \quad (17)$$

where  $Z(\mathbf{c})$  is a partition function. From Eq. (17) and by rearranging  $q(\mathbf{x}_{0:1}|\mathbf{x})$  in the inside term, we have

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_{0:1}|\mathbf{x})} [\bar{R}(\mathbf{x}_{0:1}, \mathbf{c}) - \beta \log Z(\mathbf{c})] &= \mathbb{E}_{q(\mathbf{x}_{0:1}|\mathbf{x})} \left[ \beta \log \frac{p_\theta(\mathbf{x}_{0:1}|\mathbf{c})}{q(\mathbf{x}_{0:1}|\mathbf{x})} - \beta \log \frac{p_{\text{ref}}(\mathbf{x}_{0:1}|\mathbf{c})}{q(\mathbf{x}_{0:1}|\mathbf{x})} \right] \\ &= \beta (D_{\text{KL}}(q(\mathbf{x}_{0:1}|\mathbf{x}) \| p_{\text{ref}}(\mathbf{x}_{0:1}|\mathbf{c})) - D_{\text{KL}}(q(\mathbf{x}_{0:1}|\mathbf{x}) \| p_\theta(\mathbf{x}_{0:1}|\mathbf{c}))). \end{aligned} \quad (18)$$

Note that the KL divergence satisfies following (see [27] for details):

$$\frac{d}{dt} D_{\text{KL}}(q(\mathbf{x}_{t:1}|\mathbf{x}) \| p_\theta(\mathbf{x}_{t:1}|\mathbf{c})) = \frac{1}{2} \lambda'_t \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2]. \quad (19)$$

By taking integration of Eq. (19) over  $t \in (1, 0)$ , one can rewrite  $R(\mathbf{x}, \mathbf{c})$  as follows:

$$R(\mathbf{x}, \mathbf{c}) = \frac{\beta}{2} \mathbb{E}_{t \sim \mathcal{U}(0, 1), \epsilon \sim \mathcal{N}(0, 1)} [\lambda'_t (\|\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2 - \|\epsilon_{\text{ref}}(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2)], \quad (20)$$

For a triplet  $(\mathbf{c}, \mathbf{x}^+, \mathbf{x}^-)$ , we consider following upper bound of a training objective for any convex function  $g : \mathbb{R} \rightarrow \mathbb{R}$ :

$$\begin{aligned} \bar{\ell}(\theta) &= g(R(\mathbf{x}^+, \mathbf{c}) - R(\mathbf{x}^-, \mathbf{c})) \\ &= g\left(\frac{\beta}{2} \mathbb{E}_{t, \epsilon^+, \epsilon^-} [\lambda'_t (\|\epsilon_\theta(\mathbf{x}_t^+; \mathbf{c}, t) - \epsilon^+\|_2^2 - \|\epsilon_{\text{ref}}(\mathbf{x}_t^+; \mathbf{c}, t) - \epsilon^+\|_2^2 - \|\epsilon_\theta(\mathbf{x}_t^-; \mathbf{c}, t) - \epsilon^-\|_2^2 + \|\epsilon_{\text{ref}}(\mathbf{x}_t^-; \mathbf{c}, t) - \epsilon^-\|_2^2)]\right) \\ &\leq \mathbb{E}_{t, \epsilon^+, \epsilon^-} \left[ g\left(\frac{1}{2} \beta \lambda'_t (\|\epsilon_\theta(\mathbf{x}_t^+; \mathbf{c}, t) - \epsilon^+\|_2^2 - \|\epsilon_{\text{ref}}(\mathbf{x}_t^+; \mathbf{c}, t) - \epsilon^+\|_2^2 - \|\epsilon_\theta(\mathbf{x}_t^-; \mathbf{c}, t) - \epsilon^-\|_2^2 + \|\epsilon_{\text{ref}}(\mathbf{x}_t^-; \mathbf{c}, t) - \epsilon^-\|_2^2)\right)\right], \end{aligned}$$

where  $t \sim \mathcal{U}(0, 1)$ ,  $\epsilon^+ \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ,  $\epsilon^- \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , and the last inequality comes from the Jensen's inequality. Using the equation we defined in our main paper, *i.e.*,

$$R_\theta(\mathbf{x}_t, \mathbf{c}, t) = \lambda'_t (\|\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2 - \|\epsilon_{\text{ref}}(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2), \quad (21)$$

we derive following training objectives for DPO, IPO, and CaPO:

$$\begin{aligned} \ell_{\text{DPO}}(\theta) &= \mathbb{E}_{t, \epsilon^+, \epsilon^-} \left[ -\log \sigma(\beta (R_\theta(\mathbf{x}^+, \mathbf{c}, t) - R_\theta(\mathbf{x}^-, \mathbf{c}, t))) \right] \\ \ell_{\text{IPO}}(\theta) &= \mathbb{E}_{t, \epsilon^+, \epsilon^-} \left[ \left( 1 - \beta (R_\theta(\mathbf{x}^+, \mathbf{c}, t) - R_\theta(\mathbf{x}^-, \mathbf{c}, t)) \right)^2 \right] \\ \ell_{\text{CaPO}}(\theta) &= \mathbb{E}_{t, \epsilon^+, \epsilon^-} \left[ \left( R(\mathbf{x}^+, \mathbf{c}) - R(\mathbf{x}^-, \mathbf{c}) - \beta (R_\theta(\mathbf{x}^+, \mathbf{c}, t) - R_\theta(\mathbf{x}^-, \mathbf{c}, t)) \right)^2 \right], \end{aligned} \quad (22)$$

where  $R(\mathbf{x}, \mathbf{c})$  is a reward from the external reward model.

**Independent noise sampling.** Note that in original Diffusion-DPO paper [63], the author proposed to use same noise for  $\mathbf{x}^+$  and  $\mathbf{x}^-$ , *i.e.*,  $\epsilon^+ = \epsilon^-$ , while we sample independent noise for  $\epsilon^+$  and  $\epsilon^-$ . We believe this is more theoretically grounded, and empirically found that it has slightly better performance than using the same noise (even for DPO and IPO).

### A.3. Loss weighting

In practice, we multiply  $w_t$  to the noise-prediction loss for diffusion preference optimization. One can consider this as setting timestep-wise different  $\beta_t = \beta w_t$ , *i.e.*, giving different regularization hyperparameters at each time  $t \in (0, 1)$ . Thus, we have

$$R_\theta(\mathbf{x}_t, \mathbf{c}, t) = w_t \lambda'_t (\|\epsilon_\theta(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2 - \|\epsilon_{\text{ref}}(\mathbf{x}_t; \mathbf{c}, t) - \epsilon\|_2^2), \quad (23)$$

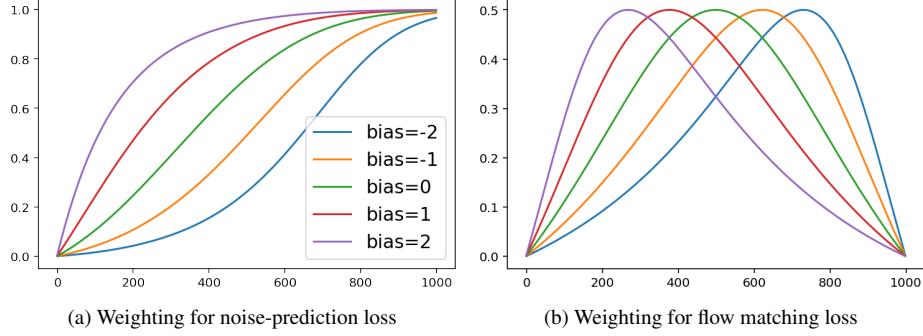


Figure 5. **Loss weighting.** We plot the weighting function with bias  $b \in \{-2, -1, 0, 1, 2\}$  for each noise prediction loss and flow matching loss.

and applies to each DPO, IPO, and CaPO loss. As we mentioned in Sec. 4.4 in our main draft, we use sigmoid loss weighting [27], where the loss weights are sigmoid function of log-SNR  $\lambda_t$  with bias  $b$ :

$$w_t = w(\lambda_t) = \frac{1}{1 + \exp(b - \lambda_t)}. \quad (24)$$

Note that SDXL uses a modified DDPM schedule [17], where  $\beta = (\sqrt{\beta_0} + \frac{t}{T-1}(\sqrt{\beta_{T-1}} - \sqrt{\beta_0}))^2$ , and  $\alpha_t = (\prod_{s=0}^t(1 - \beta_s))^{1/2}$ . Since it is impractical to compute  $\lambda'_t$ , we simply set it as constant (*i.e.*, linear  $\lambda_t$ , which empirically holds when  $\lambda_t \in [-15, 5]$ , and for  $\lambda_t > 0.5$  the weight  $w_t$  is close to 0, so one can ignore it).

SD3-M uses a rectified flow scheduler [38], where  $\lambda_t = 2 \log(\frac{1-t}{t})$ . Note that we have

$$\mathbb{E}_{t \sim \mathcal{U}(0,1), \epsilon} [\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\epsilon - \mathbf{x})\|_2^2] = \mathbb{E}_{\lambda \sim \mathcal{U}(\lambda_{\min}, \lambda_{\max}), \epsilon} [e^{-\lambda/2} \|\mathbf{v}_\theta(\mathbf{x}_\lambda, \lambda) - \epsilon\|_2^2], \quad (25)$$

where  $\mathbf{x}_\lambda$  denotes forward process of  $\mathbf{x}$  with log-SNR value  $\lambda$ , and  $\lambda_{\min}$  and  $\lambda_{\max}$  denotes the minimal and maximal value for log-SNR (see [27] Appendix D.3 for details). As such, multiplying  $w = \sigma(-\lambda)$  to noise-prediction loss is equivalent to multiplying  $(e^{\lambda/2} + e^{-\lambda/2})^{-1}$  to flow matching objective:

$$\sigma(-\lambda) \|\mathbf{v}_\theta(\mathbf{x}_\lambda, \lambda) - \epsilon\|_2^2 = \sigma(-\lambda) \cdot \frac{e^{-\lambda/2}}{e^{-\lambda/2}} \|\mathbf{v}_\theta(\mathbf{x}_\lambda, \lambda) - \epsilon\|_2^2 = \frac{1}{e^{\lambda/2} + e^{-\lambda/2}} \|\mathbf{v}_\theta(\mathbf{x}_\lambda, \lambda) - \epsilon\|_2^2. \quad (26)$$

If we shift with bias  $b$ , it becomes  $w_\lambda = (e^{-(\lambda-b)/2} + e^{-(\lambda-b)/2})^{-1}$ . In Fig. 5, we plot loss weighting functions for noise-prediction loss and flow matching loss with different bias  $b \in \{-2, -1, 0, 1, 2\}$ . In practice, we select  $\lambda \sim \mathcal{U}[-10, 10]$  and multiply  $w_\lambda$  to flow matching loss. Note that multiplying  $w_\lambda$  has a similar effect in log-normal sampling proposed in [11, 22], where we empirically find similar performance. To ensure consistency with SDXL experiments, we use loss weighting instead of logit-normal sampling for SD3-M experiments.

## B. Implementation Details

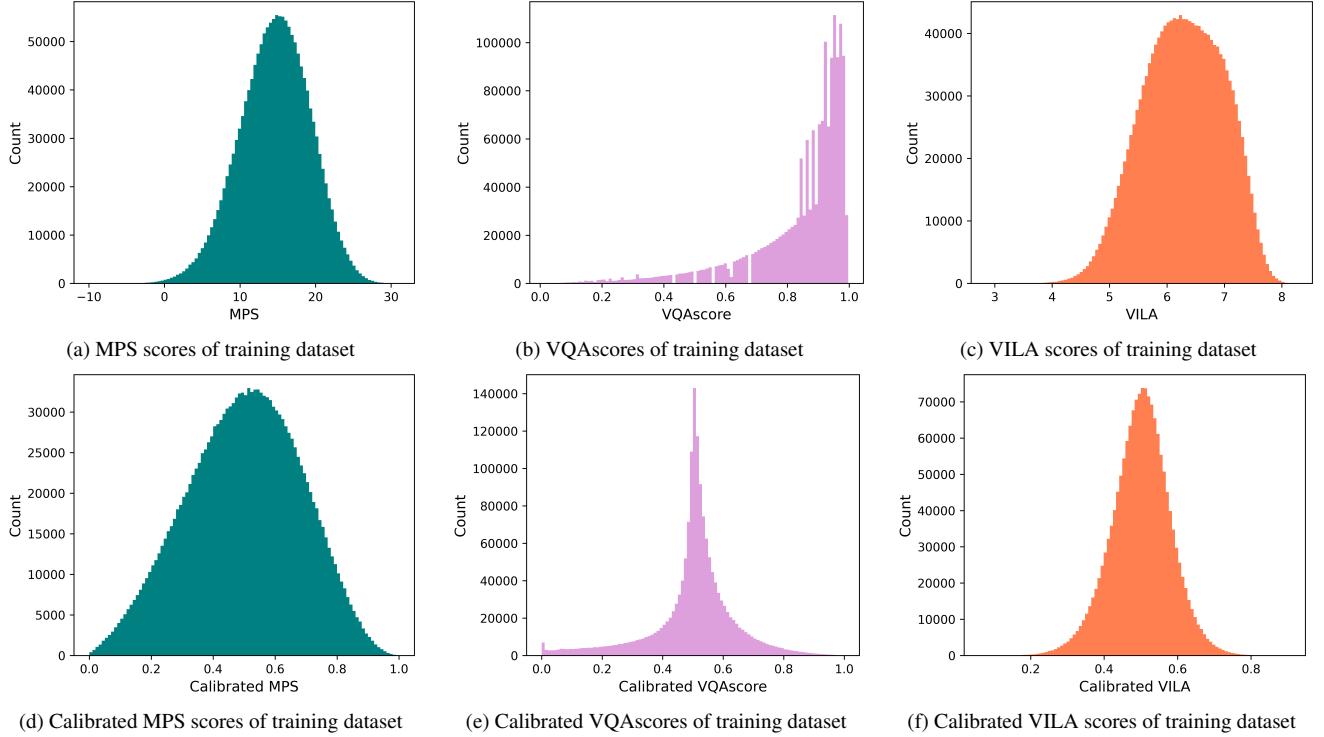
### B.1. Dataset

**Reward models.** For reward models learned by fine-tuning CLIP models (*e.g.*, Pickscore [28], MPS [71]), we compute the reward by the dot product between the image embedding and the text embedding. To compute MPS score, we additionally multiply the text embedding from condition textual description (*e.g.*, textual description for aesthetic quality). For VQAscore, we use CLIP-FlanT5-XXL [37], and compute the score by probability of "Yes" token given the image and question provided to the model:

$$P("Yes" | \mathbf{x}, "Does this figure shows \{prompt\}? Please answer yes or no."), \quad (27)$$

where  $\mathbf{x}$ ,  $\text{prompt}$  are image and text input. While VQAscore is not a Bradley-Terry model, we simply approximate the win-rate by following:

$$\mathbb{P}(\mathbf{x} \succ \mathbf{x}' | \mathbf{c}) = \frac{s(\mathbf{x}, \mathbf{c})^\alpha}{s(\mathbf{x}, \mathbf{c})^\alpha + s(\mathbf{x}', \mathbf{c})^\alpha}, \quad (28)$$



**Figure 6. Training dataset score distribution.** We plot the histogram of rewards (top row) and calibrated rewards (bottom row) of training dataset. By using calibration, the scores are centered and bounded in range  $[0, 1]$ .

where  $s(\mathbf{x}, \mathbf{c})$  is a VQAscore and  $\alpha > 0$  is a hyperparameter to control the temperature. We find  $\alpha = 1$  works well in our experiments. Lastly, VILA-R score [24] outputs the aesthetic score between 1-10, and we apply the Bradley-Terry model to compute the win-rate. In Fig. 6, we plot the histogram of reward scores and calibrated rewards of our training dataset.

**Training dataset.** We use 100K prompts from DiffusionDB [64] and generate  $N = 16$  images per prompt. For SDXL, we use DDIM [58] scheduler, guidance scale of 7.5 and sampling steps of 50. For SD3-M, we use the DPM solver [39] for flow-based models, guidance scale of 5.0 and sampling steps of 50. Furthermore, as described in [11], we shift the timestep schedules to reside more on higher timesteps, *i.e.*, we set  $t \leftarrow \frac{ts}{1+t(s-1)}$  with shift scale  $s = 3.0$ .

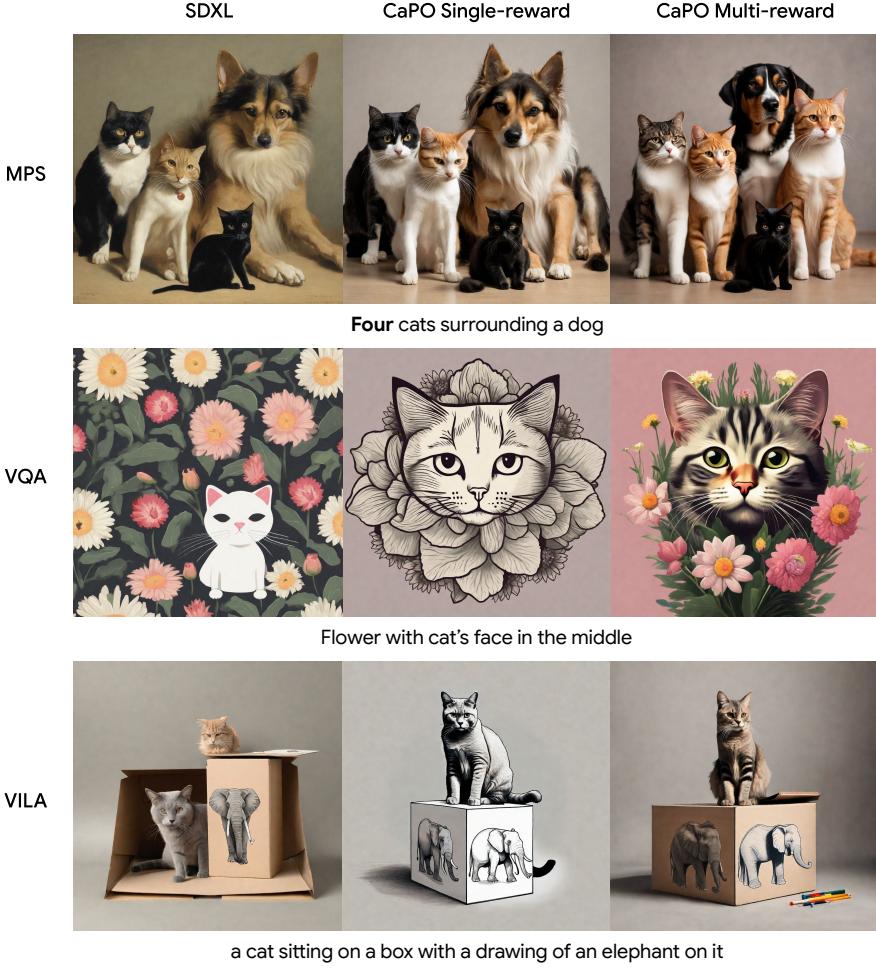
## B.2. Training and evaluation

**Training configuration.** Throughout experiments, we use Jax [14] and train models using the Optax library on TPU chips. For both SDXL and SD3-M experiments, we use Adam [26] optimizer. Regarding training configuration for SDXL experiments, we use batch size of 256, learning rate of 1e-5 with linear warmup for first 1000 steps, and train for maximum 10000 steps. For SD3-M, we use batch size of 256, learning rate of 1.5e-5 with linear warmup for first 1000 steps, and train for maximum 5000 steps. We choose hyperparameter  $\beta$  by sweeping over  $\{300, 500, 1000\}$  for CaPO,  $\{500, 1000, 2000\}$  for IPO, and  $\{2000, 3000, 4000\}$  for DPO when training SDXL model. For SD3-M, we sweep over  $\{30, 50, 100\}$  for CaPO,  $\{50, 100, 200\}$  for IPO, and  $\{100, 200, 300\}$  for DPO. For all training, we use sigmoid loss weighting with  $b = 1.5$  for SDXL and  $b = -1.0$  for SD3-M (including all DPO, IPO, and CaPO). During training, we generate images using subset of Parti prompts at each 1000-th iteration, and choose the final model with maximum validation win-rate (average of win-rates for multi-reward signals).

**Model soup.** For model merging experiments, we follow [51]. Specifically, suppose  $\theta_0$  be weights of a pretrained model and  $\theta_1, \theta_2$  be weights of fine-tuned models. Then the spherical linear interpolation (SLERP) between  $\theta_1$  and  $\theta_2$  is given by

$$\text{SLERP}(\theta_0, \theta_1, \theta_2, \lambda) = \theta_0 + \frac{\sin((1-\lambda)\Omega)}{\sin(\Omega)}(\theta_1 - \theta_0) + \frac{\sin(\lambda\Omega)}{\sin(\Omega)}(\theta_2 - \theta_0), \quad (29)$$

where  $\Omega$  is the angle between two task vectors  $\theta_1 - \theta_0$  and  $\theta_2 - \theta_0$ , and  $\lambda \in (0, 1)$  is a coefficient. To merge three fine-tuned models, we first merge two models with  $\lambda = 0.5$  to obtain  $\theta_{12} = \text{SLERP}(\theta_0, \theta_1, \theta_2, 0.5)$ , then merge  $\theta_{12}$  and  $\theta_3$  with  $\lambda = 1/3$  to obtain final model.



**Figure 7. Effect of multi-reward CaPO.** We demonstrate the qualitative comparison between CaPO with single-reward with each MPS, VQAscore, and VILA, and CaPO with multi-reward on SDXL. We see that optimizing with single-reward improves upon the base model, yet multi-reward CaPO shows the best overall improvement. For example, while using VQAscore alone improves the image-text alignment, the image aesthetics are significantly improved when using multi-reward. Also, when using VILA score, the image aesthetics improve, but it often loses the image-text alignment (*e.g.*, the image becomes drawing style, while only the elephant should be in a drawing style).

**Evaluation.** For evaluation, we generate images with the same configuration as in Sec. B.1 for different benchmark prompt dataset. For SDXL, we use Parti [69] prompts, and for SD3-M we use DPG-bench [20]. Then we compute the win-rate against the base model by comparing one-by-one comparison for each image, *e.g.*, if we have  $K$  images from base model and  $K$  images from fine-tuned model, we make  $K^2$  comparison and count the number of win and divide by  $K^2$ . We also report the average reward scores. Remark that the average reward scores and win-rate could show different trends, as the model achieves a higher score gain for some prompts, but it fails to improve on others. Thus, we found win-rate is a more general metric to see the generalization over different prompts.

**Benchmark evaluation.** We use GenEval [16] and T2I-Compbench [21] to evaluate our models. For T2I-Compbench, we use BLIP-VQA model [34] to evaluate Color, Shape, Texture, Complex, and UniDet [72] for Spatial, and CLIP [46] for Non Spatial. For baselines, we compare with the state-of-the-art open-source text-to-image diffusion models Flux-dev (12B) [29], Flux-schnell (12B) [29], and Stable Diffusion 3.5-Large (8B) [11]. Since those models are much larger than SDXL (2.6B) and SD3 (2B), we remark that this is not a fair comparison, yet we show the comparable performance of our method.

	MPS	VQA	VILA
Constant weighting	54.9	53.6	55.7
Sigmoid weighting ( $b = 0$ )	57.0	53.9	66.9
Sigmoid weighting ( $b = -1.0$ )	59.0	55.7	69.3

Table 6. **Ablation on loss weighting for SD3-M.** We show the results of CaPO multi-reward fine-tuning SD3-M with constant weighting (*i.e.*,  $-w_t \lambda'_t = 1$ ), and sigmoid weighting by varying bias  $b = 0.0, -1.0$ . Similar to SDXL, using sigmoid weighting shows better results than constant weighting, and  $b = -1.0$  performs the best.

CaPO+SDXL SDXL Improvement			CaPO+SDXL Diffusion-DPO Improvement			CaPO+SD3-M SD3-M Improvement		
54.5%	45.5%	<b>+10%</b>	52.0%	48.0%	<b>+4%</b>	53.5%	46.5%	<b>+7%</b>

Table 7. **User study results.** We report the win-rate from the user study by using 200 images. We compare CaPO+SD3-M vs SD3-M, CaPO+SDXL vs SDXL, and CaPO+SDXL vs Diffusion-DPO [63]. CaPO achieves consistent win against baseline.

## C. Additional ablation study

**Effect of multi-reward.** We demonstrate the effect of multi-reward CaPO compared to single-reward CaPO. As we demonstrated in Tab. 1 and Tab. 2 in our main draft, the single-reward model achieves the best score in which they have trained with, but the other metrics score below the multi-reward cases. We showcase the qualitative examples on the effect of multi-reward preference optimization compared to single-reward cases in Fig. 7. We notice that single-reward fine-tuning is often imperfect, *e.g.*, fine-tuning with only VILA score loses image-text alignment, and fine-tuning with only VQAscore lacks image aesthetics. On the other hand, multi-reward fine-tuning complements those issues and improves the overall image quality.

**Loss weighting for SD3-M.** We show the effect of loss weighting when training SD3-M models. Similar to SDXL, we compare the results of CaPO multi-reward fine-tuning with different bias parameters. In Tab. 6, we show that sigmoid weighting with bias  $b = -1.0$  shows the best result, outperforming the constant weighting counterpart. Note that for SDXL,  $b = 1.5$  performs the best, while for SD3-M, negative bias  $b = -1.0$  performs the best. Remark that as SD3-M performs diffusion modeling on  $16 \times 128 \times 128$ , and SDXL performs on  $4 \times 128 \times 128$ , the bias shifts toward negative as the total variance becomes higher, and the log-SNR should be increased [18, 19].

**User study evaluation.** We conduct additional user evaluations to compare our method with base models. For SDXL vs CaPO+SDXL, we randomly select 200 prompts from Parti prompt dataset [69], and for SD3-M vs CaPO+SD3-M, we randomly select 200 prompts from GenAI bench prompt dataset [33]. Additionally, we compare CaPO+SDXL with Diffusion-DPO [63] again with 200 randomly selected prompts. We give following instructions to the raters:

- Instruction: Given the text below, pick the left or the right image with better looking.
- Good example: Images are beautiful and following text description.
- Bad example: Images are not looking good or not following text description.

We use Amazon mechanical Turk [2] and 5 raters answered to each pair. In Tab. 7, we show the results of user study. We observe that CaPO+SD3-M and CaPO+SDXL consistently outperform SD3-M and SDXL, respectively. Also, CaPO+SDXL outperforms Diffusion-DPO, yet the margin is smaller than CaPO+SDXL vs SDXL.

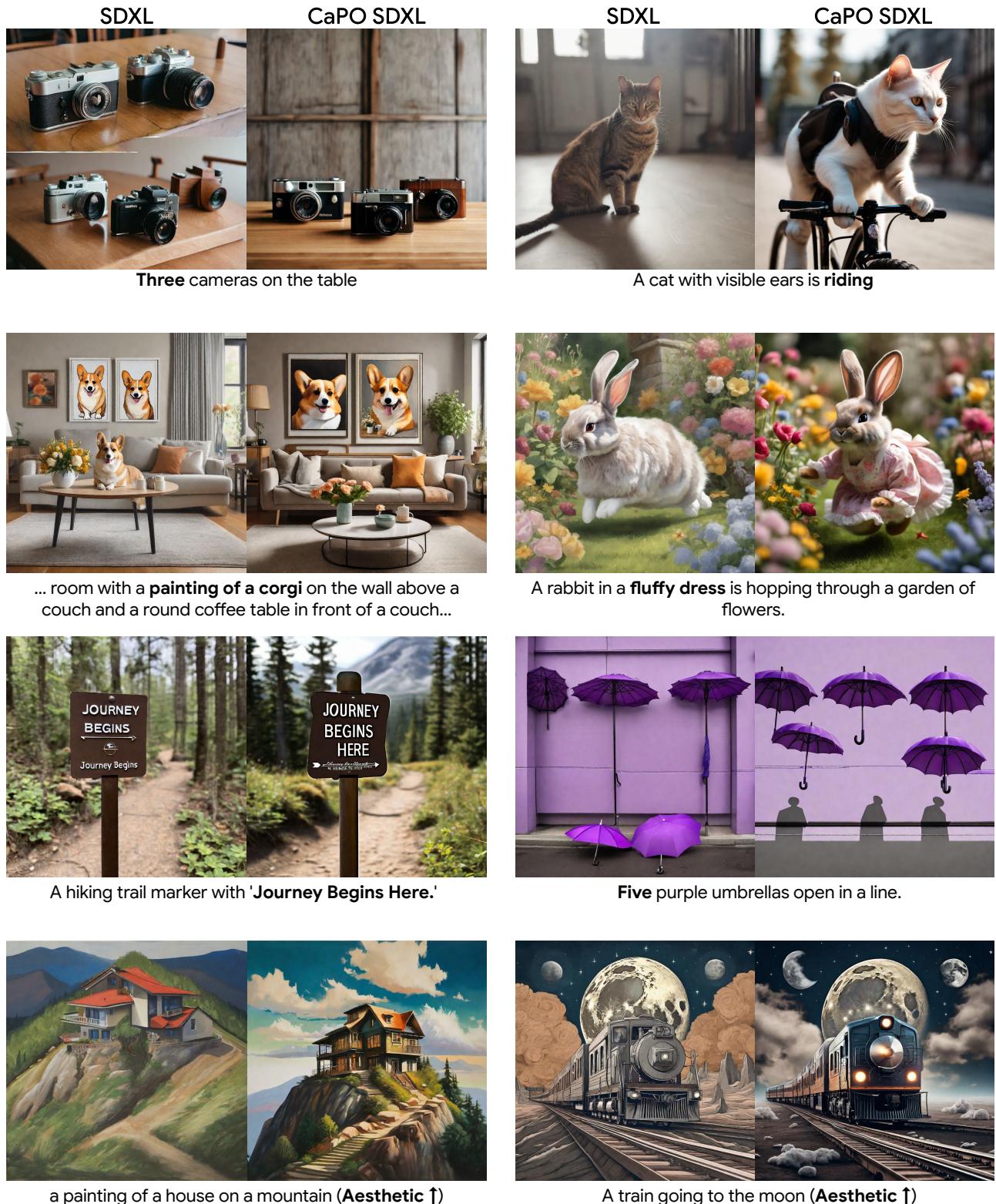


Figure 8. **Additional qualitative comparison between CaPO SDXL and SDXL.** We provide additional qualitative comparison between CaPO SDXL and SDXL. The CaPO SDXL model demonstrates better image-text alignment (e.g., counting, attribute binding, etc), as well as image aesthetics (e.g., artistic style, detail, etc). We bold the text to highlight the prompts that demonstrate improvement in image-text alignment, and (**Aesthetic ↑**) to demonstrate the improvement in image aesthetic quality.

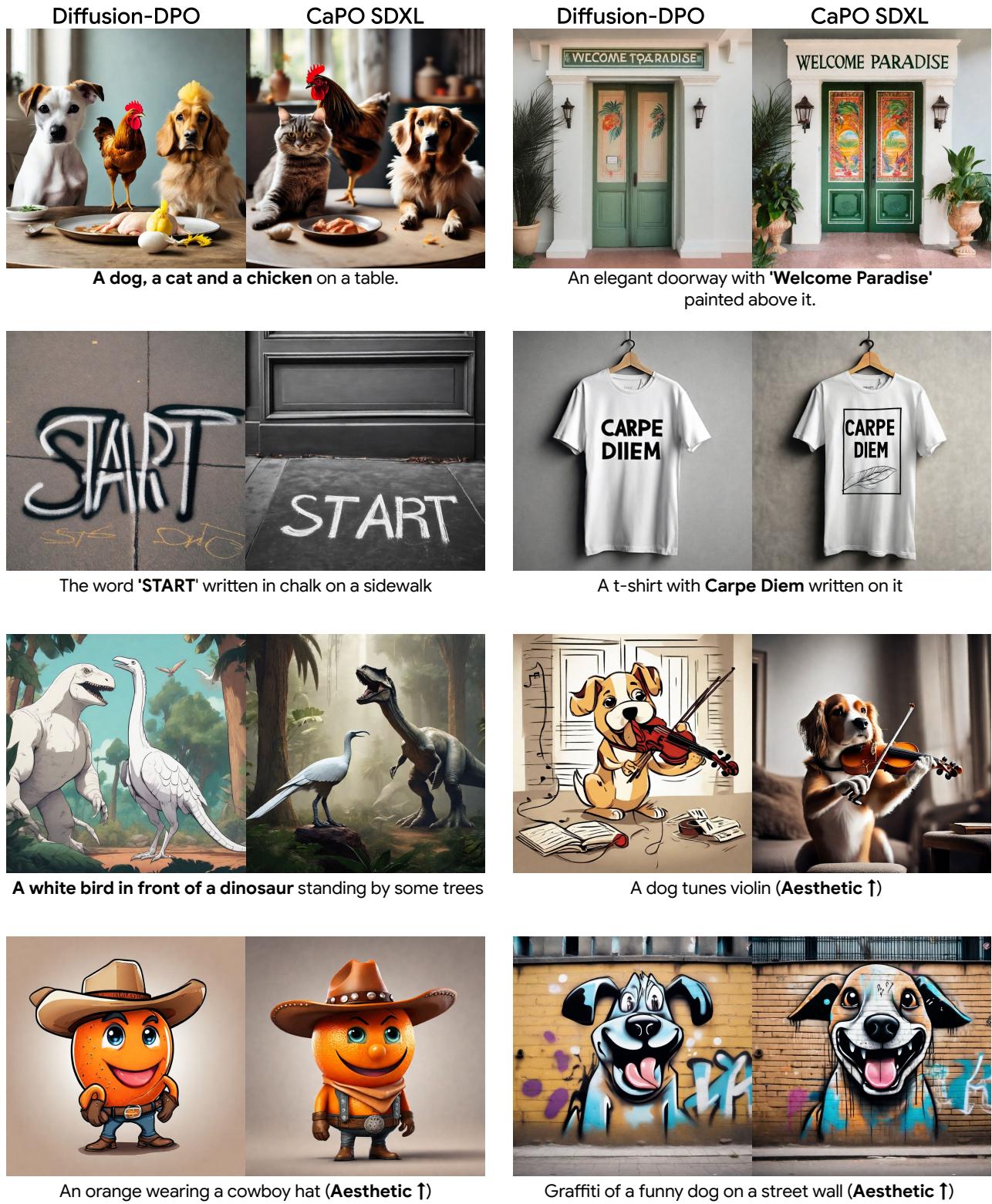


Figure 9. **Additional qualitative comparison between CaPO SDXL and Diffusion-DPO [63].** We provide additional qualitative comparison between CaPO SDXL and Diffusion-DPO [63]. CaPO SDXL shows better image-text alignment and image aesthetics compared to Diffusion-DPO without using any human annotated data. We bold the text to highlight the prompts that demonstrate improvement in image-text alignment, and (**Aesthetic ↑**) to demonstrate the improvement in aesthetic quality.

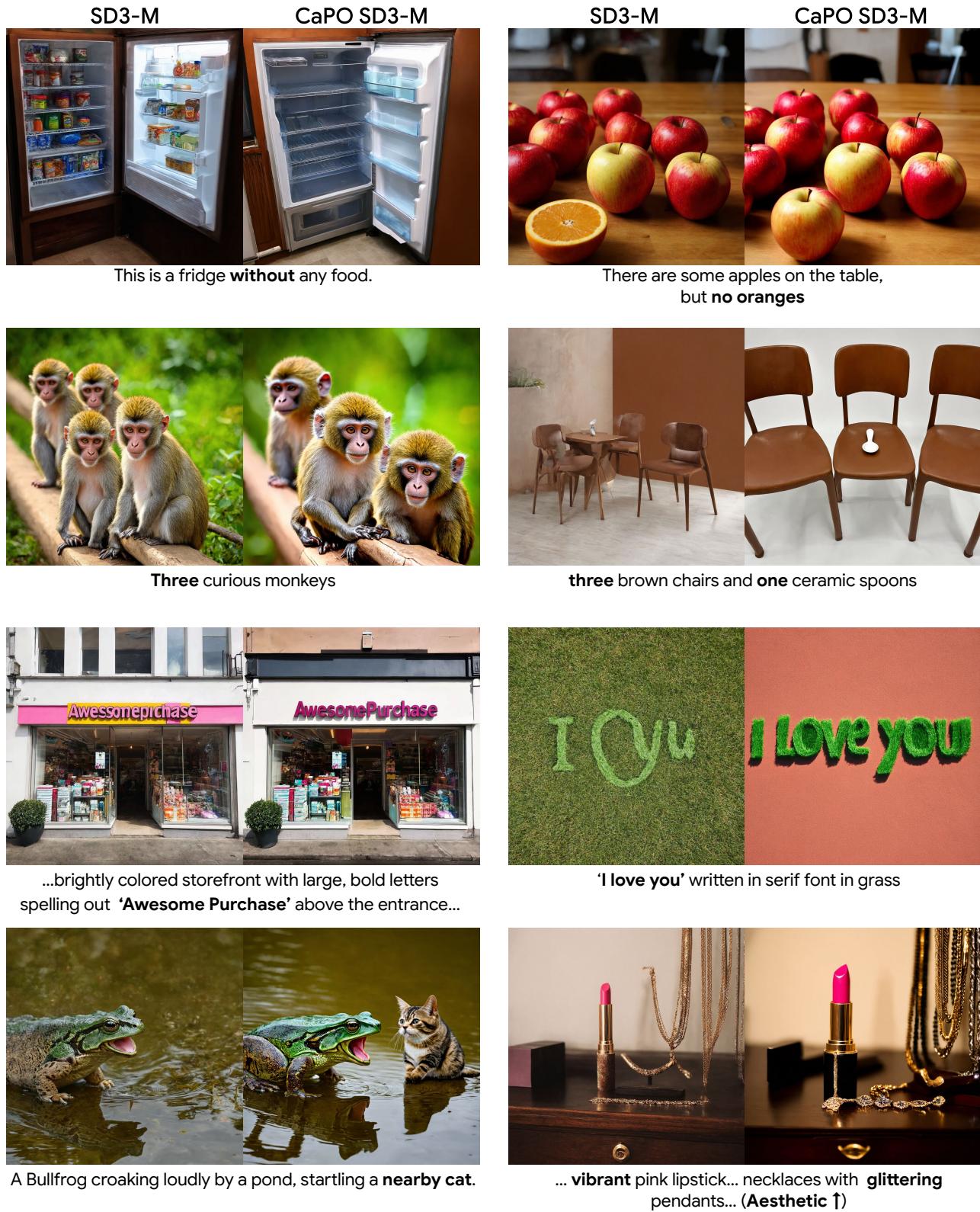


Figure 10. **Additional qualitative comparison between CaPO SD3-M and SD3-M.** We provide additional qualitative comparison between CaPO SD3-M and SD3-M. CaPO SD3-M shows better image-text alignment, e.g., negation (first row), counting (second row), visual text rendering (third row). Also it demonstrates better image aesthetics (fourth row right). We bold the text to highlight the prompts that demonstrate improvement in image-text alignment, and (**Aesthetic** ↑) to demonstrate the improvement in image aesthetic quality.