Divergence Minimization Preference Optimization for Diffusion Model Alignment

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion models have achieved remarkable success in generating realistic and versatile images from text prompts. Inspired by the recent advancements of language models, there is an increasing interest in further improving the models by aligning with human preferences. However, we investigate alignment from a divergence minimization perspective and reveal that existing preference optimization methods are typically trapped in suboptimal mean-seeking optimization. In this paper, we introduce Divergence Minimization Preference Optimization (DMPO), a novel and principled method for aligning diffusion models by minimizing reverse KL divergence, which asymptotically enjoys the same optimization direction as original RL. We provide rigorous analysis to justify the effectiveness of DMPO, and conduct comprehensive experiments to validate its empirical strength across both human evaluations and automatic metrics. Our extensive results show that diffusion models fine-tuned with DMPO can consistently outperform or match existing techniques, specifically outperforming all existing diffusion alignment baselines by *at least 64.6%* in PickScore across all evaluation datasets, demonstrating the method's superiority in aligning generative behavior with desired outputs. Overall, DMPO unlocks a robust and elegant pathway for preference alignment, bridging principled theory with practical performance in diffusion models.

1 Introduction

Diffusion models [9, 31, 33, 34] have emerged as a leading approach for text-to-image (T2I) generation [27, 22, 28], offering a scalable and robust framework for synthesizing images from natural language descriptions. These models typically adopt a single-stage training approach, learning the data distribution from large-scale web-crawled text-image pairs [29, 24, 21]. In contrast, Large Language Models (LLMs) have demonstrated remarkable success using a two-stage training paradigm [1, 6]. After pretraining on vast and noisy web data, LLMs undergo a crucial second phase of fine-tuning on smaller but more specialized datasets. This two-stage approach allows the models to first develop broad capabilities and then align with user preferences to fulfill diverse human needs [40, 6]. The fine-tuning phase enhances model responsiveness to human expectations without significantly diminishing the broad capabilities acquired during pretraining. These alignment techniques have provided tremendous inspiration for the text-to-image domain, where leveraging human preference data for a given prompt holds great promise of unlocking diffusion models' ability to align with human expectations. However, diffusion models and LLMs operate under fundamentally different mechanisms—LLMs rely on autoregressive factorization while diffusion models are built with a chain of Markov transitions. As a result, porting these two-stage alignment methods to diffusion architectures remains a significant challenge.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

In recent years, various RL-based methods have been developed for diffusion model alignment [18**?** , 5, 35, 16, 25]. Early-stage works focused on fine-tuning diffusion models through Reinforcement Learning from Human Feedback (RLHF) [20] after large-scale pretraining [18**?** , 5]. These approaches typically first fit a reward model on human preference data, and then optimize the diffusion model to generate images that receive high reward scores while avoiding excessive deviation from the original model. However, building reliable reward models for diverse tasks presents challenges, often requiring substantial datasets and considerable training resources [36, 37, 38]. To address this issue, recent research has focused on alignment techniques without reliance on reward models. Diffusion-DPO [35] was the first to avoid the reward model by extending the formulation of direct preference optimization (DPO) to diffusion models. However, our analysis reveals that this method corresponds to minimizing the variational bound of the forward Kullback-Leibler (KL) divergence between the target and model distributions, which is known to encourage compromised mean-seeking behavior [19, 11]. As a result, the resulting model often tends to cover diverse human intent but fails to capture exact modes of human preference, resulting in blurry or diluted generations. Followup works [16, 42] further propose learning with binary feedback instead of preference comparison, or fine-tuning the diffusion models from a score matching perspective. Nevertheless, none of these models challenge DPO's underlying formulation and thus the alignment performance degrades to a weak alignment method similar to supervised fine-tuning (SFT). As a result, developing principled and exact alignment methods for diffusion models remains an open problem.

In this paper, we first provide the formal analysis that the Diffusion-DPO objective is equivalent to optimizing the variational upper bound of KL divergence between target and model distributions. Based on our observation, we revisit the diffusion alignment framework from the new distribution matching perspective and argue that, compared to minimizing forward KL, optimizing reverse KL divergence provides a more mode-seeking objective. Such an objective enables more precise optimization towards the major mode in target distribution, thereby more accurately capturing the true structure of human preferences. Based on this insight, we propose Divergence Minimization Preference Optimization (DMPO), a method that fine-tunes diffusion models by minimizing the reverse KL between the model distribution and the theoretical optimal distribution. Though sharing the same optimal solution as Diffusion-DPO, in the practical scenario with limited network expressivity, DMPO enables higher human preferences alignment behavior by pushing the probability mass to the main characteristics of the target distribution. Furthermore, we theoretically show that under mild assumptions, DMPO optimizes the policy distribution in the same direction as the original RL objective which provides justification for the rationality and effectiveness of our proposed algorithm.

Through extensive experiments on generation and editing tasks, we demonstrate that DMPO can consistently outperform current alignment methods for diffusion models in both automatic evaluation metrics and human evaluations, exhibiting precise and reliable preference alignment capabilities for generative diffusion models. The main contributions of this paper can be summarized as: (1) The first proposal of fine-tuning diffusion models through minimizing divergence based on human preferences, pioneering a new perspective for designing preference learning algorithms for diffusion models; (2) Theoretical analysis demonstrating that minimizing reverse KL achieves more precise alignment and the conditions and rationality for applying this to diffusion models; and (3) Experimental evidence showing that DMPO can more accurately understand prompts, enhance the diversity of generative models, and outperform existing preference learning baseline methods without any additional computational cost.

2 Background

2.1 Diffusion Models

Suppose real data sample $\mathbf{x}_0$ follows data distribution $q(\mathbf{x}_0)$, denoising diffusion models [9] are generative models which have a discrete-time reverse process with a Markov structure $p_\theta(\mathbf{x}_{0:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ where

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(x_{t-1}; \mu_\theta(\mathbf{x}_t), \sigma^2_{t|t-1}\frac{\sigma^2_{t-1}}{\sigma^2_t} I). \tag{1}$$

Training the models is performed by minimizing the associated evidence lower bound (ELBO) [13, 32]:

$$L_{\text{Diff}} = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon},t,\mathbf{x}_t}\left[\omega(\lambda_t)\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2\right], \tag{2}$$

with $\boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I})$, $t\sim\mathcal{U}(0,T)$, $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t;\alpha_t\mathbf{x}_0,\sigma_t^2\boldsymbol{I})$. $\lambda_t = \alpha_t^2/\sigma_t^2$ is a signal-to-noise ratio [13], and $\omega(\lambda_t)$ is a pre-specified weighting function (typically chosen to be constant [9, 33]).

2.2 Reinforcement Learning from Human Feedback (RLHF)

RLHF is a foundational paradigm for aligning large-scale generative models with human preferences. The standard RLHF pipeline consists of three stages: (1) supervised fine-tuning (SFT) on curated instruction data, (2) learning a reward model from human preference comparisons, and (3) reinforcement learning to fine-tune the policy using the learned reward. In the RL phase, a parameterized policy $p_\theta(\mathbf{x}_0|c)$ (input condition $c \sim \mathcal{D}_c$) is optimized to maximize expected reward under the guidance of a fixed reward function $r(c,\mathbf{x}_0)$, typically trained using a Bradley–Terry model [3] on human-labeled preference pairs. To prevent the policy from drifting too far from a reference distribution $p_{\text{ref}}(\mathbf{x}_0|c)$ (often obtained from the SFT stage) the objective includes a KL-regularization term:

$$\max_{p_\theta} \mathbb{E}_{c\sim\mathcal{D}_c,\,\mathbf{x}_0\sim p_\theta(\mathbf{x}_0|c)}\left[r(c,\mathbf{x}_0)\right] - \beta\mathbb{D}_{\text{KL}}\left(p_\theta(\mathbf{x}_0|c)\,\|\,p_{\text{ref}}(\mathbf{x}_0|c)\right), \tag{3}$$

where $\beta > 0$ is a hyperparameter controlling regularization. Analytically, the optimal policy under this objective corresponds to a Boltzmann-rational distribution over the reference model [23]:

$$p^*(\mathbf{x}_0|c) = \frac{1}{Z(c)}\,p_{\text{ref}}(\mathbf{x}_0|c)\exp\left(\frac{1}{\beta}\,r(c,\mathbf{x}_0)\right), \tag{4}$$

where $Z(\mathbf{x})$ is the partition function. The goal of RLHF is to approximate this optimal policy p^* using the trainable policy p_θ.

Direct Preference Optimization (DPO). DPO bypasses the explicit reward learning and reinforcement learning steps in RLHF by directly modeling the optimal conditional distribution over outputs. Specifically, given preference pairs $(\mathbf{x}_0^w, \mathbf{x}_0^\ell)$ from $\mathcal{D}$, reward model can be rewrited as

$$r(c,\mathbf{x}_0) = \beta\log\frac{p_\theta(\mathbf{x}_0|c)}{p_{\text{ref}}(\mathbf{x}_0|c)} + \beta\log Z(c), \tag{5}$$

and substituting into the preference-based objective Equation (3) yields the DPO loss:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{c,\mathbf{x}_0^w,\mathbf{x}_0^\ell}\left[\log\sigma\left(\beta\left(\log\frac{p_\theta(\mathbf{x}_0^w|c)}{p_{\text{ref}}(\mathbf{x}_0^w|c)} - \log\frac{p_\theta(\mathbf{x}_0^l|c)}{p_{\text{ref}}(\mathbf{x}_0^l|c)}\right)\right)\right] \tag{6}$$

This formulation allows the model to be trained directly via preference comparisons, without the need to estimate a reward model or compute policy gradients.

3 Method

3.1 Revisiting Diffusion-DPO

Diffusion-DPO is the first method to apply DPO to diffusion models. The approach proposes to minimize Equation (3) of diffusion models by replacing the KL-divergence term with its upper bound $\mathbb{D}_{\text{KL}}\left[p_\theta(\mathbf{x}_{0:T}|c)\,\|\,p_{\text{ref}}(\mathbf{x}_{0:T}|c)\right]$, which is joint KL-divergence on sampling path $\mathbf{x}_{0:T}$. Based on the binary DPO formulation Equation (6), given text prompt c, image pairs $\mathbf{x}_0^w, \mathbf{x}_0^\ell$, Diffusion-DPO formulates the RLHF objective into the following objective:

$$\mathcal{L}_{\text{DPO-Diffusion}}(\theta) = -\mathbb{E}_{(\mathbf{x}_0^w,\mathbf{x}_0^\ell)\sim\mathcal{D}}\log\sigma\left(\beta\,\mathbb{E}_{\substack{\mathbf{x}_{1:T}^w\sim p_\theta(\mathbf{x}_{1:T}^w|\mathbf{x}_0^w)\\ \mathbf{x}_{1:T}^\ell\sim p_\theta(\mathbf{x}_{1:T}^\ell|\mathbf{x}_0^\ell)}}\left[\log\frac{p_\theta(\mathbf{x}_{0:T}^w)}{p_{\text{ref}}(\mathbf{x}_{0:T}^w)} - \log\frac{p_\theta(\mathbf{x}_{0:T}^\ell)}{p_{\text{ref}}(\mathbf{x}_{0:T}^\ell)}\right]\right), \tag{7}$$

which allows the diffusion model to be optimized using pairwise preference data. By leveraging the convexity of the $-\log\sigma$ function and further simplifying the DPO objective across two denoising trajectories, such loss can be decomposed as per-step alignment loss. This enables the model to efficiently align with human preferences through direct optimization of the denoising process.

In this paper, we further revisit Diffusion-DPO from the distribution matching perspective. Interestingly, we obtain the following theoretical result:

Theorem 1 *(informal) Generalizing Diffusion-DPO from the pairwise preference setting to the multi-sample setting with preference data sampled from the reference policy p_{ref}, we have that the gradient of Diffusion-DPO objective Equation (7) satisfies:*

$$\nabla_\theta \mathcal{L}_{\textit{DPO-Diffusion}}(\theta) = \nabla_\theta \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathbb{D}_{KL}\left(\hat{p}^*(\mathbf{x}_{0:T}|c) \parallel \hat{p}_\theta(\mathbf{x}_{0:T}|c)\right)\right], \tag{8}$$

where $\hat{p}^*(\mathbf{x}_{0:T}|c) \propto p_{ref}(\mathbf{x}_{0:T}|c)\exp(r(\mathbf{x}_{0:T}, c))$ and $\hat{p}_\theta(\mathbf{x}_{0:T}|c) \propto p_\theta(\mathbf{x}_{0:T}|c)^\beta p_{ref}(\mathbf{x}_{0:T}|c)^{1-\beta}$, i.e., Diffusion-DPO optimize the forward KL divergence $\mathbb{D}_{KL}\left(\hat{p}^*(\mathbf{x}_{0:T}|c) \parallel \hat{p}_\theta(\mathbf{x}_{0:T}|c)\right)$.

This states that Diffusion-DPO approximately minimizes a forward KL divergence between the whole sampling trajectory distribution of the preference-optimal policy and the learned policy. Extending this to the practical setting, Diffusion-DPO applies preference optimization on each step of the denoising process instead of the whole trajectory, which serves as an upper bound on the alignment objective. As a result, Diffusion-DPO similarly performs forward KL minimization to gradually align the model toward the optimal preference policy.

3.2 Our Method

While DPO optimizes a forward KL divergence between the optimal policy and the learned policy, this formulation introduces a key limitation: it enforces a mean-seeking behavior. Specifically, minimizing the Forward KL penalizes the learned policy p_θ harshly whenever it assigns low probability to any output favored by the optimal policy p^*. This encourages p_θ to cover the full support of p^*, often resulting in diffuse generations that fail to concentrate on highly preferred samples. In contrast, minimizing the reverse KL divergence provides a more targeted form of alignment. The reverse KL is given by $\mathbb{D}_{\mathrm{KL}}\big(\hat{p}_\theta(\mathbf{x}_{0:T}|c) \,\|\, \hat{p}^*(\mathbf{x}_{0:T}|c)\big)$, and only penalizes the model when it places mass in regions unsupported by p^*. This drives the model toward the high-reward modes of p^*, promoting sharper and more precise alignment with human preferences.

Based on this insight, we propose *Divergence Minimization Preference Optimization (*DMPO*)*, which replaces the *forward KL* with a *reverse KL* objective to align diffusion models with human preferences by minimizing a divergence between the learned policy and the latent optimal policy derived from Equation (4). According to the optimal policy p^* in Equation (4), we can rewrite the reward function in the form of $r(c, \mathbf{x}_{0:T}) = \log \hat{p}^*(\mathbf{x}_{0:T}|c) - \log p_{\mathrm{ref}}(\mathbf{x}_{0:T}|c)$. Besides, we can define the relative logits $g_\theta(c, \mathbf{x}_{0:T}) = \log \hat{p}_\theta(\mathbf{x}_{0:T}|c) - \log p_{\mathrm{ref}}(\mathbf{x}_{0:T}|c) = \beta(\log p_\theta(\mathbf{x}_{0:T}) - \log p_{\mathrm{ref}}(\mathbf{x}_{0:T}))$. Putting $r(c, \mathbf{x}_{0:T})$ and $g_\theta(c, \mathbf{x}_{0:T})$ into $\mathbb{D}_{\mathrm{KL}}\big(\hat{p}_\theta(\mathbf{x}_{0:T}|c) \,\|\, \hat{p}^*(\mathbf{x}_{0:T}|c)\big)$, the divergence objective becomes:

$$\mathcal{L}(\theta) = \mathbb{D}_{\mathrm{KL}}\left(\hat{p}_\theta(\mathbf{x}_{0:T}|c) \,\|\, \hat{p}^*(\mathbf{x}_{0:T}|c)\right) = \mathbb{E}_{\hat{p}_\theta}[\log\left(\frac{\hat{p}_\theta}{p_{\mathrm{ref}}} \cdot \frac{p_{\mathrm{ref}}}{\hat{p}_*}\right)] = \mathbb{E}_{p_{\mathrm{ref}}}\left[p^{g_\theta}\log\left(\frac{p^{g_\theta}}{p^r}\right)\right], \tag{9}$$

where p^{g_θ} and p^r are unnormalized distributions proportional to $\exp(g_\theta)$ and $\exp(r)$ over K samples respectively. Formly they can be written as :

$$p^{g_\theta}\left(i \mid \{\mathbf{x}_{0:T}\}_{1:K}, c\right) = \frac{e^{g_\theta(c, \{\mathbf{x}_{0:T}\}_i)}}{\sum_{j=1}^K e^{g_\theta(c, \{\mathbf{x}_{0:T}\}_j)}}, \quad p^r\left(i \mid \{\mathbf{x}_{0:T}\}_{1:K}, c\right) = \frac{e^{r(c, \{\mathbf{x}_{0:T}\}_i)}}{\sum_{j=1}^K e^{r(c, \{\mathbf{x}_{0:T}\}_j)}}. \tag{10}$$

Learning with Pair-wise Preference Data. We consider the practical objective given a dataset $\mathcal{D}$ consisting of text prompts c and preference-labeled output pairs $(\mathbf{x}_0^w, \mathbf{x}_0^\ell)$, where $\mathbf{x}_0^w$ is preferred over $\mathbf{x}_0^\ell$. By setting $K = 2$ in Equation (10), Equation (9) can be expressed as:"

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}}\left[\sigma(u(\theta))\log\frac{\sigma(u(\theta))}{1-\alpha} + \sigma(-u(\theta))\log\frac{\sigma(-u(\theta))}{\alpha}\right], \tag{11}$$

where $u(\theta) = g_\theta(c, \mathbf{x}_{0:T}^w) - g_\theta(c, \mathbf{x}_{0:T}^\ell) = \beta \cdot \mathbb{E}_{\substack{\mathbf{x}_{1:T}^w\sim p_\theta(\mathbf{x}_{1:T}^w|\mathbf{x}_0^w) \\ \mathbf{x}_{1:T}^l\sim p_\theta(\mathbf{x}_{1:T}^l|\mathbf{x}_0^l)}}\left[\log\frac{p_\theta(\mathbf{x}_{0:T}^w)}{p_{\mathrm{ref}}(\mathbf{x}_{0:T}^w)} - \log\frac{p_\theta(\mathbf{x}_{0:T}^l)}{p_{\mathrm{ref}}(\mathbf{x}_{0:T}^l)}\right]$.
Note we omit c as it's always independent of the path $\mathbf{x}_{0:T}$ as it does not affect the derivation (More details are included in Appendix A)

Since $p_\theta(\mathbf{x}_{0:T})$ is defined over the full trajectory $\mathbf{x}_{0:T}$ is computationally intractable, and the marginal likelihood $p_\theta(\mathbf{x}_{0:T})$ can not be evaluated exactly, to address this, we instead aim to estimate an upper bound of the original objective by applying Jensen's inequality. Before doing so, we examine the convexity of the function $f(u)$ which is extracted from Equation (11):

$$f(u) = \sigma(u)\log\frac{\sigma(u)}{1-\alpha} + \sigma(-u)\log\frac{\sigma(-u)}{\alpha}. \tag{12}$$

To analyze the convexity of Equation (12), we compute its second derivative $f''(u)$. Since α should be a very small positive constant, we find that there always exists an interval $(h_1(\alpha), h_2(\alpha))$ within which the function $f(u)$ maintains a strictly positive second derivative (e.g., when $\alpha = 10^{-3}$, $h_1(\alpha) = -0.28$, $h_2(\alpha) = 7.90$). In practice, the model does not choose preference data and is typically initialized near $u = 0$. As the optimization progresses, the optimization gradually pushes u in the positive direction, as the model increasingly focuses on modeling the winner distribution. The optimal value is reached when $u \to \log \frac{1-\alpha}{\alpha}$, which corresponds to the point where the model prediction matches the prior preference distribution. We also verify that for any $\alpha \in (0, 1)$, the right endpoint $h_2(\alpha)$ is always greater than $\log \frac{1-\alpha}{\alpha}$. Hence, the practical optimization trajectory $u \in (0, \log \frac{1-\alpha}{\alpha})$ always lies within the convexity-preserving interval $(h_1(\alpha), h_2(\alpha))$.

We made a mild assumption that the interval $(h_1(\alpha), h_2(\alpha))$ guaranteeing the convexity of the function is both wide enough for alignment. A more detailed theoretical derivation of this interval as a function of α is provided in Appendix A. Under this assumption we can use the Jensen inequality to move the expectation of $\mathbf{x}$ in Equation (11) outside:

$$\mathcal{L}(\theta) \leq \mathbb{E}_{\substack{(\mathbf{x}_0^w, \mathbf{x}_0^l) \sim \mathcal{D}, t \sim \mathcal{U}(0,T) \\ \mathbf{x}_t^w \sim p_\theta(\mathbf{x}_{t-1}^w, \mathbf{x}_t^w | \mathbf{x}_0^w) \\ \mathbf{x}_t^l \sim p_\theta(\mathbf{x}_{t-1}^l, \mathbf{x}_t^l | \mathbf{x}_0^l)}} \left[\sigma(u_t(\theta)) \log \frac{\sigma(u_t(\theta))}{1-\alpha} + \sigma(-u_t(\theta)) \log \frac{\sigma(-u_t(\theta))}{\alpha} \right], \tag{13}$$

where $u_t(\theta) = \beta T \left(\log \frac{p_\theta(\mathbf{x}_{t-1}^w | \mathbf{x}_t^w)}{p_{\text{ref}}(\mathbf{x}_{t-1}^w | \mathbf{x}_t^w)} - \log \frac{p_\theta(\mathbf{x}_{t-1}^l | \mathbf{x}_t^l)}{p_{\text{ref}}(\mathbf{x}_{t-1}^l | \mathbf{x}_t^l)} \right)$. In order to avoid intractable true posterior and enable efficient training, we approximate the reverse process $p_\theta(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{x}_0)$ using the forward distribution $q(\mathbf{x}_{1:T} | \mathbf{x}_0)$, which leads to the final loss:

$$\mathcal{L}_{\text{DMPO}}(\theta) = \mathbb{E}_{\substack{(\mathbf{x}_0^w, \mathbf{x}_0^l) \sim \mathcal{D}, t \sim \mathcal{U}(0,T) \\ \mathbf{x}_t^w \sim q(\mathbf{x}_t^w | \mathbf{x}_0^w) \\ \mathbf{x}_t^l \sim q(\mathbf{x}_t^l | \mathbf{x}_0^l)}} \left[\sigma(u_t(\theta)) \log \frac{\sigma(u_t(\theta))}{1-\alpha} + \sigma(-u_t(\theta)) \log \frac{\sigma(-u_t(\theta))}{\alpha} \right], \tag{14}$$

where $u_t(\theta)$ is defined as the following form:

$$\begin{aligned} u_t(\theta) = & -\beta T \Big(\mathbb{D}_{\text{KL}}(q(\mathbf{x}_{t-1}^w | \mathbf{x}_0^w, \mathbf{x}_t^w) \,\|\, p_\theta(\mathbf{x}_{t-1}^w | \mathbf{x}_t^w)) - \mathbb{D}_{\text{KL}}(q(\mathbf{x}_{t-1}^w | \mathbf{x}_0^w, \mathbf{x}_t^w) \,\|\, p_{\text{ref}}(\mathbf{x}_{t-1}^w | \mathbf{x}_t^w)) \\ & - \mathbb{D}_{\text{KL}}(q(\mathbf{x}_{t-1}^l | \mathbf{x}_0^l, \mathbf{x}_t^l) \,\|\, p_\theta(\mathbf{x}_{t-1}^l | \mathbf{x}_t^l)) + \mathbb{D}_{\text{KL}}(q(\mathbf{x}_{t-1}^l | \mathbf{x}_0^l, \mathbf{x}_t^l) \,\|\, p_{\text{ref}}(\mathbf{x}_{t-1}^l | \mathbf{x}_t^l)) \Big) \\ = & -\beta T \omega(\lambda_t) \Big(\|\epsilon^w - \epsilon_\theta(\mathbf{x}_t^w, t)\|_2^2 - \|\epsilon^w - \epsilon_{\text{ref}}(\mathbf{x}_t^w, t)\|_2^2 - \left(\|\epsilon^l - \epsilon_\theta(\mathbf{x}_t^l, t)\|_2^2 - \|\epsilon^l - \epsilon_{\text{ref}}(\mathbf{x}_t^l, t)\|_2^2 \right) \Big) \end{aligned}$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, $t \sim \mathcal{U}(0, T)$, $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \alpha_t \mathbf{x}_0, \sigma_t^2 \boldsymbol{I})$. $\lambda_t = \alpha_t^2 / \sigma_t^2$ is a signal-to-noise ratio, $\omega(\lambda_t)$ is a pre-specified weighting function). T is the maximum of training timesteps, we merge T and $\omega(\lambda_t)$ into β finally.

3.3 Theoretical Analysis

In this section, we further provide the theoretical analysis of the DMPO method by showing its connection with the original RLHF objective. Specifically, we show that the optimization direction of DMPO aligns with that of maximizing the RLHF objective in Equation (3). Let the RLHF loss for diffusion path $\mathbf{x}_{0:T}$ be denoted as $\mathcal{L}_{\text{RLHF}}$:

$$\mathcal{L}_{\text{RLHF}}(\theta) = \mathbb{E}_{c \sim \mathcal{D}_c, \, \mathbf{x}_{0:T} \sim p_\theta(\mathbf{x}_{0:T} | c)} \left[r(c, \mathbf{x}_0) \right] - \beta \mathbb{D}_{\text{KL}} \left[p_\theta(\mathbf{x}_{0:T} | c) \,\|\, p_{\text{ref}}(\mathbf{x}_{0:T} | c) \right]. \tag{15}$$

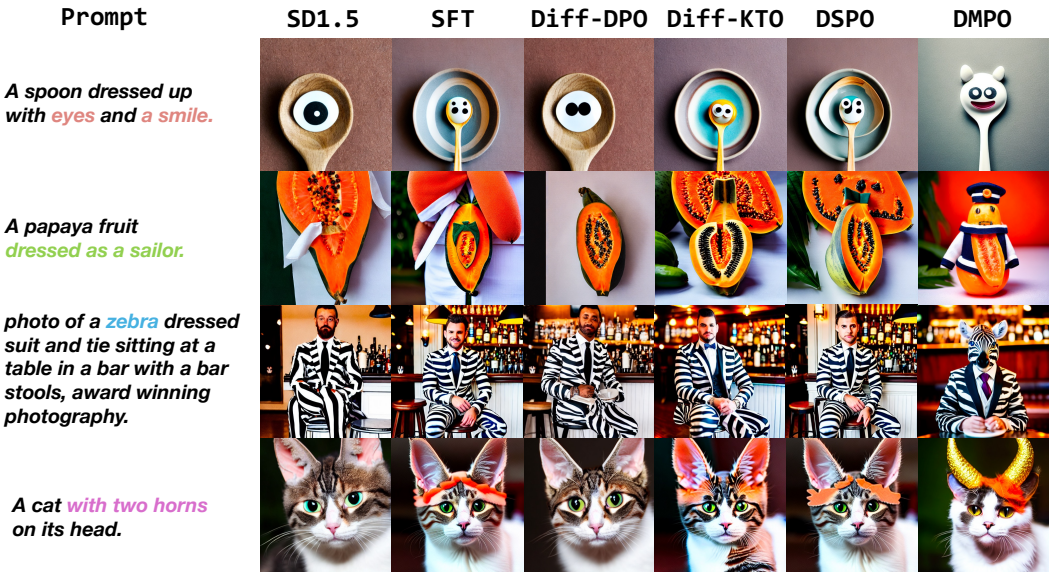
Starting from the divergence-based formulation in Equation (9), we formally have the following connection of the DMPO objective and the RL method objective.

Theorem 2 *Generalizing* DMPO *from the pairwise preference setting to the multi-sample setting with preference data sampled from the reference policy p_{ref}, when $\beta = 1$, we will have that the gradient of the* DMPO *objective satisfies:*

$$\nabla_\theta \mathcal{L}_{\text{DMPO}}(\theta) = \nabla_\theta \mathbb{E}_{c \sim \mathcal{D}_c, \, \mathbf{x}_{0:T} \sim p_\theta(\mathbf{x}_{0:T} | c)} \left[\mathbb{D}_{\text{KL}}(\hat{p}_\theta(\mathbf{x}_{0:T} | c) \,\|\, \hat{p}^*(\mathbf{x}_{0:T} | c)) \right] = -\nabla_\theta \mathcal{L}_{RLHF}(\theta). \tag{16}$$

Theorem 2 states that the optimization direction for $p_\theta(\mathbf{x}_{0:T})$ aligns with that of RLHF, which theoretically justifies that the objective of our DMPO is consistent with reinforcement learning–based alignment under specific conditions. The full technical derivation is provided in Appendix A. This highlights that though derived from a divergence minimization view, DMPO is consistent with the optimization direction of the original RLHF method.

Figure 1: We show the images generated by different models for various prompts which are selected from Pick-a-Pic V2, Parti-Prompt and HPS V2. "Diff" represents "Diffusion" for simplicity.



4 Related Work

Diffusion model alignment seeks to steer generative outputs toward human preferences by embedding reinforcement learning objectives within the denoising process [39, 38, 18]. Early efforts such as DDPO [2] leverage task-specific, hand-crafted reward functions—e.g., promoting compressibility—to fine-tune pretrained diffusion backbones. More recently, DPOK [7] replaces these bespoke rewards with feedback signals distilled from AI agents trained on vast human-preference corpora. Diffusion-DPO [35] adapts the core DPO framework, directly ingesting pairwise preference data to update the diffusion model. Diffusion-KTO [16] extends utility-based preference optimization to diffusion models using only binary human feedback and DSPO [42]. DSPO derives a novel step-wise alignment method for diffusion models from the perspective of score-based modeling. However, these approaches cannot precisely align the learned policy with the optimal policy derived from RLHF objective. In this work, we instead approach alignment by minimizing the divergence between the learned policy and the optimal policy, and adopt a reverse KL objective to explicitly align them. Efficient Exact Optimization (EXO) [11] and f-divergence Preference Optimization (f-PO) [8] also explored aligning language models (LMs) using reverse KL. However, all these studies are limited to autoregressive language models, while our method focuses on diffusion models and provides fundamentally different inspiration for aligning the Markov chains.

5 Experiments

5.1 Experimental Setting

Dataset and Baselines. We select Stable Diffusion v1.5 (SD1.5) and Stable Diffusion XL Base 1.0 (SDXL) as our base models. We train DMPO on the Pick-a-Pic V2 [14] dataset, which consists of pairwise preferences for images generated by SDXL-beta and Dreamlike. After excluding the $\sim$ 12% of pairs with ties, we end up with 851,293 preference pairs across 58,960 unique prompts. Our baselines include diffusion models fine-tuned on the Pick-a-Pic V2 dataset using various alignment methods, based on either SD1.5 or SDXL: Diffusion-DPO [35], Diffusion-KTO [16], DSPO [42], MAPO [10]. We also compare against the original SD1.5 and SDXL model as well as a supervised fine-tuning (SFT) SD1.5 model. We implement SFT and train DSPO by their released repo, and for Diffusion-DPO, KTO, and MAPO, we use the officially released checkpoints.

Training Details. We detail the implementation setup for DMPO in this section. Training is distributed across 4 NVIDIA A100 40GB GPUs, each processing a local batch of 2 pairs, with gradient accumulation over 256 steps to achieve the desired global batch size 2048. All models are

Table 1: Reward Score comparisons on Pick-a-Pic V2, HPS V2 and Parti-Prompt datasets for all baselines versus SD1.5, best results are in **boldface**. For simplicity, "Diff" represents "Diffusion".

Dataset	Method	Pick Score(↑)	HPS(↑)	CLIP(↑)	Aesthetics (↑)	Image Reward (↑)
Pick-a-Pic V2	SD1.5	0.2066	0.2612	0.3254	5.3200	-0.1478
	DMPO	**0.2165**	**0.2705**	**0.3453**	**5.6304**	**0.5415**
	SFT	0.2124	0.2701	0.3431	5.5213	0.4953
	Diff-DPO	0.2106	0.2642	0.3337	5.4729	0.0750
	Diff-KTO	0.2120	0.2701	0.3383	5.5848	0.5341
	DSPO	0.2120	0.2696	0.3396	5.6022	0.4683
HPS V2	SD1.5	0.2089	0.2672	0.3458	5.4249	-0.1175
	DMPO	**0.2195**	0.2768	**0.3629**	**5.7997**	**0.6350**
	SFT	0.2159	0.2771	0.3537	5.7356	0.5607
	Diff-DPO	0.2133	0.2706	0.3529	5.6014	0.1271
	Diff-KTO	0.2154	**0.2775**	0.3547	5.7381	0.5652
	DSPO	0.2155	0.2772	0.3551	5.7483	0.5385
Parti Prompt	SD1.5	0.2139	0.2679	0.3322	5.3115	0.0196
	DMPO	**0.2205**	**0.2758**	**0.3483**	**5.5438**	**0.6614**
	SFT	0.2172	0.2751	0.3401	5.5277	0.5293
	Diff-DPO	0.2163	0.2700	0.3382	5.3866	0.2243
	Diff-KTO	0.2173	**0.2758**	0.3408	5.5098	0.5551
	DSPO	0.2174	0.2755	0.3382	5.5291	0.5021

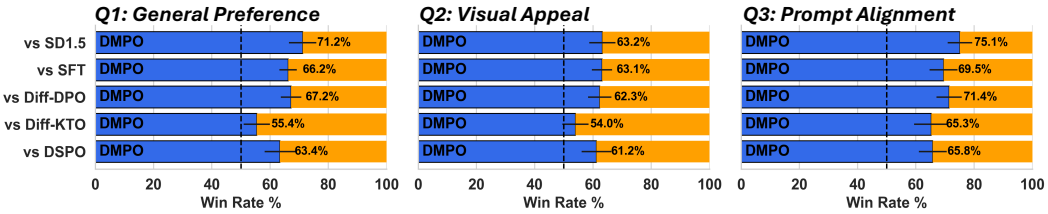
Table 2: (a) Win rate (%) comparisons on Pick-a-Pic V2, HPS V2 and Parti-Prompt datasets for all baselines versus SD1.5. (b) Win rate comparison between DMPO versus other baselines, win rates that surpass 50 % are in green, below 50 % are in red. For simplicity, "Diff" represents "Diffusion".

Dataset	Method1	Method2	Pick Score	HPS	CLIP	Aesthetics	Image Reward
Pick-a-Pic V2	DMPO	SD1.5	**82.40**	**84.80**	**65.60**	**75.60**	**80.40**
	SFT	SD1.5	74.60	82.60	61.60	72.40	77.80
	Diff-DPO	SD1.5	73.20	70.00	61.20	69.40	64.20
	Diff-KTO	SD1.5	73.60	84.60	58.60	74.00	80.00
	DSPO	SD1.5	74.60	84.80	61.60	73.60	76.20
	DMPO	SFT	67.20	50.60	52.80	53.20	50.20
	DMPO	Diff-DPO	73.40	77.20	56.80	64.80	68.40
	DMPO	Diff-KTO	69.00	50.80	55.20	55.20	50.20
	DMPO	DSPO	68.40	50.80	54.80	52.20	50.80
HPS V2	DMPO	SD1.5	**85.50**	86.50	**61.00**	**78.75**	79.75
	SFT	SD1.5	77.00	89.00	56.00	73.75	**81.50**
	Diff-DPO	SD1.5	77.50	69.25	57.75	69.5	64.50
	Diff-KTO	SD1.5	75.25	**89.50**	55.00	75.75	79.25
	DSPO	SD1.5	74.50	89.00	58.00	78.25	78.75
	DMPO	SFT	67.25	47.50	56.00	57.75	53.50
	DMPO	Diff-DPO	74.00	74.25	58.00	70.00	72.00
	DMPO	Diff-KTO	68.00	47.50	56.50	55.50	53.50
	DMPO	DSPO	65.75	50.00	59.00	57.00	54.00
Parti Prompt	DMPO	SD1.5	**76.72**	82.67	**62.01**	**71.69**	**75.25**
	SFT	SD1.5	66.36	83.88	54.35	71.51	72.37
	Diff-DPO	SD1.5	67.16	64.86	54.84	60.66	63.11
	Diff-KTO	SD1.5	65.56	84.38	54.41	69.97	73.86
	DSPO	SD1.5	67.30	**85.17**	53.31	70.22	73.71
	DMPO	SFT	64.64	51.31	57.23	51.69	56.00
	DMPO	Diff-DPO	69.30	72.54	58.15	65.62	69.55
	DMPO	Diff-KTO	67.03	48.40	59.13	55.15	55.46
	DMPO	DSPO	66.12	47.14	59.25	53.49	55.88

trained at fixed square resolutions using a learning rate of 1×10^{-5}, with a linear warmup spanning the first 500 steps. We select the best-performing model from training for evaluation; for DPMO with SD1.5 as the base model, we set the smoothing coefficient $\alpha = 0.01$ and $\beta = 2000$, while for DPMO with SDXL as the base model, we use $\alpha = 0.01$ and $\beta = 6000$. For more details on the model's behavior under various parameter settings, please refer to Appendix B.4.

Evaluation Details. We evaluate the performance of DPMO through both automated preference metrics and human user studies. For **general alignment evaluation**, we choose test prompts from

Figure 2: User study Results. DMPO significantly outperforms all baselines in human evaluation across three evaluation questions.



Pick-a-Pic V2, HPS V2, and PartiPrompt, and compare DMPO against all baselines on text-to-image generation task. For **image editing evalution**, we evaluate editing performance on two standard benchmarks, TEd-bench [12] and InstructPix2Pix [4] Bench by using SDEdit [17] pipeline and compare DMPO against all baselines. To ensure a fair comparison, we ensure consistency across all models, i.e., setting the guidance scale to 7.5, and the number of sampling steps to 50. For automated evaluation, we report three aspects: (1) Reward Score directly output by the reward models, (2) Win rates comparing DPMO and all baselines against the original base model, and (3) Pairwise win rates comparing DPMO directly with each baseline. The preference metrics include: **PickScore** [15], **HPS V2** [37], **CLIP** [26], **LAION Aesthetics Classifier** [30], and **ImageReward** [38], all of which are caption-aware models trained to predict human preference scores based on an image and its associated prompt. In addition, we conduct user study to compare DPMO with existing baselines. Detailed information on user annotations and questions in user study are provided in the Appendix B.1.

5.2 Text-to-Image Alignment Result

In this section, we present all DPMO and baseline text-to-image alignment results based on models trained with SD1.5. We further extend our study by conducting the same text-to-image alignment experiments with SDXL and by evaluating image editing performance using SD1.5. Additional results for both settings are provided in Appendix C.

Qualitative Result. Figure 1 presents a qualitative comparison of DMPO and other baseline models under the same prompt and. Compared to alternative methods, DMPO consistently demonstrates a stronger ability to capture the semantic intent of the prompt, producing outputs that are both more accurate and higher in quality. For instance, in the first row, only DMPO successfully renders the concept of a “smile”, while in the second row, it is the only model that correctly depicts "the papaya dressed as a sailor". More qualitative results can be found in the Appendix C.

Quantitative Result. Tables 1 and 2 report the reward-model scores for DMPO across all the test sets. From these tables, we observe that, for every reward metric, the DMPO-tuned models consistently outperform their respective base models (SD1.5). Generally, DMPO yields higher reward-model scores on Pick-a-Pic V2, HPS V2, and Parti-Prompt test set than almost all methods trained on pairwise preference data. In terms of Win rate comparison, it outperforms all other baseline models trained on Pick-a-Pic V2 by a win rate of at least 64.6% evaluated with PickScore. Moreover, Figure 2 shows the results of the user study shows human evaluators consistently prefer generations from the DMPO whether compared against the original SD1.5 or against all other baselines—further underscoring the effectiveness of the divergence minimization objective.

6 Conclusion

In this work, we propose *Divergence Minimization Preference Optimization (*DMPO*)*, a novel and theoretically grounded framework for aligning diffusion models with human preferences. Unlike prior methods that rely on forward KL divergence or implicit reward modeling, DMPO leverages reverse KL divergence to achieve mode-seeking behavior, enabling more precise and robust preference alignment. We provide a rigorous derivation of DMPO under the diffusion framework, including the formulation of alignment intervals. Through comprehensive experiments across multiple datasets and reward models, we demonstrate that DMPO consistently outperforms existing alignment baselines, showing stronger adherence to user preference. DMPO offers a principled and effective solution to the challenge of preference alignment in diffusion models, bridging theoretical insights and practical performance. We hope this work inspires further research into divergence-based alignment objectives for text-to-image generative models.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

[3] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[4] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[5] X. Dai, J. Hou, C.-Y. Ma, S. Tsai, J. Wang, R. Wang, P. Zhang, S. Vandenhende, X. Wang, A. Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023.

[6] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[7] Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee. DPOK: reinforcement learning for fine-tuning text-to-image diffusion models. In *NeurIPS*, 2023.

[8] J. Han, M. Jiang, Y. Song, S. Ermon, and M. Xu. f-po: Generalizing preference optimization with f-divergence minimization. *arXiv preprint arXiv:2410.21662*, 2024.

[9] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[10] J. Hong, S. Paul, N. Lee, K. Rasul, J. Thorne, and J. Jeong. Margin-aware preference optimization for aligning diffusion models without reference. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025.

[11] H. Ji, C. Lu, Y. Niu, P. Ke, H. Wang, J. Zhu, J. Tang, and M. Huang. Towards efficient exact optimization of language model alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[12] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, 2023.

[13] D. P. Kingma, T. Salimans, B. Poole, and J. Ho. Variational Diffusion Models. In *NeurIPS*, 2021.

[14] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy. Pick-a-pic: an open dataset of user preferences for text-to-image generation. In *NeurIPS*, 2023.

[15] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *arXiv preprint arXiv:2305.01569*, 2023.

[16] S. Li, K. Kallidromitis, A. Gokul, Y. Kato, and K. Kozuka. Aligning diffusion models by optimizing human utility. In *NeurIPS*, 2024.

[17] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022.

[18] Z. Miao, J. Wang, Z. Wang, Z. Yang, L. Wang, Q. Qiu, and Z. Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *CVPR*, 2024.

[19] T. Minka et al. Divergence measures and message passing. 2005.

[20] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[21] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[22] P. Pernias, D. Rampas, M. L. Richter, C. Pal, and M. Aubreville. Würstchen: An Efficient Architecture for Large-Scale Text-to-Image Diffusion Models. In *ICLR*, 2024.

[23] J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *ICML*, 2007.

[24] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[25] M. Prabhudesai, A. Goyal, D. Pathak, and K. Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv e-prints*, pages arXiv–2310, 2023.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.

[27] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[28] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-Shot Text-to-Image Generation. In *ICML*, 2021.

[29] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *CVPR*, 2022.

[30] C. Schuhmann. LAION-AESTHETICS. https://laion.ai/blog/laion-aesthetics/, 2022. Accessed: 2025-08-26.

[31] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[32] Y. Song, C. Durkan, I. Murray, and S. Ermon. Maximum Likelihood Training of Score-Based Diffusion Models. In *NeurIPS*, 2021.

[33] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

[34] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021.

[35] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion Model Alignment Using Direct Preference Optimization. In *CVPR*, 2024.

[36] Z. Wang, Y. Dong, O. Delalleau, J. Zeng, G. Shen, D. Egert, J. J. Zhang, M. N. Sreedhar, and O. Kuchaiev. HelpSteer 2: Open-source dataset for training top-performing reward models. In *NeurIPS*, 2024.

[37] X. Wu, Y. Hao, K. Sun, Y. Chen, F. Zhu, R. Zhao, and H. Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[38] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. ImageReward: learning and evaluating human preferences for text-to-image generation. In *NeurIPS*, 2023.

[39] K. Yang, J. Tao, J. Lyu, C. Ge, J. Chen, W. Shen, X. Zhu, and X. Li. Using Human Feedback to Fine-tune Diffusion Models without Any Reward Model. In *CVPR*, 2024.

[40] Q. A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y.-C. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, Z. Qiu, S. Quan, and Z. Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

[41] J. Yu, Y. Xu, J. Y. Koh, T. Luong, G. Baid, Z. Wang, V. Vasudevan, A. Ku, Y. Yang, B. K. Ayan, B. Hutchinson, W. Han, Z. Parekh, X. Li, H. Zhang, J. Baldridge, and Y. Wu. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *Transactions on Machine Learning Research*, 2022.

[42] H. Zhu, T. Xiao, and V. G. Honavar. DSPO: Direct Score Preference Optimization for Diffusion Model Alignment. In *ICLR*, 2025.

Code Availability. All code is available at an anonymous GitHub repository, which can reproduce all the experimental results in this paper:
https://anonymous.4open.science/r/dmpo-nips25/

Overview Here, we provide an overview of the Appendix below:

- §A presents theoretical analysis about all the theorems and our method.
- §B details the evaluation, ablation study and includes pseudo-code for reproducibility.
- §C provide more quantitative and qualitative results for DMPO fine-tuned on different base models.

A Proofs and Derivations

A.1 Proof of Theorem 1

In this section, we provide a detailed proof of Theorem 1, demonstrating that the objective of DPO corresponds to the forward KL divergence. Starting from Equation (10) and Equation (9), we extend the DPO loss $\mathcal{L}_{dpo}(K=2)$ from pair preference data to K samples selection, where the sigmoid needs to be replaced with softmax to present the soft distribution. Utilizing the definition of $\hat{p}_\theta(\mathbf{x}_{0:T})$:

$$\frac{\hat{p}_\theta(\mathbf{x}_{0:T}|\mathbf{c})}{p_{ref}(\mathbf{x}_{0:T}|\mathbf{c})} \propto \left(\frac{p_\theta(\mathbf{x}_{0:T}|\mathbf{c})}{p_{ref}(\mathbf{x}_{0:T}|\mathbf{c})}\right)^\beta$$

and by combining Equation (5), we can extend the dpo loss to:

$$\begin{aligned}
\mathcal{L}_{\text{dpo}}(\theta) &= \mathbb{E}_{c\sim\mathcal{D}_c}\mathbb{E}_{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_{1:K}|c)}\left[-\sum_{i=1}^{K}\frac{e^{r(c,\{\mathbf{x}_{0:T}\}_i)}}{\sum_{j=1}^{K}e^{r(c,\{\mathbf{x}_{0:T}\}_j)}}\log\left(\frac{e^{\beta\log\frac{p_\theta(\{\mathbf{x}_{0:T}\}_i|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_i|c)}}}{\sum_{j=1}^{K}e^{\beta\log\frac{p_\theta(\{\mathbf{x}_{0:T}\}_j|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_j|c)}}}\right)\right] \\
&= \mathbb{E}_{c\sim\mathcal{D}_c}\mathbb{E}_{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_{1:K}|c)}\left[-\sum_{i=1}^{K}\frac{e^{r(c,\{\mathbf{x}_{0:T}\}_i)}}{\sum_{j=1}^{K}e^{r(c,\{\mathbf{x}_{0:T}\}_j)}}\log\left(\frac{\log\frac{p_\theta(\{\mathbf{x}_{0:T}\}_i|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_i|c)}^\beta}{\sum_{j=1}^{K}\log\frac{p_\theta(\{\mathbf{x}_{0:T}\}_j|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_j|c)}^\beta}\right)\right] \\
&= \mathbb{E}_{c\sim\mathcal{D}_c}\mathbb{E}_{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_{1:K}|c)}\left[-\sum_{i=1}^{K}\frac{e^{r(c,\{\mathbf{x}_{0:T}\}_i)}}{\sum_{j=1}^{K}e^{r(c,\{\mathbf{x}_{0:T}\}_j)}}\log\left(\frac{\frac{\hat{p}_\theta(\{\mathbf{x}_{0:T}\}_i|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_i|c)}}{\sum_{j=1}^{K}\frac{\hat{p}_\theta(\{\mathbf{x}_{0:T}\}_j|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_j|c)}}\right)\right] \qquad (17)
\end{aligned}$$

When $K\to\infty$, for arbitrary function g, the estimate $\frac{1}{K}\sum_{i=1}^{K}g(\{\mathbf{x}_{0:T}\}_i)$ is unbiased since $\{\{\mathbf{x}_{0:T}\}_i\}_{i=1}^{K}$ are sampled from $p_{\text{ref}}(\cdot|c)$,, i.e.,

$$\lim_{K\to\infty}\frac{1}{K}\sum_{i=1}^{K}g(\{\mathbf{x}_{0:T}\}_i) = \mathbb{E}_{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}[g(\mathbf{x}_{0:T})].$$

For g is the relative $g(c,\mathbf{x}_{0:T}) = \frac{\hat{p}_\theta(\mathbf{x}_{0:T}|c)}{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}$, and $e^{r(c,\{\mathbf{x}_{0:T}\}_j)}$, we have:

$$\begin{aligned}
\sum_{j=1}^{K}\frac{\hat{p}_\theta(\{\mathbf{x}_{0:T}\}_j|c)}{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_j|c)} &= K\mathbb{E}_{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}\left[\frac{\hat{p}_\theta(\mathbf{x}_{0:T}|c)}{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}\right] = K, \\
\sum_{j=1}^{K}e^{r(c,\{\mathbf{x}_{0:T}\}_j)} &= K\mathbb{E}_{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}\left[e^{r(c,\mathbf{x}_{0:T})}\right] = KZ(c)
\end{aligned}$$

So that we can simplify the Equation (17):

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{dpo}}(\theta) &= \nabla_\theta \mathbb{E}_{c\sim\mathcal{D}_c}\left[-\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{p_{\text{ref}}(\{\mathbf{x}_{0:T}\}_i|c)}\left[\frac{e^{r(c,\{\mathbf{x}_{0:T}\}_i)}}{Z(c)}\log\frac{\hat{p}_\theta(\{\mathbf{x}_{0:T}\}_i|c)}{\hat{p}^*(\{\mathbf{x}_{0:T}\}_i|c)}\right]\right] \\
&= \nabla_\theta \mathbb{E}_{c\sim\mathcal{D}_c}\left[-\mathbb{E}_{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}\left[\frac{e^{r(c,\mathbf{x}_{0:T})}}{Z(c)}\log\frac{\hat{p}_\theta(\mathbf{x}_{0:T}|c)}{\hat{p}^*(\mathbf{x}_{0:T}|c)}\right]\right] \\
&= \nabla_\theta \mathbb{E}_{c\sim\mathcal{D}_c}\left[-\sum_{\mathbf{x}_{0:T}\in\mathcal{D}} p_{ref}(\mathbf{x}_{0:T}|c)\frac{e^{r(c,\mathbf{x}_{0:T})}}{Z(c)}\log\frac{\hat{p}_\theta(\mathbf{x}_{0:T}|c)}{\hat{p}^*(\mathbf{x}_{0:T}|c)}\right] \\
&= \nabla_\theta \mathbb{E}_{c\sim\mathcal{D}_c}\left[-\sum_{\mathbf{x}_{0:T}\in\mathcal{D}} \hat{p}^*(\mathbf{x}_{0:T}|c)\log\frac{\hat{p}_\theta(\mathbf{x}_{0:T}|c)}{\hat{p}^*(\mathbf{x}_{0:T}|c)}\right] \\
&= \nabla_\theta \mathbb{E}_{c\sim\mathcal{D}_c}\left[\mathbb{D}_{\text{KL}}\left(\hat{p}^*(\mathbf{x}_{0:T}|c)\parallel\hat{p}_\theta(\mathbf{x}_{0:T}|c)\right)\right],
\end{aligned}
$$

which completes the proof of Theorem 1.

A.2 Convexity of Equation (12)

We analyze the convexity by firstly computing the second derivative of Equation (12):

$$
f''(u) = \frac{e^u}{(1+e^{-u})^3}\left[-ue^u + (\log\frac{1-\alpha}{\alpha}+1)e^u + u + 1 - \log\frac{1-\alpha}{\alpha}\right]. \tag{18}
$$

Let $g(u) = -ue^u + (\log\frac{1-\alpha}{\alpha}+1)e^u + u + 1 - \log\frac{1-\alpha}{\alpha}$, then the sign of $f(u)$ is determined by the sign of $g(u)$, i.e., $\text{sgn}(f(u)) = \text{sgn}(g(u))$. We analyze the derivatives of $g(u)$:

$$
g'(u) = e^u\left(\log\frac{1-\alpha}{\alpha} - u\right) + 1, \quad g''(u) = e^u\left(\log\frac{1-\alpha}{\alpha} - u - 1\right). \tag{19}
$$

Clearly, $g''(u) = 0$ when $u_0 = \log\frac{1-\alpha}{\alpha} - 1$:

- For $u < u_0$, we have $g''(u) > 0$;
- For $u > u_0$, we have $g''(u) < 0$.

Therefore, $g'(u)$ attains its maximum at $u = u_0$. Moreover, we observe that:

$$
\lim_{u\to-\infty} g'(u) = 0^+, \quad \lim_{u\to+\infty} g'(u) = -\infty.
$$

Thus, by the intermediate value theorem, there exists a unique $u_1 > u_0$ such that $g'(u_1) = 0$. It follows that:

- $g'(u) > 0$ for $u < u_1$,
- $g'(u) < 0$ for $u > u_1$.

Therefore, $g(u)$ attains its global maximum at $u = u_1$. Since $g(u_0) = 2e^{u_0} > 0$, we have:

$$
g(u_1) > g(u_0) > 0.
$$

Furthermore, observe that $g(0) \equiv 2$, which is strictly positive. Additionally, we have:

$$
\lim_{u\to\pm\infty} g(u) = -\infty.
$$

Hence, by continuity and the monotonicity of $g(u)$, we conclude that for any $\alpha \in (0,1)$, there exists an interval $(h_1(\alpha), h_2(\alpha))$ such that

$$
h_1(\alpha) < 0 < h_2(\alpha) \quad \text{and} \quad g(u) > 0 \quad \text{for all} \quad u \in (h_1(\alpha), h_2(\alpha)).
$$

This proves that $f''(u) > 0$ within this interval.

Moreover, consider the scenario where α is small (e.g., $\alpha \in (0, 0.1]$), in this case, we can view the objective as a cross-entropy loss favoring the class with probability $1-\alpha$. The corresponding optimal logit is

$$u = \log\left(\frac{1-\alpha}{\alpha}\right).$$

Substituting this into $g(u)$, we obtain

$$g\left(\log\left(\frac{1-\alpha}{\alpha}\right)\right) = \frac{1}{\alpha} > 0.$$

This implies that, under this low-α regime, the interval in which $g(u) > 0$ is not only guaranteed to exist but also sufficiently large to cover the optimizer's likely solution. Therefore, the curvature condition $f''(u) > 0$ holds throughout the practically relevant interval.

A.3 Proof of Theorem 2

We first start by rearranging $\mathcal{L}_{\text{RLHF}}(p_\theta)$ from Equation (16) :

$$\begin{aligned}
\mathcal{L}_{\text{RLHF}}(\theta) &= \mathbb{E}_{c\sim\mathcal{D}_c}\left(\mathbb{E}_{p_\theta(\mathbf{x}_{0:T}|c)}\left[r(c,\mathbf{x}_{0:T})\right] - \beta\mathbb{D}_{\text{KL}}\left[p_\theta(\mathbf{x}_{0:T}|c)\|p_{\text{ref}}(\mathbf{x}_{0:T}|c)\right]\right) \\
&= \mathbb{E}_{c\sim\mathcal{D}_c}\left(\mathbb{E}_{p_\theta(\mathbf{x}_{0:T}|c)}\left[r(c,\mathbf{x}_{0:T})\right] - \beta\mathbb{E}_{p_\theta(\mathbf{x}_{0:T}|c)}\left[\log\frac{p_\theta(\mathbf{x}_{0:T}|c)}{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}\right]\right) \\
&= \mathbb{E}_{c\sim\mathcal{D}_c}\left(\beta\mathbb{E}_{\mathbf{x}_{0:T}\sim p_\theta(\cdot|c)}\left[\log e^{r(c,\mathbf{x}_{0:T})}\right] - \beta\mathbb{E}_{\mathbf{x}_{0:T}\sim p_\theta(\cdot|c)}\left[\log\frac{p_\theta(\mathbf{x}_{0:T}|c)}{p_{\text{ref}}(\mathbf{x}_{0:T}|c)}\right]\right) \\
&= \mathbb{E}_{c\sim\mathcal{D}_c}\mathbb{E}_{\mathbf{x}_{0:T}\sim p_\theta(\cdot|c)}\left[\beta\log\left(\frac{p_{\text{ref}}(\mathbf{x}_{0:T}|c)e^{r(c,\mathbf{x}_{0:T})}}{p_\theta(\mathbf{x}_{0:T}|c)}\right)\right]
\end{aligned}$$

Utilizing the definition of $\hat{p}^*$,we substitute $p_{\text{ref}}(\mathbf{x}_{0:T}|c)e^{r(c,\mathbf{x}_{0:T})}$ into the expression of $\mathcal{L}_{\text{RLHF}}(\theta)$:

$$\begin{aligned}
\mathcal{L}_{\text{RLHF}}(\theta) &= \mathbb{E}_{c\sim\mathcal{D}_c}\mathbb{E}_{\mathbf{x}_{0:T}\sim p_\theta(\cdot|c)}\left[\beta\log\left(\frac{H\cdot\hat{p}^*(\mathbf{x}_{0:T}|c)}{p_\theta(\mathbf{x}_{0:T}|c)}\right)\right] \\
&= \beta\mathbb{E}_{c\sim\mathcal{D}_c}\left[-\mathbb{D}_{\text{KL}}\left(p_\theta(\mathbf{x}_{0:T}|c)\|\hat{p}^*(\mathbf{x}_{0:T}|c)\right) + H\right].
\end{aligned}$$

where H is a constant independent of θ. Note that when β = 1. p_θ can be also $\hat{p}_\theta$ as the definition. Hence when β = 1, the final RLHF loss will be :

$$\mathcal{L}_{\text{RLHF}}(\theta) = \mathbb{E}_{c\sim\mathcal{D}_c}\left[-\mathbb{D}_{\text{KL}}\left(\hat{p_\theta}(\mathbf{x}_{0:T}|c)\|\hat{p}^*(\mathbf{x}_{0:T}|c)\right) + H\right], \tag{20}$$

Then we compute the gradient of above loss, as H is a constant independent of θ, finall the RLHF loss will be

$$\nabla_\theta\mathcal{L}_{\text{RLHF}}(\theta) = -\nabla_\theta\mathcal{L}_{\text{DMPO}}(\theta) \tag{21}$$

so that we complete the proof of Theorem 2.

B Experimental Details

B.1 Details of Evaluation

Reward Score. For each generated or edited image I_1 and its corresponding target prompt c, we compute an automated *reward score*:

$$R(I_1, c) \;=\; r\big(I_1, c\big),$$

where r is a pretrained vision–language reward model that outputs a scalar alignment score. We directly report the mean reward score over the entire test set for Table 1, Table 3 and Table 5:

$$\text{Reward Score} = \frac{1}{N}\sum_{i=1}^{N} R\big(I_1^{(i)}, p^{(i)}\big).$$

Win rate using Reward Score. Given a set of N image–prompt pairs $\{(I_1^{(i)}, I_2^{(i)}, p^{(i)})\}_{i=1}^{N}$, where $I_1^{(i)}$ and $I_2^{(i)}$ are the outputs from method 1 and 2 respectively for prompt $p^{(i)}$, we compute the reward score for each image using a pretrained reward model r:

$$R_1^{(i)} = r\big(I_1^{(i)}, p^{(i)}\big), \quad R_2^{(i)} = r\big(I_2^{(i)}, p^{(i)}\big).$$

We then assign a win to the method with the higher reward score on each instance:

$$\delta_i = \begin{cases} 1, & R_1^{(i)} > R_2^{(i)}, \\ 0, & \text{otherwise.} \end{cases}$$

The overall reward-based win rate of method 1 over method 2 is calculated as:

$$\text{WinRate}_{1>2} = \frac{1}{N} \sum_{i=1}^{N} \delta_i \times 100\%.$$

Details of user study.

We randomly sampled a total of 100 prompts—equally drawn from the test set of Pick-a-Pic V2, HPS V2, and Parti-Prompt—and generated one image per prompt by different models for the user study. These image–prompt pairs were then presented to human annotators via a custom web-based interface for blind evaluation.

Specifically, we include all baselines in our comparison. We pair the outputs generated by our DMPO-finetuned SD1.5 model with those from the following methods: Diffusion-DPO, Diffusion-KTO, DSPO, and SFT-finetuned SD1.5. This results in a total of 500 image pairs constructed from the 100 prompts.

Each comparison is evaluated by **2 to 3 human annotators**, yielding a total of **1500 judgments**. For each comparison, the annotator is shown a prompt and two images generated by different models, and is asked to answer the three evaluation questions: *Q1. General Preference*: Which image do you prefer given the prompt? *Q2. Visual Appeal*: Which image is more visually appealing regardless of the prompt? and *Q3. Prompt Alignment*: Which image better matches the text description?. We sampled 100 test prompts from Pick-a-Pic V2, HPS V2, and PartiPrompt, and evaluated DMPO (based on SD1.5), SD1.5, and all baselines. We report the final win rates by aggregating the human preferences across all comparisons, as shown in Figure 2.

B.2 Details of Datasets

Pick-a-Pic [14]: This comprehensive dataset captures real user preferences from the Pick-a-Pic web platform, where users generate images from text prompts. The collection encompasses more than 500,000 preference examples derived from over 35,000 unique prompts. Each entry consists of a text prompt, two AI-generated images, and user feedback indicating their preferred image or marking cases where no clear preference exists. The dataset incorporates outputs from various generative models, including Stable Diffusion 2.1, Dreamlike Photoreal 2.05, and multiple Stable Diffusion XL configurations, utilizing different classifier-free guidance parameters during generation.

Parti-Prompts [41]: This benchmark dataset features more than 1,600 carefully crafted English prompts designed to evaluate text-to-image model capabilities. The prompts cover diverse categories and present varied challenges, enabling comprehensive assessment of model performance across multiple evaluation criteria.

HPS V2 [37]: This preference-based collection contains 98,807 generated images sourced from 25,205 distinct prompts. The dataset structure allows for multiple image generations per prompt, with users selecting their preferred output while other variations serve as negative examples. The distribution includes 23,722 prompts with four associated images, 953 prompts with three images, and 530 prompts paired with two images.

TEd-bench [12]: Textual Editing Benchmark is a text-to-image editing benchmark that provides pairs of real images and corresponding editing prompts. For example, given a source image of a dog running, the target prompt might be "A cat running on the grass", in which case the model is expected to modify the original image so that it depicts the desired semantics. We evaluate both

existing baselines and our proposed method on TEd-bench by applying an automated reward model to score each edited image, and report the average reward as our editing performance metric.

InstructPix2Pix [4]: This specialized collection focuses on instruction-based image editing capabilities. The dataset enables models to modify existing images according to natural language instructions, such as "make the clouds rainy." The system processes both the editing instruction and the original image to produce the desired modifications. Our experimental evaluation utilized 1,000 test samples from this dataset to assess instruction-following image editing performance. Note that although reward models such as CLIP [26] may not comprehensively capture the semantics of instruction-style prompts when scoring edited images, we maintain that if the key terms from the instruction appear faithfully in the generated image, this constitutes a valid comparison.

B.3 Pseudo code of DMPO

```
def loss(model, ref_model, x_w, x_l, c, alpha, beta):
    """
        Calculate the DMPO loss for preferred image pair.
    model: Diffusion model with prompt and timestep conditioning.
    ref_model: Frozen reference model.
    x_w: Preferred latent.
    x_l: Less preferred latent.
    c: Conditioning input (e.g., caption text).
    alpha: smoothing coefficient.
    beta: Regularization strength.
    """
    t = torch.randint(0, 1000)
    noise = torch.randn_like(x_w)
    noisy_x_w = add_noise(x_w, noise, t)
    noisy_x_l = add_noise(x_l, noise, t)
    model_w_pred = model(noisy_x_w, c, t)
    model_l_pred = model(noisy_x_l, c, t)
    ref_w_pred = ref_model(noisy_x_w, c, t)
    ref_l_pred = ref_model(noisy_x_l, c, t)
    model_w_err = (model_w_pred - noise).norm().pow(2)
    model_l_err = (model_l_pred - noise).norm().pow(2)
    ref_w_err = (ref_w_pred - noise).norm().pow(2)
    ref_l_err = (ref_l_pred - noise).norm().pow(2)

    w_diff = model_w_err - ref_w_err
    l_diff = model_l_err - ref_l_err

    inside_term = -1 * beta * (w_diff - l_diff)
    loss_1 = torch.sigmoid(-inside_term) * (torch.logsigmoid(-
        inside_term) - torch.log(alpha))
    loss_2 = torch.sigmoid(inside_term) * (torch.logsigmoid(
        inside_term) - torch.log(1-alpha))
    loss = loss_1 + loss_2
    return loss
```

B.4 Ablation Study

We conduct ablation studies to understand the sensitivity and behavior of our alignment objective: the effect of β and the effect of α. Figure 3a illustrates the performance as β increases. As β decreases, the optimization objective degenerates into a pure reward function, leading to a drop in performance. Conversely, as β increases, the KL-divergence penalty becomes overly restrictive, greatly limiting the model's capacity to adapt. Figure 3b illustrates the performance as α increases. Regarding the smoothing coefficient α, we set α to be a small positive number since it represents the probability of the less preferred sample to avoid numerical instability. Therefore, we set $a \in \{0.1, 0.01, 0.001\}$ for ablation study. As shown in Appendix A, we observe that as α decreases, the second derivative of the objective $f(u)$ in Equation (12) increases for $u > 0$, resulting in a looser upper bound. This may weaken alignment precision. In particular, when α is very small, the reward term becomes smoothed

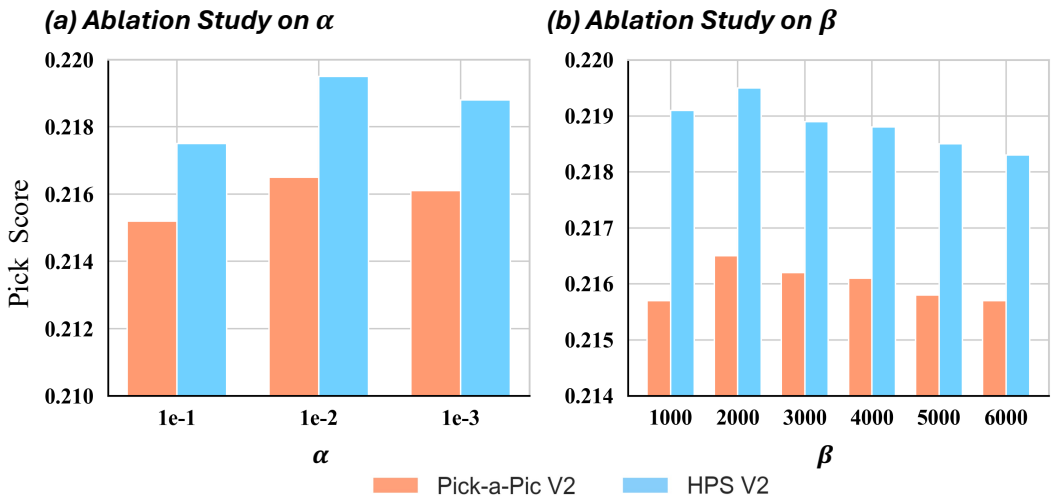


Figure 3: **Ablation study** on parameters α and β, evaluated on the Pick-a-Pic V2 and HPS V2 test sets. (a) Effect of β: **With α fixed at 0.01**, model performance first increases and then decreases as β increases. (b) Effect of α: **With β fixed at 2000**, performance also first increases and then decreases as α increases.

by the coefficient α, causing the model to occasionally favor less-preferred samples in preference pairs.

C More Experimental Results

C.1 More Qualitative Results

Text-to-Image Alignment Qualitative Results To further demonstrate the effectiveness of DMPO, we provide additional qualitative results on text-to-image alignment and image editing tasks. These results in Figure 4, Figure 5, Figure 6, and Figure 7 clearly show that DMPO consistently produces superior outputs compared to baseline methods. We list the prompts used in the Figures as follows:

Prompts used in Figure 4:

- a painting of a fox in the style of starry night.
- Cyberpunk cat.
- The image is a vibrant and intricate illustration of a man, with a focus on his shoulder and head, created using inkpen and Unreal Engine technology.
- A cat with two horns on its head.
- A spoon dressed up with eyes and a smile.
- A papaya fruit dressed as a sailor.
- A giant dinosaur frozen into a glacier and recently discovered by scientists, cinematic still.
- photo of a zebra dressed suit and tie sitting at a table in a bar with a bar stools, award winning photography.

Prompts used in Figure 5:

- an image of a photo model wearing a red lace dress, standing in the jungle.
- a child in the air while jumping on a trampoline.
- Two cups of coffee, one with latte art of the words "LOVE" written in one. The other has latte art of the words "PEACE" written in the other.
- anthropomorphic coffee bean drinking coffee.

- 16-year-old teenager wearing a white bear-ear hat with a smirk on their face.
- Frontal portrait of an anime girl with pink hair and sunglasses wearing a white t-shirt.
- A giant dinosaur frozen into a glacier and recently discovered by scientists, cinematic still.
- Of the lunar module landing on a hydrogen lake on Titan, through a foggy yellow smog.

Prompts used in Figure 6:

- A girl with pink pigtails and face tattoos.
- A cat with two horns on its head.
- A hand-drawn cute gnome holding a pumpkin in an autumn disguise, portrayed in a detailed close-up of the face with warm lighting and high detail.
- A cute puppy leading a session of the United Nations, newspaper photography.
- A purple raven flying over Big Sur, light fog, deep focus+closeup, hyper-realistic, volumetric lighting, dramatic lighting, beautiful composition, intricate details, instagram, trending, photograph, film grain and noise, 8K, cinematic, post-production.
- photo of a zebra dressed suit and tie sitting at a table in a bar with a bar stools, award winning photography.

Prompts used in Figure 7:

- A photo of a bird spreading wings.
- A half eaten pizza.
- A jumping horse.
- A goat and a cat hugging.
- A teddy bear holding a cup.
- Image of a cat wearing a floral shirt.
- A door with a pet entrance.
- A photo of a cat wearing a hat.

Image Editing Qualitative Results

Beyond improving alignment in image generation tasks, DMPO also significantly enhances the model's capabilities in image editing, particularly for text-guided image editing scenarios. This improvement stems from the model's strengthened ability to interpret and execute complex textual instructions. Figure 7 presents representative qualitative editing results on TEd-bench. In the second row, where the input prompt is "A half eaten pizza.", only DMPO generates an image that is both semantically faithful and highly visually appealing. In the fourth row, only DMPO correctly understands and represents the content 'hugging'.

C.2 More Quantitative Results

We also conduct quantitative evaluations and analyses for all experiments based on SD1.5 and SDXL. As shown in Table 3, Table 4, Table 5 and Table 6, DMPO demonstrates strong generalization ability, consistently outperforming other baselines in both reward model scores and win rates across both base models.

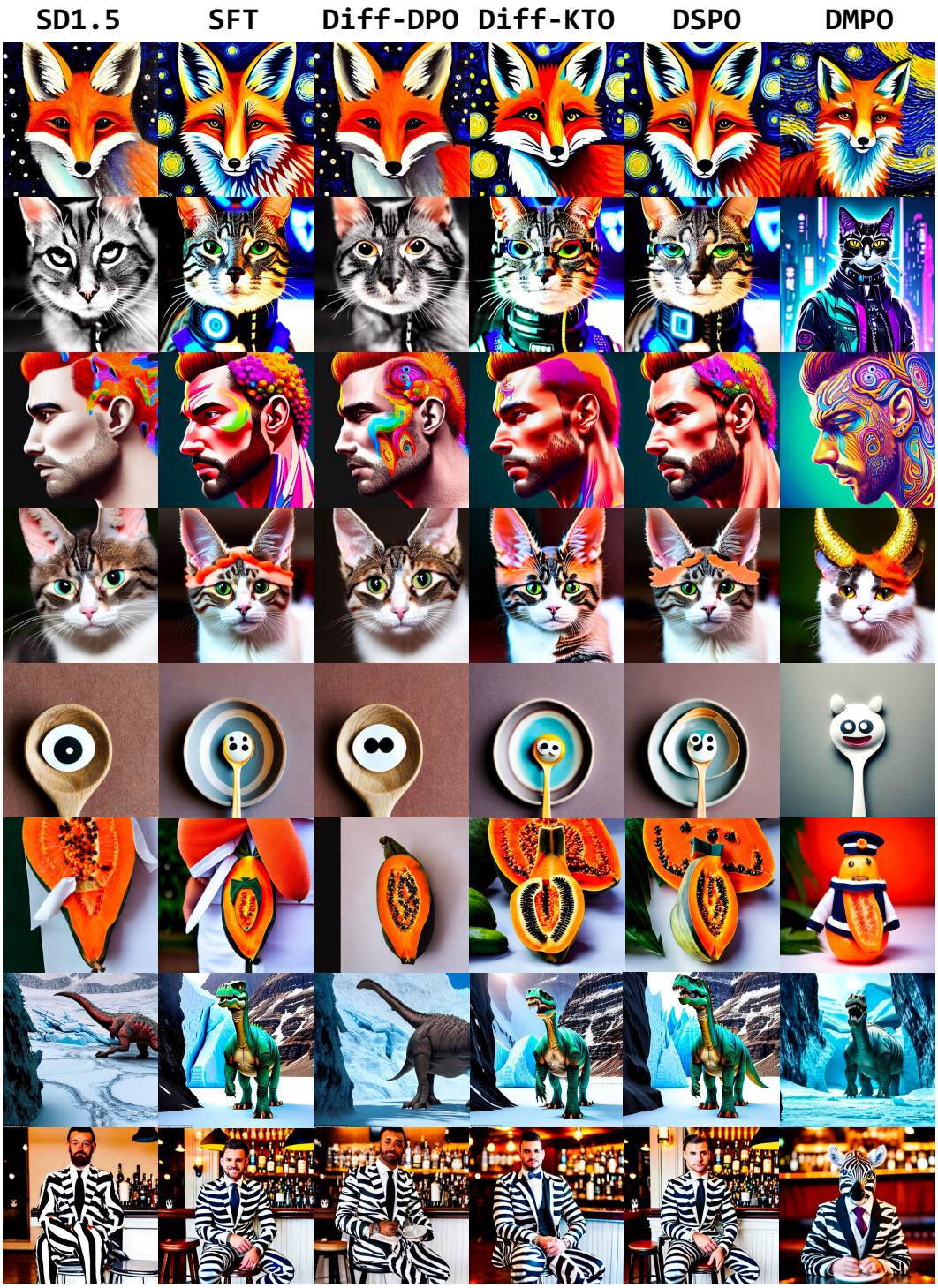


Figure 4: Images generated by different models (based on SD1.5) for various prompts which are selected from Pick-a-Pic V2, Parti-Prompt and HPS V2. Prompts used can be found in Appendix C.1

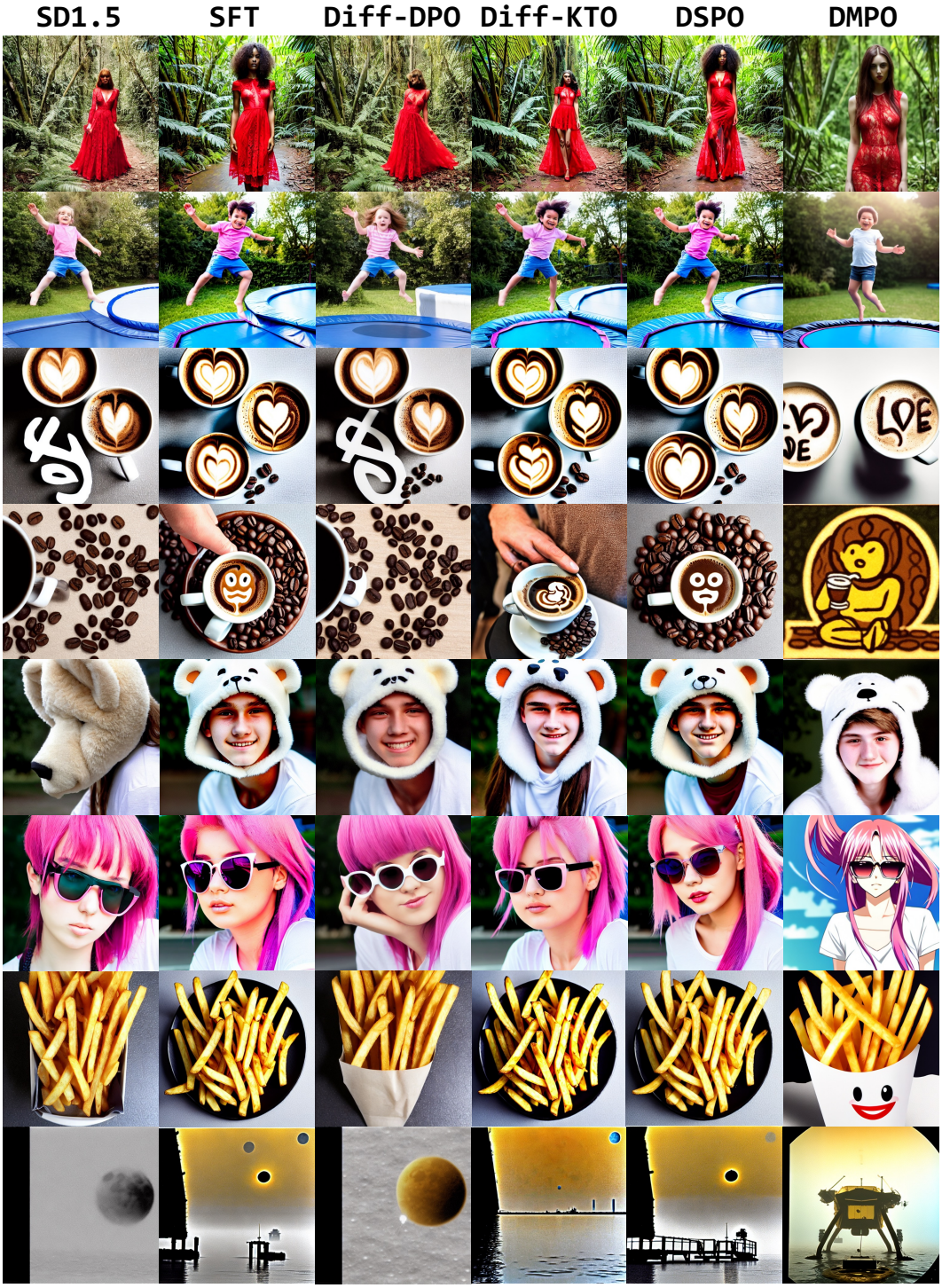


Figure 5: Images generated by different models (based on SD1.5) for various prompts which are selected from Pick-a-Pic V2, Parti-Prompt and HPS V2. Prompts used can be found in Appendix C.1.

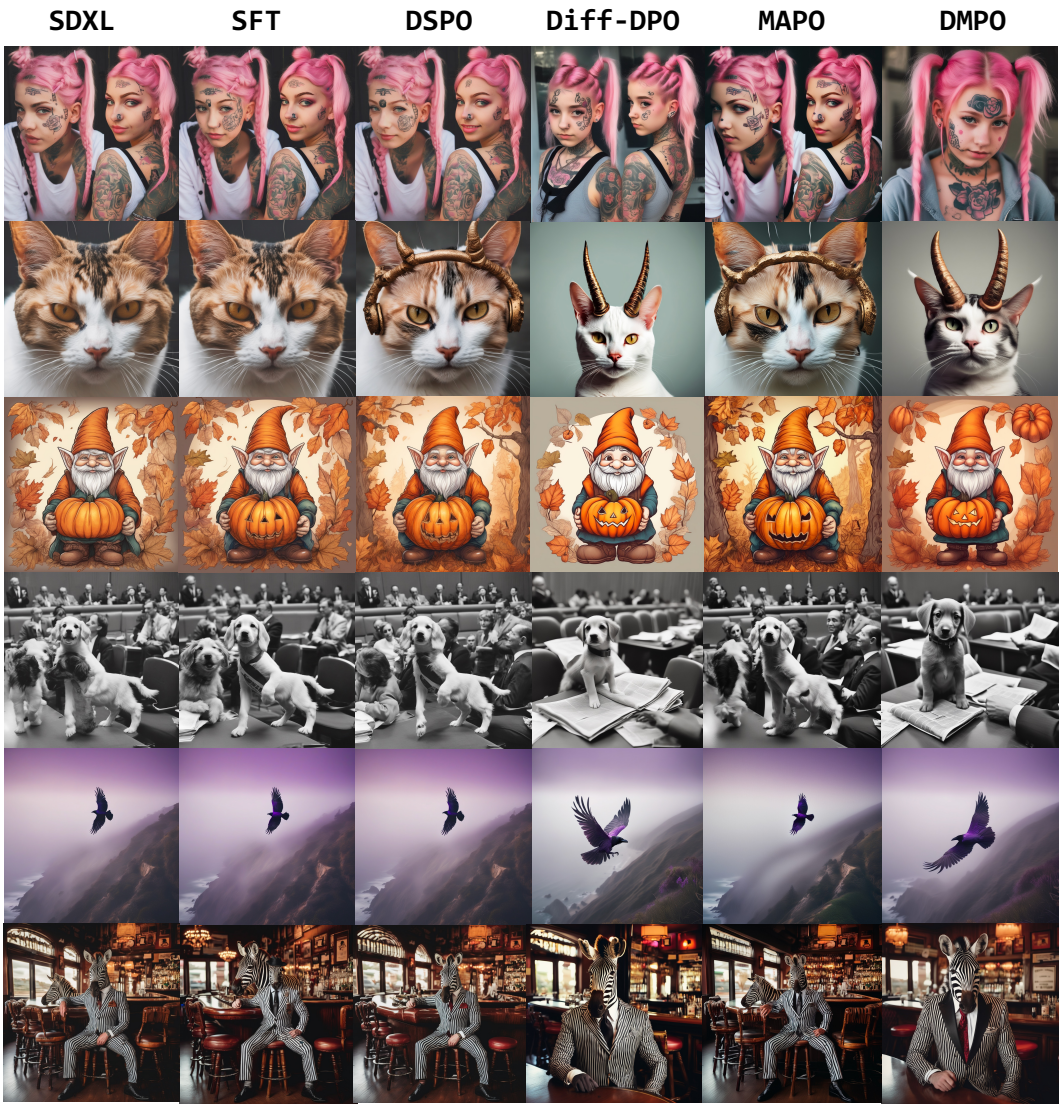


Figure 6: Images generated by different models (based on SDXL) for various prompts which are selected from Pick-a-Pic V2, Parti-Prompt and HPS V2. Prompts used can be found in Appendix C.1.

Table 3: Reward Score comparisons on TEd-bench and Instructpix2pix datasets for all baselines versus SD1.5, best results are in **boldface**. For simplicity, "Diff" represents "Diffusion".

Dataset	Method	Pick Score(↑)	HPS(↑)	CLIP(↑)	Aesthetics (↑)	Image Reward (↑)
TEd-bench	SD1.5	0.2165	0.2743	0.3043	5.4194	-0.0078
	DMPO	**0.2218**	**0.2796**	**0.3301**	5.5982	**0.5619**
	SFT	0.2185	0.2789	0.3061	**5.6185**	0.1869
	DPO	0.2182	0.2756	0.3065	5.4676	0.0601
	KTO	0.2190	0.2795	0.3125	5.6142	0.3298
	DSPO	0.2183	0.2787	0.3094	5.6022	0.1822
Instructpix2pix	SD1.5	0.2044	0.2561	0.2557	5.4923	-0.4589
	DMPO	**0.2090**	**0.2618**	**0.2879**	5.4849	**-0.0151**
	KTO	0.2073	0.2612	0.2700	5.7312	-0.0623
	DPO	0.2054	0.2572	0.2613	5.5066	-0.3727
	SFT	0.2076	0.2610	0.2680	**5.8001**	-0.0986
	DSPO	0.2076	0.2611	0.2687	5.7972	-0.0825

Figure 7: Images edited by different models (based on SD1.5) for various prompts which are selected from TEd-bench. Prompts used can be found in Appendix C.1.

Table 4: (a) Win rate (%) comparisons about imaging editing on TEd-bench datasets for all baselines (fine-tuned on SD1.5) versus SD1.5. (b) Win rate (%) comparison between DMPO versus other baselines, win rates that surpass 50 % are in green, below 50 % are in red. For simplicity, "Diff" represents "Diffusion".

Dataset	Method1	Method2	Pick Score	HPS	CLIP	Aesthetics	Image Reward
TEd-bench	DMPO	SD1.5	**80.00**	88.00	**71.00**	**84.00**	**85.00**
	SFT	SD1.5	67.00	83.00	46.00	83.00	71.00
	DPO	SD1.5	75.00	69.00	56.00	62.00	55.00
	KTO	SD1.5	64.00	**94.00**	54.00	83.00	69.00
	DSPO	SD1.5	59.00	81.00	50.00	84.00	71.00
	DMPO	SFT	64.00	51.00	71.00	48.00	66.00
	DMPO	DPO	74.00	77.00	70.00	63.00	78.00
	DMPO	KTO	60.00	48.00	64.00	50.00	53.00
	DMPO	DSPO	67.00	50.00	68.00	51.00	69.00
Instructpix2pix	DMPO	SD1.5	**81.30**	80.60	**75.50**	58.50	**78.90**
	SFT	SD1.5	77.00	85.80	64.90	79.30	78.00
	DPO	SD1.5	68.50	67.70	60.50	54.50	61.10
	KTO	SD1.5	73.40	**84.70**	66.10	**72.80**	74.80
	DSPO	SD1.5	77.40	85.10	64.80	79.50	78.10
	DMPO	SFT	63.30	51.60	69.70	40.30	52.70
	DMPO	DPO	75.90	76.60	73.90	54.40	72.40
	DMPO	KTO	63.60	52.20	66.50	43.90	51.10
	DMPO	DSPO	61.40	51.10	68.20	40.60	49.00

Table 5: Reward Score comparisons on Pick-a-Pic V2, HPS V2 and Parti-Prompt datasets for all baselines versus SDXL, best results are in **boldface**. For simplicity, "Diff" represents "Diffusion".

Dataset	Method	Pick Score(↑)	HPS(↑)	CLIP(↑)	Aesthetics (↑)	Image Reward (↑)
Pick-a-Pic V2	SDXL	0.2203	0.2661	0.3609	5.9892	0.5111
	DMPO	**0.2264**	**0.2730**	**0.3741**	5.9422	**0.8563**
	SFT	0.2224	0.2675	0.3624	5.9239	0.5834
	Diff-DPO	0.2256	0.2709	0.3722	5.9890	0.7584
	MAPO	0.2213	0.2682	0.3615	**6.1196**	0.6226
	DSPO	0.2256	0.2684	0.3615	5.9598	0.6831
HPS V2	SDXL	0.2271	0.2730	0.3775	6.1125	0.6749
	DMPO	**0.2320**	**0.2804**	**0.3914**	6.1883	**1.0169**
	SFT	0.2277	0.2762	0.3784	6.0638	0.6998
	Diff-DPO	0.2314	0.2777	0.3890	6.1546	0.9610
	MAPO	0.2279	0.2765	0.3801	**6.2134**	0.7839
	DSPO	0.2285	0.2754	0.3795	6.0545	0.7385
Parti Prompt	SDXL	0.2249	0.2714	0.3551	5.7648	0.6010
	DMPO	**0.2290**	**0.2780**	**0.3769**	5.8598	**1.0399**
	SFT	0.2257	0.2701	0.3531	5.7239	0.6953
	Diff-DPO	0.2286	0.2763	0.3688	5.7900	0.9515
	MAPO	0.2250	0.2732	0.3561	**5.9089**	0.7031
	DSPO	0.2256	0.2714	0.3596	5.7598	0.7600

Table 6: (a) Win rate (%) comparisons on Pick-a-Pic V2, HPS V2 and Parti-Prompt datasets for all baselines (fine-tuned on SDXL) versus SDXL. (b) Win rate (%) comparison between DMPO versus other baselines, win rates that surpass 50 % are in green, below 50 % are in red. For simplicity, "Diff" represents "Diffusion".

Dataset	Method1	Method2	Pick Score	HPS	CLIP	Aesthetics	Image Reward
Pick-a-Pic V2	DMPO	SDXL	**75.20**	**80.20**	61.60	48.00	**69.80**
	SFT	SDXL	54.20	67.50	53.50	43.20	61.80
	Diff-DPO	SDXL	74.50	79.00	**61.80**	51.00	69.00
	MAPO	SDXL	55.60	68.00	51.20	66.60	64.20
	DSPO	SDXL	56.30	69.20	52.60	44.80	60.50
	DMPO	SFT	65.80	72.00	60.20	53.20	66.20
	DMPO	Diff-DPO	51.50	67.00	50.80	45.40	61.20
	DMPO	MAPO	66.20	74.00	61.60	35.80	64.60
	DMPO	DSPO	63.40	73.40	63.40	50.30	63.20
HPS V2	DMPO	SDXL	67.75	**87.25**	**63.75**	47.75	**71.25**
	SFT	SDXL	54.75	70.00	53.00	61.25	60.00
	Diff-DPO	SDXL	**70.50**	81.25	61.75	59.25	68.75
	MAPO	SDXL	54.00	79.50	52.00	**70.00**	62.75
	DSPO	SDXL	53.25	71.25	55.25	57.35	64.75
	DMPO	SFT	60.75	65.50	58.75	50.00	63.50
	DMPO	Diff-DPO	48.00	72.00	53.00	43.00	52.50
	DMPO	MAPO	62.25	75.25	59.50	38.50	66.75
	DMPO	DSPO	63.75	73.00	59.75	48.25	62.25
Parti Prompt	DMPO	SDXL	**68.72**	**79.96**	**61.46**	51.69	**71.75**
	SFT	SDXL	50.44	66.58	50.75	52.14	56.69
	Diff-DPO	SDXL	66.89	76.89	60.64	54.96	69.30
	MAPO	SDXL	49.94	67.34	50.80	**69.24**	59.44
	DSPO	SDXL	52.44	65.17	53.31	53.22	58.71
	DMPO	SFT	62.22	66.31	59.23	50.69	53.46
	DMPO	Diff-DPO	51.07	61.89	52.77	45.62	52.14
	DMPO	MAPO	64.22	70.22	62.01	44.21	66.24
	DMPO	DSPO	60.12	63.08	58.25	51.49	57.52