Self-NPO: Data-Free Diffusion Model Enhancement via Truncated Diffusion Fine-Tuning

Fu-Yun Wang[1], **Keqiang Sun**[1], **Yao Teng**[2], **Xihui Liu**[2], **Jiale Yuan**[4], **Jiaming Song**[3], **Hongsheng Li**[1]
[1]The Chinese University of Hong Kong
[2]Hong Kong University
[3]Luma AI
[4]Shanghai Jiaotong University
fywang0126@gmail.com

Abstract

Diffusion models have demonstrated remarkable success in various visual generation tasks, including image, video, and 3D content generation. Preference optimization (PO) is a prominent and growing area of research that aims to align these models with human preferences. While existing PO methods primarily concentrate on producing favorable outputs, they often overlook the significance of classifier-free guidance (CFG) in mitigating undesirable results. Diffusion-NPO addresses this gap by introducing negative preference optimization (NPO), training models to generate outputs opposite to human preferences and thereby steering them away from unfavorable outcomes through CFG. However, prior NPO approaches rely on costly and fragile procedures for obtaining explicit preference annotations (*e.g.*, manual pairwise labeling or reward model training), limiting their practicality in domains where such data are scarce or difficult to acquire. In this work, we propose Self-NPO, specifically **truncated diffusion fine-tuning**, a data-free approach of negative preference optimization by directly learning from the model itself, eliminating the need for manual data labeling or reward model training. This data-free approach is highly efficient (less than 1% training cost of Diffusion-NPO) and achieves comparable performance to Diffusion-NPO in a data-free manner. We demonstrate that Self-NPO integrates seamlessly into widely used diffusion models, including SD1.5, SDXL, and CogVideoX, as well as models already optimized for human preferences, consistently enhancing both their generation quality and alignment with human preferences.

Code — https://github.com/G-U-N/Diffusion-NPO

Introduction

Over the past few years, diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Song et al. 2020) have made significant strides in various visual generation tasks, including image (Karras et al. 2022; Rombach et al. 2022; Wang et al. 2024c; Podell et al. 2023; Dhariwal and Nichol 2021; Meng et al. 2021; Teng et al. 2024; Ma et al. 2024; Li et al. 2024; Sun et al. 2024b), video (Blattmann et al. 2023; Mao et al. 2024; Shi et al. 2024; Singer et al. 2022; Chen et al. 2024; Bian et al. 2025; Pokle, Geng, and Kolter 2022; Wang et al. 2024b, 2025b, 2024a), and 3D generation (Gao et al. 2024; Poole et al. 2022; Li et al. 2025; Yan, Chen, and Wang 2025; Sun et al. 2024a; Lai et al. 2025). However, diffusion models trained on massive unfiltered data (*e.g.*, image-text pairs) often generate results that do not align well with human preferences. The growing interest in aligning these models for human preference-aligned generation has led to the development of preference optimization (PO) methods. In general, current PO methods can be categorized into four types: (i) Differentiable Reward (DR) evaluates images from iterative denoising using a pre-trained reward model, refining diffusion models through gradient-based backpropagation to align with the reward model (Xu et al. 2024; Prabhudesai et al. 2024; Zhang et al. 2024b; Wu et al. 2023c,a, 2024; Clark et al. 2023). (ii) Reinforcement Learning (RL) models the denoising process as a Markov decision process, using techniques like PPO to optimize preferences by dynamically generating and evaluating images to maximize rewards (Puterman 2014; Sutton 2018; Schulman et al. 2017). (iii) Direct Preference Optimization (DPO) streamlines training by fine-tuning with paired preference datasets, eliminating separate reward models or online evaluation, often using high-quality data as positive examples

Figure 1: Visual comparison on Stable Diffusion XL. Prompts: "A cute blue cat", "An attractive young woman rolling her eyes", and "Close-up of a head with smoke from ears, watching a smartphone, dynamic angle".

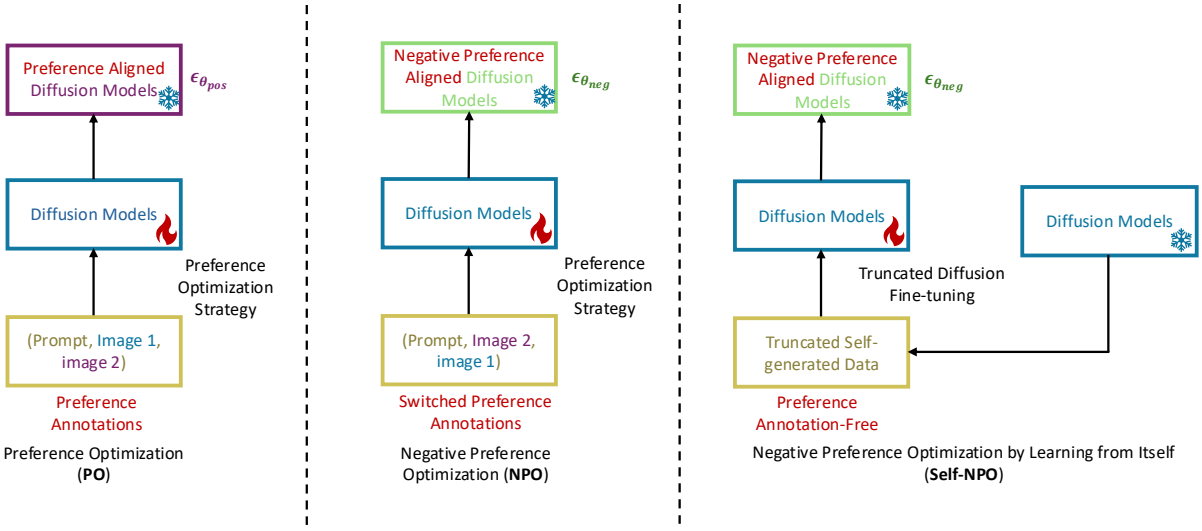


Figure 2: Motivation of Self-NPO. Preference Optimization (PO) optimizes the preference aligned model by training on preference annotations. Negative Preference Optimization (NPO) optimizes the negative preference aligned model by training on switched preference annotations. Self-NPO optimizes the negative preference aligned model by truncated diffusion fine-tuning (TDFT) on truncated self-generated data.

and self-generated data as negatives (Rafailov et al. 2024; Wallace et al. 2024; Yuan et al. 2024; Deng et al. 2024; Gu et al. 2024; Zhang et al. 2024a). (iv) Negative Preference Optimization (Diffusion-NPO) trains models to avoid poor outputs by opposing human preferences, improving performance through classifier-free guidance (CFG) to steer away from undesired results (Wang et al. 2025a; Ho and Salimans 2022; Karras et al. 2024; Shen et al. 2024; Ahn et al. 2024).

However, we argue that a major limitation of existing methods is their reliance on data sources, *ie*, ***explicit preference annotations***, such as the expensive manual collection of preference data pairs (Kirstain et al. 2023; Wu et al. 2023b) and the challenging and fragile process of reward model training (Xu et al. 2024; Wu et al. 2023b; Schuhmann 2022). This heavy dependency on preference annotations makes these approaches less practical, particularly in domains where obtaining or labeling preference data is expensive or infeasible.

In this work, we introduce **Self-NPO**, specifically, truncated diffusion fine-tuning (TDFT), a novel approach for negative preference optimization (NPO) that can be done in a data-free manner. Specifically, our training objective is to apply negative preference optimization to a pre-trained model, making its predictions more aligned with the opposite direction of human preferences. This facilitates unconditional/negative-conditional outputs in classifier-free guidance, thus reducing the probability of generating results that do not align with human preferences.

To implement Self-NPO, we have three key observations:

1. **Controlled weakening of generative ability.** People typically do not prefer corrupted generated images. For example, blurry details or structural errors make an image less appealing. Therefore, weakening the generative ability of a pre-trained diffusion model is, to some extent, equivalent to negative preference optimization. However, this weakening of generative ability cannot be arbitrary. In classifier-free guidance, we typically have the equation

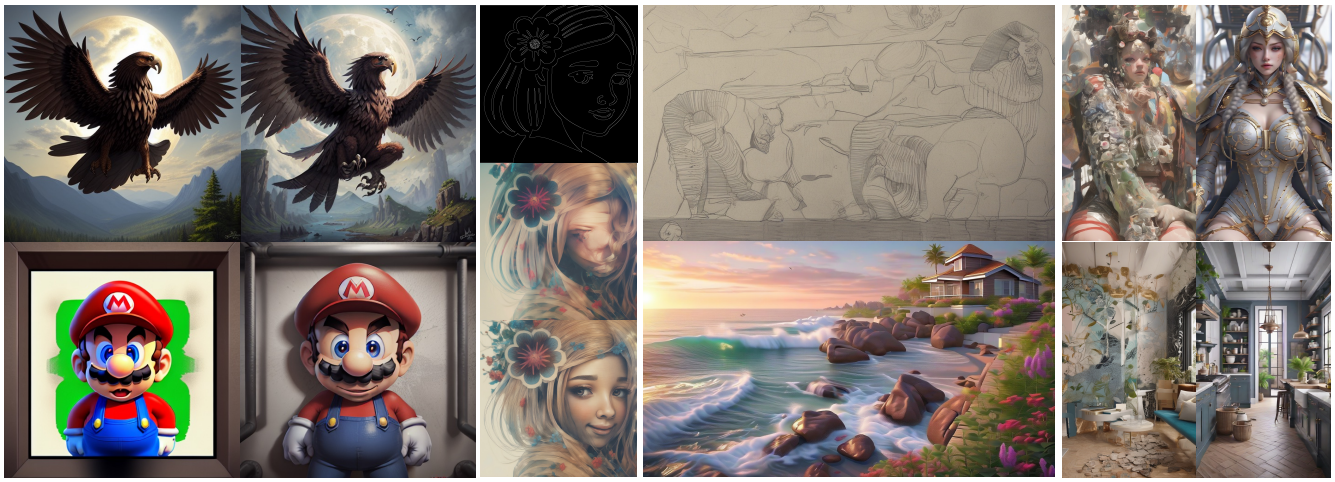
$$\boldsymbol{\epsilon}^{\omega} = (\omega+1)\boldsymbol{\epsilon}_{\boldsymbol{\theta}_{pos}}(\mathbf{x}_t, t, \boldsymbol{c}) - \omega\boldsymbol{\epsilon}_{\boldsymbol{\theta}_{neg}}(\mathbf{x}_t, t, \boldsymbol{c}') \quad (1)$$

   To avoid altering the variance of the epsilon predictions, $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_{neg}}(\mathbf{x}_t, t, \boldsymbol{c}')$ should maintain a certain level of correlation with $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_{pos}}(\mathbf{x}_t, t, \boldsymbol{c})$. For example, for two completely independent Gaussian noises, the above operation would result in a variance of $2\omega^2 + 2\omega + 1$.
2. **Weakening via self-generated data.** This weakening, which satisfies the above conditions, can be achieved by learning from the model's own generated data, which can be regarded as a target distribution regularized weakening.
3. **Efficient tuning without full diffusion.** We do not need to fully execute iterative diffusion generation process to implement self-learning. We propose a novel tuning strategy, termed as **truncated diffusion fine-tuning**, which allows us to fine tune diffusion models on the partially excuted iterative diffusion generation results. This avoids the huge generation costs of preparing data for self-learning.

We demonstrate that Self-NPO can be seamlessly integrated with existing models, such as SD1.5 (Rombach et al. 2022), SDXL (Podell et al. 2023), and CogVideoX (Yang et al. 2024), as well as models previously fine-tuned through preference optimization, consistently improving their alignment with human preferences.

Preliminary

Preliminary of CFG. CFG has become a necessary and important technique for improving generation quality and text



a) Plug-and-paly b) Layout control c) Unconditional

Figure 3: Applications of Self-NPO. Prompts: "A giant eagle monster art", "Mario in prison", and "Girl, pretty, flower, hair, smile". Results below/right use Self-NPO; above/left without it.

alignment of diffusion models. For convenience, we focus our discussion on the general formal of diffusion models, *i.e.*, $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ (Kingma et al. 2021). Suppose we learn a score estimator from a epsilon prediction neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{c}, t)$, and we have $\nabla_{\mathbf{x}_t} \log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}; t) = -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sigma_t}$. The sample prediction at timestep t of the score estimator is formulated as

$$\hat{\mathbf{x}}_0 = \frac{1}{\alpha_t}(\mathbf{x}_t + \sigma^2 \nabla_{\mathbf{x}_t} \log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}; t)) \,. \tag{2}$$

Applying the CFG is equivalent to add an additional score term, that is, we replace $\nabla_{\mathbf{x}_t} \log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}; t)$ in Eq. 2 with the following term,

$$\nabla_{\mathbf{x}_t} \log \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}; t) + \nabla_{\mathbf{x}_t} \log \left[\frac{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}; t)}{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}'; t)} \right]^{\omega} , \tag{3}$$

where ω is to control the strength of CFG, $\boldsymbol{c}$ and $\boldsymbol{c}'$ are conditional and unconditional/negative-conditional inputs, respectively. It is apparent that the generation will be pushed to high probability region of $\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}; t)$ and relatively low probability region of $\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_t|\boldsymbol{c}'; t)$. Write the above equation into the epsilon format, and then we have

$$\boldsymbol{\epsilon}^{\omega}_{\boldsymbol{\theta}} = (\omega + 1)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{c}, t) - \omega \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{c}', t) \,. \tag{4}$$

Diffusion-NPO (Wang et al. 2025a)**.** Diffusion models use classifier-free guidance (CFG) to improve generation quality by combining conditional and negative-conditional outputs to favor preferred results. Traditional preference optimization aligns outputs with human preferences but overlooks avoiding undesirable outputs. Negative Preference Optimization (NPO), introduced in Diffusion-NPO, trains an additional model to counteract human preferences, reducing undesired outputs via CFG.

Training with Diffusion-NPO. NPO adapts existing preference optimization methods without new datasets or strategies. For reward-based methods, the negative preference reward is:

$$R_{\text{NPO}}(\mathbf{x}, \boldsymbol{c}) = 1 - R(\mathbf{x}, \boldsymbol{c}), \tag{5}$$

where $R(\mathbf{x}, \boldsymbol{c}) \in [0, 1]$ is a reward model (*e.g.*, HPSv2). For DPO-based methods, preference pairs $r = (\mathbf{x}_0, \mathbf{x}_1, \boldsymbol{c})$, with $\mathbf{x}_1$ preferred, are reversed:

$$r_{\text{NPO}} = (\mathbf{x}_1, \mathbf{x}_0, \boldsymbol{c}). \tag{6}$$

Inference with Diffusion-NPO. Let $\boldsymbol{\theta}$ denote the base model weights, with $\boldsymbol{\eta}$ and $\boldsymbol{\delta}$ representing the weight offsets after positive and negative preference optimization, respectively. The preference optimized model and negative preference optimized models can be represented as: $\boldsymbol{\theta}_{pos} = \boldsymbol{\theta} + \boldsymbol{\eta}$, $\boldsymbol{\theta}_{neg} = \boldsymbol{\theta} + \boldsymbol{\delta}$. Classifier-free guidance is then applied as: $\boldsymbol{\epsilon}^{\omega}_{\boldsymbol{\theta}} = (\omega + 1)\boldsymbol{\epsilon}_{\boldsymbol{\theta}_{pos}} - \omega \boldsymbol{\epsilon}_{\boldsymbol{\theta}_{neg}}$. However, this often results in significant output discrepancies. To mitigate this, Diffusion-NPO modifies the negative weights to include a combination of the positive and negative offsets:

$$\boldsymbol{\theta}_{neg} = \boldsymbol{\theta} + \alpha\boldsymbol{\eta} + \beta\boldsymbol{\delta}, \quad \alpha, \beta \in [0, 1], \tag{7}$$

which ensures more stable and correlated outputs.

Methodology

NPO via self-generated data

Reinforcement Learning with Human Feedback (RLHF) (Rafailov et al. 2024; Wallace et al. 2024; Griffith et al. 2013; Liu et al. 2025) aims to optimize a conditional distribution $\mathbb{P}_\theta(\mathbf{x}_0 \mid \boldsymbol{c})$, where $\boldsymbol{c} \sim \mathcal{D}_c$, in order to maximize the expectation value of the associated reward model $R(\mathbf{x}_0, \boldsymbol{c})$. Simultaneously, the optimization imposes a regularization term that penalizes deviations from a reference distribution $\mathbb{P}_{\text{ref}}(\mathbf{x}_0 \mid \boldsymbol{c})$. Formally, the objective function is formulated as follows:

$$\begin{aligned} &\max_{\mathbb{P}_\theta} \mathbb{E}_{\boldsymbol{c}\sim\mathcal{D}_c, \mathbf{x}_0\sim\mathbb{P}_\theta(\mathbf{x}_0|\boldsymbol{c})} \left[R(\mathbf{x}_0, \boldsymbol{c})\right] \\ &-\beta D_{\text{KL}} \left[\mathbb{P}_\theta(\mathbf{x}_0 \mid \boldsymbol{c}) \,\|\, \mathbb{P}_{\text{ref}}(\mathbf{x}_0 \mid \boldsymbol{c})\right] , \end{aligned} \tag{8}$$

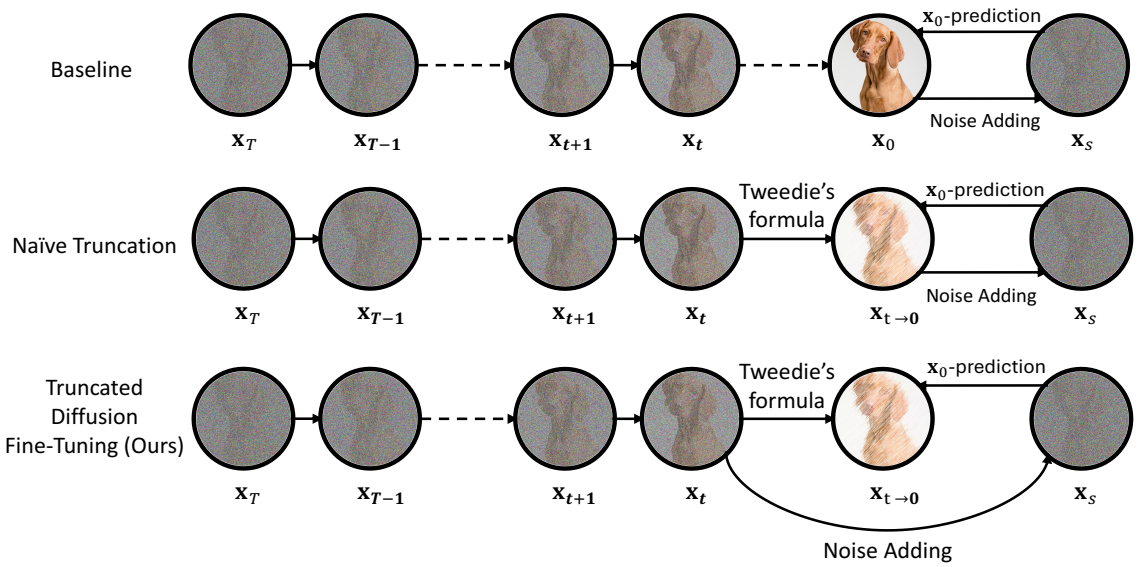


Figure 4: Truncated diffusion fine-tuning (TDFT). Baseline method requires full simulation of generation process, introducing huge amount of generation costs. Naive truncation reduces the simulation costs but introduces distribution mismatch. TDFT reduces the simulation costs and maintains the target distribution.

where β is a hyper-parameter that governs the balance between the reward maximization and the regularization effect imposed by the KL-divergence. The regularization is crucial because it prevents the model from straying too far from the distribution where the reward model is reliable, while also preserving generation diversity and preventing mode collapse to a single high-reward sample. Replacing $R(\mathbf{x}_0, \boldsymbol{c})$ with $R_{\text{NPO}}(\mathbf{x}_0, \boldsymbol{c})$ yields the learning objective for Negative Preference Optimization (NPO):

$$\begin{aligned}\max_{\mathbb{P}_\theta} \mathbb{E}_{\boldsymbol{c}\sim\mathcal{D}_c, \mathbf{x}_0\sim\mathbb{P}_\theta(\mathbf{x}_0|\boldsymbol{c})}\left[R_{\text{NPO}}(\mathbf{x}_0, \boldsymbol{c})\right] \\ -\beta D_{\text{KL}}\left[\mathbb{P}_\theta(\mathbf{x}_0 \mid \boldsymbol{c}) \parallel \mathbb{P}_{\text{ref}}(\mathbf{x}_0 \mid \boldsymbol{c})\right] .\end{aligned} \tag{9}$$

where the first term encourages the samples generated from $\mathbb{P}_{\boldsymbol{\theta}}$ to have low reward scores, while the second term constrains the learned distribution to remain close to $\mathbb{P}_{\text{ref}}$.

Observing that a diffusion model can naturally produce undesirable outputs (*e.g.*, disordered compositions, incoherent structures, blurry details), such low-quality samples inherently yield low reward scores. Consequently, exploiting self-generated data provides a straightforward path toward distribution-regularized negative preference optimization. Fig. 2 summarizes the core idea and situates it among related methods. This approach is justified on two grounds: 1) **Distribution preservation.** Learning from self-generated data preserves the original distributional properties of the model. 2) **Reward reduction.** Self-generated samples exhibit low reward scores for several reasons: a) *Hallucination.* Generation of unrealistic objects, artifacts, or contextually inconsistent outputs, which result in low reward scores due to poor quality or implausibility of the generated content. b) *Mode collapse.* The model may fail to generate a diverse set of outputs, leading to a lack of coverage across the entire data distribution and, consequently, lower reward scores due to insufficient variety. c) *Optimization.* Imperfect optimization within the diffusion model itself can produce suboptimal results, such as blurry or noisy details, which leads to lower reward scores due to the misalignment between generated outputs and the desired target distribution. *Indeed, we find it is helpful to intentionally corrupt the ODE sampling process to some extent, thus increasing the probability of reward reduction. We have more discussion and empirical evidence in the supplementary material.*

However, despite its conceptual simplicity, fine-tuning a diffusion model on self-generated data can be computationally expensive, as it requires generating large amounts of training samples. To address this, we introduce a *truncated diffusion fine-tuning* strategy. This method updates the model using partially generated diffusion samples, thereby avoiding the high costs of performing full diffusion simulations during training.

Truncated diffusion fine-tuning

Baseline: Fine-tuning on fully generated data. We denote the reference diffusion model for data generation as $\mathbb{P}_{\text{ref}}(\mathbf{x}_0 \mid \boldsymbol{c})$, which models the adjacent timesteps transition conditional distribution $\mathbb{P}_{\text{ref}}(\mathbf{x}_{t-1}^{ref} \mid \mathbf{x}_t^{ref}, \boldsymbol{c})$ for $t = 1, 2, \ldots, T$. We can achieve data sampling by iteratively call the transition distribution,

$$\mathbf{x}_T^{ref} \to \mathbf{x}_{T-1}^{ref} \to \ldots \mathbf{x}_t^{ref} \to \ldots \mathbf{x}_1^{ref} \to \mathbf{x}_0^{ref} , \tag{10}$$

where $\mathbf{x}_T^{ref} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. After obtaining the $\mathbf{x}_t^{ref}$, the learning of $\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_0 \mid \boldsymbol{c})$ is equivalent to the learning a $\mathbf{x}_0$-prediction neural network $\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{c})$ with standard diffu-

sion loss following previous work:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t,\boldsymbol{\epsilon}} \|\mathbf{x}_0^{ref} - \boldsymbol{f}_{\boldsymbol{\theta}}((\mathbf{x}_0^{ref})_t, t, \boldsymbol{c})\|_2^2, \tag{11}$$

where t is sampled from $\{1, 2, \ldots, T\}$, and $(\mathbf{x}_0^{ref})_t = \alpha_t \mathbf{x}_0^{ref} + \sigma_t \boldsymbol{\epsilon}$.

Our method: Truncated Diffusion Fine-tuning. According to the Tweedie's formula, for the semi-implicit distribution $\mathbb{P}_{\text{ref}}(\mathbf{x}_t^{ref} \mid \boldsymbol{c})$ induced by the reference diffusion model, the expected value of $\mathbf{x}_0$ given $\mathbf{x}_t^{ref}$, based on the conditional $\mathbb{P}(\mathbf{x}_0|\mathbf{x}_t^{ref}) = \frac{\mathbb{P}(\mathbf{x}_t^{ref}|\mathbf{x}_0)\mathbb{P}_{\text{ref}}(\mathbf{x}_0|\boldsymbol{c})}{\mathbb{P}_{\text{ref}}(\mathbf{x}_t^{ref}|\boldsymbol{c})}$ according to Bayes' rule, is related to the score of $\mathbb{P}_{\text{ref}}(\mathbf{x}_t^{ref} \mid \boldsymbol{c})$ (Efron 2011; Zhou et al. 2024). Specifically, the expectation is:

$$\mathbb{E}_{\mathbf{x}_0 \sim \mathbb{P}(\mathbf{x}_0|\mathbf{x}_t^{ref})}[\mathbf{x}_0|\mathbf{x}_t^{ref}] = \mathbf{x}_t + \sigma_t^2 \nabla_{\mathbf{x}_t} \log \mathbb{P}_{\text{ref}}(\mathbf{x}_t^{ref} \mid \boldsymbol{c}). \tag{12}$$

In this equation, the expectation $\mathbb{E}_{\mathbf{x}_0 \sim \mathbb{P}(\mathbf{x}_0|\mathbf{x}_t^{ref})}[\mathbf{x}_0|\mathbf{x}_t^{ref}]$ is computed as an integral over $\mathbf{x}_0$, and it is expressed as the current state $\mathbf{x}_t^{ref}$ plus a term involving the score function $\nabla_{\mathbf{x}_t^{ref}} \log \mathbb{P}_{\text{ref}}(\mathbf{x}_t^{ref} \mid \boldsymbol{c})$, where σ_t^2 represents the noise variance. For notation simplicity, we denote $\mathbf{x}_{t\to 0}^{ref}$ as $\mathbb{E}_{\mathbf{x}_0 \sim \mathbb{P}(\mathbf{x}_0|\mathbf{x}_t^{ref})}[\mathbf{x}_0|\mathbf{x}_t^{ref}]$ without introducing ambiguity.

Therefore, essentially, we can obtain the expectation of $\mathbf{x}_0$ of $\mathbf{x}_t^{ref}$ (*i.e.*, $\mathbf{x}_{t\to 0}^{ref}$) at arbitrary intermediate timesteps. Our idea is to replace the $\mathbf{x}_0^{ref}$ as adopted in the baseline with $\mathbf{x}_{t\to 0}^{ref}$, which eliminates the needs for fully simulation of generation process. Specifically, we can truncate the whole denoising process into

$$\mathbf{x}_T^{ref} \to \mathbf{x}_{T-1}^{ref} \to \ldots \mathbf{x}_t^{ref} \xrightarrow{\text{Tweedie's formula}} \mathbf{x}_{t\to 0}^{ref}, \tag{13}$$

Further more, by incorporating real data for consideration, we can further truncate the left part by replacing the denoising trajectory from $\mathbf{x}_T^{ref}$ to $\mathbf{x}_{t'}^{ref}$ with the noise injection from real data, *i.e.*,

$$\mathbf{x}_{t'}^{ref} \to \mathbf{x}_{t'-1}^{ref} \to \ldots \mathbf{x}_t^{ref} \xrightarrow{\text{Tweedie's formula}} \mathbf{x}_{t\to 0}^{ref}, \tag{14}$$

where $\mathbf{x}_{t'}^{ref}$ is obtained by adding noise from real data following the forward process $\mathbb{P}(\mathbf{x}_t \mid \mathbf{x}_0)$. When $t' = T$ and $t = 1$, we have $\mathbf{x}_{t'}^{ref} = \mathbf{x}_T^{ref}$ and $\mathbf{x}_{t\to 0}^{ref} = \mathbf{x}_{1\to 0}^{ref} = \mathbf{x}_0^{ref}$. In this case, truncated diffusion fine-tuning essentially reduces to the baseline method. Thus, truncated diffusion fine-tuning can be viewed as a generalization of the baseline method, as it eliminates the need for fully solving the denoising process numerically, leading to significant improvements in training efficiency.

Solving the distribution discrepancy. It is worthy noting that $\mathbf{x}_{t\to 0}^{ref}$ indeed has a different distribution with $\mathbf{x}_0^{ref} \sim \mathbb{P}_{\text{ref}}(\mathbf{x}_0 \mid \boldsymbol{c})$. This is because $\mathbb{P}(\mathbf{x}_{t\to 0}^{ref} \mid \mathbf{x}_t^{ref}, \boldsymbol{c})$ can be a very smooth distribution, and $\mathbf{x}_{t\to 0}^{ref}$ is the weighted average value of many potential $\mathbf{x}_0^{ref} \sim \mathbb{P}_{\text{ref}}(\mathbf{x}_0^{ref} \mid \boldsymbol{c})$. Therefore, directly adopting the vanilla diffusion fine-tuning on $\mathbf{x}_{t\to 0}^{ref}$ can not achieve equivalent effect as the baseline method. We solve this distribution discrepancy by adopting a different noise adding strategy. Specifically, instead of directly adding noise to $\mathbf{x}_{t\to 0}^{ref}$ to obtain $\mathbf{x}_s$, *i.e.*,

$$\mathbf{x}_s = \alpha_s \mathbf{x}_{t\to 0}^{ref} + \sigma_s \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{15}$$

we choose to obtain $\mathbf{x}_s$ by adding noise to $\mathbf{x}_t^{ref}$, *i.e.*,

$$\mathbf{x}_s = \frac{\alpha_s}{\alpha_t} \mathbf{x}_t^{ref} + \sqrt{\sigma_s^2 - \sigma_t^2 \frac{\alpha_s^2}{\alpha_t^2}} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{16}$$

Then we optimize the diffusion model $\boldsymbol{f}_{\boldsymbol{\theta}}$ (corresponding to $\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}_0 \mid \boldsymbol{c})$) following the standard diffusion training with $\mathbf{x}_{t\to 0}^{ref}$ as $\mathbf{x}_0$-prediction target, *i.e.*,

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_s \sim \mathbb{P}(\mathbf{x}_s|\mathbf{x}_t^{ref}),\, \mathbf{x}_t^{ref} \sim \mathbb{P}_{\text{ref}}(\mathbf{x}_t^{ref}|\boldsymbol{c})} \|\mathbf{x}_{t\to 0}^{ref} - \boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, \boldsymbol{c})\|_2^2, \tag{17}$$

Theoretical grounding

We theoretically show the reasonability of proposed TDFT from three aspects:

1. ***Distribution of*** $\mathbf{x}_s$. We show that the distribution of $\mathbf{x}_s$ is equivalent to the noise-perturbed distribution of $\mathbb{P}_{\text{ref}}(\mathbf{x}_0^{\text{ref}} \mid \boldsymbol{c})$ (See Thm. 1).
2. ***Gradient equivalent optimization target***. We show that our optimization objective, Eq. (17), has a gradient-equivalent learning objective, which is the expectation value of all $\mathbf{x}_{t\to 0}^{\text{ref}}$ (See Thm. 2).
3. ***Equivalent optimization objective of diffusion models***. We show that the gradient-equivalent learning objective, as proved in 2), is equivalent to the learning objective of a standard diffusion model (See Thm. 3).

Proof Sketch: Due to the limited space, we encourage readers to review the detailed proofs in Thm. 1, Thm. 2, and Thm. 3. These proofs rigorously establish the equivalence of distributions, optimization objectives, and their connections to diffusion models, forming the foundation for understanding the validity and effectiveness of our proposed approach.

Experiments

Validation setup

To better evaluate the performance of our method, we tested three baseline models: a) Stable Diffusion v1-5, a text-to-image generation model with approximately 860 million parameters, widely used for creating high-quality images from textual prompts in tasks like digital art and concept visualization; b) Stable Diffusion XL (SDXL), an advanced version with around 2.6 billion parameters, enabling higher-resolution image generation (*e.g.*, 1024x1024) and improved compositional coherence; c) CogVideoX-5B, a video generation model with about 5 billion parameters, designed for tasks like video synthesis and frame interpolation, leveraging its capacity for temporal modeling. Therefore, our testing covers diffusion models of varying sizes and encompasses mainstream text-to-image and text-to-video tasks, ensuring a comprehensive and well-designed baseline.

Method	P.S.	HPS.	I.R.	AES.
SDXL	22.06	28.04	0.6246	6.114
SDXL + NPO	22.25	**28.98**	**0.6831**	6.136
SDXL + Self-NPO	**22.26**	28.24	0.6697	**6.226**
Diff.-DPO	22.57	29.76	0.8624	6.099
Diff.-DPO + NPO	**22.69**	**30.48**	**0.9210**	6.112
Diff.-DPO + Self-NPO	22.67	29.83	0.8784	**6.179**
Juggernaut	22.66	30.25	0.9778	6.021
Juggernaut + Self-NPO	**22.77**	**30.56**	**0.9921**	**6.031**

Table 1: Quantitative performance comparison with stable diffusion XL based models. All metrics are tested with official weights.

Comparison

Quantitative comparison. For text-to-image generation, we evaluate our method quantitatively by adhering to prior research and utilizing the 'test_unique' split from Pick-a-pic as the benchmark for testing (Kirstain et al. 2023). We utilize PickScore (Kirstain et al. 2023) (**P.S.**), HPSv2.1 (Wu et al. 2023a) (**HPS.**), ImageReward (Xu et al. 2024) (**I.R.**), and Laion-Aesthetic (Schuhmann 2022) (**AES.**) as our evaluation metrics. The outcomes of this quantitative assessment are detailed in Tab. 2 and Tab. 1. These tables reveal that integrating Self-NPO with the base model and its preference-optimized variants consistently improves the aesthetic quality of the generated images. Beyond presenting average scores, as shown in Fig. 6, we also compute the percentage of samples generated from the same prompt that attain a higher preference score. The results produced with Self-NPO markedly surpass those generated without it. For text-to-video generation, we provide detailed experimental comparisons and demonstrations in the supplementary material, where our method achieves clear performance improvements on the majority of metrics.

Qualitative comparison. We present the generated results in different scenarios in Fig. 1, and Fig. 3, demonstrating that self-NPO consistently improves both the generation visual quality and fidelity.

User study. We evaluate generation quality across four areas: color and lighting, high-frequency details, low-frequency composition, and text-image alignment. In Color and Lighting, users assess whether the images feature natural, visually pleasing color schemes and lighting. For high-frequency details, users look at the sharpness and texture details, especially edges and fine elements. In low-frequency composition, they focus on the overall structure and balance of the image. Finally, for text-image alignment, users judge how well the generated image matches the input text. The user study uses prompts from Pickapic validation_unique dataset, with different models generating images based on the same random seed. Users choose between "No Preference" (Draws), "Self-NPO is better" (Wins), or "Self-NPO is worse" (Losses) for each pair of images. We collect responses from 9 volunteers, each evaluating 100 pairs of images generated by Dreamshaper, totaling 900 votes. Results, shown in Fig. 5, reveal that NPO significantly improves high-frequency details, enhances color and lighting preferences, and helps with composition. It also shows an improvement in text-image alignment.

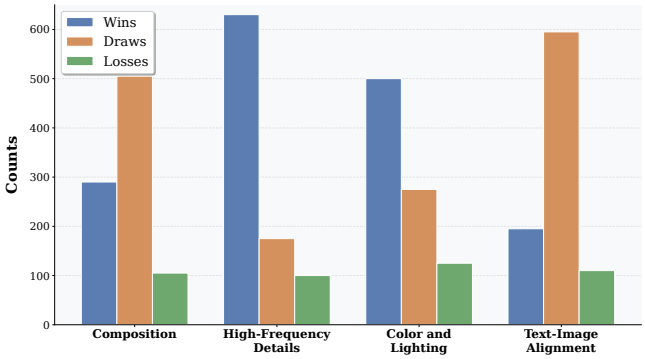


Figure 5: User study analysis.

Applications

Plug-and-play. Our method is not only applicable to the original stable diffusion-based models and their fine-tuned versions optimized through preference optimization, but also extends directly to high-quality stylized models fine-tuned on private datasets. As shown in Fig. 3, when leveraging the Dreamshaper model alongside the self-NPO-optimized model as an unconditional predictor, we observe substantial improvements in both image quality and aesthetic appeal.

Controllable generation. Moreover, our approach seamlessly integrates with controllable generation techniques. By incorporating controllable plugins like T2I-Adapter (Mou et al. 2024), we significantly boost the model's generation quality under layout-controlled conditions.

Unconditional generation. It is well-established that unconditional generation typically falls short of conditional generation in terms of quality. This disparity is even more pronounced in text-to-image diffusion models, where, without conditional input, the model often struggles to produce anything other than incoherent, low-quality images. However, with the self-NPO-enhanced model, we can generate relatively high-quality images even in the absence of conditional guidance.

Training efficiency comparison

We report the training costs of Self-NPO. With the default truncated simulation steps set to 5, we trained Stable Diffusion v1-5 for 1,000 iterations with a batch size of 80. The overall training time was approximately 0.5 hours on 4 A800 GPUs (2 A100 GPU hours). For the baseline (*i.e.*, full simulation, $K = 25$) described in our paper, training for 1,000 iterations required approximately 2.6 hours on 4 A800 GPUs (10.4 A800 GPU hours), which is around 5 times longer than our default setting.

According to the official GitHub page of Diffusion-DPO, the official weights of Diffusion-DPO were trained with a batch size of 2,048 for 2,000 iterations, taking 24 hours on 16 A100 GPUs (384 A100 GPU hours, approximately 192 times longer than our default setting). Since DPO-based

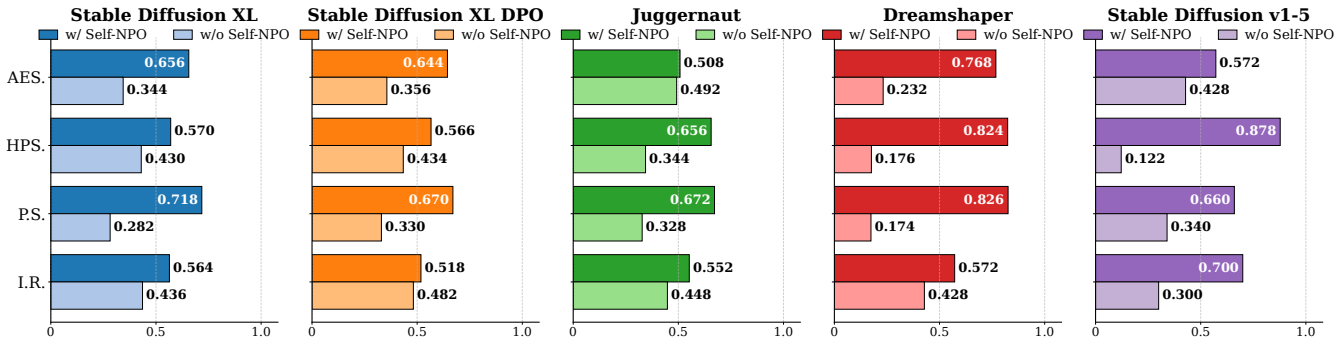


Figure 6: Winning ratio comparison. Self-NPO works seamlessly with SD1.5, SDXL, and those optimized for preferences, consistently improving both their generation quality and alignment with human preferences.

Method	P.S.	HPS.	I.R.	AES.
*DDPO	21.06	24.91	0.0817	5.591
*D3PO	20.76	23.97	-0.1235	5.527
*Diff.-SPO	21.41	26.85	0.1738	5.946
SD-1.5	20.75	26.84	0.1064	5.539
SD-1.5 + NPO	21.26	27.36	0.2028	5.667
SD-1.5 + Self-NPO	21.00	27.04	0.2816	5.609
Diff.-DPO	21.12	25.93	0.2651	5.648
Diff.-DPO + NPO (reg= 500)	**21.58**	27.60	0.3101	5.762
Diff.-DPO + NPO (reg= 1000)	21.43	27.36	0.3472	**5.773**
Diff.-DPO + Self-NPO	21.34	**27.78**	**0.4085**	5.710
SePPO	21.51	28.45	0.5981	5.892
SePPO + Self-NPO	**21.73**	**30.28**	**0.6744**	**6.014**
DreamShaper	21.85	28.85	0.6819	6.143
DreamShaper + NPO ($\alpha = 1.0$)	22.30	30.13	0.7258	6.234
DreamShaper + NPO ($\alpha = 0.6$)	**22.39**	29.92	0.6034	6.492
DreamShaper + Self-NPO ($\alpha = 1.0$)	22.20	**30.40**	**0.8038**	6.196
DreamShaper + Self-NPO ($\alpha = 0.9$)	22.36	30.39	0.7738	6.345
DreamShaper + Self-NPO ($\alpha = 0.8$)	22.34	29.96	0.6548	**6.562**

Table 2: Quantitative performance comparison with stable diffusion v1-5 based models. * means the metrics are copied from Diffusion-NPO. Other metrics are tested with official weights.

Diffusion-NPO can be trained by simply reversing the preference pair, it incurs the same training cost. However, when Diffusion-NPO is trained with LoRA, our tests show that the training time is reduced to 153.6 A800 GPU hours, approximately 76.8 times longer than our default setting.

Conclusions

In this paper, we explore existing diffusion model-based preference optimization techniques, including vanilla preference optimization and negative preference optimization. We observe that these methods require large amounts of explicit preference annotations or fragile reward model training, which involve costly manual data labeling and reward model training. To address this, we introduce Self-NPO (Negative Preference Optimization), a method that learns directly from data generated by the diffusion model itself, thus eliminating the need for explicit preference optimization annotations. Additionally, we propose Truncated Diffusion Fine-tuning, which reduces the dependency on full data generation simulations, significantly improving training efficiency. Extensive experimental results validate the effectiveness of Self-NPO.

Limitations: Since Self-NPO does not rely on explicit preference annotations, it might have a lower performance bound compared to NPO. However, it is important to highlight that by eliminating the need for explicit annotations, Self-NPO broadens the scope of preference optimization, particularly in domains where acquiring such annotations is challenging or impractical.

Method	Training	GPU Hours	GPU Type
Diffusion-NPO	Full weight	384	A100
Diffusion-NPO	LoRA	153.6	A800
Baseline	Full weight	10.4	A800
Ours	Full weight	2	A800

Table 3: Training costs for different methods.

Acknowledgements

This study was supported in part by National Key R&D Program of China Project 2022ZD0161100, in part by the Centre for Perceptual and Interactive Intelligence, a CUHK-led InnoCentre under the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government, in part by NSFC-RGC Project N_CUHK498/24, and in part by Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515130008, XW).

References

Ahn, D.; Cho, H.; Min, J.; Jang, W.; Kim, J.; Kim, S.; Park, H. H.; Jin, K. H.; and Kim, S. 2024. Self-Rectifying Diffusion Sampling with Perturbed-Attention Guidance. *arXiv preprint arXiv:2403.17377*.

Bian, W.; Huang, Z.; Shi, X.; Li, Y.; Wang, F.-Y.; and Li, H. 2025. GS-DiT: Advancing Video Generation with Pseudo 4D Gaussian Fields through Efficient Dense 3D Point Tracking. *arXiv preprint arXiv:2501.02690*.

Blattmann, A.; Rombach, R.; Ling, H.; Dockhorn, T.; Kim, S. W.; Fidler, S.; and Kreis, K. 2023. Align your latents: High-resolution video synthesis with latent diffusion mod-

els. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22563–22575.

Chen, H.; Zhang, Y.; Cun, X.; Xia, M.; Wang, X.; Weng, C.; and Shan, Y. 2024. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7310–7320.

Clark, K.; Vicol, P.; Swersky, K.; and Fleet, D. J. 2023. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*.

Deng, Y.; Lu, P.; Yin, F.; Hu, Z.; Shen, S.; Gu, Q.; Zou, J. Y.; Chang, K.-W.; and Wang, W. 2024. Enhancing large vision language models with self-training on image comprehension. *Advances in Neural Information Processing Systems*, 37: 131369–131397.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

Efron, B. 2011. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496): 1602–1614.

Gao, R.; Holynski, A.; Henzler, P.; Brussee, A.; Martin-Brualla, R.; Srinivasan, P.; Barron, J. T.; and Poole, B. 2024. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*.

Griffith, S.; Subramanian, K.; Scholz, J.; Isbell, C. L.; and Thomaz, A. L. 2013. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26.

Gu, Y.; Wang, Z.; Yin, Y.; Xie, Y.; and Zhou, M. 2024. Diffusion-rpo: Aligning diffusion models through relative preference optimization. *arXiv preprint arXiv:2406.06382*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; and Salimans, T. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577.

Karras, T.; Aittala, M.; Kynkäänniemi, T.; Lehtinen, J.; Aila, T.; and Laine, S. 2024. Guiding a Diffusion Model with a Bad Version of Itself. *arXiv preprint arXiv:2406.02507*.

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J. 2021. Variational diffusion models. *Advances in neural information processing systems*, 34: 21696–21707.

Kirstain, Y.; Polyak, A.; Singer, U.; Matiana, S.; Penna, J.; and Levy, O. 2023. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 36652–36663.

Lai, Z.; Zhao, Y.; Zhao, Z.; Liu, H.; Wang, F.; Shi, H.; Yang, X.; Lin, Q.; Huang, J.; Liu, Y.; et al. 2025. Unleashing Vecset Diffusion Model for Fast Shape Generation. *arXiv preprint arXiv:2503.16302*.

Li, S.; Sun, K.; Lai, Z.; Wu, X.; Qiu, F.; Xie, H.; Miyata, K.; and Li, H. 2024. Ecnet: Effective controllable text-to-image diffusion models. *arXiv preprint arXiv:2403.18417*.

Li, Z.; Hu, M.; Zheng, Q.; and Jiang, X. 2025. Connecting Consistency Distillation to Score Distillation for Text-to-3D Generation. In *European Conference on Computer Vision*, 274–291. Springer.

Liu, J.; Liu, G.; Liang, J.; Yuan, Z.; Liu, X.; Zheng, M.; Wu, X.; Wang, Q.; Qin, W.; Xia, M.; et al. 2025. Improving Video Generation with Human Feedback. *arXiv preprint arXiv:2501.13918*.

Ma, Y.; Xu, W.; Zhao, C.; Sun, K.; Jin, Q.; Zhao, Z.; Fan, C.; and Hu, Z. 2024. Storynizor: Consistent Story Generation via Inter-Frame Synchronized and Shuffled ID Injection. *arXiv preprint arXiv:2409.19624*.

Mao, X.; Jiang, Z.; Wang, F.-Y.; Zhu, W.; Zhang, J.; Chen, H.; Chi, M.; and Wang, Y. 2024. Osv: One step is enough for high-quality image to video generation. *arXiv preprint arXiv:2409.11367*.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Mou, C.; Wang, X.; Xie, L.; Wu, Y.; Zhang, J.; Qi, Z.; and Shan, Y. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 4296–4304.

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

Pokle, A.; Geng, Z.; and Kolter, J. Z. 2022. Deep Equilibrium Approaches to Diffusion Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 37975–37990. Curran Associates, Inc.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Prabhudesai, M.; Mendonca, R.; Qin, Z.; Fragkiadaki, K.; and Pathak, D. 2024. Video Diffusion Alignment via Reward Gradients. *arXiv preprint arXiv:2407.08737*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Schuhmann, C. 2022. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics/. Accessed: 2023-11-10.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shen, D.; Song, G.; Xue, Z.; Wang, F.-Y.; and Liu, Y. 2024. Rethinking the Spatial Inconsistency in Classifier-Free Diffusion Guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9370–9379.

Shi, X.; Huang, Z.; Wang, F.-Y.; Bian, W.; Li, D.; Zhang, Y.; Zhang, M.; Cheung, K. C.; See, S.; Qin, H.; et al. 2024. Motion-I2V: Consistent and Controllable Image-to-Video Generation with Explicit Motion Modeling. *arXiv e-prints*, arXiv–2401.

Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; et al. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sun, K.; Jourabloo, A.; Bhalodia, R.; Meshry, M.; Rong, Y.; Yang, Z.; Nguyen-Phuoc, T.; Haene, C.; Xu, J.; Johnson, S.; et al. 2024a. GenCA: A Text-conditioned Generative Model for Realistic and Drivable Codec Avatars. *arXiv preprint arXiv:2408.13674*.

Sun, K.; Pan, J.; Ge, Y.; Li, H.; Duan, H.; Wu, X.; Zhang, R.; Zhou, A.; Qin, Z.; Wang, Y.; et al. 2024b. Journeydb: A benchmark for generative image understanding. *Advances in Neural Information Processing Systems*, 36.

Sutton, R. S. 2018. Reinforcement learning: An introduction. *A Bradford Book*.

Teng, Y.; Wu, Y.; Shi, H.; Ning, X.; Dai, G.; Wang, Y.; Li, Z.; and Liu, X. 2024. DiM: Diffusion Mamba for Efficient High-Resolution Image Synthesis. *arXiv preprint arXiv:2405.14224*.

Wallace, B.; Dang, M.; Rafailov, R.; Zhou, L.; Lou, A.; Purushwalkam, S.; Ermon, S.; Xiong, C.; Joty, S.; and Naik, N. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8228–8238.

Wang, F.-Y.; Huang, Z.; Bergman, A. W.; Shen, D.; Gao, P.; Lingelbach, M.; Sun, K.; Bian, W.; Song, G.; Liu, Y.; et al. 2024a. Phased Consistency Model. *arXiv preprint arXiv:2405.18407*.

Wang, F.-Y.; Huang, Z.; Bian, W.; Shi, X.; Sun, K.; Song, G.; Liu, Y.; and Li, H. 2024b. AnimateLCM: Computation-Efficient Personalized Style Video Generation without Personalized Video Data. arXiv:2402.00769.

Wang, F.-Y.; Shui, Y.; Piao, J.; Sun, K.; and Li, H. 2025a. Diffusion-NPO: Negative Preference Optimization for Better Preference Aligned Generation of Diffusion Models. In *The Thirteenth International Conference on Learning Representations*.

Wang, F.-Y.; Wu, X.; Huang, Z.; Shi, X.; Shen, D.; Song, G.; Liu, Y.; and Li, H. 2025b. Be-your-outpainter: Mastering video outpainting through input-specific adaptation. In *European Conference on Computer Vision*, 153–168. Springer.

Wang, F.-Y.; Yang, L.; Huang, Z.; Wang, M.; and Li, H. 2024c. Rectified diffusion: Straightness is not your need in rectified flow. *arXiv preprint arXiv:2410.07303*.

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

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

Wu, X.; Hao, Y.; Zhang, M.; Sun, K.; Huang, Z.; Song, G.; Liu, Y.; and Li, H. 2024. Deep Reward Supervisions for Tuning Text-to-Image Diffusion Models. *arXiv preprint arXiv:2405.00760*.

Wu, X.; Sun, K.; Zhu, F.; Zhao, R.; and Li, H. 2023c. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2096–2105.

Xu, J.; Liu, X.; Wu, Y.; Tong, Y.; Li, Q.; Ding, M.; Tang, J.; and Dong, Y. 2024. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36.

Yan, R.; Chen, Y.; and Wang, X. 2025. Consistent Flow Distillation for Text-to-3D Generation. *arXiv preprint arXiv:2501.05445*.

Yang, Z.; Teng, J.; Zheng, W.; Ding, M.; Huang, S.; Xu, J.; Yang, Y.; Hong, W.; Zhang, X.; Feng, G.; et al. 2024. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*.

Yuan, H.; Chen, Z.; Ji, K.; and Gu, Q. 2024. Self-play fine-tuning of diffusion models for text-to-image generation. *arXiv preprint arXiv:2402.10210*.

Zhang, D.; Lan, G.; Han, D.-J.; Yao, W.; Pan, X.; Zhang, H.; Li, M.; Chen, P.; Dong, Y.; Brinton, C.; et al. 2024a. SePPO: Semi-Policy Preference Optimization for Diffusion Alignment. *arXiv preprint arXiv:2410.05255*.

Zhang, Y.; Tzeng, E.; Du, Y.; and Kislyuk, D. 2024b. Large-scale Reinforcement Learning for Diffusion Models. *arXiv preprint arXiv:2401.12244*.

Zhou, M.; Zheng, H.; Wang, Z.; Yin, M.; and Huang, H. 2024. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *Forty-first International Conference on Machine Learning*.