FESTDPO: FEW-STEP GENERATOR ALIGNMENT WITH DIRECT PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step generative models can generate high-fidelity samples within a few function evaluations. Despite this efficiency, generated samples may not exhibit desirable properties. When these properties are difficult to encode as an explicit reward function, direct preference optimization (DPO) can align generative models using pairwise preference feedback without training a separate reward model. However, extending DPO to few-step generative models is challenging because few-step generative models are generally implicit, making the likelihood evaluation required by DPO intractable. To address this challenge, we introduce **Few-step DPO (FestDPO)**, an extension of DPO for few-step generative models that leverages nonparametric likelihood estimation from empirical samples. By exploiting the fast sampling capabilities of few-step generative models, our approach makes sample-based approximation of DPO loss computationally feasible. Furthermore, the sample-based formulation makes FestDPO agnostic to the model family and sampling procedure. Our toy experiment demonstrates that FestDPO matches the reward-tilted target distribution across four few-step generators. For real-world tasks, we evaluate FestDPO in two domains: text-to-image generation and protein backbone generation. In text-to-image generation, FestDPO outperforms preference optimization baselines in both win rates against the base models and human evaluation scores. In protein backbone generation, it achieves a higher β-sheet fraction and better structural designability than the baselines.

1 INTRODUCTION

Recently emerging few-step generative models (Song et al., 2023; Frans et al., 2025; Zhou et al., 2025; Geng et al., 2026; Deng et al., 2026) enable fast generation with high-quality samples. However, generated samples may not satisfy user preferences (Xu et al., 2023; Wu et al., 2023), including undesirable content (Park et al., 2024; Liu et al., 2025) or potentially harmful biological molecules (Das et al., 2021). Encoding these desiderata in an explicit reward function can be difficult, whereas pairwise comparisons provide a way to express preferences for alignment (Christiano et al., 2017; Wirth et al., 2017). Direct Preference Optimization (DPO; Rafailov et al., 2023) provides a solution for aligning generative models with preference pairs, operating without explicit reward models, and avoiding a potentially unstable online reinforcement learning (RL) loop. However, extending DPO to few-step generative models is challenging since the implicit nature of few-step models makes the likelihood evaluation (Ai et al., 2026) required by DPO intractable.

To overcome this limitation, we exploit the inherent sampling efficiency of few-step generative models. We found that the fast sampling speed makes sample-based nonparametric density estimation (Silverman, 2018) computationally feasible. Building on this, we propose **Few-step DPO (FestDPO)**, a direct preference optimization method tailored to few-step generative models. DPO aims to sample from a reward-tilted unnormalized density, where the reward is derived from the Bradley-Terry preference model (Bradley & Terry, 1952). To approximate DPO with few-step generative models, we replace its intractable likelihood terms with sample-based density estimation. The sample-based approach offers a key advantage: FestDPO is agnostic to the sampling procedure and pre-trained model family, allowing it to be applied to a broad range of few-step generative models without model-specific modifications. For the domains where distance in the original data space fails to reflect underlying structure, such as semantic similarity in images or rotational invariance in protein structures, we estimate densities in the embedding space of a pretrained encoder.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

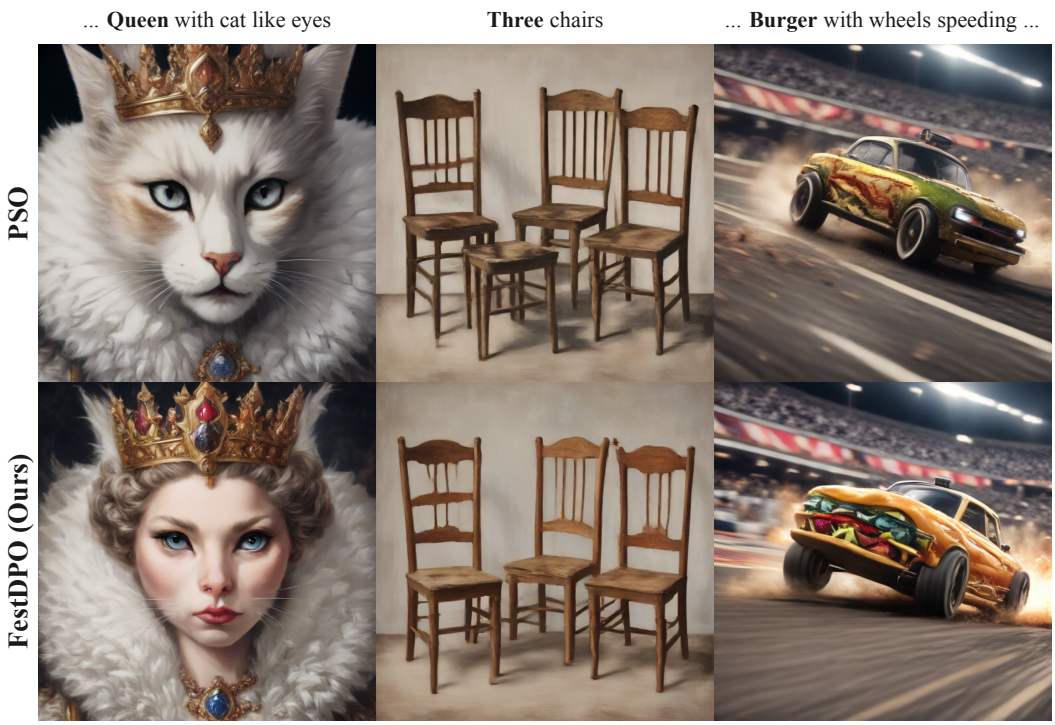

Figure 1: Qualitative comparison of PSO (Miao et al., 2025) and FestDPO on three text prompts: "a gorgeous **queen** with cat-like eyes" (Left), "**three** chairs" (Middle), and "3D digital illustration, **burger with wheels** speeding on the race track, supercharged, detailed, hyperrealistic, 4K" (Right).

We evaluate FestDPO in a 1-D toy setting, text-to-image generation (Rombach et al., 2022; Esser et al., 2024), and protein backbone generation (Woo et al., 2026), demonstrating effective preference optimization across tasks. In the toy setting, we show that FestDPO learns to sample from the unnormalized target density induced by the latent reward in the Bradley-Terry model across diverse few-step generative models. In text-to-image generation at 512×512 resolution, FestDPO achieves the highest win rates among preference optimization baselines when fine-tuning both SDXL-Turbo (Sauer et al., 2024) and SDXL-DMD2 (Yin et al., 2024a), and receives the highest human ratings. Our ablation studies further show that FestDPO is robust to the choice of feature encoder and to the number of samples used for density estimation, indicating that sample-based estimation with a finite number of samples is effective for preference optimization, even in high-dimensional domains such as images and proteins. In protein backbone generation, we demonstrate that FestDPO on a few-step protein backbone generator (Woo et al., 2026) outperforms the baselines in structural designability and control over secondary structure composition of the protein.

2 Related Work

2.1 Few-step generative models

Few-step generative models enable efficient sampling within a few function evaluations, addressing the expensive computational costs of diffusion (Ho et al., 2020; Song et al., 2020) and flow-based models (Liu et al., 2022; Lipman et al., 2022). These models broadly fall into two categories. The first comprises direct noise-to-data mappings, such as VAEs (Kingma & Welling, 2013), GANs (Goodfellow et al., 2020), and recently proposed drifting models (Deng et al., 2026). The second category is flow map models (Song et al., 2023; Kim et al., 2024; Boffi et al., 2024; Frans et al., 2025; Geng et al., 2026; Zhou et al., 2025; Boffi et al., 2026), including distilled generators (Luo et al., 2023; Sauer et al., 2024; Yin et al., 2024b), which learn jump operators between intermediate states between noise and data. Despite advances in sampling speed, many of these models are implicit and do not provide tractable likelihoods (Ai et al., 2026) required by RL and probabilistic inference-based alignment methods (Uehara et al., 2024; 2025). Noise-to-data generators also do not exhibit denoising trajectories, which are required for score matching-based approaches (Honavar, 2025).

2.2 ALIGNMENT

Alignment is a core technique for steering generative models towards desirable output, broadly applied for natural language (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022), images (Black et al., 2024; Fan et al., 2023; Clark et al., 2024; Domingo i Enrich et al., 2025; Kang et al., 2026; Lee et al., 2026c; Lee & Ye, 2026), robotic control (Ren et al., 2025), and scientific design (Gu et al., 2024; Lee et al., 2026b; Su et al., 2026). Many alignment methods rely on an explicit reward function, either predefined or learned from a labeled dataset. DPO (Rafailov et al., 2023) instead optimizes generators directly from offline preference pairs under a Bradley-Terry preference model (Bradley & Terry, 1952), avoiding both an explicit reward model and an online RL loop. However, extending DPO-based alignment methods to emerging few-step generators is not straightforward: prior works often rely on diffusion-specific formulations (Wallace et al., 2024; Yang et al., 2024a; Croitoru et al., 2025; Liu et al., 2026) or require likelihood evaluation on a denoising trajectory (Yang et al., 2024b; Liang et al., 2025; Honavar, 2025; Miao et al., 2025). Our concurrent work, DrPO (Jiang et al., 2026), uses preference pairs to construct drift fields (Deng et al., 2026) for aligning few-step generators, but its connection to the DPO objective remains unestablished. We instead develop a sample-based approximation to DPO and establish its asymptotic consistency, clarifying the corresponding reward-tilted target distribution.

2.3 SAMPLING FROM UNNORMALIZED DENSITY

Sampling from unnormalized density has been a fundamental problem of machine learning (Hinton, 2002; LeCun et al., 2006). Given access to reward or energy evaluations, sampling methods can approximate the target distribution through Monte Carlo estimation (Hastings, 1970; Del Moral et al., 2006), or by learning amortized samplers, including GFlowNets (Bengio et al., 2021; 2023; Choi et al., 2025) and diffusion samplers (Zhang & Chen, 2021; Akhound-Sadegh et al., 2024; Havens et al., 2025; 2026). The sampling perspective also applies to alignment: the solution of KL-regularized reward maximization is equivalent to sampling from a reward-tilted unnormalized density (Korbak et al., 2022). Prior methods adopt a sampling perspective to align generative models (Venkatraman et al., 2024; Domingo i Enrich et al., 2025; Lee et al., 2026a) using explicit reward evaluations. FestDPO instead learns a few-step implicit sampler that approximates the reward-tilted distribution from a fixed preference dataset, without evaluating a reward function during training.

3 BACKGROUND

3.1 PREFERENCE OPTIMIZATION

Given an input x, let $y_w \succ y_l \mid x$ denote a preference for y_w over y_l. Learning from preferences (Christiano et al., 2017; Ziegler et al., 2019) aims to align model behavior with the latent reward $r^\star$ that governs the preference outcomes. The Bradley–Terry (BT) model (Bradley & Terry, 1952) formulates how the latent reward $r^\star$ determines preference distribution $p^\star$:

$$p^\star\left(y_w \succ y_l \mid x\right) = \sigma\left(r^\star\left(x, y_w\right) - r^\star\left(x, y_l\right)\right), \tag{1}$$

where σ is the logistic function. Given a pairwise preference dataset $\mathcal{D}$, the standard preference optimization objective is the expected reward with a reverse KL divergence penalty to a reference generative model to mitigate reward over-optimization (Gao et al., 2023):

$$\mathcal{J}(\pi) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(\cdot|x)}\left[r^\star(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi(y \mid x) \| \pi_{\mathrm{ref}}(y \mid x)\right]. \tag{2}$$

where $\beta > 0$ controls the regularization strength and π, π_{ref} denotes a policy and fixed reference policy respectively. Early approaches (Stiennon et al., 2020; Ouyang et al., 2022) train a reward model $\hat{r}$ to approximate the latent reward $r^\star$ by maximizing the likelihood in Equation (1). The policy is then optimized using proximal policy optimization (PPO) (Schulman et al., 2017) with the learned reward model $\hat{r}$ instead of the latent reward $r^\star$. While effective at aligning models with human preferences, online RL with $\hat{r}$ requires a separate stage to train the reward model and suffers from the instability and hyperparameter sensitivity of online RL (Rafailov et al., 2023).

3.2 Direct Preference Optimization

DPO (Rafailov et al., 2023) avoids training a separate reward model and an online RL loop by optimizing the generative model directly from preference pairs. DPO objective is derived from the closed-form solution to the KL-regularized objective (Peng et al., 2019; Korbak et al., 2022):

$$\pi^\star(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r^\star(x,y)\right), \tag{3}$$

where $Z(x) = \int \pi_{\text{ref}}(y \mid x)\exp(\frac{1}{\beta}r^\star(x,y))dy$ is the normalizing constant. Then, the latent reward r^* can be expressed as follows:

$$r^\star(x,y) = \beta \log \frac{\pi^\star(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \tag{4}$$

Substituting Equation (4) into the Equation (1) cancels the intractable $Z(x)$ and yields the DPO objective for a parameterized policy π_θ:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]. \tag{5}$$

Minimizing Equation (5) trains π_θ to match the optimal reward-tilted distribution in Equation (3) (see Appendix A for details).

4 Extend DPO towards Few-step Generative Models

While DPO offers a promising way to align generative models with preference pairs, its extension to few-step generators remains underexplored. A main challenge is that many few-step generators are implicit models, for which the likelihood evaluation required by the DPO loss is intractable (Ai et al., 2026). Existing DPO extensions for diffusion models rely on Evidence Lower Bound (ELBO)-based likelihood surrogates (Wallace et al., 2024; Yang et al., 2024a) or tractable likelihood evaluation of the denoising transition (Honavar, 2025), which is usually unavailable for few-step implicit generators.

Consequently, extending DPO to few-step generative models requires a new formulation. Given fast sampling and algorithmic diversity, we found that employing sample-based nonparametric estimation to construct a tractable surrogate objective of DPO can lead to the following advantages: (i) making effective use of the computational efficiency of few-step sampling; (ii) being agnostic to the pre-training objective and sampling procedure. These observations lead us to propose **Few-step DPO (FestDPO)**, a sample-based approximation of DPO for few-step generative models.

5 Methods

In this section, we introduce **Few-step DPO (FestDPO)**, a direct preference optimization method tailored to few-step generative models. Figure 2 provides an overview of FestDPO: using samples from the trainable generator and the reference generator, we estimate the intractable likelihoods at y_w and y_l through a sample-based approach. Then, we employ estimated likelihoods to approximate DPO objective, leading to the FestDPO objective in Equation (8). Theoretically, we establish conditions for the FestDPO loss to converge in probability to the DPO loss, and show that the unique minimizer of the FestDPO loss matches the reward-tilted distribution.

5.1 FestDPO

Following Rafailov et al. (2023), our goal is to fine-tune a few-step generator to sample from the reward-tilted unnormalized density defined by the latent reward $r^\star$:

$$\pi^\star(y \mid x) \propto \pi_{\text{ref}}(y \mid x)\exp\left(r^\star(x,y)/\beta\right). \tag{6}$$

To learn a sampler for Equation (6) without exact likelihood evaluation, we approximate the likelihood terms in the DPO loss Equation (5) using sample-based nonparametric density estimation

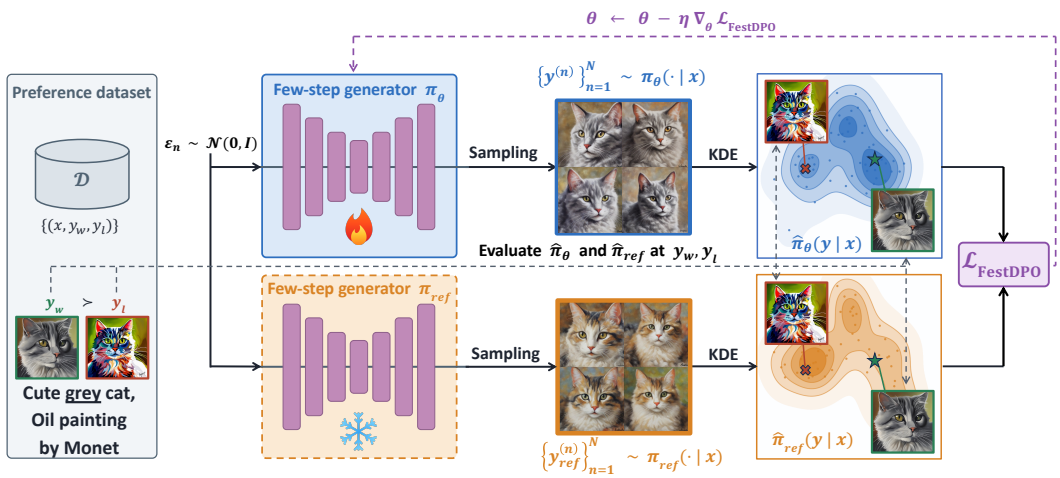

Figure 2: **Overview of FestDPO.** For a prompt x and a preference pair $(y_w \succ y_l)$, we draw N samples from both the trainable policy π_θ and the frozen reference policy π_{ref}. We then construct kernel density estimates $\hat{\pi}_\theta$ and $\hat{\pi}_{\text{ref}}$ and evaluate them at y_w and y_l. These estimates yield a tractable surrogate for the DPO loss (Equation (8)), which we minimize to align π_θ with the preference data.

(Silverman, 2018; Liu et al., 2022). Thanks to fast sampling from few-step generative models, sample-based estimation becomes practically feasible. For simplicity, we use kernel density estimation (KDE) to approximate the likelihood $\pi_\theta(y \mid x)$ as follows:

$$\hat{\pi}_\theta(y \mid x) = \mathbb{E}_{y' \sim \pi_\theta}[k_h(y, y')], \tag{7}$$

where k_h is a Gaussian kernel with bandwidth h. We estimate the reference density π_{ref} analogously. Replacing every likelihood term in Equation (5) with Equation (7) yields a FestDPO loss: a tractable surrogate of DPO loss for few-step implicit generators:

$$\mathcal{L}_{\text{FestDPO}}(\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\hat{\pi}_\theta(y_w \mid x)}{\hat{\pi}_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\hat{\pi}_\theta(y_l \mid x)}{\hat{\pi}_{\text{ref}}(y_l \mid x)} \right) \right]. \tag{8}$$

In practice, we draw N independent noise $\{\epsilon_n\}_{n=1}^N$ and pass them through both the trainable generator and the frozen reference generator simultaneously. We use the resulting samples to construct density estimates and evaluate the FestDPO loss on minibatches of preference pairs. Gradients propagate through f_θ, while f_{ref} remains fixed. The resulting gradient signal steers the current generator, pulling it toward the preferred samples while pushing it away from the unpreferred samples.

5.2 Theoretical Analysis

In this section, we theoretically analyze the connection between FestDPO and DPO. Under the consistency of kernel density estimation, we show that the FestDPO objective asymptotically recovers the DPO objective. We further establish that minimizers of the FestDPO objective are consistent for the DPO minimizer.

Proposition 1 (Asymptotic Equivalence of FestDPO and DPO)**.** *Let $\hat{\pi}_\theta$ and $\hat{\pi}_{ref}$ be KDEs constructed from N samples with bandwidth h, using a kernel K satisfying the KDE consistency conditions (Silverman, 2018). Suppose that $h \to 0$ and $Nh^d \to \infty$ as $N \to \infty$, and that, for almost every (x, y_w, y_l) under $\mathcal{D}$, $\pi_\theta(\cdot \mid x)$ and $\pi_{\text{ref}}(\cdot \mid x)$ are continuous and strictly positive at y_w and y_l. If the per-example FestDPO losses are uniformly integrable, then*

$$\mathcal{L}_{\text{FestDPO}}(\theta) \to \mathcal{L}_{\text{DPO}}(\theta). \tag{9}$$

Proposition 1 shows that, as the number of samples from the model and reference increases, the FestDPO objective converges to the DPO objective. We further show that FestDPO minimizer converges to the DPO minimizer, which corresponds to the optimal target distribution induced by the KL-regularized reward maximization objective in Appendix A.

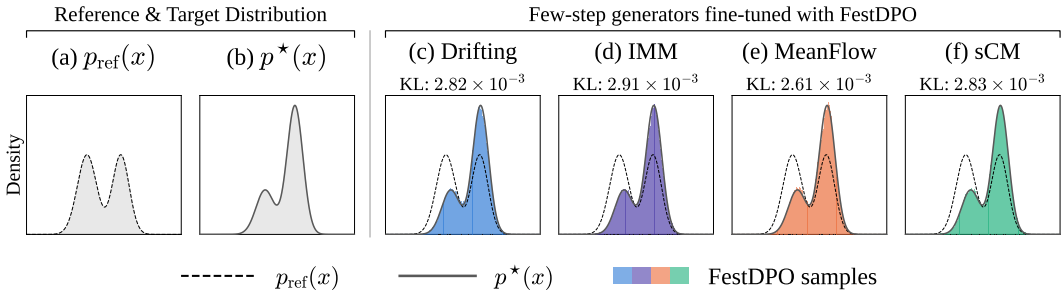

Figure 3: **FestDPO aligns diverse few-step generators.** (a) The reference distribution $p_{\text{ref}}(x)$ and (b) the reward-tilted target distribution $p^\star(x) \propto p_{\text{ref}}(x)\exp(r^\star(x))$. (c)–(f) Distributions of Drifting, IMM, MeanFlow, and sCM fine-tuned with FestDPO. Across all four generators, the fine-tuned distributions closely match $p^\star(x)$, with KL divergences to the target of approximately 3×10^{-3}.

Proposition 2 (Consistency of the FestDPO Minimizer)**.** *Let (Θ, d) be a metric space. Suppose that the convergence in Proposition 1 holds uniformly over Θ and that the population DPO objective admits a unique, well-separated minimizer $\theta^\star$. Then, for any sequence of $o_p(1)$-approximate minimizers $\hat{\theta}^\star$ of the FestDPO objective,*

$$d(\hat{\theta}^\star, \theta^\star) \xrightarrow{p} 0, \qquad N \to \infty. \tag{10}$$

The detailed derivations for Proposition 1 and 2 are provided in Appendix B and C, respectively.

5.3 Kernel Density Estimation in Feature Space

Kernel similarities in raw space may poorly capture the image semantics (Theis et al., 2015) or depend on arbitrary rotations and translations of protein structures (Jumper et al., 2021; Satorras et al., 2021). Motivated by kernel-based distribution matching in embedding spaces (Bińkowski et al., 2018; Deng et al., 2026; Lee et al., 2026a), we compute kernel similarity in the embedding space of a pretrained encoder ϕ to better reflect semantic and geometric similarities. Specifically, we replace $\hat{\pi}_\theta(\cdot \mid x)$ in Equation (8) with

$$\hat{\pi}_{\theta,\phi}(y \mid x) = \mathbb{E}_{y' \sim \pi_\theta(\cdot|x)}\left[k_h\left(\phi(y), \phi\left(y'\right)\right)\right] \tag{11}$$

6 Experiments

In this section, we evaluate FestDPO in three tasks: a 1D toy setting with diverse few-step generative models, text-to-image generators, and protein backbone generation. These experiments demonstrate that FestDPO is capable of sampling from preference-tilted distributions across tasks, from one-dimensional data to high-resolution images and protein structures on SE(3) manifold. Implementation details and hyperparameters for all experiments are provided in Section E

6.1 Toy Experiments

To evaluate the capability of FestDPO in sampling from a tilted target distribution defined by the latent reward model: $p^\star(x) \propto p_{\text{ref}}(x)\exp(r^\star(x))$, we perform a synthetic experiment on a 1D Gaussian Mixture Model. We first pre-train four few-step generators, namely Drifting (Deng et al., 2026), IMM (Zhou et al., 2025), MeanFlow (Geng et al., 2026), and sCM (Lu & Song, 2025). We construct an offline preference dataset by drawing pairs of samples from each pre-trained generator and labeling them with the Bradley-Terry model under $r^\star$, and fine-tune each generator with FestDPO on this dataset. As shown in Figure 3, FestDPO closely matches the target distribution across all generators, achieving a KL divergence of approximately 3×10^{-3}, while preserving the multi-modal structure of the target. Detailed descriptions are provided in Appendix E.1.

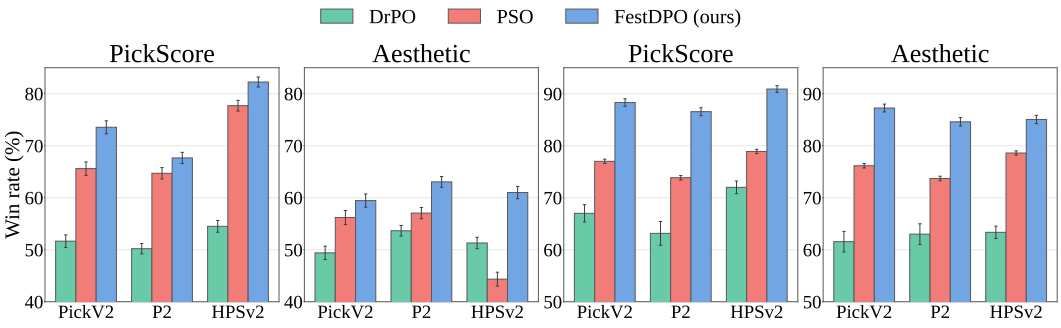

Figure 4: Win rate comparison against (a) SDXL-Turbo and (b) SDXL-DMD2 on Pick-a-Pic v2, Parti-Prompts, and HPSv2 evaluation datasets. Each bar represents the mean win rate (%) of each method for PickScore and Aesthetic score, and error bars denote the standard deviation.

6.2 Text-to-image generation

We evaluate FestDPO for aligning text-to-image models with user preferences. We first describe the experimental setup in Section 6.2.1, then report the main results in Section 6.2.2, and finally present ablation studies in Section 6.2.3.

6.2.1 Experimental setup

Baselines. We compare FestDPO with PSO (Miao et al., 2025) and DrPO (Jiang et al., 2026), which are the preference optimization methods for fine-tuning a few-step generator from offline preference pairs. We defer a detailed description of the baselines to Appendix D.

Base Models & Preference Dataset. We fine-tune two few-step text-to-image generators: SDXL-Turbo (Sauer et al., 2024) and SDXL-DMD2 (Yin et al., 2024a). All methods are trained on the training split of Pick-a-Pic v2 (Kirstain et al., 2023), an open dataset of text-to-image prompts and preferences of real users over generated images.

Evaluation. We evaluate the generated images using PickScore (Kirstain et al., 2023) and Aesthetic Score (Schuhmann & Beaumont, 2023). We also report pairwise win rates against the corresponding base models. We then evaluate on 500 prompts sampled from each of three sets: the held-out Pick-a-Pic v2 test split (PickV2), Parti-Prompts (P2; Yu et al., 2022), and the HPSv2 benchmark (Wu et al., 2023). We further conduct a human study on 80 of these prompts, reported in Section 6.2.4.

Feature Encoder. Our default feature extractor is the 340M-parameter Latent-MAE encoder (He et al., 2022) from Drifting Models (Deng et al., 2026). To assess the sensitivity of FestDPO to the choice of feature representation, we further conduct ablations using DINOv2 (Oquab et al., 2023) and CLIP (Radford et al., 2021) features in Section 6.2.3.

6.2.2 Main Results

We report the win rates of FestDPO, PSO, and DrPO over base models on Pick-a-Pic v2, PartiPrompts, and HPSv2 evaluation datasets in Figure 4. For each base model, we compute the win rate as the percentage of prompts on which the fine-tuned model scores higher than the base model, separately for PickScore and Aesthetic score. FestDPO achieves the highest win rate on every prompt dataset under PickScore and Aesthetic. Figure 5 further shows the anytime performance on SDXL-DMD2. FestDPO rapidly improves PickScore within the first 150 steps and remains stable, whereas both baselines show slower and smaller improvements. We report the absolute scores and win rates on Pick-a-Pic v2, PartiPrompts, and HPSv2 evaluation datasets in Appendix F and provide the qualitative comparison in Appendix G.

6.2.3 Ablation Study

In this section, We conduct ablation studies on HPSv2 with SDXL-Turbo. Since FestDPO employs nonparametric density estimation, we first study the effect of the number of samples N used to

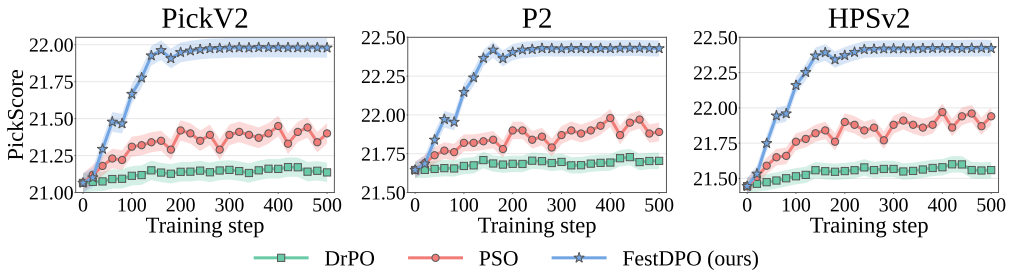

Figure 5: PickScore of DrPO, PSO, and FestDPO during fine-tuning of SDXL-DMD2 on Pick-a-Pic v2, Parti-Prompts, and HPSv2. Lines and shaded regions denote the mean and standard deviation.

Table 1: Sensitivity analysis on N.

N	PickScore ↑	Aesthetic ↑
$N = 8$	23.06 ± 1.32	6.13 ± 0.68
$N = 12$ (default)	23.13 ± 1.35	6.11 ± 0.66
$N = 24$	23.14 ± 1.37	6.07 ± 0.63

Table 2: Ablation on the feature encoder ϕ.

Feature Encoder	PickScore ↑	Aesthetic ↑
Latent MAE (default)	23.13 ± 1.35	6.11 ± 0.66
CLIP	23.17 ± 1.35	6.08 ± 0.64
DINOv2	23.07 ± 1.34	6.08 ± 0.65

construct the KDE. As shown in Table 1, increasing N from 8 to 24 improves PickScore, while Aesthetic scores remain approximately stable across N. These results indicate that FestDPO is robust to the choice of N, with even a small number of samples yielding substantial gains over the pretrained model. We further investigate the sensitivity of FestDPO to the feature encoder. Table 2 compares alignment performance using Latent-MAE (Deng et al., 2026), CLIP (Radford et al., 2021), and DINOv2 (Oquab et al., 2023) as pretrained encoders for density estimation. FestDPO achieves comparable performance across all three encoders, demonstrating that its effectiveness is robust to the choice of feature encoder. Results on Pick-a-Pic v2, Parti-Prompts, and HPSv2 are provided in Section F.

6.2.4 Human Evaluations

We conduct a human evaluation study to assess the prompt alignment and aesthetic quality of the generated samples. As shown in Table 3, FestDPO receives the highest ratings for both prompt alignment and aesthetics. The results indicate that human annotators consistently preferred images generated by FestDPO over those from the baselines. Details of the human evaluation protocol are provided in Appendix H.

Table 3: Human evaluation results (mean rating). Best results are in **bold**.

Method	Align. ↑	Aesth. ↑
SDXL-Turbo	3.683	3.149
PSO	3.767	3.286
DrPO	3.696	3.158
FestDPO (ours)	**4.199**	**3.860**

6.3 Protein Backbone Generation

We evaluate FestDPO on few-step protein backbone generation (Woo et al., 2026) to assess the capability of aligning the few-step generative model on the SE(3) manifold.

6.3.1 Experiment Setup

Base model and baselines. We fine-tune Riemannian MeanFlow (RMF; Woo et al., 2026), a few-step flow map-based protein backbone generator defined on the $\mathrm{SE}(3)$ manifold. Specifically, we fine-tune the RMF-S variant with single-step generation. We employ DrPO (Jiang et al., 2026) as a baseline and exclude PSO (Miao et al., 2025) since extending PSO to flow maps is not trivial.

Evaluation. We optimize two reward functions: (i) self-consistency root mean square distance (scRMSD) for structural designability, and (ii) ratio of β-sheet secondary structure (SS-match). scRMSD evaluates whether a generated protein backbone is realistic. To compute scRMSD, we inverse-fold the generated structure using ProteinMPNN (Dauparas et al., 2022), predict the 3D structure of the resulting sequence with ESMFold (Lin et al., 2023), and calculate the RMSD against

Table 4: β-sheet optimization results.

Method	β-sheet % ↑	TM-diversity ↓
RMF-S (base)	0.153 ± 0.006	**0.281 ± 0.006**
DrPO	0.218 ± 0.005	0.294 ± 0.002
FestDPO	**0.295 ± 0.014**	0.292 ± 0.002

Table 5: scRMSD optimization results.

Method	scRMSD ↓	TM-diversity ↓
RMF-S (base)	9.303 ± 0.549	**0.281 ± 0.006**
DrPO	9.010 ± 0.383	0.305 ± 0.001
FestDPO	**7.435 ± 0.384**	0.294 ± 0.002

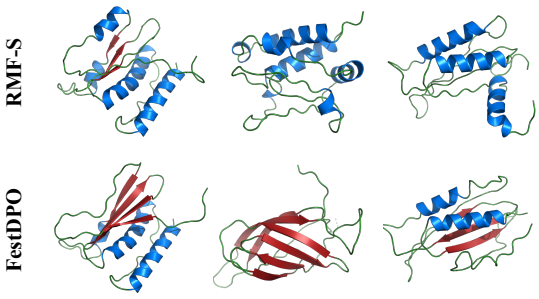

Figure 6: Qualitative comparison between generated protein backbone structures.

the original generated backbone. Motivated by SS rewards in prior works (Huguet et al., 2024; Venkatraman et al., 2025; Su et al., 2026), our second reward encourages the formation of β-sheet secondary structure within the protein. Secondary structures are commonly grouped into α-helices, β-strands, and coils. We use DSSP (Kabsch & Sander, 1983) and P-SEA (Labesse et al., 1997) to assign secondary-structure labels to generated backbones. For each target, we construct a separate offline dataset of 8,000 preference pairs from RMF-S samples, using one-step generation for SS-match and ten-step generation for scRMSD. Within each pair, we label the sample with higher SS-strand reward or lower scRMSD as preferred. Following Woo et al. (2026), we measure structural similarity between the generated backbones through pairwise TM-score (Zhang & Skolnick, 2004).

Feature Encoder. We use the pretrained ProteinMPNN encoder (Dauparas et al., 2022) to obtain structural features invariant to global rotations and translations. The encoder remains frozen throughout training and defines the feature space used for kernel density estimation.

6.3.2 Results

As illustrated in Table 4 and Table 5, FestDPO achieves a higher β-strand residue fraction and lower scRMSD than both base model and DrPO. FestDPO exhibits better structural diversity than DrPO, although diversity decreases relative to the base model. Figure 6 compares backbones generated by the base model and FestDPO after optimization with the SS-match reward, showing increased β-sheet content in the generated protein structure.

7 Discussion

Conclusion. We introduce FestDPO, a sample-based direct preference optimization method for few-step generative models. FestDPO approximates DPO objective with sample-based nonparametric estimation, exploiting the fast sampling speed of few-step generative models. We theoretically show that the FestDPO objective converges in probability to the DPO objective as the number of samples grows, and that its minimizer is consistent. In a 1-D toy experiment, FestDPO learns to sample from the unnormalized target density across diverse few-step generative models, showing the agnosticism of FestDPO. In text-to-image generation, FestDPO achieves the highest win rates against the base model across Pick-a-Pic v2, Parti-Prompts, and HPSv2 datasets and receives the highest human ratings. In protein backbone generation, FestDPO achieves a higher β-strand fraction and lower scRMSD than both the base model and baseline.

Limitations. FestDPO employs sample-based nonparametric density estimation, which is asymptotically consistent as the number of samples grows. In practice, however, we empirically show that sample-based estimation with finite samples is effective for preference optimization, also in high-dimensional domains such as images and proteins. Since density estimation directly in the high-dimensional data space is challenging, FestDPO performs it in the feature space of a pretrained encoder, which captures the underlying structure of the data. While this design introduces a dependence on the choice of feature encoder, our ablation study shows that FestDPO remains robust across different pretrained encoders.

AI USE STATEMENT

Large Language Models were employed exclusively for two auxiliary tasks: (1) minor polishing of the manuscript text for improving grammar and readability, and (2) limited assistance in code implementation for debugging syntax or refactoring functions. Importantly, LLMs did not contribute to the conception of the research problem, the development of the core methodology, or the design and execution of experiments. All critical ideas, methods, and analyses presented in this paper are the original work of the authors.

REFERENCES

Xinyue Ai, Yutong Kelly He, Albert Gu, Russ Salakhutdinov, Zico Kolter, Nicholas Boffi, and Max Simchowitz. Joint distillation for fast likelihood evaluation and sampling in flow-based models. In *International Conference on Learning Representations*, volume 2026, pp. 44940–44977, 2026.

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, et al. Iterated denoising energy matching for sampling from boltzmann densities. *arXiv preprint arXiv:2402.06121*, 2024.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in neural information processing systems*, 34:27381–27394, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Nicholas Boffi, Michael Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *Advances in Neural Information Processing Systems*, 38: 33346–33382, 2026.

Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv preprint arXiv:2406.07507*, 2024.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Sanghyeok Choi, Sarthak Mittal, Víctor Elvira, Jinkyoo Park, and Esmeralda S Whitammer. Reinforced sequential monte carlo for amortised sampling. *arXiv preprint arXiv:2510.11711*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, Nicu Sebe, and Mubarak Shah. Curriculum direct preference optimization for diffusion and consistency models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2824–2834. IEEE, 2025.

Payel Das, Tom Sercu, Kahini Wadhawan, Inkit Padhi, Sebastian Gehrmann, Flaviu Cipcigan, Vijil Chenthamarakshan, Hendrik Strobelt, Cicero Dos Santos, Pin-Yu Chen, et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nature Biomedical Engineering*, 5(6):613–623, 2021.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Luc Devroye. Nonparametric density estimation. *The L_1 View*, 1985.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, volume 2025, pp. 34668–34684, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2026.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Siyi Gu, Minkai Xu, Alexander Powers, Weili Nie, Tomas Geffner, Karsten Kreis, Jure Leskovec, Arash Vahdat, and Stefano Ermon. Aligning target-aware molecule diffusion models with exact energy optimization. *Advances in neural information processing systems*, 37:44040–44063, 2024.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

Aaron Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Brandon Wood, Daniel Levine, Bin Hu, Brandon Amos, Brian Karrer, et al. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. *arXiv preprint arXiv:2504.11713*, 2025.

Aaron Havens, Brian Karrer, and Neta Shaul. Flow sampling: Learning to sample from unnormalized densities via denoising conditional processes. *arXiv preprint arXiv:2605.03984*, 2026.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Vasant G Honavar. Dspo: Direct score preference optimization for diffusion model alignment. *ICLR*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein generation. *Advances in neural information processing systems*, 37:33007–33036, 2024.

Zhou Jiang, Yandong Wen, and Zhen Liu. Drifting preference optimization for one-step generative models. *arXiv preprint arXiv:2606.02521*, 2026.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.

Hyeongyu Kang, Jaewoo Lee, Woocheol Shin, Kiyoung Om, and Jinkyoo Park. Diffusion fine-tuning via reparameterized policy gradient of the soft q-function. In *International Conference on Learning Representations*, volume 2026, pp. 78690–78725, 2026.

Dongjun Kim, Chieh-Hsin Lai, WeiHsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, volume 2024, pp. 44493–44525, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Tomasz Korbak, Ethan Perez, and Christopher Buckley. Rl with kl penalties is better viewed as bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, 2022.

Gilles Labesse, Nathalie Colloc'h, Joël Pothier, and J-P Mornon. P-sea: a new efficient assignment of secondary structure from cα trace of proteins. *Bioinformatics*, 13(3):291–295, 1997.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Jaewoo Lee, Hyeongyu Kang, Dohyun Kim, Kyuil Sim, Woocheol Shin, Minsu Kim, Taeyoung Yun, Jeongjae Lee, Sanghyeok Choi, Tabitha Edith Lee, et al. Aligning few-step generative models by amortizing sample-based variational inference. *arXiv preprint arXiv:2605.26552*, 2026a.

Jaewoo Lee, Minsu Kim, Sanghyeok Choi, Inhyuck Song, Sujin Yun, Hyeongyu Kang, Woocheol Shin, Taeyoung Yun, Kiyoung Om, and Jinkyoo Park. Diffusion alignment as variational expectation-maximization. In *International Conference on Learning Representations*, volume 2026, pp. 31062–31093, 2026b.

Jeongjae Lee and Jong Chul Ye. Pcpo: Proportionate credit policy optimization for preference alignment of image generation models. In *International Conference on Learning Representations*, volume 2026, pp. 126948–126982, 2026.

Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models. *arXiv preprint arXiv:2604.17415*, 2026c.

Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13199–13208. IEEE, 2025.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *Advances in Neural Information Processing Systems*, 38:82155–82192, 2026.

Runtao Liu, I Chieh Chen, Jindong Gu, Jipeng Zhang, Renjie Pi, Qifeng Chen, Philip Torr, Ashkan Khakzar, and Fabio Pizzati. Alignguard: scalable safety alignment for text-to-image generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17024–17034. IEEE, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, volume 2025, pp. 50611–50649, 2025.

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *Advances in Neural Information Processing Systems*, 36:76525–76546, 2023.

Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning timestep-distilled diffusion model using pairwise sample optimization. In *International Conference on Learning Representations*, volume 2025, pp. 13809–13830, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Yong-Hyun Park, Sangdoo Yun, Jin-Hwa Kim, Junho Kim, Geonhui Jang, Yonghyun Jeong, Junghyo Jo, and Gayoung Lee. Direct unlearning optimization for robust and safe text-to-image models. *Advances in Neural Information Processing Systems*, 37:80244–80267, 2024.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

David Pollard. *A user's guide to measure theoretic probability*, volume 1. Cambridge University Press Cambridge, 2002.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

Christoph Schuhmann and Romain Beaumont. Laion-aesthetics, 2022. *URL https://laion.ai/blog/laion-aesthetics/. Accessed*, pp. 11–10, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Xingyu Su, Xiner Li, Masatoshi Uehara, Sunwoo Kim, Yulai Zhao, Gabriele Scalia, Ehsan Hajiramezanali, Tommaso Biancalani, Degui Zhi, and Shuiwang Ji. Iterative distillation for reward-guided fine-tuning of diffusion models in biomolecular design. In *International Conference on Learning Representations*, volume 2026, pp. 139443–139469, 2026.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in neural information processing systems*, 37:76080–76114, 2024.

Siddarth Venkatraman, Mohsin Hasan, Minsu Kim, Luca Scimeca, Marcin Sendera, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Outsourced diffusion sampling: Efficient posterior inference in latent spaces of generative models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 61212–61239. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/venkatraman25a.html.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Dongyeop Woo, Marta Skreta, Seonghyun Park, Kirill Neklyudov, and Sungsoo Ahn. Riemannian meanflow. *arXiv preprint arXiv:2602.07744*, 2026.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8941–8951. IEEE, 2024a.

Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference. *arXiv preprint arXiv:2402.08265*, 2024b.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *arXiv preprint arXiv:2111.15141*, 2021.

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Optimal Distribution of the KL-Regularized Objective

Following Rafailov et al. (2023), we first derive the optimal policy of the KL-regularized reward maximization objective.

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(\cdot|x)}[r(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi(y \mid x)\|\pi_{\mathrm{ref}}(y \mid x)\right] \tag{12}$$

where $r^\star(x,y)$ denotes the latent reward function, π_{ref} the reference policy, and π the policy to be optimized. For $\beta > 0$, the objective in Equation (12) can be rewritten as

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(\cdot|x)}[r(x,y)] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi(y \mid x)\|\pi_{\mathrm{ref}}(y \mid x)\right] \tag{13}$$

$$= \max_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[r(x,y) - \beta\log\frac{\pi(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)}\right] \tag{14}$$

$$= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\log\frac{\pi(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} - \frac{1}{\beta}r(x,y)\right] \tag{15}$$

$$= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\log\frac{\pi(y \mid x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)} - \log Z(x)\right] \tag{16}$$

where the partition function denotes $Z(x) = \int \pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r^\star(x,y)\right)dy$. Because the partition function does not depend on the policy π, we can define the closed-form solution as

$$\pi^\star(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right). \tag{17}$$

Rearranging Equation (17), the reward can be expressed in terms of the corresponding optimal policy:

$$r^\star(x,y) = \beta\log\frac{\pi^\star(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta\log Z(x). \tag{18}$$

Under the Bradley–Terry preference model, the probability that y_w is preferred over y_l is given by

$$P(y_w \succ y_l \mid x) = \sigma\left(r^\star(x,y_w) - r^\star(x,y_l)\right). \tag{19}$$

Substituting Equation (18) into Equation (19), the partition function $Z(x)$ cancels, yielding

$$P(y_w \succ y_l \mid x) = \sigma\left[\beta\log\frac{\pi^\star(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi^\star(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right]. \tag{20}$$

DPO objective is then obtained by parameterizing this preference probability and maximizing its likelihood on the preference dataset. Therefore, under the Bradley–Terry model, DPO objective and the KL-regularized reward maximization objective share the same optimal policy, given by the reward-tilted distribution in Equation (17) (Rafailov et al., 2023).

B Proof of Proposition 1

Proof. For simplicity, define the normalized KDEs using N samples:

$$\hat{\pi}_\theta(y \mid x) = \frac{1}{N}\sum_{n=1}^{N} k_h\left(y, y^{(n)}\right), \quad \hat{\pi}_{\mathrm{ref}}(y \mid x) = \frac{1}{N}\sum_{n=1}^{N} k_h\left(y, y_{\mathrm{ref}}^{(n)}\right), \tag{21}$$

where $y^{(n)} \sim \pi_\theta(\cdot \mid x)$ and $y_{\mathrm{ref}}^{(n)} \sim \pi_{\mathrm{ref}}(\cdot \mid x)$, and $k_h(y,z) = h^{-d}K((y-z)/h)$ for $y, z \in \mathbb{R}^d$, where K is a strictly positive bounded Borel kernel satisfying the standard conditions for pointwise KDE consistency (Devroye, 1985; Silverman, 2018).

As $N \to \infty$, suppose the bandwidth h satisfies $h \to 0$, $Nh^d \to \infty$. Under these conditions, for almost every $(x, y_w, y_l) \sim \mathcal{D}$ at which the corresponding densities are continuous and strictly positive at y_w and y_l, the KDE consistency property (Devroye, 1985; Silverman, 2018) gives

$$\hat{\pi}_\theta(y \mid x) \xrightarrow{p} \pi_\theta(y \mid x) \ \text{ and } \ \hat{\pi}_{\mathrm{ref}}(y \mid x) \xrightarrow{p} \pi_{\mathrm{ref}}(y \mid x), \quad \text{as } N \to \infty. \tag{22}$$

By the continuous mapping theorem (Van der Vaart, 2000), the log-density ratios satisfy

$$\log \frac{\hat{\pi}_\theta(y_w \mid x)}{\hat{\pi}_{\text{ref}}(y_w \mid x)} \xrightarrow{p} \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}, \quad \log \frac{\hat{\pi}_\theta(y_l \mid x)}{\hat{\pi}_{\text{ref}}(y_l \mid x)} \xrightarrow{p} \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}, \tag{23}$$

Combining the above,

$$\beta \log \frac{\hat{\pi}_\theta(y_w \mid x)}{\hat{\pi}_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\hat{\pi}_\theta(y_l \mid x)}{\hat{\pi}_{\text{ref}}(y_l \mid x)} \quad \xrightarrow{p} \quad \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}.$$

Since $z \mapsto -\log \sigma(z)$ is continuous, the continuous mapping theorem yields

$$\ell_{\text{FestDPO}}(x, y_w, y_l) \xrightarrow{p} \ell_{\text{DPO}}(x, y_w, y_l). \tag{24}$$

By assumption, the sequence of per-example FestDPO losses is uniformly integrable. For instance, this assumption holds if the corresponding log-density ratios are uniformly bounded in N, which is guaranteed when the estimated densities are uniformly bounded above and bounded away from zero at the evaluated samples. Together with the convergence in probability established above, this implies convergence in L^1 (Pollard, 2002):

$$\mathbb{E}\left[|\ell_{\text{FestDPO}}(x, y_w, y_l) - \ell_{\text{DPO}}(x, y_w, y_l)|\right] \longrightarrow 0. \tag{25}$$

Using the inequality $|\mathbb{E}[X]| \le \mathbb{E}[|X|]$, we then obtain

$$|\mathcal{L}_{\text{FestDPO}}(\theta) - \mathcal{L}_{\text{DPO}}(\theta)| = |\mathbb{E}\left[\ell_{\text{FestDPO}} - \ell_{\text{DPO}}\right]| \tag{26}$$

$$\le \mathbb{E}\left[|\ell_{\text{FestDPO}} - \ell_{\text{DPO}}|\right] \longrightarrow 0. \tag{27}$$

Hence, the FestDPO objective converges to the DPO objective:

$$\mathcal{L}_{\text{FestDPO}}(\theta) \longrightarrow \mathcal{L}_{\text{DPO}}(\theta). \tag{28}$$

□

C PROOF OF PROPOSITION 2

Proof. Let $\theta^\star = \arg\min_{\theta \in \Theta} \mathcal{L}_{\text{DPO}}(\theta)$ denote the minimizer of DPO objective. By the assumed uniform version of Proposition 1 and the well-separatedness assumption, we have

$$\sup_{\theta \in \Theta} |\mathcal{L}_{\text{FestDPO}}(\theta) - \mathcal{L}_{\text{DPO}}(\theta)| \xrightarrow{p} 0, \tag{29}$$

and, for every $\epsilon > 0$,

$$\inf_{\theta:\, d(\theta, \theta^\star) \ge \epsilon} \mathcal{L}_{\text{DPO}}(\theta) > \mathcal{L}_{\text{DPO}}(\theta^\star), \tag{30}$$

where the second condition is the minimization analogue of the condition in Van der Vaart (2000, Theorem 5.7). Since $\hat{\theta}^\star$ is an $o_p(1)$-approximate minimizer of FestDPO,

$$\mathcal{L}_{\text{FestDPO}}(\hat{\theta}^\star) \le \mathcal{L}_{\text{FestDPO}}(\theta^\star) + o_p(1) \tag{31}$$

where $o_p(1)$ denotes a sequence of random variables that converges to zero in probability as $N \to \infty$. By Equation (29) above,

$$\mathcal{L}_{\text{FestDPO}}(\hat{\theta}^\star) \le \mathcal{L}_{\text{DPO}}(\theta^\star) + o_p(1) \tag{32}$$

$$\mathcal{L}_{\text{DPO}}(\hat{\theta}^\star) - \mathcal{L}_{\text{DPO}}(\theta^\star) \le \mathcal{L}_{\text{DPO}}(\hat{\theta}^\star) - \mathcal{L}_{\text{FestDPO}}(\hat{\theta}^\star) + o_p(1) \tag{33}$$

$$\le \sup_{\theta} |\mathcal{L}_{\text{FestDPO}}(\theta) - \mathcal{L}_{\text{DPO}}(\theta)| + o_p(1) \xrightarrow{\text{P}} 0. \tag{34}$$

Since $\theta^\star$ minimizes the population DPO objective, the left-hand side is nonnegative. Hence,

$$\mathcal{L}_{\text{DPO}}(\hat{\theta}^\star) - \mathcal{L}_{\text{DPO}}(\theta^\star) \xrightarrow{p} 0. \tag{35}$$

By the well-separatedness condition, this implies (Van der Vaart, 2000)

$$d(\hat{\theta}^\star, \theta^\star) \xrightarrow{p} 0. \tag{36}$$

□

D BASELINE DETAILS

D.1 PSO

PSO (Miao et al., 2025) fine-tunes a distilled model $p_\theta(x_0 \,|\, c)$ of $N = 1 \sim 4$ steps. Since applying diffusion loss makes blurry outputs, it maximizes a relative margin between a target sample $x_0^\tau \sim p_{\text{data}}$ and a reference sample $x_0^\rho \sim p_\theta$ from the model being tuned, regularized by the pretrained distilled model p_{pre} with weight β:

$$\mathcal{L} = -\mathbb{E}\left[\log\sigma\left(\beta\log\frac{p_\theta(x_0^\tau \mid c)}{p_{\text{pre}}(x_0^\tau \mid c)} - \beta\log\frac{p_\theta(x_0^\rho \mid c)}{p_{\text{pre}}(x_0^\rho \mid c)}\right)\right], \tag{37}$$

where c is condition. In Equation (37), Evaluating $p_\theta(x_0 \mid c)$ would require marginalizing over the intermediate states which is generally intractable. PSO therefore focus on the relative margin to joint likelihoods of whole trajectories. Specifically, the forward diffusion process yields the data trajectory, and the reverse generative process yields the reference one:

$$\mathcal{L}_{\text{PSO}} = -\mathbb{E}\left[\log\sigma\left(\beta\sum_n\left(\log\frac{p_\theta(x_{t_{n-1}}^\tau \mid x_{t_n}^\tau, c)}{p_{\text{pre}}(x_{t_{n-1}}^\tau \mid x_{t_n}^\tau, c)} - \log\frac{p_\theta(x_{t_{n-1}}^\rho \mid x_{t_n}^\rho, c)}{p_{\text{pre}}(x_{t_{n-1}}^\rho \mid x_{t_n}^\rho, c)}\right)\right)\right]. \tag{38}$$

PSO makes these transition densities explicit by casting few-step denoising as an MDP with state (x_{t_n}, t_n), action $x_{t_{n-1}}$, and Gaussian policy $\mathcal{N}(\mu_\theta, \sigma_{t_n}^2 I)$. By substituting the MDP action-state conditional distribution, final objective for PSO is

$$\begin{aligned}\mathcal{L}_{\text{PSO}} = -\mathbb{E}\Bigg[\log\sigma\Bigg(-\beta\cdot\sum_{n=2}^{N}\Bigg(&\left(\left\|\epsilon^\tau - \epsilon_\theta\left(x_{t_n}^\tau, t_n, c\right)\right\|^2 - \left\|\epsilon^\tau - \epsilon_{\text{pre}}\left(x_{t_n}^\tau, t_n, c\right)\right\|^2\right)\\ &-\frac{1}{2\sigma_{t_n}^2}\left(\left\|x_{t_{n-1}}^\rho - \mu_\theta\left(x_{t_n}^\rho, t_n, c\right)\right\|^2 - \left\|x_{t_{n-1}}^\rho - \mu_{\text{pre}}\left(x_{t_n}^\rho, t_n, c\right)\right\|^2\right)\Bigg)\Bigg)\Bigg].\end{aligned} \tag{39}$$

In our offline setting, the reference samples are pre-collected offline dataset instead of being drawn from the current model, and their trajectories are approximated by the diffusion forward process. A detailed explanation is provided in Miao et al. (2025).

D.2 DRPO

DrPO (Jiang et al., 2026) aligns a deterministic one-step generator $x = g_\theta(\epsilon, c)$ by kernel k. Given positive $\mathcal{A}^+$ and negative feature samples $\mathcal{A}^-$, they define a non-parametric (dipole) reward with strength γ:

$$R_{\text{dipole}}(z) = \exp(E(z)), \quad E(z) = \gamma\sum_{j=1}^{M}\left[k\left(z, a_j^+\right) - k\left(z, a_j^-\right)\right], \tag{40}$$

where $a_j^+ \in \mathcal{A}^+$ and $a_j^- \in \mathcal{A}^-$ denotes positive and negative sample. Assuming non-negative differentable reward $R(z)$, given $x = g_\theta(\epsilon, c)$ and $z = \phi(x)$, the pathwise gradient of the reference-regularized soft RL objective is

$$\nabla_\theta J = \mathbb{E}_\epsilon\left[\Big(\nabla_z\log R(z) + \lambda\big(\nabla_z\log p_{\text{pre}}(z) - \nabla_z\log p_\theta(z)\big)\Big)\nabla_\theta z\right], \tag{41}$$

where $\nabla_\theta z = \nabla_x\phi(x)\nabla_\theta g_\theta(\epsilon, c)$ and ϕ denotes the feature encoder. The reward score term reduces to $V_{\text{pref}} = \nabla_z\log R_{\text{dipole}}(z)$, and the reference term can be represented by a mean-shift difference between reference features $\mathcal{R}$ and current model features $\mathcal{Z}$. Setting $V_{\text{DrPO}} = V_{\text{pref}} + \lambda(\hat{\mu}_{\mathcal{R}} - \hat{\mu}_{\mathcal{Z}})$, the generator is trained as in drifting models by regressing each feature onto a stop-gradient target with velocity scale η:

$$z_i^\star = \text{sg}\big(z_i + \eta\, V_{\text{DrPO}}(z_i)\big), \qquad \mathcal{L}_{\text{DrPO}} = \frac{1}{2K}\sum_{i=1}^{K}\left\|z_i - z_i^\star\right\|_2^2. \tag{42}$$

In our offline setting, we replace the online feature sets $\mathcal{A}^+$ and $\mathcal{A}^-$ with features of positive and negative samples from the dataset. A detailed explanation is provided in Jiang et al. (2026).

E EXPERIMENTS DETAILS

E.1 TOY EXPERIMENTS

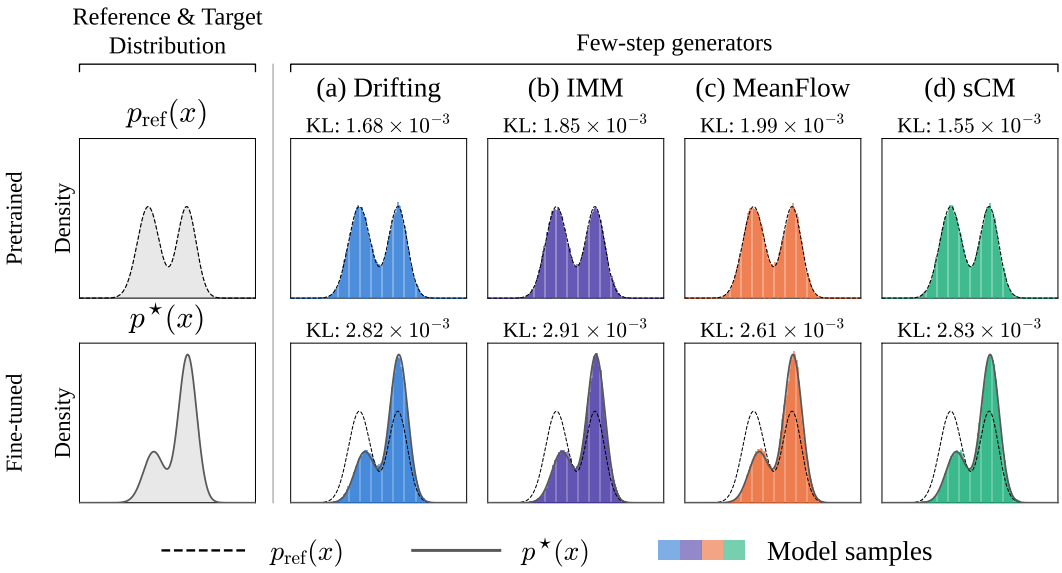

Figure 7: **FestDPO aligns diverse few-step generators with the target distribution.** Left of the solid line: the reference distribution $p_{\text{ref}}(x)$, a mixture of Gaussians, and the reward-tilted target distribution $p^\star(x) \propto p_{\text{ref}}(x)\exp(r^\star(x))$. Right of the solid line: distributions of (a) Drifting model, (b) IMM, (c) MeanFlow, and (d) sCM) before (top) and after (bottom) fine-tuning with FestDPO. Dashed and solid curves denote $p_{\text{ref}}(x)$ and $p^\star(x)$, respectively. We report the KL divergence (KL) between the fine-tuned and target distributions.

Reference, Reward, and Target distribution. The reference distribution $p_{\text{ref}}(x)$ is a two-component Gaussian mixture with means $(-0.625, 0.25)$, variances $(0.0625, 0.046875)$, and weights $(0.5364, 0.4636)$. The reward $r^\star(x)$ is a Gaussian density $\mathcal{N}(x; 0.5, 0.5)$, which peaks slightly to the right of the right mode of $p_{\text{ref}}(x)$. The target distribution is the reward-tilted distribution $p^\star(x) \propto p_{\text{ref}}(x)\exp(r^\star(x))$, which is itself a Gaussian mixture available in closed form. Hence, forward KL divergence is computed against the exact target.

Dataset. For pre-training, we draw 10^5 i.i.d. samples from $p(x)$ once and use this fixed dataset for all methods. For fine-tuning, we construct a fixed offline preference dataset of 10^5 pairs, shared across all methods, by drawing two independent samples from the reference model and labeling them with the Bradley-Terry model, i.e., y_a is preferred with probability $\sigma(\log r(y_a) - \log r(y_b))$

Pre-training. All methods share the same backbone: a time-conditioned residual MLP with 1-dimensional input and output, a 128-dimensional sinusoidal time embedding, SiLU activations, and four residual blocks. The hidden width is 256 (512 for sCM). The pre-training hyperparameters are summarized in Table 6.

Fine-tuning with FestDPO. Both log-densities π_{ref} and π_θ are estimated with a Gaussian KDE (Equation (7)): $\log \hat{\pi}_\theta$ and $\log \hat{\pi}_{\text{ref}}$ are computed from N samples respectively. The loss is averaged over multiple KDE bandwidths, listed in Table 7. Averaging over bandwidths at multiple scales provides informative gradients commonly used in kernel-based distribution matching (Bińkowski et al., 2018). Under this setup, the optimum of the DPO objective with $\beta = 1$ is exactly $p^\star(x) \propto p(x)\, r(x)$. The fine-tuning hyperparameters are summarized in Table 7.

Sampling. We use 1 sampling step for Drifting and 4 steps for IMM, MeanFlow, and sCM, both during fine-tuning and at evaluation. All samples are generated with EMA weights, and each reported metric is computed from 10^5 samples per checkpoint.

Table 6: Pre-training hyperparameters.

	Drifting	IMM	MeanFlow	sCM
Training steps	300k	100k	300k	100k
Optimizer	AdamW	RAdam	Adam	AdamW
Learning rate	2×10^{-4}	1×10^{-4}	2×10^{-4}	3×10^{-5}
(β_1, β_2)	(0.9, 0.95)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.99)
Weight decay	0.01	0	0	0.01
Batch size	1024	4096	1024	16384
Gradient clip	2.0	–	–	10.0
EMA decay	0.999	0.9999	0.9999	0.999
# Parameters	0.53M	0.92M	0.92M	3.48M

Table 7: Fine-tuning hyperparameters of FestDPO.

	Drifting	IMM	MeanFlow	SCM
Fine-tuning steps	4,000	4,000	8,000	4,000
Learning rate	1×10^{-4}	1×10^{-4}	3×10^{-5}	1×10^{-4}
β	1.0	1.0	1.0	1.0
KDE bandwidths	0.02 / 0.06 / 0.2	0.02 / 0.06 / 0.2	0.01 / 0.03 / 0.1	0.02 / 0.06 / 0.2
# Samples ($M = N$)	2048	2048	8192	2048
Pairs per step	512	512	512	512
Gradient clip	1.0	1.0	1.0	1.0

E.2 Text-to-image generation

E.2.1 Datasets Details

In this section, we provide a detailed explanation of the datasets.

Pick-a-Pic v2 (Kirstain et al., 2023): Pick-a-Pic v2 collects human preferences from users of a text-to-image web application, where each sample pairs a prompt with two generated images and a preference label. It provides 851,293 pairs across 58,960 prompts, with images generated by Stable Diffusion 2.1, Dreamlike Photoreal 2.0, and Stable Diffusion XL variants (Rombach et al., 2022) using diverse classifier-free guidance scales (Ho & Salimans, 2022). For evaluation, we use all 500 prompts in the test split.

Parti-Prompts (Yu et al., 2022): Parti-Prompts (P2) is a dataset of 1,632 English prompts designed to probe text-to-image models across a broad range of capabilities. Each prompt is annotated with two labels: a *Category*, which specifies its broad subject domain, and a *Challenge*, which identifies the aspect that makes the prompt difficult to render faithfully. For evaluation, we construct a subset of 500 prompts by sampling an equal number of prompts from each category.

HPSv2 (Wu et al., 2023): HPSv2 contains approximately 798K human preference choices over 434K image pairs generated by a variety of text-to-image models, together with real images. It also provides a benchmark of 3,200 prompts evenly divided into four styles: Animation, Concept-art, Painting, and Photo. For evaluation, we sample 125 prompts from each style, resulting in 500 prompts in total.

E.2.2 Implementation details

In this section, we describe the experimental setup and implementation details of FestDPO. We apply LoRA (Hu et al., 2021) fine-tuning to the base few-step generator (SDXL-Turbo and SDXL-DMD2), using a rank of 16 and a scaling factor of 1 across all experiments, and conduct all main experiments with single-step generation (NFE $= 1$). We use the AdamW optimizer (Kingma & Ba, 2014) with a prompt batch size of 512. The learning rate is initialized at 3×10^{-4} and decayed to 1×10^{-6} following a cosine schedule over the first 300 steps, followed by a constant learning rate of 1×10^{-6}. The regularization coefficient β is fixed to 0.5 throughout all experiments. For each prompt, FestDPO constructs the KDE from $N = 12$ generated samples. Since the density is estimated in the

feature space of a pretrained encoder (Latent MAE by default), we find a simple Gaussian kernel to be sufficient, and the kernel bandwidth is tuned separately for each base model. All experiments are conducted on 8 NVIDIA RTX 3090 GPUs. Further details on the hyperparameter settings can be found in Table 8.

Table 8: Hyperparameters for FestDPO.

	SDXL-Turbo	SDXL-DMD2
KDE bandwidth h	0.15, 0.2	0.2, 0.3
β	0.5	
# of KDE samples N	12	
LoRA rank / scale	16 / 1	
Optimizer	AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$)	
Weight decay	0.01	
Initial learning rate	3×10^{-4}	
Final learning rate	1×10^{-6}	
Gradient clip norm	1.0	
Max steps	1,000	
Prompt batch size	512	

Baseline implementation details. Building on the official implementations of PSO and DrPO, we adapt both methods to our offline setting and tune their key hyperparameters as described below.

- **PSO** (Miao et al., 2025): Pairwise Sample Optimization fine-tunes timestep-distilled diffusion models by increasing the relative likelihood of preferred images over reference images, where the likelihood is approximated by the denoising loss along the diffusion trajectory. Following Miao et al. (2025), we tune the learning rate over $\{10^{-5}, 10^{-4}\}$ and the regularization coefficient β over $\{1, 10, 50, 100\}$, and select $LR = 10^{-4}$ and $\beta = 10$.
- **DrPO** (Jiang et al., 2026): DrPO aligns few-step generators by constructing a drift field from preference pairs, which attracts generated samples toward preferred samples and repels them from dispreferred ones, with a regularization term weighted by λ that keeps the model close to the reference. We tune the learning rate over $\{10^{-5}, 10^{-4}\}$ and λ over $\{0.1, 0.2, 0.3, 0.5\}$, and select $LR = 10^{-4}$ and $\lambda = 0.1$. For a fair comparison, we use the same feature encoder (Latent MAE) and kernel bandwidth as FestDPO.

E.3 Protein backbone generation

Base model and optimization targets. We use the RMF-S variant of Riemannian MeanFlow (Woo et al., 2026) as the base generator and frozen reference model. We fine-tune separate generators for two optimization targets: SS-match, adapted from Su et al. (2026), and self-consistency root mean square deviation (scRMSD). The SS-match reward favors a higher fraction of residues in β-strands. We assign secondary-structure labels using DSSP (Kabsch & Sander, 1983) and group them into helices, strands, and coils. For scRMSD, we use ProteinMPNN (Dauparas et al., 2022) to design an amino-acid sequence for each generated backbone and ESMFold (Lin et al., 2023) to predict its structure. We then compute the Cα RMSD between the generated and predicted backbones after structural alignment, favoring lower values as a proxy for structural designability.

Model checkpoints. We use the RMF-S checkpoint and its accompanying config from the released weights archive[1], together with the official Protein-RMF implementation[2]. For both inverse folding and feature extraction, we use the Cα-only ProteinMPNN checkpoint from the official repository[3]. This checkpoint uses 48 neighbors and was trained with coordinate noise of 0.20Å.

[1] https://zenodo.org/records/18582218
[2] https://github.com/dongyeop3813/Protein-RMF
[3] https://github.com/dauparas/ProteinMPNN

The ProteinMPNN encoder remains frozen during FestDPO training. For refolding, we use the `facebook/esmfold_v1` checkpoint from Hugging Face[4].

E.3.1 Preference Dataset Construction

For each optimization target, we construct a separate offline preference dataset from a pool of $8{,}192$ backbones of length 128 generated by the frozen RMF-S reference model. Pairs are formed only within the corresponding pool. The preference datasets remain fixed during fine-tuning, which requires no additional reward evaluations or preference annotations.

SS-match. We generate the pool using single-step sampling and score each backbone using the ceiling-normalized P-SEA β-strand fraction. For backbones of length 128, one additional strand residue changes the score by 1/(128-4)≈0.0081. We require a preference gap of at least 2/(128-4)≈0.0161, corresponding to two residues, to reduce sensitivity to single-residue differences at secondary-structure boundaries. The reference pool has a mean strand fraction of 0.152 with a standard deviation of 0.107.

scRMSD. We generate a separate pool using ten-step sampling to include backbones with better self-consistency than those obtained through single-step generation. Each backbone is scored by its negative scRMSD, so that higher scores indicate stronger self-consistency. The pool has a median scRMSD of 5.30Å, and 7.4% of its backbones satisfy the designability criterion of scRMSD below 2.0, Å. We require an scRMSD gap of at least 0.1Å between paired backbones.

Pair construction. We greedily form pairs satisfying the target-specific gap threshold and allow each backbone to appear in at most four pairs, limiting repeated use of individual samples. This procedure yields 8000 pairs per target. Within each pair, the backbone with the higher SS-match score or lower scRMSD is labeled preferred.

E.3.2 Implementation details

We fine-tune all 16.35 million parameters of RMF-S without adapters. Fine-tuning and evaluation use single-step generation ($K = 1$, NFE $= 2$) with a backbone length of 128 residues. We use AdamW (Loshchilov & Hutter, 2017) with a constant learning rate of 10^{-6} for $1{,}000$ updates, using 64 preference pairs per update. At each update, FestDPO constructs density estimates from 64 samples drawn from the trainable generator and 64 samples drawn from the frozen reference generator. Both sample sets are regenerated at every update. We apply a Gaussian kernel in the 128-dimensional feature space of the frozen ProteinMPNN encoder. Table 9 lists the hyperparameters and candidate values considered. Experiments use a single NVIDIA RTX 5090 GPU and take approximately 2 hours.

Table 9: Hyperparameters and candidate values for FestDPO on protein backbone generation.

Hyperparameter	SS-match	scRMSD
KDE bandwidth σ	0.5	0.05
DPO coefficient β	2000	20
Base model	RMF-S	
KDE samples per update	64 trainable / 64 reference	
Optimizer	AdamW	
AdamW (β_1, β_2)	$(0.9, 0.999)$	
Weight decay	10^{-12}	
Learning rate	10^{-6}	
Gradient clipping norm	1000	
Preference pairs per update	64	
Training updates	1,000	
Backbone length	128	

[4] `https://huggingface.co/facebook/esmfold_v1`

DrPO baseline. We use a Laplacian kernel for DrPO and consider kernel radii of $(0.02, 0.05, 0.2)$. We tune the reference loss weight λ over the range $[0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2]$, while fixing the anchor weight at 1.0. Each update uses 64 samples from the trainable generator and 64 reference samples. The remaining training settings, including the learning rate, number of updates, preference batch size, and random seeds, are identical to those used for FestDPO.

F Additional Experimental Results

We evaluate all methods on 500 prompts from each of Pick-a-Pic v2, Parti-Prompts, and HPSv2. For each prompt, we generate images with four different random seeds. In all experiments, we report the mean and standard deviation of PickScore (Kirstain et al., 2023) and Aesthetic score (Schuhmann & Beaumont, 2023) over each prompt set.

F.1 Main Results

Table 10: Main results with SDXL-Turbo on Pick-a-Pic v2, Parti-Prompts, HPSv2.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
Method	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
SDXL-Turbo	22.37 ± 0.06	6.03 ± 0.03	22.78 ± 0.04	5.69 ± 0.02	22.83 ± 0.05	6.09 ± 0.02
DrPO	22.38 ± 0.06	6.03 ± 0.03	22.78 ± 0.04	5.69 ± 0.02	22.85 ± 0.05	6.09 ± 0.02
PSO	22.46 ± 0.06	6.06 ± 0.02	22.86 ± 0.04	**5.72 ± 0.02**	**23.22 ± 0.05**	6.05 ± 0.02
FestDPO	**22.60 ± 0.07**	**6.07 ± 0.03**	**22.92 ± 0.05**	**5.72 ± 0.02**	23.18 ± 0.05	**6.14 ± 0.02**

Table 11: Win-rate comparison with SDXL-Turbo on Pick-a-Pic v2, Parti-Prompts, HPSv2.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
Method	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
DrPO	51.65 ± 1.21	49.40 ± 1.27	50.20 ± 1.00	53.65 ± 1.02	54.50 ± 1.13	51.30 ± 1.09
PSO	65.60 ± 1.29	56.20 ± 1.35	64.70 ± 1.09	57.05 ± 1.08	77.70 ± 1.03	44.35 ± 1.33
FestDPO	**73.55 ± 1.24**	**59.45 ± 1.27**	**67.65 ± 1.08**	**63.05 ± 1.04**	**82.25 ± 0.96**	**61.00 ± 1.18**

Table 12: Main results with SDXL-DMD2 on Pick-a-Pic v2, Parti-Prompts, HPSv2.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
Method	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
SDXL-DMD2	21.06 ± 1.46	5.37 ± 0.68	21.65 ± 1.17	5.25 ± 0.57	21.45 ± 1.28	5.58 ± 0.67
DrPO	21.17 ± 1.47	5.43 ± 0.70	21.73 ± 1.19	5.31 ± 0.57	21.61 ± 1.31	5.63 ± 0.68
PSO	21.58 ± 1.54	5.66 ± 0.70	22.06 ± 1.35	5.51 ± 0.64	22.05 ± 1.38	5.90 ± 0.62
FestDPO	**21.98 ± 1.50**	**5.87 ± 0.57**	**22.43 ± 1.16**	**5.65 ± 0.52**	**22.42 ± 1.32**	**5.99 ± 0.59**

Table 13: Win-rate comparison with SDXL-DMD2 on Pick-a-Pic v2, Parti-Prompts, HPSv2.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
Method	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
DrPO	67.00 ± 1.67	61.55 ± 1.98	63.15 ± 2.29	63.00 ± 1.99	72.00 ± 1.21	63.35 ± 1.19
PSO	77.00 ± 0.42	76.15 ± 0.43	73.85 ± 0.44	73.70 ± 0.44	78.90 ± 0.41	78.60 ± 0.41
FestDPO	**88.30 ± 0.72**	**87.25 ± 0.75**	**86.55 ± 0.76**	**84.60 ± 0.81**	**90.90 ± 0.64**	**85.05 ± 0.80**

F.2 Number of KDE samples

Since FestDPO employs nonparametric density estimation, we first study the effect of the number of samples N used to construct the KDE. Beyond HPSv2 reported in Table 1, we additionally evaluate on Pick-a-Pic v2 and Parti-Prompts. As shown in Table 14, increasing N from 8 to 24 consistently improves PickScore across all three benchmarks, while Aesthetic scores remain approximately stable. These results indicate that FestDPO is robust to the choice of N, with even a small number of samples yielding substantial gains over the pretrained model.

Table 14: Sensitivity analysis on the number of KDE samples with SDXL-Turbo.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
# of KDE samples	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
SDXL-Turbo	22.37 ± 1.46	6.03 ± 0.61	22.78 ± 1.20	5.69 ± 0.57	22.83 ± 1.29	6.09 ± 0.69
$N = 8$	22.53 ± 1.51	6.06 ± 0.62	22.91 ± 1.24	5.74 ± 0.58	23.06 ± 1.32	6.13 ± 0.68
$N = 12$	22.60 ± 1.52	6.07 ± 0.62	22.92 ± 1.25	5.73 ± 0.59	23.13 ± 1.35	6.11 ± 0.66
$N = 24$	22.61 ± 1.50	6.02 ± 0.60	22.94 ± 1.24	5.72 ± 0.57	23.14 ± 1.37	6.07 ± 0.63

F.3 Feature encoder for KDE

We next conduct an ablation on the feature encoder used for kernel density estimation. Table 15 compares three pretrained encoders: Latent MAE, CLIP, and DINOv2. FestDPO achieves comparable performance across the three pretrained encoders, demonstrating that its effectiveness is robust to the choice of pretrained encoder. We adopt Latent MAE as the default encoder, as it attains consistently strong results on both PickScore and Aesthetic score, and apply the same encoder to DrPO for a fair comparison.

Table 15: Ablation on the feature encoder with SDXL-Turbo.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
Feature encoder	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
SDXL-Turbo	22.37 ± 1.46	6.03 ± 0.61	22.78 ± 1.20	5.69 ± 0.57	22.83 ± 1.29	6.09 ± 0.69
Latent MAE	22.60 ± 1.52	6.07 ± 0.62	22.92 ± 1.25	5.73 ± 0.59	23.13 ± 1.35	6.11 ± 0.66
CLIP	22.66 ± 1.48	5.99 ± 0.57	22.91 ± 1.22	5.72 ± 0.56	23.17 ± 1.35	6.08 ± 0.64
DINOv2	22.54 ± 1.49	6.02 ± 0.61	22.92 ± 1.22	5.71 ± 0.59	23.07 ± 1.34	6.08 ± 0.65

Number of function evaluations (NFE). While our main experiments are reported with 1-step generation, we further examine whether FestDPO remains effective in the few-step regime. As shown in Table 16, FestDPO with two function evaluations (NFE $= 2$) achieves performance comparable to the 1-step setting, with consistent improvements in PickScore across all three benchmarks. This suggests that FestDPO benefits from the improved sample quality afforded by additional function evaluations.

Table 16: Performance of FestDPO with SDXL-Turbo under different NFE.

	Pick-a-Pic v2		Parti-Prompts		HPSv2	
NFE	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑	PickScore ↑	Aesthetic ↑
1	22.60 ± 1.52	6.07 ± 0.62	22.92 ± 1.25	5.73 ± 0.59	23.13 ± 1.35	6.11 ± 0.66
2	22.63 ± 1.48	5.98 ± 0.59	23.07 ± 1.22	5.67 ± 0.54	23.14 ± 1.33	6.06 ± 0.67

G QUALITATIVE RESULTS

In this section, we present qualitative comparisons of FestDPO with PSO and DrPO for text-to-image generation across three prompts sets: Pick-a-Pic v2, PartiPrompts, and HPS v2. All methods use SDXL-Turbo as the base model.

G.1 PICK-A-PIC V2

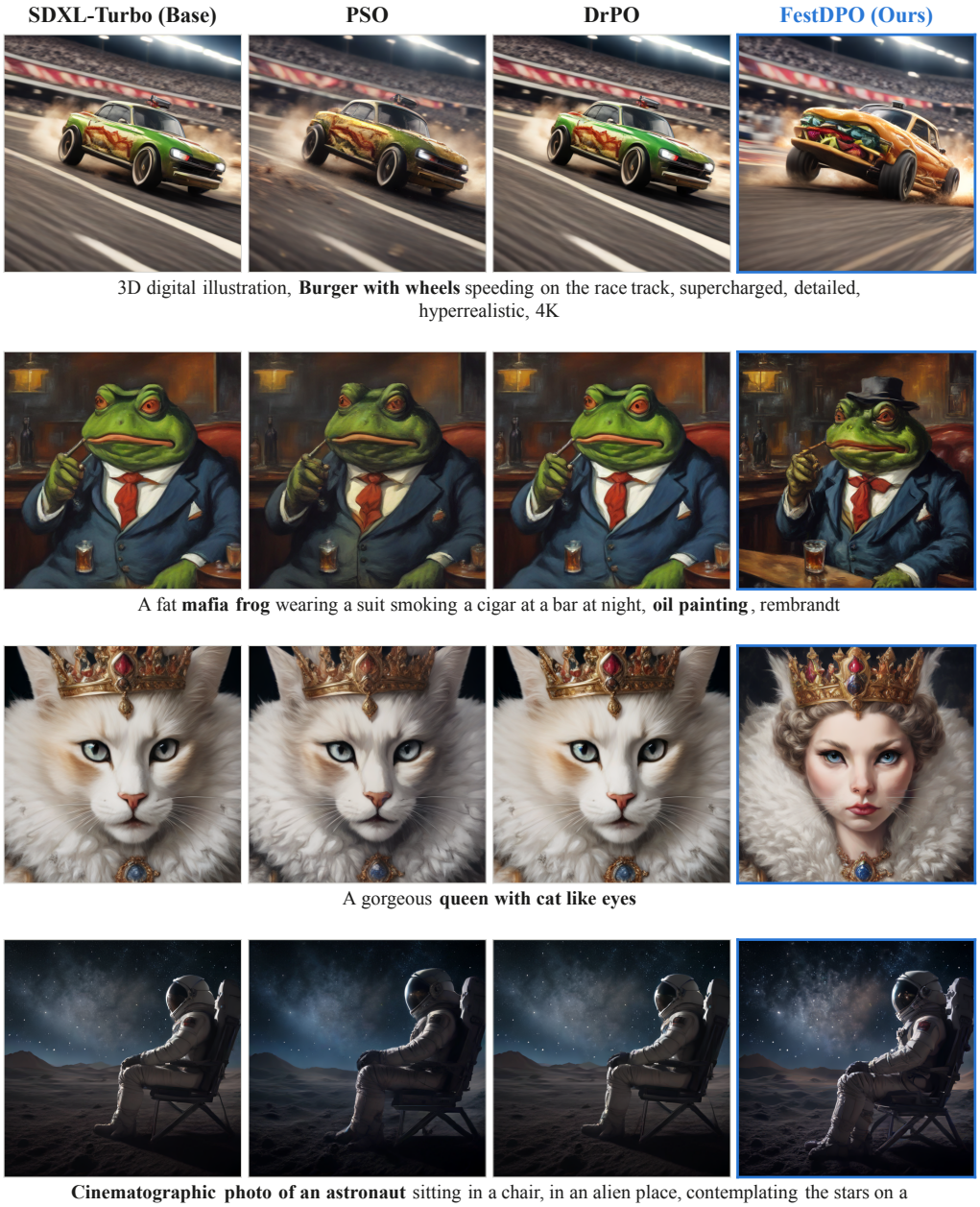

Figure 8: Qualitative comparison of FestDPO with PSO and DrPO on Pick-a-Pic V2.

G.2 PARTI-PROMPTS

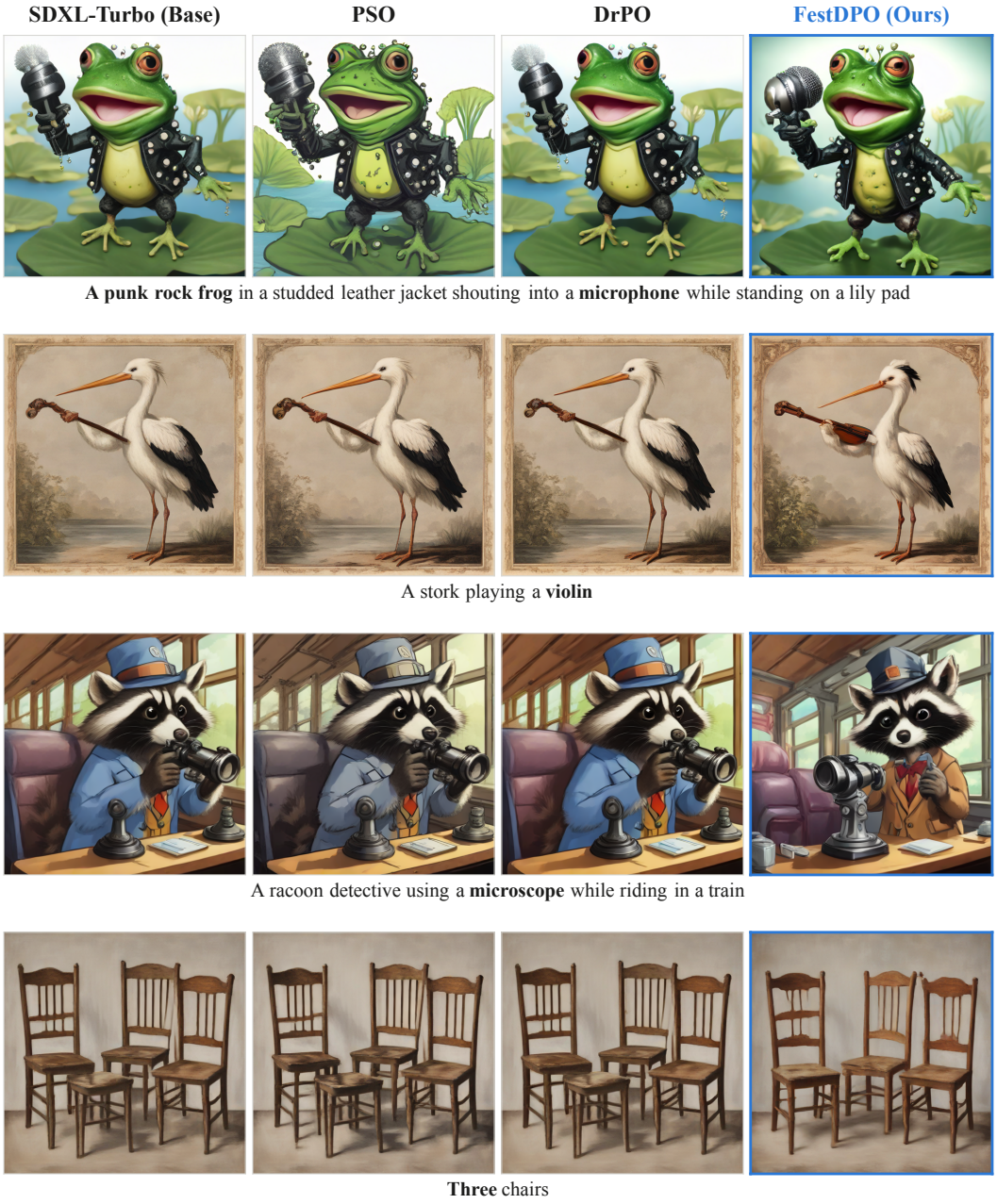

Figure 9: Qualitative comparison of FestDPO with PSO and DrPO on Parti-Prompts.

G.3 HPSv2

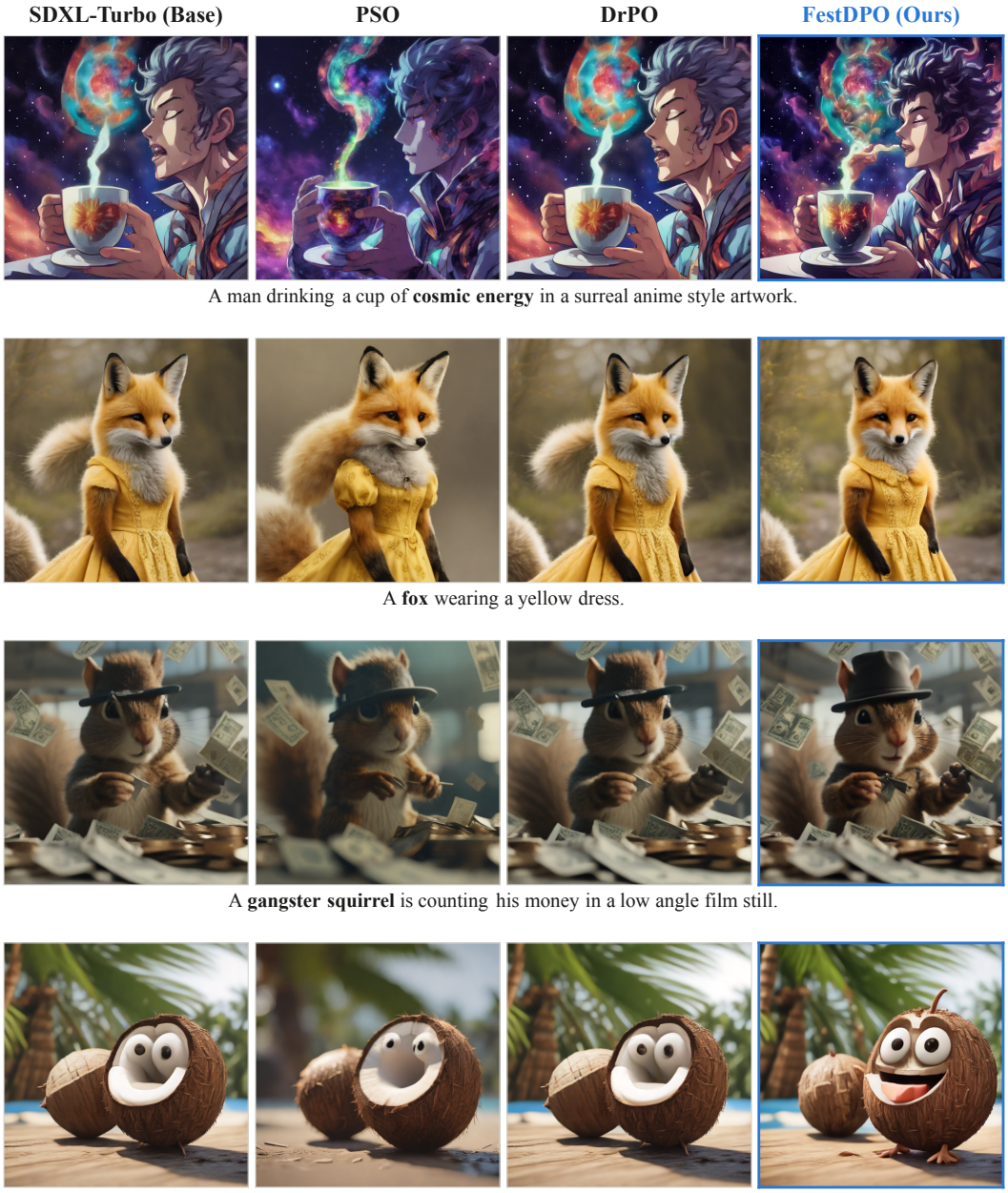

Figure 10: Qualitative comparison of FestDPO with PSO and DrPO on HPSv2.

H HUMAN EVALUATION

Evaluation protocol. We conduct the human evaluation using the annotation interface shown in Figure 11. The evaluation includes 80 prompts randomly drawn from the three prompt sets used in our experiments: 29 from Pick-a-Pic v2, 20 from Parti-Prompts, and 31 from HPSv2. For each prompt, annotators are presented with four images generated by SDXL-Turbo, PSO, DrPO, and FestDPO and rate each image on a 1–5 scale for prompt alignment and aesthetics. To prevent annotators from identifying which method generates each image, all images are anonymized. For each annotator, we independently randomize both the prompt order and the arrangement of the four images.

Participants. We recruited 15 annotators from collaborators, lab members, and friends of the authors for the human evaluation. In total, we collected 9,600 ratings (15 annotators $\times$ 320 images $\times$ 2 evaluation criteria). The median completion time per annotator is about 36 minutes.

Results and reliability. As shown in Table 17, FestDPO achieves the highest mean ratings for both prompt alignment and aesthetics across all three prompt sets. The improvements over all baselines are statistically significant ($p < 0.001$, Wilcoxon signed-rank test (Wilcoxon, 1945)). The pairwise comparison in Figure 12 further shows consistently higher win rates for FestDPO against all baselines on both criteria. Specifically, FestDPO achieves win rates of 76–84% for prompt alignment and 72–81% for aesthetics against the three baselines. Moreover, all 15 annotators individually assign FestDPO the highest mean rating on both criteria. The mean pairwise Spearman correlation between annotators is 0.35 for prompt alignment and 0.32 for aesthetics. Finally, the mean ratings differ by at most 0.06 across the four display positions, suggesting that the results are not substantially affected by image placement.

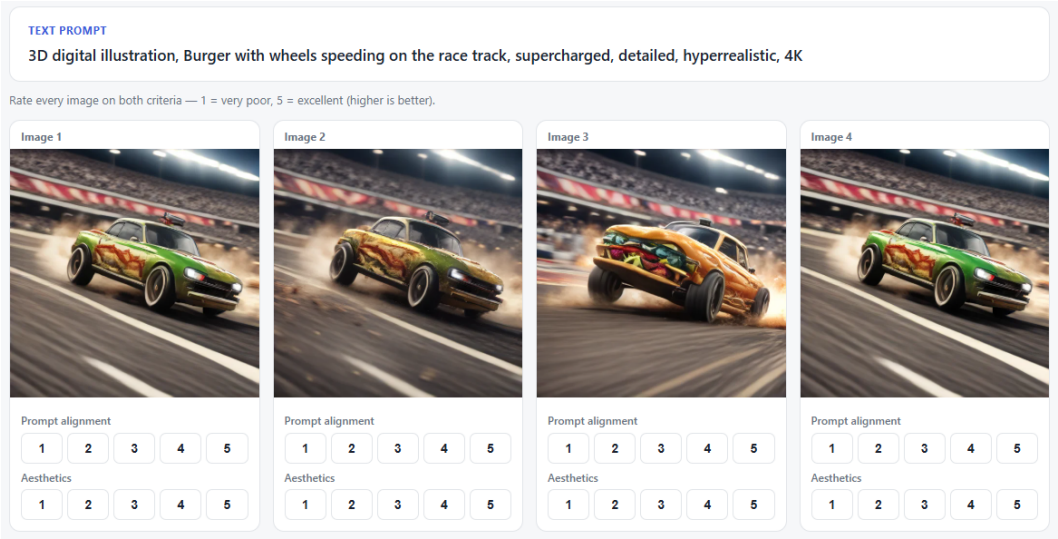

Figure 11: Screenshot of the annotation interface used for human evaluation. For each text prompt, four images generated by different methods are displayed side by side in randomized order. Annotators rate each image on a 1–5 scale for prompt alignment and aesthetics.

Table 17: Human evaluation results on Pick-a-Pic v2, Parti-Prompts, HPSv2.

	Overall		Pick-a-Pic v2		Parti-Prompts		HPSv2	
Method	Align. ↑	Aes. ↑	Align. ↑	Aes. ↑	Align. ↑	Aes. ↑	Align. ↑	Aes. ↑
SDXL-Turbo	3.683	3.149	3.756	3.248	3.497	3.080	3.735	3.101
PSO	3.767	3.286	3.731	3.156	3.607	3.103	3.903	3.525
DrPO	3.696	3.158	3.759	3.234	3.520	3.057	3.751	3.151
FestDPO (ours)	**4.199**	**3.860**	**4.329**	**3.936**	**4.137**	**3.793**	**4.118**	**3.832**

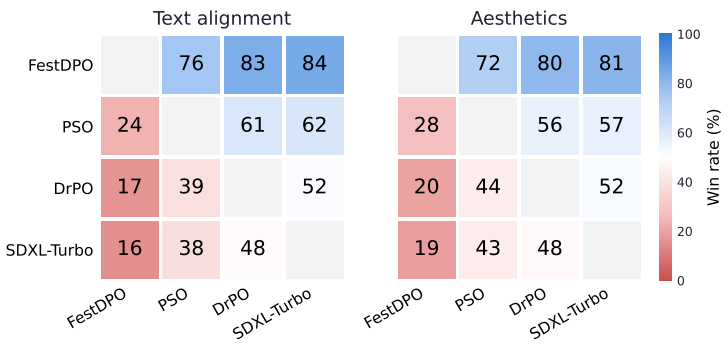

Figure 12: Pairwise win rates (%) in the human evaluation for prompt alignment and aesthetics. Each entry indicates the percentage of comparisons in which the method on the y-axis is preferred over the method on the x-axis.