Curriculum Direct Preference Optimization for Diffusion and Consistency Models
Supplementary Material

6. Detailed Preliminaries

Diffusion models. Diffusion models [11, 20, 26, 28, 51, 61, 64, 65, 71] are a class of generative models trained to reverse a process that progressively inserts Gaussian noise across T steps into the original data samples, transforming them into standard Gaussian noise. Formally, the forward process defined by:

$$x_t = \alpha_t x_0 + \sigma_t \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{9}$$

transforms the original samples $x_0 \sim p(x_0)$ into noisy versions x_t, following the noise schedule implied by the time-dependent predefined functions $(\alpha_t)_{t=1}^T$ and $(\sigma_t)_{t=1}^T$. The model is trained to estimate the noise ϵ added in the forward step defined in Eq. (9), by minimizing the following objective:

$$\mathcal{L}_{simple} = \mathbb{E}_{t\sim\mathcal{U}(1,T),\epsilon\sim\mathcal{N}(0,\mathbf{I}),x_0\sim p(x_0)} \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 . \tag{10}$$

The generation process involves denoising, starting from a sample of standard Gaussian noise, denoted as $x_T \sim \mathcal{N}(0, \mathbf{I})$. It then follows the transitions outlined in Eq. (11) to produce novel samples:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sigma^2_{t|t-1}\frac{\sigma^2_{t-1}}{\sigma^2_t}\right), \tag{11}$$

with $\mu_\theta(x_t, t) = \frac{1}{\alpha_{t|t-1}}\left(x_t - \frac{\epsilon_\theta(x_t,t)\sigma^2_{t|t-1}}{\sigma_t}\right)$, where $\sigma^2_{t|t-1} = \sigma^2_t - \alpha^2_{t|t-1}\sigma^2_{t-1}$ and $\alpha_{t|t-1} = \frac{\alpha_t}{\alpha_{t-1}}$.

The forward process can also be defined in a continuous time manner [65], as a stochastic differential equation (SDE):

$$\mathrm{d}x_t = f(t)x_t\mathrm{d}t + g(t)\mathrm{d}\omega_t, t \in [0, T], \tag{12}$$

where, given the notations from Eq. (9), we can write $f(t) = \frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}$ and $g^2(t) = \frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t} - 2\frac{\mathrm{d}\log\alpha(t)}{\mathrm{d}t}\cdot\sigma^2(t)$, and ω_t is the standard Brownian motion.

Furthermore, the diffusion process described by the SDE from Eq. (12) can be reversed by another diffusion process given by a reverse-time SDE [2, 65]. In addition to this, Song et al. [65] showed that the reverse SDE has a corresponding ordinary differential equation (ODE), called Probability flow-ODE (PF-ODE), with the following form:

$$\mathrm{d}x_t = f(t)x_t\mathrm{d}t + \frac{g^2(t)}{2\sigma(t)}\epsilon_\theta(x_t, t). \tag{13}$$

Consistency models. Consistency models [36, 63, 66] are a new class of generative models. These models operate on the idea of training a model to associate each point along a trajectory of the PF-ODE (Eq. (13)) to the trajectory's initial point, which corresponds to the denoised sample. Such models can either be trained from scratch or through distillation from a pre-trained diffusion model. In our study, we employ the distillation method, so we next detail this approach.

Given a solution trajectory $\{x_t\}_{t\in[\delta,T]}$ of the PF-ODE defined in Eq. (13), where $\delta \to 0$, the training of a consistency model $f_\phi(x_t, t)$ involves enforcing the self-consistency property across this trajectory, such that, $\forall t, t' \in [\delta, T]$, the condition $f_\phi(x_t, t) = f_\phi(x_{t'}, t')$ holds. The loss function designed to achieve this self-consistency is described as follows:

$$\mathcal{L}_{\text{CD}}(\phi) = d(f_\phi(x_{t_{n+1}}, t_{n+1}), f_{\phi^-}(\hat{x}^\theta_{t_n}, t_n)), \tag{14}$$

where d is a distance metric, $n \sim \mathcal{U}(1, N)$, N is the discretization length of the interval $[0, T]$, ϕ are the trainable parameters of the consistency model and ϕ^- is a running average of ϕ. The term $\hat{x}^\theta_{t_n}$ represents a one-step denoised version of $x_{t_{n+1}}$, obtained by applying an ODE solver on the PF-ODE. The solver operates using a pre-trained diffusion model, $\epsilon_\theta(x_{t_n}, t_n)$.

Direct Preference Optimization (DPO). Training pipelines based on Reinforcement Learning with Human Feedback (RLHF) [79] have been highly successful in aligning Large Language Models to human preferences. These pipelines feature an initial phase where a reward model is trained using examples ranked by humans, followed by a reinforcement learning phase where the policy model is fine-tuned to align with the learned reward model. In this context, Rafailov et al. [48] introduced DPO as an alternative to the previous pipeline, which bypasses the training of the reward model and directly optimizes the policy model using the ranked examples.

The training dataset contains triplets of the form (c, x^w_0, x^l_0), where x^w_0 denotes the favored sample, x^l_0 the unfavored one and c is a condition used to generate both samples. RLHF trains a reward model by maximizing the likelihood $p(x^w_0 \prec x^l_0|c)$[1], which, under the Bradley-Terry (BT) model, has the following form:

$$p_{\text{BT}}(x^w_0 \prec x^l_0|c) = \sigma(r_\varphi(x^w_0, c) - r_\varphi(x^l_0, c)), \tag{15}$$

where σ denotes the sigmoid function and r_φ is the reward model parameterized by the trainable parameters φ. The training objective for the reward model is the negative log-likelihood:

$$\mathcal{L}_{\text{BT}} = -\mathbb{E}_{x^w_0, x^l_0, c}\left[\log\sigma(r_\varphi(x^w_0, c) - r_\varphi(x^l_0, c))\right]. \tag{16}$$

After training the reward model $r_\varphi(x_0, c)$, RLHF optimizes a conditional generative model $p_\theta(x_0|c)$ to maximize the reward $r_\varphi(x_0, c)$ and, at the same time, controls the deviance from a reference model $p_{\text{ref}}(x_0, c)$ through a Kullback–Leibler (KL) divergence term:

$$\max_\theta \mathbb{E}_{c,x_0\sim p_\theta(x_0|c)}\left[r_\varphi(x_0, c) - \beta\text{KL}(p_\theta(x_0|c), p_{\text{ref}}(x_0|c))\right], \tag{17}$$

[1] $a \prec b$ denotes that a precedes b in the ranking implied by the reward model.

Algorithm 2: Curriculum DPO (for diffusion models)

Input: $\{(x_{0,i}, c)\}_{i=1}^{M}$ - the training samples, $r_\varphi(x_0, c)$ - the reward model which can be conditioned on c, B - the number of batches for splitting the set of pairs, α_t, σ_t - the parameters of the noise schedule, T - the last time step of diffusion, β - DPO hyperparameter to control the divergence from the initial pre-trained state, σ - the sigmoid function, η - the learning rate, $\{H_k\}_{k=1}^{B}$ - the number of training iterations after including the k-th batch.

Output: θ - the trained weights of the generative model.

1 $\hat{X} \leftarrow \{(x_{0,i}, c) | r_\varphi(x_{0,i}, c) \leq r_\varphi(x_{0,i-1}, c), i = \{2, 3, ..., M\}\}$; ◁ sort the samples in descending order of the rewards

2 $S \leftarrow \left\{(x_{0,i}, x_{0,j}, c) | i, j \in \{1, \dots M\}; i < j; x_{0,i}, x_{0,j} \in \hat{X}, r_\varphi(x_{0,i}, c) > r_\varphi(x^l_{0,j}, c)\right\}$; ◁ create pairs of examples using the order from $\hat{X}$

3 $\mathrm{L}_k \leftarrow \left\{\frac{(M-1)\cdot(B-k)}{B}\right\}_{k=1}^{B}$; ◁ the minimum preference limits of the batches

4 $\mathrm{R}_k \leftarrow \left\{\frac{(M-1)\cdot(B-(k-1))}{B}\right\}_{k=1}^{B}$; ◁ the maximum preference limits of the batches

5 $S_k \leftarrow \left\{(x^w_0, x^l_0, c) | (x^w_0, x^l_0) = (x_{0,i}, x_{0,j}); \mathrm{L}_k < j - i \leq \mathrm{R}_k; (x_{0,i}, x_{0,j}, c) \in S\right\}_{k=1}^{B}$; ◁ the batches of increasingly difficult pairs

6 $P \leftarrow \emptyset$; ◁ current training set

7 **foreach** $k \in \{1, \dots, B\}$ **do**

8 $\quad P \leftarrow P \cup S_k$; ◁ include a new batch in the training

9 $\quad$ **foreach** $i \in \{1, \dots, H_k\}$ **do**

10 $\quad\quad (x^w_0, x^l_0, c) \sim \mathcal{U}(P)$; $t \sim \mathcal{U}\{1, \dots, T\}$; $\epsilon^w, \epsilon^l \sim \mathcal{N}(0, \mathbf{I})$;

11 $\quad\quad x^w_t \leftarrow \alpha_t x^w_0 + \sigma_t \epsilon^w$; ◁ forward process

12 $\quad\quad x^l_t \leftarrow \alpha_t x^l_0 + \sigma_t \epsilon^l$; ◁ forward process

13 $\quad\quad \mathcal{L}_{\text{Diff-DPO}}(\theta) \leftarrow -\Big[\log\sigma\Big(-\beta T\Big(\left(\|\epsilon^w - \|^2_2 - \|\epsilon^w - \epsilon^w_{ref}(x^w_t, t, c)\|^2_2\right) - \left(\|\epsilon^l - \epsilon^l_\theta(x^l_t, t, c)\|^2_2 - \|\epsilon^l - \epsilon^l_{ref}(x^l_t, t, c)\|^2_2\right)\Big)\Big)\Big]$; ◁ DPO loss

14 $\quad\quad \theta \leftarrow \theta - \eta \frac{\partial \mathcal{L}_{\text{Diff-DPO}}}{\partial \theta}$; ◁ update the weights

15 **return** θ

where β controls the importance of the divergence term.

To derive the DPO objective, Rafailov et al. [48] write the optimal policy model p^*_θ of Eq. (17) as a function of the reward and reference model, as shown in prior works [45, 46]:

$$p^*_\theta(x_0|c) = \frac{p_{\text{ref}}(x_0|c) \cdot \exp\left(\frac{r(x_0,c)}{\beta}\right)}{Z(c)}, \tag{18}$$

where $Z(c) = \sum_{x_0} p_{\text{ref}}(x_0|c) \cdot \exp\left(\frac{r(x_0,c)}{\beta}\right)$ is a normalization constant. Further, from Eq. (18), Rafailov et al. [48] rewrite the reward as:

$$r(x_0, c) = \beta\left(\log\frac{p^*_\theta(x_0|c)}{p_{\text{ref}}(x_0|c)} + \log Z(c)\right). \tag{19}$$

Finally, the DPO objective is obtained after replacing the reward in Eq. (16) with the form from Eq. (19):

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{x^w_0, x^l_0, c}\left[\log\sigma\left(\beta\left(\log\frac{p_\theta(x^w_0|c)}{p_{\text{ref}}(x^w_0|c)} - \log\frac{p_\theta(x^l_0|c)}{p_{\text{ref}}(x^l_0|c)}\right)\right)\right], \tag{20}$$

To grasp the intuition behind $\mathcal{L}_{\text{DPO}}$, we can analyze its gradient with respect to θ:

$$\frac{\partial \mathcal{L}_{\text{DPO}}(\theta)}{\partial \theta} = -\beta\mathbb{E}_{x^w_0, x^l_0, c}\Bigg[\sigma\left(\hat{r}_\theta(x^l_0, c) - \hat{r}_\theta(x^w_0, c)\right) \cdot \left(\frac{\partial \log p_\theta(x^w_0|c)}{\partial \theta} - \frac{\partial \log p_\theta(x^l_0|c)}{\partial \theta}\right)\Bigg], \tag{21}$$

with $\hat{r}_\theta(x_0, c) = \beta \cdot \log\frac{p_\theta(x_0|c)}{p_{\text{ref}}(x_0|c)}$. By analyzing Eq. (21), as discussed in [48], it is evident that the DPO objective enhances the likelihood of favored examples, while diminishing it for the unfavored ones. The magnitude of the update is proportional to the error in $\hat{r}_\theta$. Here, the term "error" refers to the degree to which $\hat{r}_\theta$ incorrectly prioritizes the sample x^l_0.

7. Curriculum DPO for Diffusion Models

We formally present the application of Curriculum DPO to diffusion models in Algorithm 2. The initial steps 1-9, which outline the curriculum strategy, are identical with those used

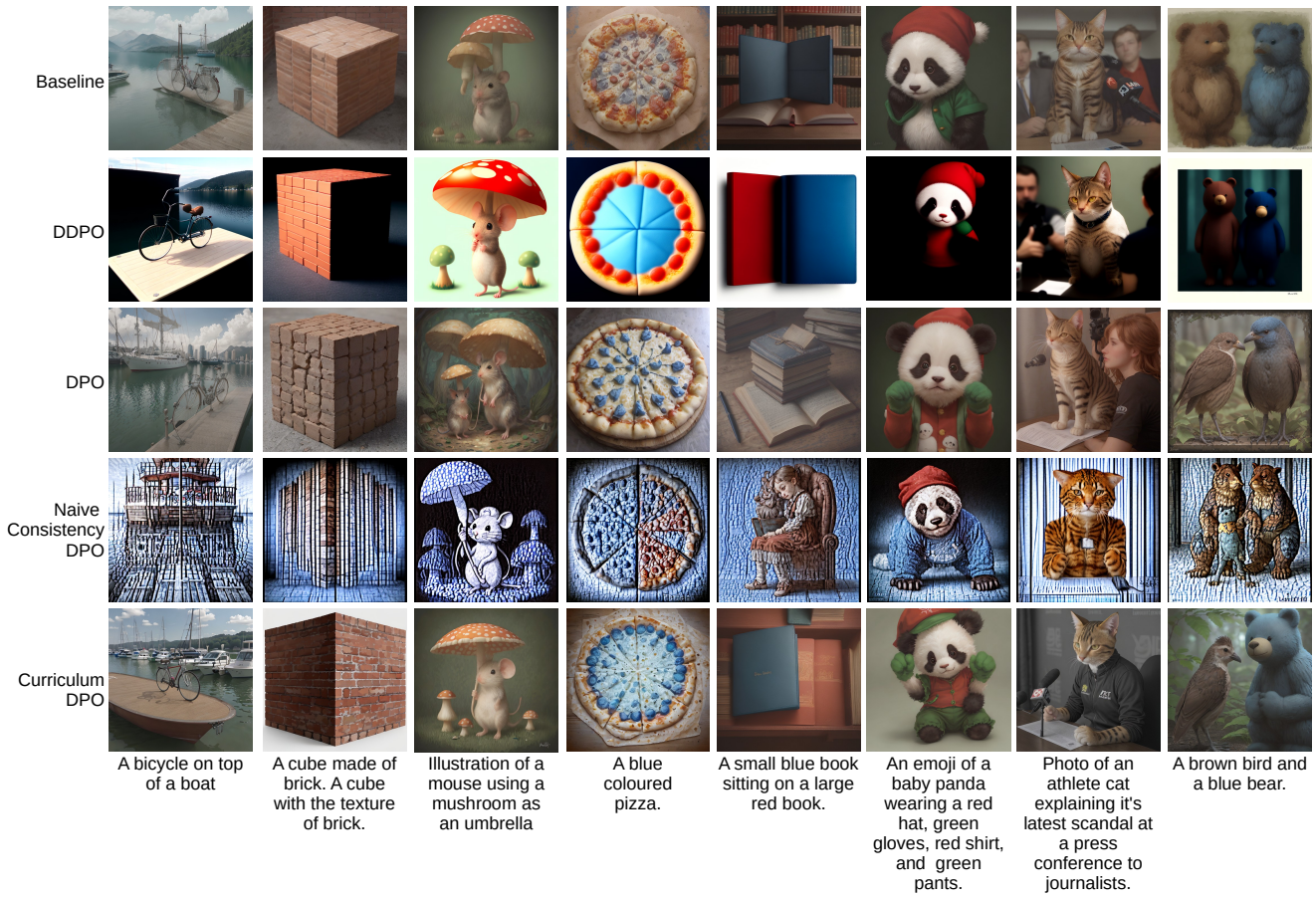


Figure 4. Qualitative results before and after fine-tuning for the text alignment task on DrawBench. The fine-tuning methods are: DDPO, DPO, Naive DPO and Curriculum DPO. Best viewed in color.

in the implementation for consistency models described in Algorithm 1. Steps 10-14 are changed to include the forward process for the preferred and less preferred samples, along with the Diffusion-DPO loss defined in Eq. (5).

8. Importance of Consistency-DPO

Our work makes two contributions: Curriculum DPO and Consistency-DPO. While the novelty and importance of Curriculum DPO is more obvious, we consider that the significance of Consistency-DPO is not immediately observable. To this end, it is important to note that the Diffusion-DPO [72] approach cannot be directly applied to consistency models. The most direct modification is to substitute the noise estimation in the Diffusion-DPO objective with the consistency distillation loss used in consistency models. However, applying this modification directly breaks the consistency property required by these models and leads to poor results, as shown in Figure 4 and further discussed in Section 10.

We found two solutions for this problem. The first is to reintegrate the consistency distillation for both preferred and non-preferred samples as separate components within the optimization function, in addition to the Consistency-DPO loss (Eq. (6)). This method, however, introduces the need for additional hyperparameters to balance these terms, which represents a significant drawback because it requires extensive hyperparameter tuning.

The second solution, which we ultimately adopt in our study, is to ensure the initial estimation for the ODE's starting point (the target in the consistency distillation loss) is a sample of the consistency model that undergoes fine-tuning. We accomplish this by replacing the Exponential Moving Average (EMA) model, that is typically used to get this estimation, with the pre-trained model from which we begin the fine-tuning process. This approach maintains the integrity of the consistency model's properties throughout the training.

We thus conclude that adapting DPO to consistency models is not trivial, since the adaptation requires a deep understanding of the framework and strong knowledge about the use of gradients.

9. More Quantitative Results

Results on Pick-a-Pic. We report additional results for Stable Diffusion on 150,000 image pairs from Pick-a-Pic (D_3) in Table 4. In this scenario, the dataset already includes pairs

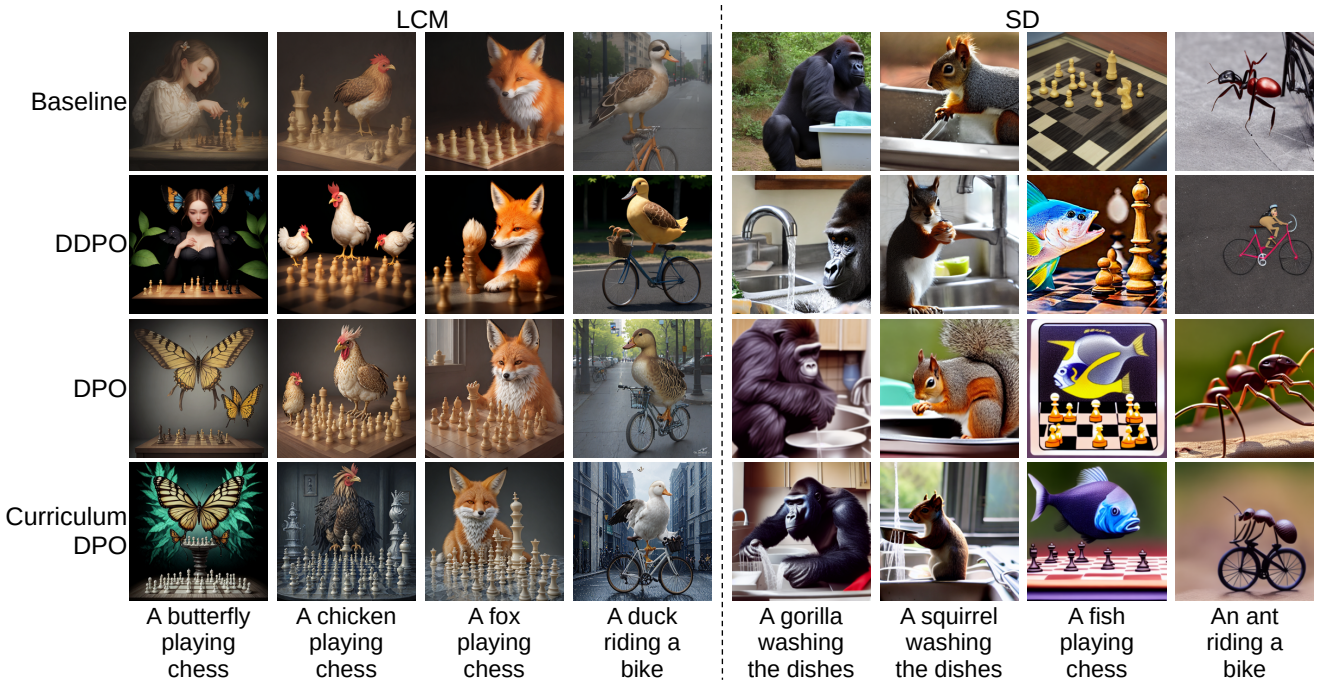


Figure 5. Qualitative results after fine-tuning with HPSv2 as the reward model (human preference). The fine-tuning alternatives are: DDPO, DPO and Curriculum DPO. Best viewed in color.

Fine-Tuning Strategy	Text Alignment	Aesthetics	Human Preference
-	0.5246	5.6675	0.2673
DDPO	0.5317	5.6764	0.2717
DPO	0.5328	5.7593	0.2725
Curriculum DPO (ours)	**0.5413**	**5.7998**	**0.2783**

Table 4. Text alignment, aesthetic and human preference scores on Pick-a-Pic (D_3), obtained by the baseline (pre-trained) Stable Diffusion model versus the three fine-tuning strategies: DDPO, DPO and Curriculum DPO. The best scores are highlighted in bold.

Fine-Tuning Strategy	LLaVA	Phi-3
-	0.6804	0.6804
DDPO	0.7629	0.7602
DPO	0.7614	0.7643
Curriculum DPO (ours)	**0.7703**	**0.7736**

Table 5. Text alignment results on dataset D_1 by using two reward models (LLaVA and Phi-3) for DDPO, DPO and Curriculum DPO applied on Stable Diffusion. The best scores are highlighted in bold.

of winning and losing images, so we only apply the reward models for the ranking described in Figure 1. The results reported on D_3 are consistent with those reported on D_1 and D_2, further highlighting the importance of curriculum learning.

Results with different reward models. In Table 5, we compare text alignment results for two alternative reward models: LLaVA [33] and Phi-3 [1]. During training, we use a reward model to extract image descriptions and then measure their similarity to the original prompts to produce winning and losing image pairs. The same similarity scores also determine the ranking used by Curriculum DPO. These experiments further confirm the superiority of Curriculum DPO over DPO and DDPO, regardless of the employed reward model.

10. More Qualitative Results

In Figures 5 and 6, we present qualitative results after fine-tuning the models with HPSv2 and LAION Aesthetics Predictor as reward models on D_1, respectively. Fine-tuning for human preference (Figure 5) generally results in generating images with more details for the LCM model. Curriculum DPO, in particular, produces better aesthetics for both foreground objects and the background. In contrast, the SD results show a better alignment with the text prompt, Curriculum DPO being the only method that generates the ant on a bike displayed in the last column. Fine-tuning for improving the visual appeal (Figure 6) returns in general, as expected, better aesthetics for the animals. However, Curriculum DPO returns several examples that look better, *e.g.* the camel in the sixth column and the dog in the third column.

In Figure 4, we show qualitative results when fine-tuning the model for text alignment on D_2. In addition to the baseline, DPO, DDPO and Curriculum DPO results, we also

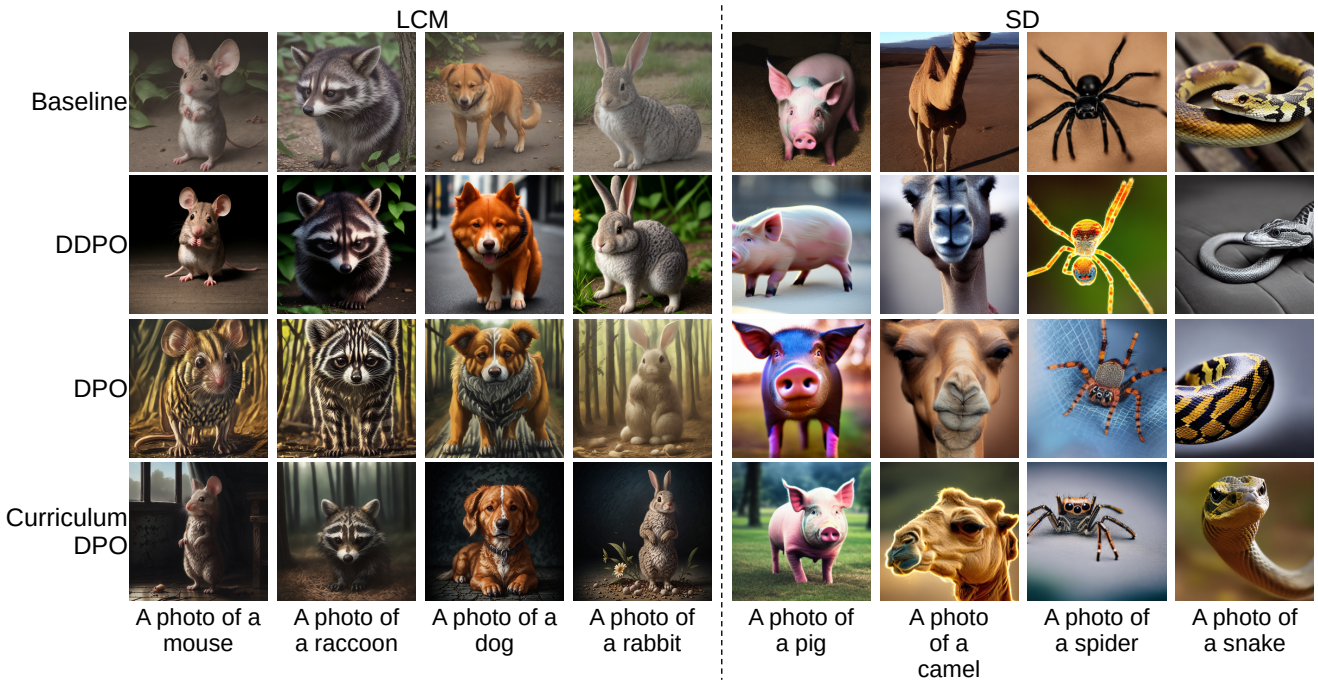
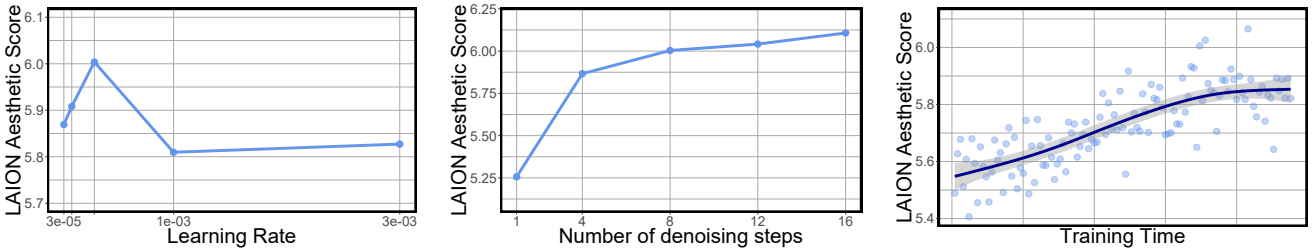


Figure 6. Qualitative results after fine-tuning with the LAION Aesthetics Predictor as the reward model. The fine-tuning alternatives are: DDPO, DPO and Curriculum DPO. Best viewed in color.



(a) Varying the learning rate for Curriculum DPO on LCM.

(b) Varying the number of LCM generation steps for Curriculum DPO.

(c) The progression of the aesthetic reward function during training with the DDPO method.

Figure 7. Additional ablation results for Curriculum DPO applied on LCM are depicted in Figure 7a and 7b. In Figure 7c, we show the evolution of the reward score when training Stable Diffusion with DDPO. All experiments are carried out on DrawBench.

include the images generated by a naive implementation of Consistency-DPO. This implementation refers to the most direct adaptation of Diffusion-DPO to consistency models. More precisely, we substitute the noise estimation in the Diffusion-DPO objective with the consistency distillation loss used in consistency models. However, applying this modification directly breaks the consistency property required by these models and leads to bad results, as illustrated in the 4th row of Figure 4.

11. Additional Ablations

Aside from the ablation results presented in Figure 3 and Table 3 from the main article, there are a few other hyperparameters involved in the training process, such as the learning rate and the number of steps used in the multi-step generation of LCM. We performed additional ablation studies on the learning rate and the number of steps used by LCM, on the DrawBench dataset, using $M = 5$ generated images per prompt. The results presented in Figures 7a and 7b demonstrate that, regardless of the chosen values, the outcomes consistently surpass the baseline (see Table 6). We emphasize that we did not try to tune these hyperparameters for Curriculum DPO to avoid overfitting in hyperparameter space. Moreover, we underline that some apparent hyperparameters directly depend on already ablated hyperparameters. For example, the hyperparameter B (ablated in Figure 3c) is the only one that influences the minimum/maximum preference limits L_k and R_k, which are computed in steps 3 and 4 of both Algorithm 1 and Algorithm 2. Note that the equations for L_k and R_k generate equally-sized batches, so the lim-

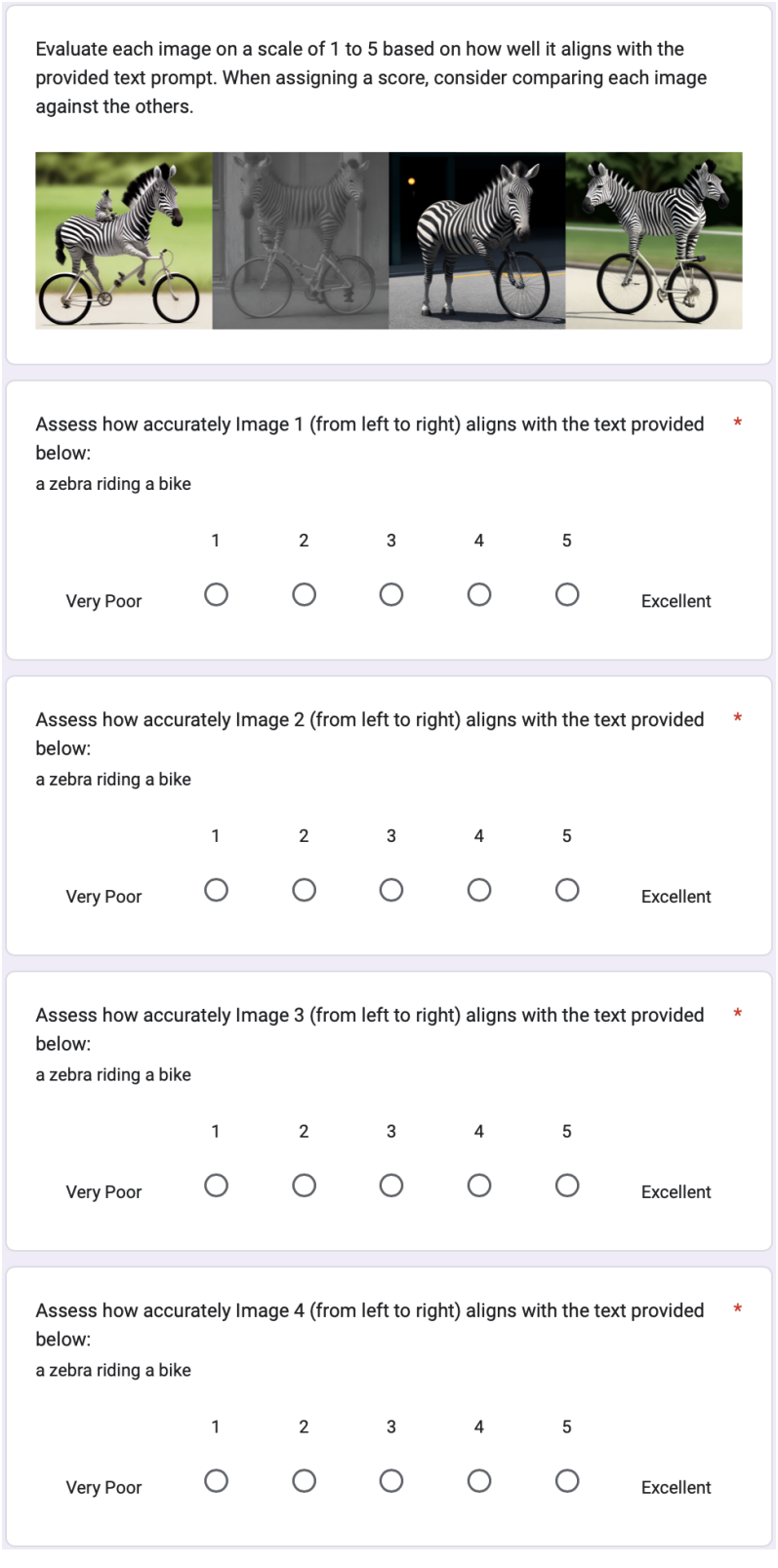


Figure 8. A screenshot of one of the annotation forms, showcasing the annotation interface for one text prompt and the four corresponding images that are generated with alternative methods. The instructions are followed by the generated images, which are displayed on the same row, side by side. The images are placed in a random order to obfuscate the methods used to generate the images. A radio button list allows the users to input the rating for each generated image. Best viewed in color.

Model	Fine-Tuning Strategy	Text Alignment	Aesthetics	Human Preference
LCM	-	0.5602	5.8038	0.2610
	DDPO	0.5627	5.9488	0.2780
	DPO	0.5639	5.9611	0.2783
	Curriculum DPO (ours)	**0.5654**	**6.0038**	**0.2793**

Table 6. Text alignment, aesthetic and human preference scores obtained on the DrawBench dataset by the baseline (pre-trained) LCM versus the three fine-tuning strategies: DDPO, DPO and Curriculum DPO. The DDPO, DPO and Curriculum DPO methods use only 5 images per prompt during optimization. The best scores are highlighted in bold.

its change only when we change the number of curriculum batches B. Therefore, ablating L_k and R_k is redundant.

In Figure 7c, we present the evolution of the reward score during the Stable Diffusion training with DDPO. For DPO and Curriculum DPO, we did not preserve the reward curves, as these methods did not involve multiple queries to the reward models. Instead, they rely solely on the original example ranking throughout the entire training process. Thus, additional queries to the reward models are unnecessary for DPO and Curriculum DPO. This represents an advantage of these methods over DDPO.

12. Human Evaluation Study

In the human evaluation study, participants were asked to rate generated images from two perspectives: prompt alignment and aesthetics. The images were generated either with SD or LCM. We created a separate annotation form containing 80 text prompts for each (task, generative model) pair, resulting in four independent annotation forms. For each generative architecture, there are four images per prompt: one from each fine-tuning strategy (DDPO, DPO and Curriculum DPO), along with another one corresponding to the pre-trained generative model. For each prompt, the images were displayed in a random order, preventing annotators from knowing which strategy was used to generate a certain image. The users were asked to rate each image with an integer grade between 1 and 5, as shown in Figure 8. The evaluation instructions were customized for each task. For text alignment, we requested the annotators to give their ratings based on how closely each generated image matches the accompanying text prompt. For aesthetics, the participants were asked to compare the images and rate each one according to their personal preference.

Since there are four images for each prompt and an annotation form comprises 80 prompts, the number of images to be annotated in one form is 320. Each form was completed by nine human evaluators, yielding a total of 2,880 annotations per experiment (form). Since we conducted the study on two generative models (SD and LCM) and two tasks (prompt alignment and aesthetics), the total number of collected annotations is 11,520.

The average time to complete the annotations for a single form is around 15-20 minutes. The nine human annotators who agreed to complete the annotation forms are either close collaborators, family members or friends of the authors. They volunteered to perform the annotations for free. To make sure that the annotations are relevant, we computed the inter-rater agreement, obtaining a Kendall Tau correlation coefficient of 0.34. This translates into 69.8% of all image pairs being concordant among annotators. Additionally, we performed statistical testing for the evaluations, and found that the voting results are statistically significant, at a p-value below 0.005.

13. Scalability

In the ablation study presented in Figure 3d of the main paper, we examine the visual appeal reward when we vary the numbers of generated images per prompt. Here, we provide a more detailed analysis of the extreme case based on 5 images per prompt, comparing Curriculum DPO with all the other fine-tuning strategies across all the three studied tasks. The results shown in Table 6 confirm that our method, Curriculum DPO, surpasses the competing methods even when the number of image samples per prompt is low. Therefore, we conclude that our training strategy does not require a high number of generated images per prompt to outperform DPO and DDPO.

14. Limitations

One limitation of our model is the introduction of additional hyperparameters, such as B or K. These might require tuning in order to find the optimal values, which involves more computing power. However, in the ablation study from Section 4, we demonstrate that Curriculum DPO outperforms all baselines for multiple hyperparameter combinations. Therefore, suboptimal hyperparameter choices can still improve the generative models.

A limitation of text-to-image generative models (as well as reward models) is the poor ability to disambiguate words in the input prompt. This can be observed especially in the prompt alignment task, where a word with multiple meanings or connotations leads to generating poor results. For example, the prompt “a turkey riding a bike” often results in images of a cooked meal instead of a live bird. Curriculum

DPO does not address this generic limitation of generative and reward models.

15. Broader Impact

Generative models can be a valuable asset in many scenarios, ranging from boosting the productivity of creative tasks to being integrated in applications that are used on a daily basis (such as image restoration or super-resolution). Nevertheless, it might also represent a great source of fake data aiming for disinformation and impersonation, especially when the model is optimized to human preferences. In the recent years, an increase in deep fake materials flooded the Internet, with attackers aiming to spread false information or even steal sensitive information by posing as another entity or person.

While we strongly believe in the benefits of very capable generative models, we are aware of the potential risks. However, we can see that governments are working very closely with academia and industry on safely developing artificial intelligence, and thus observe and support the increasing focus on models that detect AI-generated content to mitigate the aforementioned risks. Notably, the ultimate goal of the project that funded our research is to develop robust deepfake detectors.