Diffusion Model Alignment Using Direct Preference Optimization

Supplementary Material

S1. Comparisons to existing work

RL-Based Methods such as [7, 13] have shown effectiveness in operating on a limited set of prompts ($<$ 10 and $<$ 1000 respectively) but do not generalize as well to the open-vocabulary setting as shown in [9, 34]. We found this in our experiments as well, where training using the DDPO scheme on SD1.5 did not improve the win rate versus DPO. Recent post-submission work [61], however, recently demonstrated promise applying these methods at scale to smaller (SD1.5) diffusion models.

While DDPO [7] is an RL-based method as is DPOK [13], their target objective and distributional guarantees are different. Specifically, DDPO purely aims to optimize the reward function without any KL-regularization

$$\mathbb{E}_{\boldsymbol{c}\sim p(c),\boldsymbol{x}_0\sim p_\theta(\boldsymbol{x}_0|\boldsymbol{c})} r(\boldsymbol{x}_0, \boldsymbol{c}) \tag{15}$$

while DPOK adds in a term governing KL-regularization between the learned distribution and a reference distribution as in our setting. This means that DDPO is optimizing the same objective as DRaFT and AlignProp ([9, 34]) but via RL instead of gradient descent through the diffusion chain. DDPO uses early stopping in lieu of distributional control.

Additionally, through the score function policy gradient estimator employed by DDPO it is observable why the method struggles with open vocabulary generation. The gradient estimation used is

$$\nabla_\theta \mathcal{J}_{\text{DDRL}} = \mathbb{E} \sum_{t=0}^{T} \frac{p_\theta(x_{t-1} \mid x_t, \boldsymbol{c})}{p_{\theta_{\text{old}}}(x_{t-1} \mid x_t, \boldsymbol{c})} \nabla_\theta \log p_\theta(x_{t-1} \mid x_t, \boldsymbol{c})\; r(x_0, \boldsymbol{c}) \tag{16}$$

Here the trajectories $\{\boldsymbol{x}_T, \boldsymbol{x}_{T-1}, \dots, \boldsymbol{x}_0\}$ are generated by the *original* model $p_{\theta_{\text{old}}}$. In this formulation, the term $\frac{p_\theta(x_{t-1}|x_t,\boldsymbol{c})}{p_{\theta_{\text{old}}}}$ simply is an importance weighter which scales gradient contributions based on the relevance of the sample (as determined by how aligned the learned and reference model predictions are). Since the trajectories are generated by the "old" (reference) model, $r(x_0, \boldsymbol{c})$ is only a weighting in the latter term $\nabla_\theta \log p_\theta(x_{t-1} \mid x_t, \boldsymbol{c})\; r(x_0, \boldsymbol{c})$. The gradient encourages higher likelihoods for generations of high reward, but makes no distinction about the diversity of those generations. High-reward prompts can dominate the gradient trajectory, while generations considered lower-reward are ignored or discouraged. This stands in contrast to the DPO framework where the likelihood of a generation is contrasted against another with the same conditioning. This normalization across conditioning prevents sets of $\boldsymbol{c}$ being considered unimportant/undesirable and not being optimized for. In Diffusion-DPO, conditionings with all types of reward magnitudes are weighted equally towards the $\boldsymbol{x}_0^w$ and away from the $\boldsymbol{x}_0^l$.

Inference Time-Optimization namely DOODL [54], does not learn any new model parameters, instead optimizing diffusion latents to improve some criterion on the generated image similar to CLIP+VQGAN[10]. This runtime compute increases inference cost by more than an order of magnitude.

Reward Maximization Training such as [9, 34] amortize the cost of DOODL from runtime to training. They train by generating images from text prompts, computing a reward loss on the images, and backpropagating gradients through the generative process to improve the loss. While effective in the open-vocabulary setting (also training on Pick-a-Pic prompts), these methods provide no distributional guarantees (unlike the control via β in Diffusion-DPO) and suffer from mode collapse with over-training. These methods do not generalize to all reward functions, with [9] noting the inability of DRaFT to improve image-text alignment using CLIP[35] as a reward function. In contrast, Diffusion-DPO can improve image-text alignment using CLIP preference, as shown in Sec. 5.4. Furthermore, only differentiable rewards can be optimized towards in the reward maximization setting. This necessitates not only data collection but also reward model training.

Dataset Curation As discussed, models such as StableDiffusion variants [33, 39] train on laion-aesthetics [40] to bias the model towards more visually appealing outputs. Concurrent work Emu [11] takes this approach to an extreme. Instead of training on any images from a web-scale dataset which pass a certain model score threshold, they employ a multi-stage pipeline where such filtering is only the first stage. Subsequently, crowd workers filter the subset down using human judgement and at the final stage expert in photography are employed to create the dataset. While effective, this process has several drawbacks compared to Diffusion-DPO. First, necessitating training on existing data can be a bottleneck, both in terms of scale and potential applications. While [11] reports lesser text faithfulness improvements as well, these are likely due to the hand-written captions, a much more costly data collection stage than preferences. The Emu pipeline is not generalizable to different types of feedback as DPO is (e.g. outside of recaptioning it is non-obvious how such an approach can improve text-image alignment).

S1.1. Experiments

DDPO We compare a DPO-tuned SD2.1 model to the DDPO-tuned SD2.1 from CarperAI [1] on Partiprompts (general preference). The DPO model has a win rate of 59.7%, maintaining a significant margin of improvement. We additionally use the public `diffusers` DDPO implementation to train a SD1.5 model on Pick-a-Pic prompts, our DPO SD1.5 model maintains a 55.7% win rate against this model as well. Given the outperformance of DPOK by DDPO shown in [8] we do not directly compare to DPOK.

Emu We test Emu-style finetuning [11] on Pick-a-Pic by selecting the top 1% and 0.1% (∼10k and ∼1k samples respectively, comparable to Emu) of the images by PickScore and tune SDXL using hyperparameters from [11]. While such training improves upon SFT, it still falls significantly short of Diffusion-DPO, winning $< 45\%$ of head-to-head PartiPrompt comparisons vs. DPO-SDXL in both cases.

S2. Details of the Primary Derivation

Starting from Eq. (5), we have

$$\begin{aligned}
&\min_{p_\theta} -\mathbb{E}_{p_\theta(\boldsymbol{x}_0|\boldsymbol{c})} r(\boldsymbol{c}, \boldsymbol{x}_0)/\beta + \mathbb{D}_{\text{KL}}(p_\theta(\boldsymbol{x}_0|\boldsymbol{c})||p_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})) \\
\le &\min_{p_\theta} -\mathbb{E}_{p_\theta(\boldsymbol{x}_0|\boldsymbol{c})} r(\boldsymbol{c}, \boldsymbol{x}_0)/\beta + \mathbb{D}_{\text{KL}}\left(p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})||p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c})\right) \\
= &\min_{p_\theta} -\mathbb{E}_{p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})} R(\boldsymbol{c}, \boldsymbol{x}_{0:T})/\beta + \mathbb{D}_{\text{KL}}\left(p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})||p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c})\right) \\
= &\min_{p_\theta} \mathbb{E}_{p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})} \left(\log \frac{p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c}) \exp(R(\boldsymbol{c}, \boldsymbol{x}_{0:T})/\beta)/Z(\boldsymbol{c})} - \log Z(\boldsymbol{c})\right) \\
= &\min_{p_\theta} \mathbb{D}_{\text{KL}}\left(p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})\|p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c}) \exp(R(\boldsymbol{c}, \boldsymbol{x}_{0:T})/\beta)/Z(\boldsymbol{c})\right).
\end{aligned} \tag{17}$$

where $Z(\boldsymbol{c}) = \sum_{\boldsymbol{x}} p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c}) \exp\left(r(\boldsymbol{c}, \boldsymbol{x}_0)/\beta\right)$ is the partition function. The optimal $p^*_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})$ of Equation (17) has a unique closed-form solution:

$$p^*_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c}) = p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c}) \exp(R(\boldsymbol{c}, \boldsymbol{x}_{0:T})/\beta)/Z(\boldsymbol{c}),$$

Therefore, we have the reparameterization of reward function

$$R(\boldsymbol{c}, \boldsymbol{x}_{0:T}) = \beta \log \frac{p^*_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c})} + \beta \log Z(\boldsymbol{c}).$$

Plug this into the definition of r, hence we have

$$r(\boldsymbol{c}, \boldsymbol{x}_0) = \beta \mathbb{E}_{p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0, \boldsymbol{c})} \left[\log \frac{p^*_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_{0:T}|\boldsymbol{c})}\right] + \beta \log Z(\boldsymbol{c}).$$

Substituting this reward reparameterization into maximum likelihood objective of the Bradly-Terry model as Eq. (4), the partition function cancels for image pairs, and we get a maximum likelihood objective defined on diffusion models, its per-example formula is:

$$L_{\text{DPO-Diffusion}}(\theta) = -\log \sigma \left(\beta \mathbb{E}_{\boldsymbol{x}^w_{1:T} \sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0), \boldsymbol{x}^l_{1:T} \sim p_\theta(\boldsymbol{x}^l_{1:T}|\boldsymbol{x}^l_0)} \left[\log \frac{p_\theta(\boldsymbol{x}^w_{0:T})}{p_{\text{ref}}(\boldsymbol{x}^w_{0:T})} - \log \frac{p_\theta(\boldsymbol{x}^l_{0:T})}{p_{\text{ref}}(\boldsymbol{x}^l_{0:T})}\right]\right)$$

where $\boldsymbol{x}^w_0, \boldsymbol{x}^l_0$ are from static dataset, we drop $\boldsymbol{c}$ for simplicity.

An approximation for reverse process Since sampling from $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ is intractable, we utilize $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ for approximation.

$$
\begin{aligned}
L_1(\theta) &= -\log\sigma\left(\beta\mathbb{E}_{\boldsymbol{x}^w_{1:T}\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{1:T}\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}^l_0)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{0:T})}{p_{\text{ref}}(\boldsymbol{x}^w_{0:T})}-\log\frac{p_\theta(\boldsymbol{x}^l_{0:T})}{p_{\text{ref}}(\boldsymbol{x}^l_{0:T})}\right]\right)\\
&= -\log\sigma\left(\beta\mathbb{E}_{\boldsymbol{x}^w_{1:T}\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{1:T}\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}^l_0)}\left[\sum_{t=1}^{T}\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right)\\
&= -\log\sigma\left(\beta\mathbb{E}_{\boldsymbol{x}^w_{1:T}\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{1:T}\sim q(\boldsymbol{x}_{1:T}|\boldsymbol{x}^l_0)}T\mathbb{E}_t\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right)\\
&= -\log\sigma\left(\beta T\mathbb{E}_t\mathbb{E}_{\boldsymbol{x}^w_{t-1,t}\sim q(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{t-1,t}\sim q(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}^l_0)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right)\\
&= -\log\sigma\Bigg(\beta T\mathbb{E}_{t,\boldsymbol{x}^w_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0),\boldsymbol{x}^l_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)}\\
&\quad\mathbb{E}_{\boldsymbol{x}^w_{t-1}\sim q(\boldsymbol{x}_{t-1}|\boldsymbol{x}^w_t,\boldsymbol{x}^w_0),\boldsymbol{x}^l_{t-1}\sim q(\boldsymbol{x}_{t-1}|\boldsymbol{x}^l_t,\boldsymbol{x}^l_0)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\Bigg)
\end{aligned}
\tag{18}
$$

By Jensen's inequality, we have

$$
\begin{aligned}
L_1(\theta) &\leq -\mathbb{E}_{t,\boldsymbol{x}^w_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0),\boldsymbol{x}^l_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)}\log\sigma\Bigg(\\
&\qquad\beta T\mathbb{E}_{\boldsymbol{x}^w_{t-1}\sim q(\boldsymbol{x}_{t-1}|\boldsymbol{x}^w_t,\boldsymbol{x}^w_0),\boldsymbol{x}^l_{t-1}\sim q(\boldsymbol{x}_{t-1}|\boldsymbol{x}^l_t,\boldsymbol{x}^l_0)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\Bigg)\\
&= -\mathbb{E}_{t,\boldsymbol{x}^w_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0),\boldsymbol{x}^l_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)}\log\sigma\Bigg(-\beta T\big(\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_{0,t})\|p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t))-\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_{0,t})\|p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t))\\
&\qquad-\big(\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_{0,t})\|p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t))+\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_{0,t})\|p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t))\big)\big)\Bigg)
\end{aligned}
$$

Using the Gaussian parameterization of the reverse process (Eq. (1)), the above loss simplifies to:

$$
L_1(\theta)\leq-\mathbb{E}_{t,\boldsymbol{\epsilon}^w,\boldsymbol{\epsilon}^l}\log\sigma\left(-\beta T\omega(\lambda_t)\left(\ \|\boldsymbol{\epsilon}^w-\boldsymbol{\epsilon}_\theta(\boldsymbol{x}^w_t,t)\|^2-\|\boldsymbol{\epsilon}^w-\boldsymbol{\epsilon}_{\text{ref}}(\boldsymbol{x}^w_t,t)\|^2\ -\left(\|\boldsymbol{\epsilon}^l-\boldsymbol{\epsilon}_\theta(\boldsymbol{x}^l_t,t)\|^2-\|\boldsymbol{\epsilon}^l-\boldsymbol{\epsilon}_{\text{ref}}(\boldsymbol{x}^l_t,t)\|^2\right)\right)\right)
$$

where $\boldsymbol{\epsilon}^w,\boldsymbol{\epsilon}^l\sim\mathcal{N}(0,I)$, $\boldsymbol{x}_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ thus $\boldsymbol{x}_t=\alpha_t\boldsymbol{x}_0+\sigma_t\boldsymbol{\epsilon}$. Same as Eq. (2), $\lambda_t=\alpha_t^2/\sigma_t^2$ is a signal-to-noise ratio term [23], in practice, the reweighting assigns each term the same weight [19].

An alternative approximation Note that for Eq. (18) we utilize $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$ to approximate $p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$. For each step, it is to use $q(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}_0)$ to approximate $p_\theta(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}_0)$. Alternatively, we also propose to use $q(\boldsymbol{x}_t|\boldsymbol{x}_0)p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ for approximation. And this approximation yields lower error because $\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_t|\boldsymbol{x}_0)p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\|p_\theta(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}_0))=\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_t|\boldsymbol{x}_0)\|p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_0))<\mathbb{D}_{\text{KL}}(q(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}_0)\|p_\theta(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}_0))$.

$$
\begin{aligned}
L_{\text{DPO-Diffusion}}(\theta) &= -\log\sigma\left(\beta\mathbb{E}_{\boldsymbol{x}^w_{1:T}\sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{1:T}\sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^l_0)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{0:T})}{p_{\text{ref}}(\boldsymbol{x}^w_{0:T})}-\log\frac{p_\theta(\boldsymbol{x}^l_{0:T})}{p_{\text{ref}}(\boldsymbol{x}^l_{0:T})}\right]\right)\\
&= -\log\sigma\left(\beta\mathbb{E}_{\boldsymbol{x}^w_{1:T}\sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{1:T}\sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^l_0)}\left[\sum_{t=1}^{T}\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right)\\
&= -\log\sigma\left(\beta\mathbb{E}_{\boldsymbol{x}^w_{1:T}\sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{1:T}\sim p_\theta(\boldsymbol{x}_{1:T}|\boldsymbol{x}^l_0)}T\mathbb{E}_t\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right)\\
&= -\log\sigma\left(\beta T\mathbb{E}_t\mathbb{E}_{\boldsymbol{x}^w_{t-1,t}\sim p_\theta(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}^w_0),\boldsymbol{x}^l_{t-1,t}\sim p_\theta(\boldsymbol{x}_{t-1,t}|\boldsymbol{x}^l_0)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}-\log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right).
\end{aligned}
$$

By approximating $p_\theta(x_{t-1,t}|\boldsymbol{x}_0)$ with $q(\boldsymbol{x}_t|\boldsymbol{x}_0)p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, we have

$$\begin{aligned} L_2(\theta) &= -\log\sigma\left(\beta T \mathbb{E}_t \mathbb{E}_{\boldsymbol{x}^w_{t-1,t}\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0)p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}^w_t),\boldsymbol{x}^l_{t-1,t}\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}^l_t)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)} - \log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right) \\ &= -\log\sigma\left(\beta T \mathbb{E}_{t,\boldsymbol{x}^w_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0),\boldsymbol{x}^l_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)} \mathbb{E}_{\boldsymbol{x}^w_{t-1}\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}^w_t),\boldsymbol{x}^l_{t-1}\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}^l_t)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)} - \log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right). \end{aligned}$$

By Jensen's inequality, we have

$$\begin{aligned} L_2(\theta) &\leq -\mathbb{E}_{t,\boldsymbol{x}^w_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0),\boldsymbol{x}^l_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)} \log\sigma\left(\beta T \mathbb{E}_{\boldsymbol{x}^w_{t-1}\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}^w_t),\boldsymbol{x}^l_{t-1}\sim p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}^l_t)}\left[\log\frac{p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)}{p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)} - \log\frac{p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)}{p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}_t)}\right]\right) \\ &= -\mathbb{E}_{t,\boldsymbol{x}^w_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^w_0),\boldsymbol{x}^l_t\sim q(\boldsymbol{x}_t|\boldsymbol{x}^l_0)} \log\sigma\left(\beta T\left(\mathbb{D}_{\text{KL}}(p_\theta(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)\|p_{\text{ref}}(\boldsymbol{x}^w_{t-1}|\boldsymbol{x}^w_t)) - \mathbb{D}_{\text{KL}}(p_\theta(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)\|p_{\text{ref}}(\boldsymbol{x}^l_{t-1}|\boldsymbol{x}^l_t)\right)\right) \end{aligned}$$

Using the Gaussian parameterization of the reverse process (Eq. (1)), the above loss simplifies to:

$$L_2(\theta) = -\mathbb{E}_{t,\boldsymbol{\epsilon}^w,\boldsymbol{\epsilon}^l} \log\sigma\left(-\beta T\omega(\lambda_t)\left(\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}^w_t,t) - \boldsymbol{\epsilon}_{\text{ref}}(\boldsymbol{x}^w_t,t)\|^2 - \|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}^l_t,t) - \boldsymbol{\epsilon}_{\text{ref}}(\boldsymbol{x}^l_t,t)\|^2\right)\right)$$

where $\boldsymbol{\epsilon}^w, \boldsymbol{\epsilon}^l \sim \mathcal{N}(0, I)$, $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ thus $\boldsymbol{x}_t = \alpha_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$. Same as Eq. (2), $\lambda_t = \alpha_t^2/\sigma_t^2$ is a signal-to-noise ratio term [23], in practice, the reweighting assigns each term the same weight [19].

S3. Alternate Derivation: Reinforcement Learning Perspective

We can also derive our objective as a multi-step RL approach, in the same setting as [7, 13]. A Markov Decision Process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, \rho_0, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, ρ_0 is an initial state distribution, $\mathcal{P}$ is the transition dynamics and $\mathcal{R}$ is the reward function. In this formulation, at each time step t a policy $\pi(a_t|s_t)$ observes a state $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$. The environment then transitions to a next state $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$ and the returns a reward $\mathcal{R}(s_t, a_t)$.The goal of the policy is to maximize the total rewards it receives. Prior works [7, 13] map the denoising process in diffusion model generation to this formulation via:

$$\begin{aligned} \mathbf{s}_t &\triangleq (\boldsymbol{c}, \boldsymbol{x}_t, t) \\ \mathbf{a}_t &\triangleq \boldsymbol{x}_t \\ \mathcal{P}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t) &\triangleq (\delta_{\boldsymbol{c}}, \delta_{t-1}, \delta_{\boldsymbol{x}_{t-1}}) \\ \rho(\mathbf{s}_0) &\triangleq (p(\boldsymbol{c}), \delta_T, \mathcal{N}(\mathbf{0}, \mathbf{I})) \end{aligned}$$

$$\mathcal{R}(\mathbf{s_t}, \mathbf{a}_t) = \begin{cases} r(\boldsymbol{c}, \boldsymbol{x}_0) & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $\boldsymbol{c}$ is the prompt $\boldsymbol{x}_t$ is the time-step t nosy image and δ_y is the Dirac delta function with unit density at y. That is in this formulation we consider the denoising model as a policy, with each denoising step a step in an MDP. The objective of the policy is to maximize the reward (alignment with human preference) of the final image. In the derivation below, we drop the time step t for brevity. In this formulation the generative model is a policy and the denoising process is a rollout in an MDP with a sparse reward received for the final generated image. Following [13] we optimize the following objective

$$\mathbb{E}_{\boldsymbol{c}\sim\mathcal{D},p_\theta}\left[\sum_{t=T}^{0} r(\boldsymbol{c}, \boldsymbol{x}_t) - \beta D_{KL}[p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})||p_{\text{ref}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})]\right] \tag{20}$$

While prior works [7, 13] use policy gradient approaches to optimize this objective, we're going to use off-policy methods. Following Control as Variational Inference [26], we have the following

$$Q^*((\boldsymbol{x}_t, \boldsymbol{c}), \boldsymbol{x}_{t-1}) = r(\boldsymbol{c}, \boldsymbol{x}_t) + V^*(\boldsymbol{x}_{t-1}, \boldsymbol{c}) \tag{21}$$

$$V^*(\boldsymbol{x}_{t-1}, \boldsymbol{c}) = \beta\log\mathbb{E}_{p_{\text{ref}}}\left[\exp Q^*((\boldsymbol{x}_t, \boldsymbol{c}), \boldsymbol{x}_{t-1})/\beta\right] \tag{22}$$

$$p^*(\boldsymbol{x}_{t-1}|(\boldsymbol{x}_t,\boldsymbol{c})) = p_{\text{ref}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})e^{(Q^*((\boldsymbol{x}_t,\boldsymbol{c}),\boldsymbol{x}_{t-1})-V^*(\boldsymbol{x}_t,\boldsymbol{c}))/\beta} \tag{23}$$

where V^* is the optimal value function and Q^* is the optimal state-action value function (in tour definition of the denoising MDP, he policy is stochastic, but the dynamics is deterministic). Also notice that in Eq. 23 the equation is exact since the right-hand side integrates to 1. We then consider the inverse soft Bellman operator [15] and have the following

$$r(\boldsymbol{c},\boldsymbol{x}_t) = V^*(\boldsymbol{x}_{t-1},\boldsymbol{c}) - Q^*((\boldsymbol{x}_t,\boldsymbol{c}),\boldsymbol{x}_{t-1}) \tag{24}$$

However, from Eq. 23 we have

$$Q^*((\boldsymbol{x}_t,\boldsymbol{c}),\boldsymbol{x}_{t-1}) - V^*(\boldsymbol{x}_t,\boldsymbol{c}) = \log\frac{p^*(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})} \tag{25}$$

substituting in Eq. 24 we obtain:

$$r(\boldsymbol{c},\boldsymbol{x}_t) = V^*(\boldsymbol{x}_{t-1},\boldsymbol{c}) + \log\frac{p^*(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})} - V^*(\boldsymbol{x}_t,\boldsymbol{c}) \tag{26}$$

Using a telescoping sum through the diffusion chain we are left with

$$r(\boldsymbol{c},\boldsymbol{x}_0) = \sum_{t=0}^{T}\log\frac{p^*(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})}{p_{\text{ref}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{c})} - V^*(\boldsymbol{x}_T,\boldsymbol{c}) \tag{27}$$

since by definition all intermediate rewards are zero. If we assume both diffusion chains start from the same state and plug this result into the preference formulation of Eq. 3 we obtain the objective of Eq. 11. Here we optimize the same objective as prior works [7, 13], but instead of a policy gradient approach we derive our objective as an off-policy learning problem in the same MDP. This not only simplifies the algorithm significantly, but justifies our sampling choices in Eq. 13 and we do not have to sample through the entire difussion chain.

S4. Alternative Derivation: Noise-Aware Preference Model

Paralleling the original DPO formulation we consider a policy trained on maximizing the likelihood of $p(x_0|\boldsymbol{c},t,x_{\text{obs}})$ where x_{obs} is a noised version of x_0. Here x_0 is an image, $\boldsymbol{c}$ is a text caption, t is a noising scale, and x_{obs} is a corruption (noised version) of x_0. We initialize from a reference diffusion policy p_{ref}. We aim to optimize the same RL objective of Eq. (5), reprinted here for convenience:

$$\max_{p_\theta}\mathbb{E}_{\boldsymbol{c}\sim\mathcal{D},\boldsymbol{x}_0\sim p_\theta(\boldsymbol{x}_0|\boldsymbol{c})}\left[r(\boldsymbol{c},\boldsymbol{x}_0)\right] - \beta\mathbb{D}_{\text{KL}}\left[p_\theta(\boldsymbol{x}_0|\boldsymbol{c})\|p_{\text{ref}}(\boldsymbol{x}_0|\boldsymbol{c})\right] \tag{28}$$

Our policy has additional conditioning (t, x_{obs}). The latter is a noised version of x_0. Define the space of noising operators at time t as Q_t where $q_t \sim Q_t$ with $q_t(x_0) = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha}\epsilon_{q_t}, \epsilon_{q_t} \sim N(0,I)$. Here q_t refers to the linear transform corresponding with a specific gaussian draw $\sim N(0,I)$ and the set of q_t is Q_t. In general at some time level t we have $y_{\text{obs}} = q_t(x_0)$ for some $q_t \sim Q_t$ so can write the conditioning as $p(x_0|\boldsymbol{c},t,q_t(y))$. We rewrite Eq. (28) as

$$\max_{p_\theta}\mathbb{E}_{\boldsymbol{c}\sim\mathcal{D},x_0\sim p_\theta^{(gen)}(x_0|\boldsymbol{c}),t\sim\mathcal{U}\{0,T\},q_t\sim Q_T}\left(r_\phi(\boldsymbol{c},x_0) - \beta\mathbb{D}_{\text{KL}}\left[p_\theta(x_0\mid\boldsymbol{c},t,q_t(x_0)) \mid\mid p_{\text{ref}}(x_0\mid\boldsymbol{c},t,q_t(x_0))\right]\right) \tag{29}$$

$p^{(gen)}$ denoting the generative process associated with p as a diffusion model. Note that the reward model is the same formulation as in DPO. The optimal policy now becomes

$$p^*_\theta(x_0\mid\boldsymbol{c},t,q_t(x_0)) = \frac{1}{Z(\boldsymbol{c},t,q_t)}p_{\text{ref}}(x_0\mid\boldsymbol{c},t,q_t(x_0))\exp\left(\frac{1}{\beta}r(\boldsymbol{c},x_0)\right) \tag{30}$$

with Z a partition over captions, timesteps, and noising draws. Rearranging for $r(\boldsymbol{c},x_0)$ now yields

$$r(\boldsymbol{c},x_0) = \beta\log\frac{p^*_\theta(x_0\mid\boldsymbol{c},t,q_t)}{p_{\text{ref}}(x_0\mid\boldsymbol{c},t,q_t)} + \beta\log Z(\boldsymbol{c},t,q_t),\ \forall t,q_t \tag{31}$$

We have not changed the reward model formulation at all, but our policies have extra conditioning as input (which ideally the likelihoods are constant with respect to). Putting this formulation into the original Bradley-Terry framework of Eq. (3) (re-printed here)

$$p_{\text{BT}}(\boldsymbol{x}_0^w \succ \boldsymbol{x}_0^l|\boldsymbol{c}) = \sigma(r(\boldsymbol{c}, \boldsymbol{x}_0^w) - r(\boldsymbol{c}, \boldsymbol{x}_0^l)) \tag{32}$$

results in the objective:

$$\mathcal{L}_{\text{DPO}}(p_\theta; p_{\text{ref}}) = -\mathbb{E}_{(x_0^w, x_0^l \sim p^{(gen)}(\boldsymbol{c}), \boldsymbol{c} \sim \mathcal{D}, t \sim \mathcal{U}\{0,T\}, q_t \sim Q_t} \left[\log \sigma \left(\beta \log \frac{p_\theta(x_0^w \mid \boldsymbol{c}, t, q_t(x_0^w))}{p_{\text{ref}}(x_0^w \mid \boldsymbol{c}, t, q_t(x_0^w))} - \beta \log \frac{p_\theta(x_0^l \mid \boldsymbol{c}, t, q_t(x_0^l))}{p_{\text{ref}}(x_0^l \mid \boldsymbol{c}, t, q_t(x_0^l))}\right)\right] \tag{33}$$

We now consider how to compute these likelihoods. Using the notation $a_t = \sqrt{\alpha_t}$ and $b_t = \sqrt{1 - \alpha_t}$ as shorthand for commonly-used diffusion constants (α are defined as in DDIM[46]) we have

$$x_{\text{obs}} = q_t(x_0) = a_t x_0 + b_t \epsilon, \epsilon \sim \mathcal{N}(0, I) \tag{34}$$

We use Eq. 57 from DDIM[46] (along with their definition of σ_t):

$$p(x_0|x_t) = \mathcal{N}(x_0^{pred}, \sigma_t^2 I) \tag{35}$$

Our x_0^{pred} is:

$$x_0^{pred} = \frac{x_{\text{obs}} - b_t \epsilon_\theta^{pred}}{a_t} = \frac{a_t x_0 + b_t \epsilon - b_t \epsilon_\theta^{pred}}{a_t} = x_0 + \frac{b_t}{a_t}(\epsilon - \epsilon_\theta^{pred}) \tag{36}$$

Here ϵ_θ^{pred} is the output of $\epsilon_\theta(\boldsymbol{c}, t, x_{\text{obs}})$ Making the conditional likelihood:

$$p_\theta(x_0|\boldsymbol{c}, t, x_{\text{obs}}) = \mathcal{N}(x_0; x_0 + \frac{b_t}{a_t}(\epsilon - \epsilon_\theta^{pred}), \sigma_t^2 I) = \frac{1}{(2\pi\sigma_t^2)^{d/2}} e^{-\frac{b_t^2}{2a_t^2\sigma_t^2}||\epsilon - \epsilon_\theta^{pred}||_2^2} \tag{37}$$

For convenience we define

$$z_t = \frac{1}{(2\pi\sigma_t^2)^{d/2}} \tag{38}$$

$$SE = ||\epsilon - \epsilon_{pred}||_2^2 \tag{39}$$

We will decorate the latter quantity (SE) with sub/superscripts later. For now we get:

$$p_\theta(x_0|\boldsymbol{c}, t, x_{\text{obs}}) = z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2} SE} \tag{40}$$

We see to minimize

$$\mathop{\mathbb{E}}_{(x_0^w, x_0^l \sim p^{(gen)}(\boldsymbol{c}); \boldsymbol{c} \sim \mathcal{D},; t \sim \mathcal{U}\{0,T\}; q_t \sim Q_t} -\log \sigma \left(\beta \left(\log \frac{p_\theta(x_0^w|\boldsymbol{c}, t, q_t(x_0^w))}{p_{\text{ref}}(x_0^w|\boldsymbol{c}, t, q_t(x_0^w)} - \log \frac{p_\theta(x_0^l|\boldsymbol{c}, t, q_t(x_0^l))}{p_{\text{ref}}(x_0^l|\boldsymbol{c}, t, q_t(x_0^l)}\right)\right) = \tag{41}$$

$$\mathop{\mathbb{E}}_{(x_0^w, x_0^l \sim p^{(gen)}(\boldsymbol{c}); \boldsymbol{c} \sim \mathcal{D},; t \sim \mathcal{U}\{0,T\}; q_t \sim Q_t} -\log \sigma \left(\beta \left(\log \frac{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2} SE_\theta^{(w)}}}{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2} SE_{\text{ref}}^{(w)}}} - \log \frac{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2} SE_\theta^{(l)}}}{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2} SE_{\text{ref}}^{(l)}}}\right)\right) \tag{42}$$

Here we use $SE_\psi^{(d)} = ||\epsilon_{q_t} - \psi(\boldsymbol{c}, t, q_t(x_0^d))||_2^2$ to denote the L2 error in the noise prediction of model ψ operating on the noisy $q_t(x_0^d)$ with corresponding conditioning $(\boldsymbol{c}, t)$ ($d \in \{w, l\}$). Here the model associated with SE_{ref}^* is the model of the reference policy p_{ref}. Note that these SE terms are the standard diffusion training objective from Eq. (2). Continuing to simplify the above yields:

$$-\log\sigma\left(\beta\left(\log\frac{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2}SE_\theta^{(w)}}}{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2}SE_{\text{ref}}^{(w)}}}-\log\frac{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2}SE_\theta^{(l)}}}{z_t e^{-\frac{b_t^2}{2a_t^2\sigma_t^2}SE_{\text{ref}}^{(l)}}}\right)\right) \tag{43}$$

$$= -\log\sigma\left(-\beta\frac{b_t^2}{2a_t^2\sigma_t^2}\left((SE_\theta^{(w)}-SE_{\text{ref}}^{(w)})-(SE_\theta^{(l)}-SE_{\text{ref}}^{(l)})\right)\right) \tag{44}$$

We can simplify the coefficient:

$$\frac{b_t^2}{a_t^2\sigma_t^2}=\frac{1-\alpha_t}{\alpha_t}\frac{1}{\sigma_t^2}=\frac{\sigma_{t+1}^2}{\sigma_t^2}\approx 1 \tag{45}$$

Resulting in objective

$$\approx \mathop{\mathbb{E}}_{x,y_w,y_l\sim D;t;\epsilon\sim\mathcal{N}(0,I)} -\log\sigma\left(-\frac{\beta}{2}\left((SE_\theta^{(w)}-SE_{\text{ref}}^{(w)})-(SE_\theta^{(l)}-SE_{\text{ref}}^{(l)})\right)\right) \tag{46}$$

Up to the approximation of Eq. (45) this is the equivalent to Eq. (33). The log of the likelihood ratios simply take on the elegant form of a difference in diffusion training losses. Due to the equation negatives and $\log\sigma$ being a monotonic increasing function, by minimizing Eq. (46) we are aiming to minimize the inside term

$$\left((SE_\theta^{(w)}-SE_{\text{ref}}^{(w)})-(SE_\theta^{(l)}-SE_{\text{ref}}^{(l)})\right) \tag{47}$$

This can be done by minimizing $SE_\theta^{(w)}$ or maximizing $SE_\theta^{(l)}$, with the precise loss value depending on how these compare to the reference errors $SE_{\text{ref}}^{(w)}, SE_{\text{ref}}^{(l)}$. The asymmetry of the $\log\sigma$ function allows β to control the penalty for deviating from the reference distribution. A high β results in a highly assymetric distribution, disproportionately penalizing low $SE_\theta^{(l)}$ and high $SE_\theta^{(w)}$ and encouraging a p_θ to make less mistakes in implicitly scoring y_w, y_l by deviating less from the reference policy p_{ref}. We visualize the $\log\sigma$ curves in Figure S1 for several values of β.

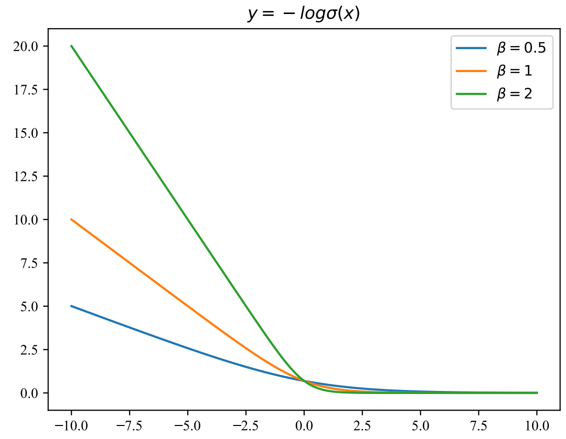

Figure S1. Visualization of $y = -\log\sigma(\beta x)$

S4.1. Reward Estimation

Finally, we note that in this formulation that if we wish to compute the noise-aware reward difference $r(\boldsymbol{c}, x_0^A) - r(\boldsymbol{c}, x_0^B)$, referring to Eq. (31) this now has form

$$r(\boldsymbol{c}, x_0^A) - r(\boldsymbol{c}, x_0^B) = \left[\beta(SE_\theta^A - SE_{\text{ref}}^A) + \beta \log Z(\boldsymbol{c}, t, q_t)\right] - \left[\beta(SE_\theta^B - SE_{\text{ref}}^B) + \beta \log Z(\boldsymbol{c}, t, q_t)\right],\ \forall \boldsymbol{c}, t, q_t \tag{48}$$

$$= \beta\left[(SE_\theta^A - SE_{\text{ref}}^A) - (SE_\theta^B - SE_{\text{ref}}^B)\right],\ \forall \boldsymbol{c}, t, q_t \tag{49}$$

$$\tag{50}$$

Which means for two images (x_0^A, x_0^B) with the same conditioning $\boldsymbol{c}$ we can estimate the reward difference using Eq. (48). When doing this it improves the estimate to average over multiple draws (t, q_t). We use this method in Table S2.

S5. β Ablation

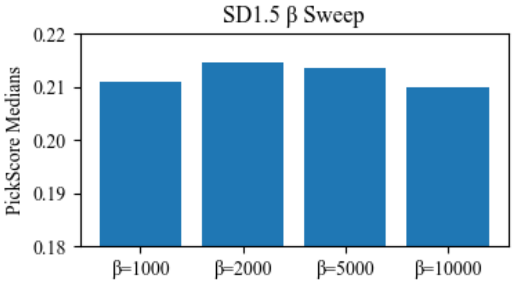

Figure S2. Median PickScores for generations on the Pick-a-Pic v2 validation set for different choices of β

For β far below the displayed regime, the diffusion model degenerates into a pure reward scoring model. Much greater, and the KL-divergence penalty greatly restricts any appreciable adaptation. Qualitative examples are shown in Figure S3

S6. Further SFT Discussions

We also partially attribute this difference in effectiveness of SFT to the gap in pretraining vs. downstream task considered in the original DPO paper [36] vs. our work. On two of the DPO LLM tasks (**sentiment generation, single-turn dialogue**), generic off-the-shelf autoregressive language models are tuned on specific tasks in the SFT stage. In the final setting, **summarization**, the SFT model has been pretrained on a similar task/dataset. In this case, finetuning on the "preferred" dataset (preferred-FT) baseline performs comparably to the SFT initialization.

This final setting is most analogous to that of Diffusion-DPO. The generic pretraining, task, and evaluation setting are all text-to-image generation. There is no task-specific domain gap and all of the settings are open-vocabulary with a broad range of styles. As such, our findings are similar to that of **summarization** in [36] where an already task-tuned model does not benefit from preferred finetuning.

S7. Ethics

The performance of Diffusion-DPO is impressive, but any effort in text-to-image generation presents ethical risks, particularly when data are web-collected. Generations of harmful, hateful, fake or sexually explicit content are known risk vectors. Beyond that, this approach is increasingly subject to the biases of the participating labelers (in addition to the biases present in the pretrained model); Diffusion-DPO can learn and propagate these preferences. As a result, a diverse and representative set of labelers is essential – whose preferences in turn become encoded in the dataset. Furthermore, a portion of user-generated Pick-a-Pic prompts are overtly sexual, and even innocuous prompts may deliver images that skew more suggestively (particularly for prompts that hyper-sexualize women). Finally, as with all text-to-image models, the image produced will not always match the prompt. Hearteningly though, some of these scenarios can be addressed at a dataset level, and data filtering is also possible.

S8. Additional Automated Metrics

Automated metrics on Pick-a-Pic validation captions are shown in Figure S4 for DPO-SDXL. The y-axis measures the fraction of head-to-head generation comparisions for a prompt that DPO-SDXL scores higher than the baseline SDXL.

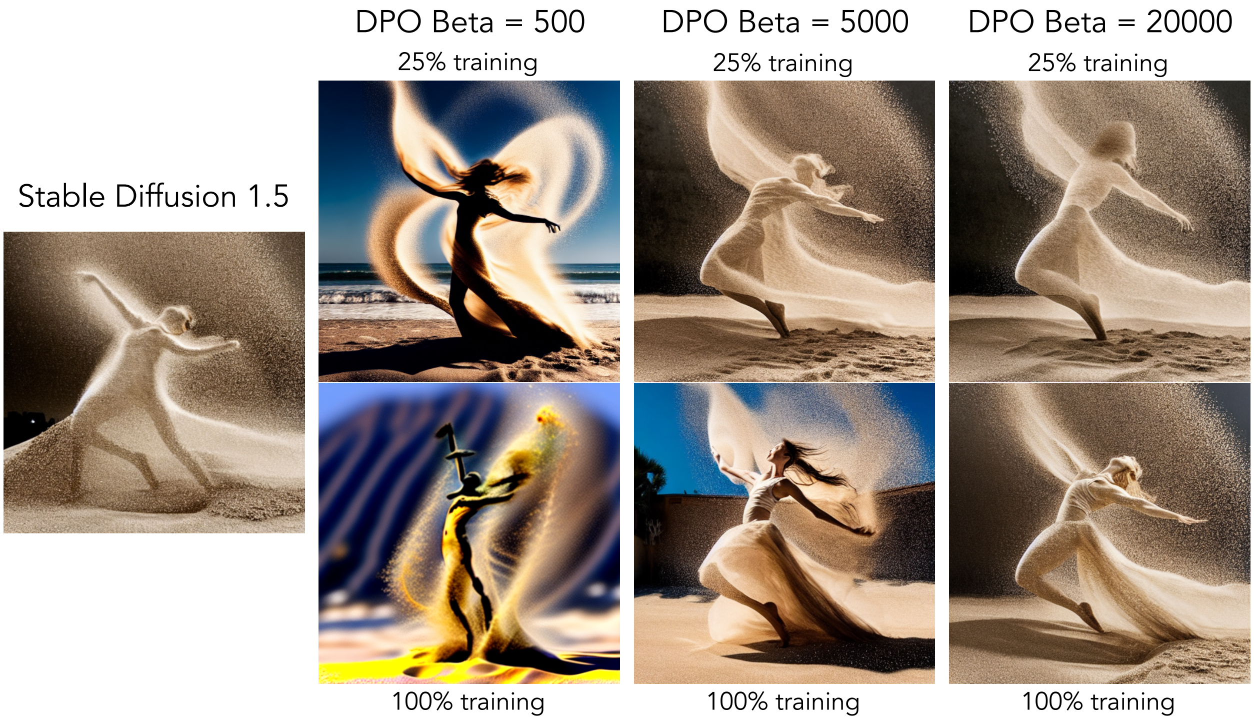

Figure S3. For too low of β, generations initially improve but subsequently degenerate. For sufficiently large β, generations remain similar to initialization.

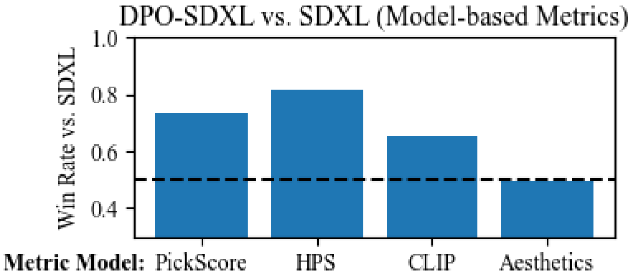

Figure S4. Box plots of automated metrics vs. SDXL baseline. All 500 unique prompts from PickScore validation set.

Property	Attribute binding			Obj. relationships		Complex
Benchmark	Color	Shape	Texture	Spatial	Non-spatial	Complex
Metric	B-VQA	B-VQA	B-VQA	UniDet	CLIP	3-in-1
SDXL	0.583	0.489	0.549	0.187	0.312	0.311
DPO-SDXL	**0.680**	**0.522**	**0.616**	**0.196**	**0.316**	**0.332**

Table S1. Proposed metrics from T2I-CompBench [20]

S9. PickScore Rejection Sampling

Rejection sampling was used in [24] as a powerful inference-time tool. 100 samples were drawn from variants of a prompt and PickScore-ranked, with the highest scored images being compared to a single random draw. PickScore selections were human-preferred 71.4% of the time. We compare using additional compute at inference vs. additional training in Figure S5. We plot the expected PickScore win rate of n draws from the reference model against a single draw from the learned (DPO) model. The mean inference compute for baseline rejection sampling to surpass the DPO-trained model is $10\times$ higher in both cases. For 7% (SDXL) and 16% (SD1.5) of the prompts even 100 draws is insufficient.

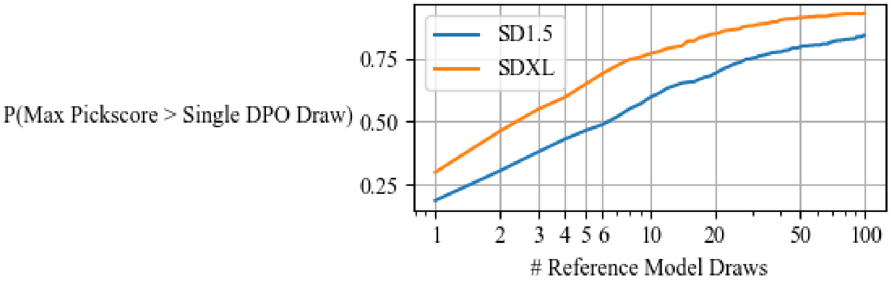

Figure S5. The number of draws from the reference model vs. the probability that maximum PickScore of the draws exceeds a single DPO generation. 500 PickScore validation prompts used. Mean (including 100s)/Median: SDXL $(13.7, 3)$, SD1.5 $(25.6, 7)$.

S10. Pseudocode for Training Objective

```
def loss(model, ref_model, x_w, x_l, c, beta):
    """
    # This is an example psuedo-code snippet for calculating the Diffusion-DPO loss
    # on a single image pair with corresponding caption

    model: Diffusion model that accepts prompt conditioning c and time conditioning t
    ref_model: Frozen initialization of model
    x_w: Preferred Image (latents in this work)
    x_l: Non-Preferred Image (latents in this work)
    c: Conditioning (text in this work)
    beta: Regularization Parameter

    returns: DPO loss value
    """
    timestep = torch.randint(0, 1000)
    noise = torch.randn_like(x_w)
    noisy_x_w = add_noise(x_w, noise, t)
    noisy_x_l = add_noise(x_l, noise, t)

    model_w_pred = model(noisy_x_w, c, t)
    model_l_pred = model(noisy_x_l, c, t)
    ref_w_pred = ref(noisy_x_w, c, t)
    ref_l_pred = ref(noisy_x_l, c, t)

    model_w_err = (model_w_pred - noise).norm().pow(2)
    model_l_err = (model_l_pred - noise).norm().pow(2)
    ref_w_err = (ref_w_pred - noise).norm().pow(2)
    ref_l_err = (ref_l_pred - noise).norm().pow(2)

    w_diff = model_w_err - ref_w_err
    l_diff = model_l_err - ref_l_err

    inside_term = -1 * beta * (w_diff - l_diff)

    loss = -1 * log(sigmoid(inside_term))

    return loss
```

S11. Additional Qualitative Results

In Figure S7 we present generations from DPO-SDXL on complex prompts from DALLE3 [17]. Other generations for miscellaneous prompts are shown in Figure S6. In Fig. S 8 and 9 we display qualitative comparison results from HPSv2 with random seeds from our human evaluation for prompt indices 200, 600, 1000, 1400, 1800, 2200, 2600, 3000.

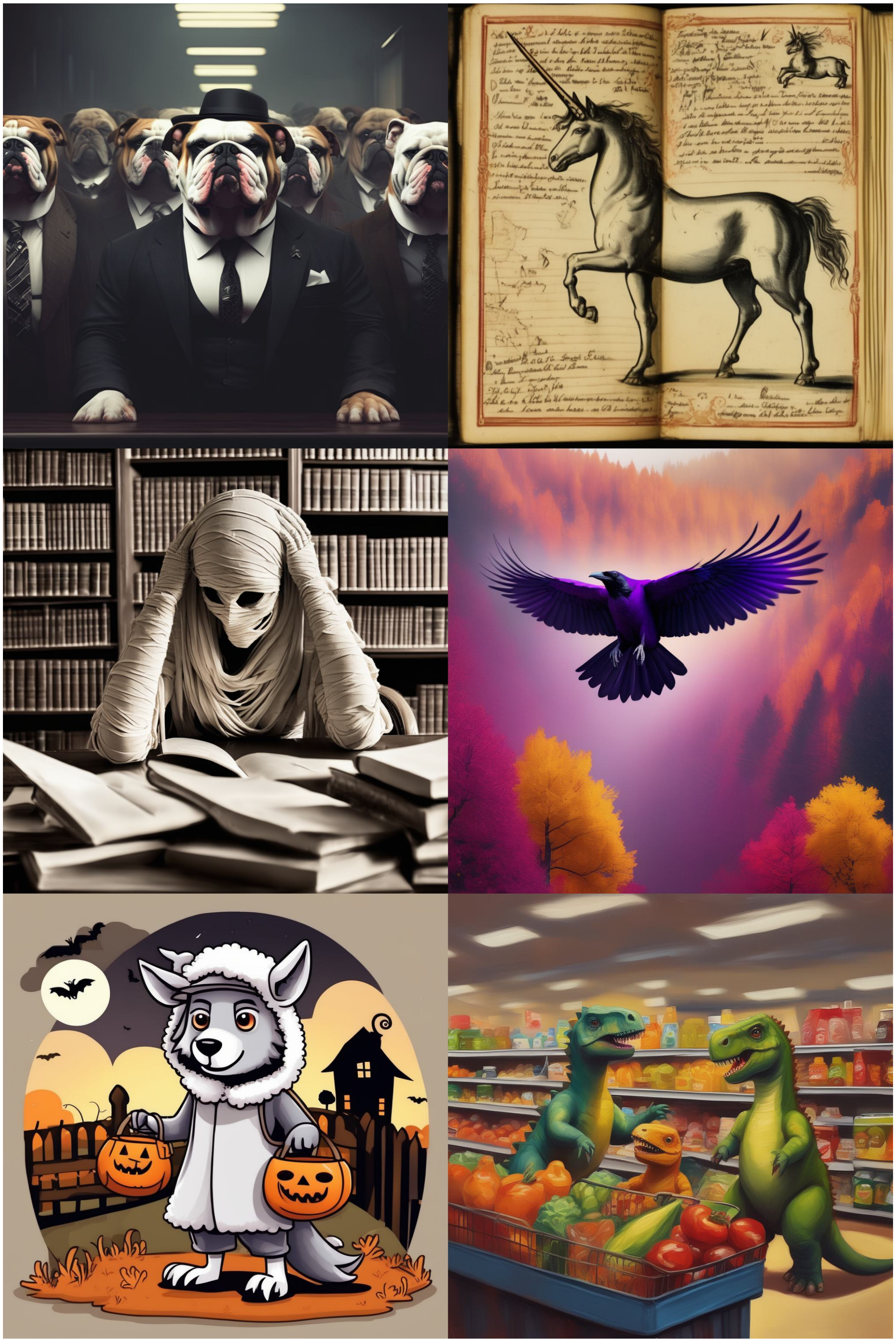

Figure S6. DPO-SDXL gens on miscellaneous prompts Prompts (clockwise) (1) A bulldog mob boss, moody cinematic feel (2) A old historical notebook detailing the discovery of unicorns (3) A purple raven flying over a forest of fall colors, imaginary documentary (4) Small dinosaurs shopping in a grocery store, oil painting (5) A wolf wearing a sheep halloween costume going trick-or-treating at the farm (6) A mummy studying hard in the library for finals, head in hands

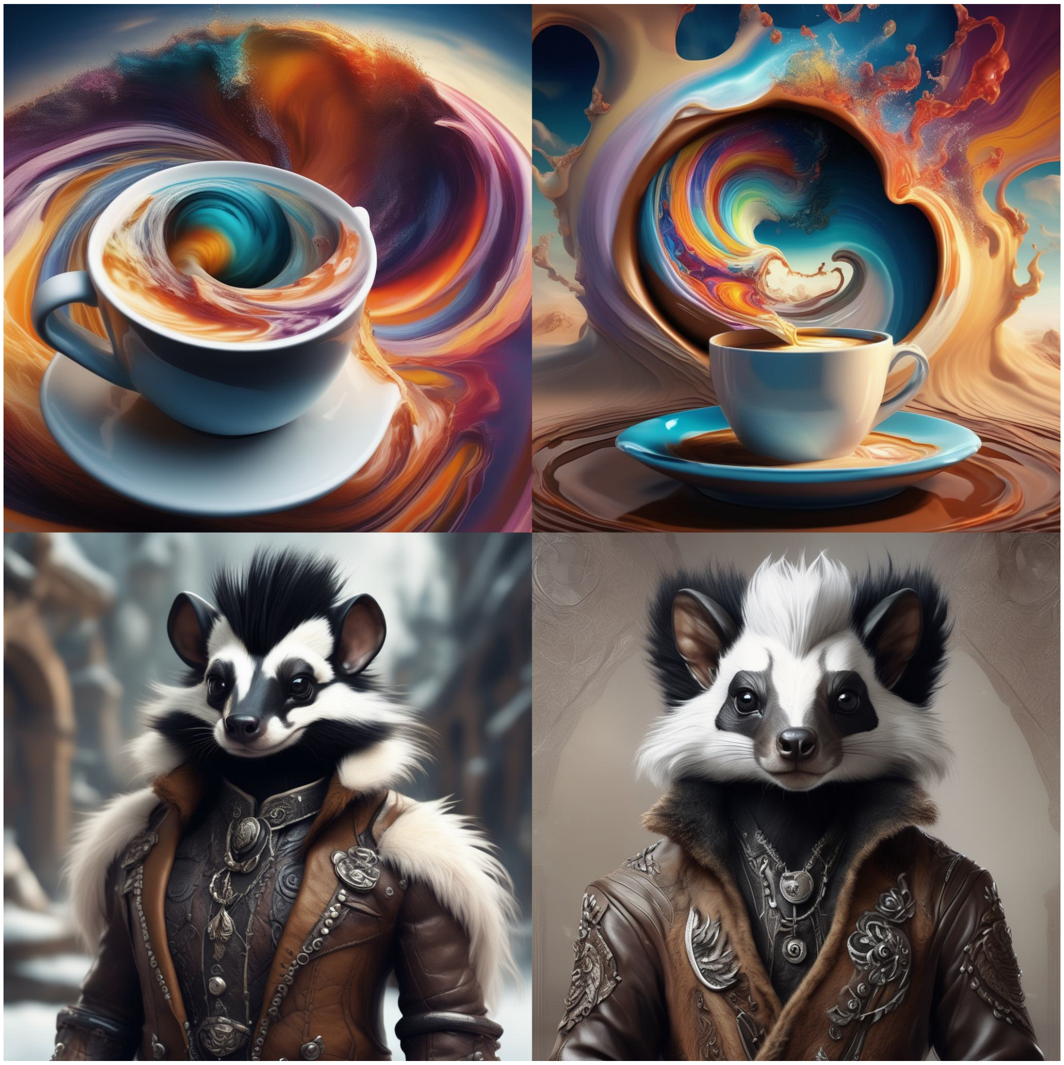

Figure S7. DPO-SDXL gens on prompts from DALLE3 [17] Prompts: (1): A swirling, multicolored portal emerges from the depths of an ocean of coffee, with waves of the rich liquid gently rippling outward. The portal engulfs a coffee cup, which serves as a gateway to a fantastical dimension. The surrounding digital art landscape reflects the colors of the portal, creating an alluring scene of endless possibilities. (2): In a fantastical setting, a highly detailed furry humanoid skunk with piercing eyes confidently poses in a medium shot, wearing an animal hide jacket. The artist has masterfully rendered the character in digital art, capturing the intricate details of fur and clothing texture"

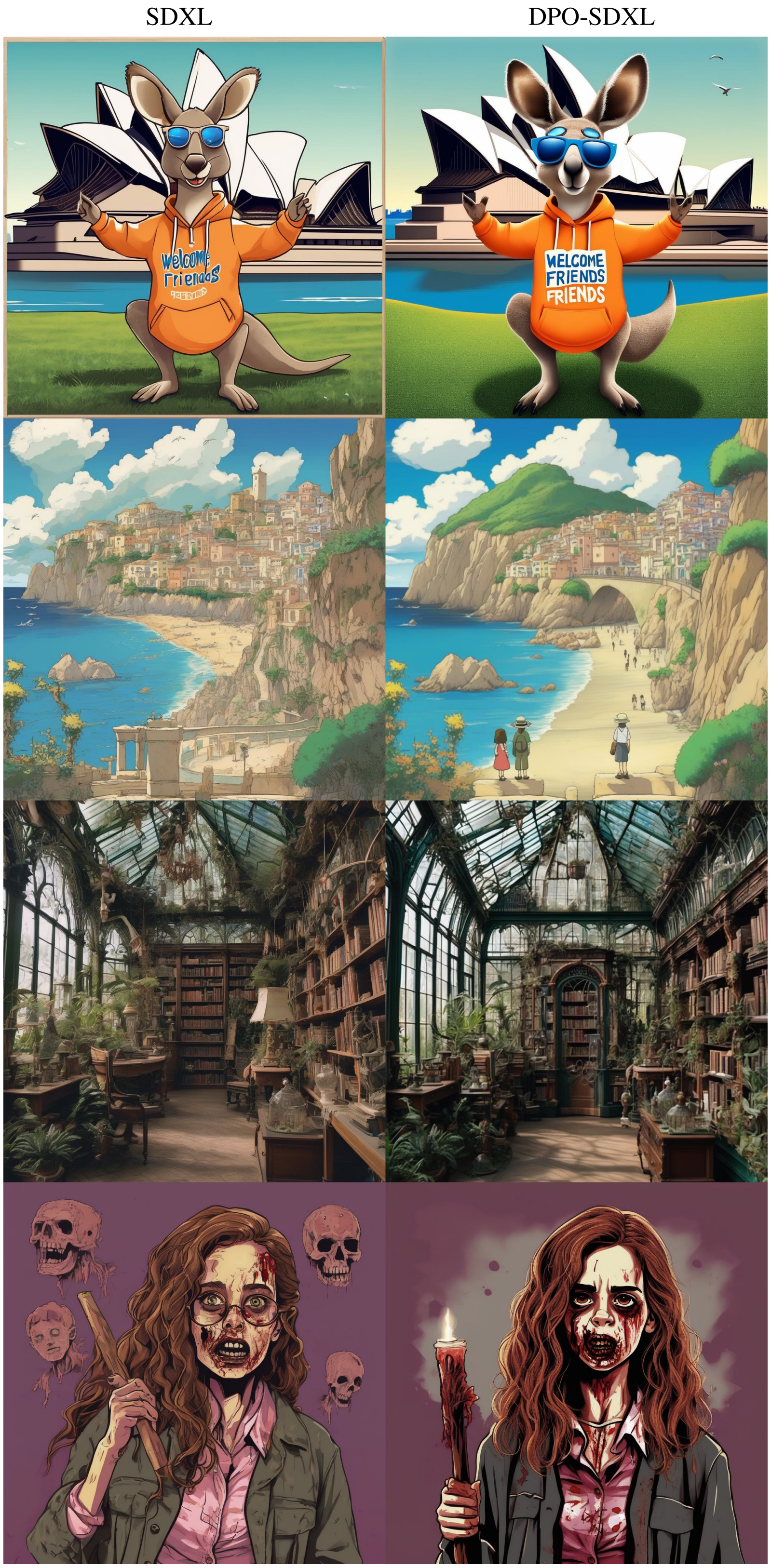

Figure S8. Prompts: (1) A kangaroo wearing an orange hoodie and blue sunglasses stands on the grass in front of the Sydney Opera House, holding a sign that says Welcome Friends. (2) Anime Costa Blanca by Studio Ghibli. (3) There is a secret museum of magical items inside a crystal greenhouse palace filled with intricate bookshelves, plants, and Victorian style decor. (4) A depiction of Hermione Granger from the Harry Potter series as a zombie.

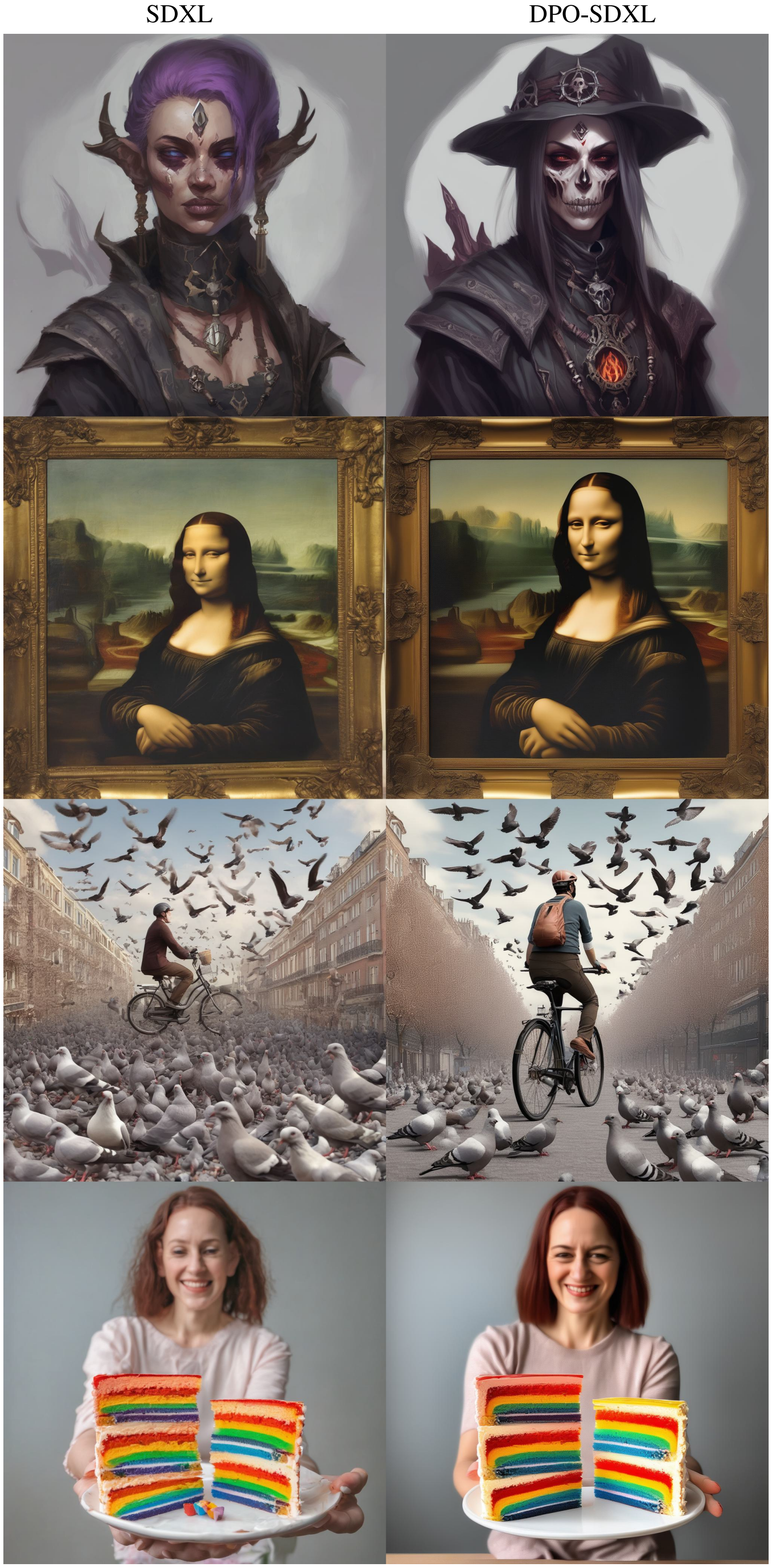

Figure S9. (1) A portrait art of a necromancer, referencing DND and War craft. (2) Monalisa painting a portrait of Leonardo Da Vinci. (3) There is a cyclist riding above all the pigeons. (4) A woman holding two rainbow slices of cake.

References

[1] https://huggingface.co/carperai/sd-2-1-pickscore-450epochs. 2

[2] Model index for researchers, 2023. 2

[3] Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Neural Information Processing Systems*, 2017. 2

[4] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021. 2

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. 2

[6] Michiel A. Bakker, Martin J. Chadwick, Hannah R. Sheahan, Michael Henry Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matthew M. Botvinick, and Christopher Summerfield. Fine-tuning language models to find agreement among humans with diverse preferences. *Neural Information processing systems*, 2022. 2

[7] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 1, 3, 4, 5

[8] Kevin Black et al. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 2

[9] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards, 2023. 1, 2, 3, 6

[10] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance, 2022. 1

[11] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 1, 3, 6, 2

[12] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2023. 2

[13] 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. *arXiv preprint arXiv:2305.16381*, 2023. 1, 3, 4, 5

[14] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022. 2

[15] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, Matthieu Geist, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Neural Information Processing Systems*, 2021. 5

[16] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022. 2

[17] Gabriel Goh, James Betker, Li Jing, Aditya Ramesh, Tim Brooks, Jianfeng Wang, Lindsey Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Prafulla Dhariwal, Casey Chu, Joy Jiao, Jong Wook Kim, Alex Nichol, Yang Song, Lijuan Wang, and Tao Xu. Improving image generation with better captions. 2023. 3, 11, 13

[18] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling, 2023. 2

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. pages 6840–6851, 2020. 3, 4

[20] Kaiyi Huang et al. T2i-compbench: A comprehensive benchmark for open-world compositional inage generation, 2023. 9

[21] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 6

[22] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023. 6

[23] Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. 2021. 3, 4

[24] 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. *arXiv preprint arXiv:2305.01569*, 2023. 3, 5, 9

[25] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback, 2023. 2, 6

[26] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018. 4

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[28] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 6, 8

[29] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022. 2

[30] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016. 2

[31] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 1

[32] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 2

[33] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 2, 3, 5, 1

[34] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. 1, 3

[35] 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*, pages 8748–8763. PMLR, 2021. 6, 1

[36] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023. 1, 2, 3, 4, 5, 8

[37] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization, 2022. 2

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable diffusion 2. https://huggingface.co/stabilityai/stable-diffusion-2. Accessed: 2023 - 11 - 16. 1

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1, 3, 5

[40] Christoph Schuhmann. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics/, 2022. Accessed: 2023 - 11-10. 3, 6, 1

[41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 2

[42] Eyal Segalis, Dani Valevski, Danny Lumen, Yossi Matias, and Yaniv Leviathan. A picture is worth a thousand words: Principled recaptioning improves image generation, 2023. 3

[43] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR, 2018. 5

[44] Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. *Neural Information Processing Systems*, 2022. 2

[45] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265, 2015. 3, 4

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 6

[47] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 3

[48] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Neural Information Processing Systems*, 2021. 3

[49] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 3

[50] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. *Neural Information Processing Systems*, 18, 2020. 2

[51] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 1

[52] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023. 2

[53] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback, 2022. 2

[54] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. *arXiv preprint arXiv:2303.13703*, 2023. 3, 1

[55] 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. 3, 5, 6

[56] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Hpsv2 github. https://github.com/tgxs002/HPSv2/tree/master, 2023. Accessed: 2023 - 11 - 15. 5

[57] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 6

[58] 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. *arXiv preprint arXiv:2304.05977*, 2023. 3

[59] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022. 5

[60] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *Neural Information Processing Systems*, 2023. 2

[61] Yinan Zhang et al. Large-scale reinforcement learning for diffusion models, 2024. 6, 1

[62] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. Slic-hf: Sequence likelihood calibration with human feedback, 2023. 2

[63] Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of rlhf in large language models part i: Ppo, 2023. 2

[64] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020. 6