

---

# Training Diffusion Models with Reinforcement Learning

---

Kevin Black<sup>\* 1</sup> Michael Janner<sup>\* 1</sup> Yilun Du<sup>2</sup> Ilya Kostrikov<sup>1</sup> Sergey Levine<sup>1</sup>

<sup>1</sup> University of California, Berkeley <sup>2</sup> Massachusetts Institute of Technology  
`{kvablock, janner, kostrikov, sergey.levine}@berkeley.edu` `yilundu@mit.edu`

## Abstract

Diffusion models are a class of flexible generative models trained with an approximation to the log-likelihood objective. However, most use cases of diffusion models are not concerned with likelihoods, but instead with downstream objectives such as human-perceived image quality or drug effectiveness. In this paper, we investigate reinforcement learning methods for directly optimizing diffusion models for such objectives. We describe how posing denoising as a multi-step decision-making problem enables a class of policy gradient algorithms, which we refer to as denoising diffusion policy optimization (DDPO), that are more effective than alternative reward-weighted likelihood approaches. Empirically, DDPO is able to adapt text-to-image diffusion models to objectives that are difficult to express via prompting, such as image compressibility, and those derived from human feedback, such as aesthetic quality. Finally, we show that DDPO can improve prompt-image alignment using feedback from a vision-language model without the need for additional data collection or human annotation.

## 1 Introduction

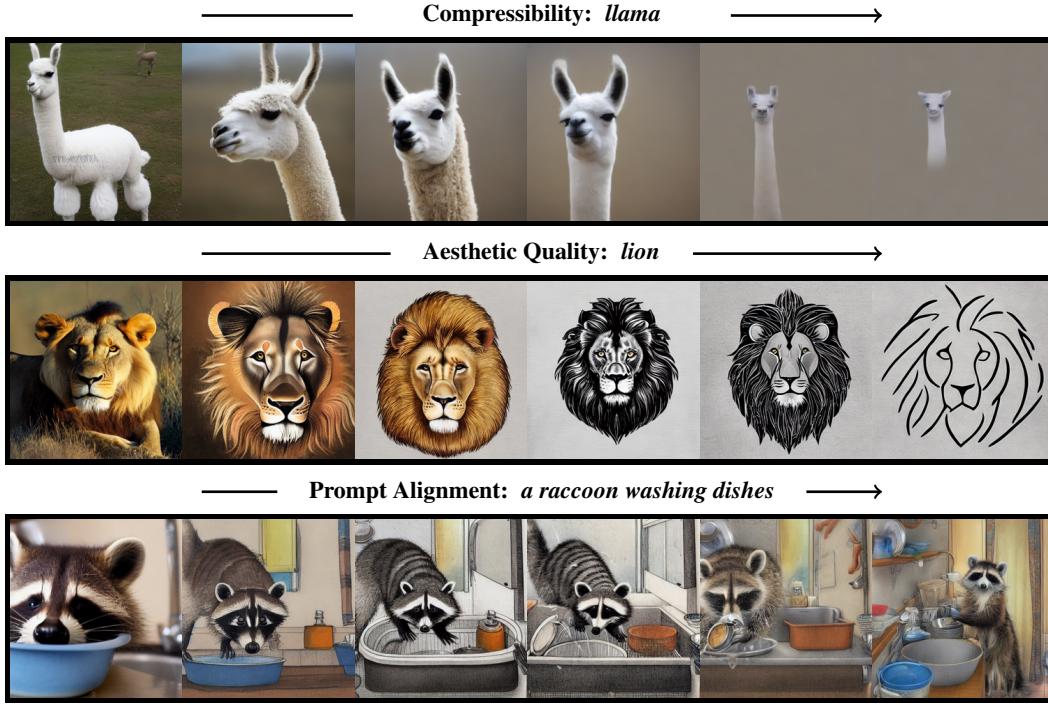
Diffusion probabilistic models [47] have recently emerged as the de facto standard for generative modeling in continuous domains. Their flexibility in representing complex, high-dimensional distributions has led to the adoption of diffusion models in applications including image and video synthesis [38, 41, 20], drug and material design [56, 55, 42], and continuous control [21, 52, 15]. The key idea behind diffusion models is to iteratively transform a simple prior distribution into a target distribution by applying a sequential denoising process. This procedure is conventionally motivated as a maximum likelihood estimation problem, with the objective derived as a variational lower bound on the model log-likelihood.

However, most use cases of diffusion models are not directly with likelihoods, but instead on a downstream objective such as human-perceived image quality or drug effectiveness. In this paper, we consider the problem of training diffusion models to satisfy such objectives directly, as opposed to matching a data distribution. This problem is challenging because exact likelihood computation with diffusion models is intractable, making it difficult to apply many conventional reinforcement learning (RL) algorithms. We instead propose to frame denoising as a multi-step decision-making task, using the exact likelihoods at each denoising step in place of the approximate likelihoods induced by a full denoising process. We then devise a policy gradient algorithm, which we refer to as denoising diffusion policy optimization (DDPO), that can optimize a diffusion model for downstream tasks using only a black-box reward function.

---

<sup>\*</sup>Denotes equal contribution.

More results on our project page: [rl-diffusion.github.io](https://rl-diffusion.github.io).



**Figure 1 (Reinforcement learning for diffusion models)** We propose a reinforcement learning algorithm, DDPO, for optimizing diffusion models on downstream objectives such as compressibility, aesthetic quality, and prompt-image alignment as determined by vision-language models. Each row shows a progression of samples for the same prompt and random seed over the course of training.

We apply our algorithm to the finetuning of large pretrained text-to-image diffusion models. Our initial evaluation focuses on tasks that are difficult to specify via prompting, such as image compressibility, and those derived from human feedback, such as aesthetic quality. However, because many reward functions of interest are difficult to specify programmatically, finetuning procedures often rely on large-scale human labeling efforts to obtain a reward signal. In the case of text-to-image diffusion, we propose a method for replacing such labeling with feedback from a vision-language model (VLM). Similar to RLAIF finetuning for language models [4], the resulting procedure allows for diffusion models to be adapted to reward functions that would otherwise require additional human annotations. We use this procedure to improve prompt-image alignment for unusual subject-setting compositions.

Our contributions are as follows. We first present the derivation and conceptual motivation of DDPO. We then document the design of various reward functions for text-to-image generation, ranging from simple computations to workflows involving large VLMs, and demonstrate the effectiveness of DDPO compared to alternative reward-weighted likelihood methods in these settings. Finally, we study the generalization and overoptimization effects resulting from our finetuning procedures.

## 2 Related Work

Denoising diffusion models [47, 19] have emerged as an effective class of generative models for modalities including images [38, 41], videos [20, 46], 3D shapes [60, 57], and robotic trajectories [21, 1, 6]. While the denoising objective is conventionally derived as an approximation to likelihood, the training of diffusion models typically departs from maximum likelihood in several ways that improve sample quality in practice [19]. Modifying the objective to more strictly optimize likelihood [33, 23] often leads to worsened image quality, as likelihood is not a faithful proxy for visual quality. In this paper, we show how diffusion models can be optimized directly for downstream objectives.

Recent progress in text-to-image diffusion models [38, 41] has enabled fine-grained high-resolution image synthesis. To further improve the controllability and quality of diffusion models, recent approaches have investigated finetuning on limited user-provided data [40], optimizing text embeddings

for new concepts [12], composing models [11, 27], adapters for additional input constraints [58], and inference-time techniques such as classifier [10] and classifier-free [18] guidance.

A number of works have studied using human feedback to optimize models in settings such as simulated robotic control [7], game-playing [24], machine translation [32], citation retrieval [28], browsing-based question-answering [31], summarization [49, 61], instruction-following [34], and alignment with specifications [3]. Recently, Lee et al. [25] studied the alignment of text-to-image diffusion models to human preferences using a method based on reward-weighted likelihood maximization, and posited that finetuning with RL is a promising direction for future work. In our comparisons, their method roughly corresponds to one iteration of the RWR method, though precise implementation details are likely different. Our results demonstrate that DDPO significantly outperforms even multiple iterations of weighted likelihood maximization (RWR-style) optimization. More generally, our aim is not to study learning from human feedback per se, but general algorithms compatible with a variety of reward functions.

### 3 Preliminaries

In this section, we provide a brief background on diffusion models and the RL problem formulation.

#### 3.1 Diffusion Models

In this work, we consider conditional diffusion probabilistic models [47, 19], which represent a distribution over data  $\mathbf{x}_0$  conditioned on context  $\mathbf{c}$  as the result of sequential denoising. The denoising procedure is trained to reverse a Markovian forward process  $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ , which iteratively adds noise to the data. Reversing the forward process can be accomplished by training a forward process posterior mean predictor  $\mu_\theta(\mathbf{x}_t, t, \mathbf{c})$  for all  $t \in \{0, 1, \dots, T\}$  with the following simplified objective:

$$\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E} [\|\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t, \mathbf{c})\|^2] \quad (1)$$

where  $\tilde{\mu}$  is a weighted average of  $\mathbf{x}_0$  and  $\mathbf{x}_t$ . This objective is justified as maximizing a variational lower bound on the model log-likelihood [19].

Sampling from a diffusion model begins with sampling  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and using the reverse process  $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})$  to produce a trajectory  $\{\mathbf{x}_T, \mathbf{x}_{T-1}, \dots, \mathbf{x}_0\}$  ending with a sample  $\mathbf{x}_0$ . The reverse process depends not only on the predictor  $\mu_\theta$  but also the choice of sampler. Most popular samplers [19, 48] use an isotropic Gaussian reverse process with a fixed timestep-dependent variance:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{x}_t, t, \mathbf{c}), \sigma_t^2 \mathbf{I}). \quad (2)$$

#### 3.2 Markov Decision Processes and Reinforcement Learning

A Markov decision process (MDP) is a formalization of sequential decision-making problems. An MDP is defined by a tuple  $(\mathcal{S}, \mathcal{A}, \rho_0, P, R)$ , in which  $\mathcal{S}$  is the state space,  $\mathcal{A}$  is the action space,  $\rho_0$  is the distribution of initial states,  $P$  is the transition kernel, and  $R$  is the reward function. At each timestep  $t$ , the agent observes a state  $\mathbf{s}_t \in \mathcal{S}$ , takes an action  $\mathbf{a}_t \in \mathcal{A}$ , receives a reward  $R(\mathbf{s}_t, \mathbf{a}_t)$ , and transitions to a new state  $\mathbf{s}_{t+1} \sim P(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$ . An agent acts according to a policy  $\pi(\mathbf{a} \mid \mathbf{s})$ .

As the agent acts in the MDP, it produces trajectories, which are sequences of states and actions  $\tau = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_T, \mathbf{a}_T)$ . The reinforcement learning (RL) objective for the agent is to maximize  $\mathcal{J}_{\text{RL}}(\pi)$ , the expected cumulative reward over trajectories sampled from its policy:

$$\mathcal{J}_{\text{RL}}(\pi) = \mathbb{E}_{\tau \sim p(\cdot \mid \pi)} \left[ \sum_{t=0}^T R(\mathbf{s}_t, \mathbf{a}_t) \right].$$

### 4 Reinforcement Learning Training of Diffusion Models

We now describe how RL algorithms can be used to train diffusion models. We present two classes of methods, one based on prior work and one novel, and show that each corresponds to a different mapping of the denoising process to the MDP framework.

## 4.1 Problem Statement

We assume a pre-existing diffusion model, which may be pretrained or randomly initialized. If we choose a fixed sampler, the diffusion model induces a sample distribution  $p_\theta(\mathbf{x}_0 \mid \mathbf{c})$ . The denoising diffusion RL objective is to maximize a reward signal  $r$  defined on the samples and contexts:

$$\mathcal{J}_{\text{DDRL}}(\theta) = \mathbb{E}_{\mathbf{c} \sim p(\mathbf{c}), \mathbf{x}_0 \sim p_\theta(\cdot \mid \mathbf{c})} [r(\mathbf{x}_0, \mathbf{c})]$$

for some context distribution  $p(\mathbf{c})$  of our choosing.

## 4.2 Reward-Weighted Regression

To optimize  $\mathcal{J}_{\text{DDRL}}$  with minimal changes to standard diffusion model training, we can use the denoising objective  $\mathcal{L}_{\text{DDPM}}$  (Equation 1), but with training data sampled from the model itself and a per-sample loss weighting that depends on the reward  $r(\mathbf{x}_0, \mathbf{c})$ . Lee et al. [25] describe a single-round version of this procedure for diffusion models, but in general this approach can be performed for multiple rounds of alternating sampling and training, leading to a simple RL method. We refer to this general class of algorithms as reward-weighted regression (RWR) [36].

A standard weighting scheme uses exponentiated rewards to ensure nonnegativity,

$$w_{\text{RWR}}(\mathbf{x}_0, \mathbf{c}) = \frac{1}{Z} \exp(\beta R(\mathbf{x}_0, \mathbf{c})),$$

where  $\beta$  is an inverse temperature and  $Z$  is a normalization constant. We also consider a simplified weighting scheme that uses binary weights,

$$w_{\text{sparse}}(\mathbf{x}_0, \mathbf{c}) = \mathbb{1}[R(\mathbf{x}_0, \mathbf{c}) \geq C],$$

where  $C$  is a reward threshold determining which samples are used for training. The sparse weights may be desirable because they eliminate the need to retain every sample from the model.

Within the RL formalism, the RWR procedure corresponds to the following one-step MDP:

$$\mathbf{s} \triangleq \mathbf{c} \quad \mathbf{a} \triangleq \mathbf{x}_0 \quad \pi(\mathbf{a} \mid \mathbf{s}) \triangleq p_\theta(\mathbf{x}_0 \mid \mathbf{c}) \quad \rho_0(\mathbf{s}) \triangleq p(\mathbf{c}) \quad R(\mathbf{s}, \mathbf{a}) \triangleq r(\mathbf{x}_0, \mathbf{c})$$

with a transition kernel  $P$  that immediately leads to an absorbing termination state. Therefore, maximizing  $\mathcal{J}_{\text{DDRL}}(\theta)$  is equivalent to maximizing  $\mathcal{J}_{\text{RL}}(\pi)$  in this MDP.

Weighting a maximum likelihood objective by  $w_{\text{RWR}}$  approximately optimizes  $\mathcal{J}_{\text{RL}}(\pi)$  subject to a KL divergence constraint on the policy [30]. However,  $\mathcal{L}_{\text{DDPM}}$  is not an exact maximum likelihood objective, but is derived from a reweighted variational bound. Therefore, RWR algorithms applied to  $\mathcal{L}_{\text{DDPM}}$  optimize  $\mathcal{J}_{\text{DDRL}}$  via two levels of approximation. Thus, this methodology provides us with a starting point, but might underperform for complex objectives.

## 4.3 Denoising Diffusion Policy Optimization

RWR relies on an approximate maximum likelihood objective because it ignores the sequential nature of the denoising process, only using the final samples  $\mathbf{x}_0$ . In this section, we show that when the sampler is fixed, the denoising process can be reframed as a *multi-step* MDP. This allows us to directly optimize  $\mathcal{J}_{\text{DDRL}}$  using policy gradient estimators. We refer to the resulting class of algorithms as denoising diffusion policy optimization (DDPO) and present two variants.

**Denoising as a multi-step MDP.** We map the iterative denoising procedure to the following MDP:

$$\begin{aligned} s_t &\triangleq (\mathbf{c}, t, \mathbf{x}_t) & \pi(a_t \mid s_t) &\triangleq p_\theta(x_{t-1} \mid x_t, \mathbf{c}) & P(s_{t+1} \mid s_t, a_t) &\triangleq (\delta_c, \delta_{t-1}, \delta_{x_{t-1}}) \\ a_t &\triangleq \mathbf{x}_{t-1} & \rho_0(s_0) &\triangleq (p(\mathbf{c}), \delta_T, \mathcal{N}(\mathbf{0}, \mathbf{I})) & R(s_t, a_t) &\triangleq \begin{cases} r(\mathbf{x}_0, \mathbf{c}) & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

in which  $\delta_y$  is the Dirac delta distribution with nonzero density only at  $y$ . Trajectories consist of  $T$  timesteps, after which  $P$  leads to a termination state. The cumulative reward of each trajectory is equal to  $r(\mathbf{x}_0, \mathbf{c})$ , so maximizing  $\mathcal{J}_{\text{DDRL}}(\theta)$  is equivalent to maximizing  $\mathcal{J}_{\text{RL}}(\pi)$  in this MDP.

The benefit of this formulation is that, if we use a standard sampler parameterized as in Equation 2, the policy  $\pi$  becomes an isotropic Gaussian as opposed to an arbitrarily complicated distribution

induced by the entire denoising procedure. This simplification allows for the evaluation of exact action likelihoods and gradients of these likelihoods with respect to the diffusion model parameters.

**Policy gradient estimation.** With access to likelihoods and likelihood gradients, we can make Monte Carlo estimates of the policy gradient  $\nabla_{\theta} \mathcal{J}_{\text{DDRL}}$ . DDPO alternates collecting trajectories  $\{\mathbf{x}_T, \mathbf{x}_{T-1}, \dots, \mathbf{x}_0\}$  via sampling and updating parameters via gradient ascent on  $\mathcal{J}_{\text{DDRL}}$ .

The first variant of DDPO, which we call DDPO<sub>SF</sub>, uses the score function policy gradient estimator, also known as the likelihood ratio method or REINFORCE [53, 29]:

$$\hat{g}_{\text{SF}} = \mathbb{E} \left[ \sum_{t=0}^T \nabla_{\theta} \log p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{c}, t, \mathbf{x}_t) r(\mathbf{x}_0, \mathbf{c}) \right] \quad (3)$$

where the expectation is taken over denoising trajectories generated by the current policy  $p_{\theta}$ .

This estimator is unbiased. However, it only allows for one step of optimization per round of data collection, as the gradients must be estimated using data from the current policy. To perform multiple steps of optimization, we may use an importance sampling estimator [22]:

$$\hat{g}_{\text{IS}} = \mathbb{E} \left[ \sum_{t=0}^T \frac{p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{c}, t, \mathbf{x}_t)}{p_{\theta_{\text{old}}}(\mathbf{x}_{t-1} \mid \mathbf{c}, t, \mathbf{x}_t)} \nabla_{\theta} \log p_{\theta}(\mathbf{x}_{t-1} \mid \mathbf{c}, t, \mathbf{x}_t) r(\mathbf{x}_0, \mathbf{c}) \right] \quad (4)$$

where  $\theta_{\text{old}}$  are the parameters used to collect the data, and the expectation is taken over denoising trajectories generated by the corresponding policy  $p_{\theta_{\text{old}}}$ . This estimator also becomes inaccurate if  $p_{\theta}$  deviates too far from  $p_{\theta_{\text{old}}}$ , which can be addressed using trust regions [44] to constrain the size of the update. In practice, we implement the trust region by clipping the importance weights, as introduced in proximal policy optimization [45]. We call this variant DDPO<sub>IS</sub>.

## 5 Reward Functions for Text-to-Image Diffusion

In this work, we evaluate our methods on text-to-image diffusion. Text-to-image diffusion serves as a valuable test environment for reinforcement learning experiments due to the availability of large pretrained models and the versatility of using diverse and visually interesting reward functions.

The choice of reward function is one of the most important decisions in practical applications of RL. In this section, we outline our selection of reward functions for text-to-image diffusion models. We study a spectrum of reward functions of varying complexity, ranging from those that are straightforward to specify and evaluate to those that capture the complexity of real-world downstream tasks.

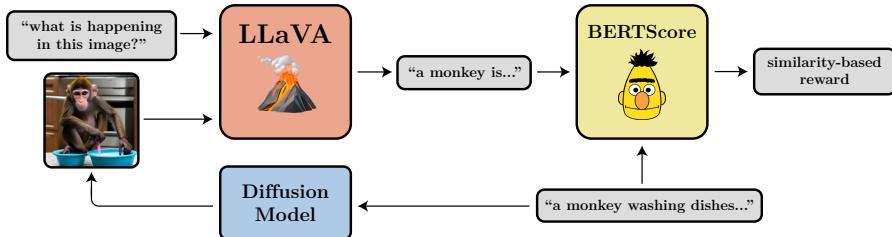
### 5.1 Compressibility and Incompressibility

The capabilities of text-to-image diffusion models are limited by the co-occurrences of text and images in their training distribution. For instance, images are rarely captioned with their file size, making it impossible to specify a desired file size via prompting. This limitation makes reward functions based on file size a convenient case study: they are simple to compute, but not controllable through the conventional workflow of likelihood maximization and prompt engineering.

We fix the resolution of diffusion model samples at 512x512, such that the file size is determined solely by the compressibility of the image. We define two tasks based on file size: compressibility, in which the file size of the image after JPEG compression is minimized, and incompressibility, in which the same measure is maximized.

### 5.2 Aesthetic Quality

To capture a reward function that would be useful to a human user, we define a task based on perceived aesthetic quality. We use the LAION aesthetics predictor [43], which is trained on 176,000 human image ratings. The predictor is implemented as a linear model on top of CLIP embeddings [37]. Annotations range between 1 and 10, with the highest-rated images mostly containing artwork. Since the aesthetic quality predictor is trained on human judgments, this task constitutes reinforcement learning from human feedback [34, 7, 61].



**Figure 2 (VLM reward function)** Illustration of the VLM-based reward function for prompt-image alignment. LLaVA [26] provides a short description of a generated image; the reward is the similarity between this description and the original prompt as measured by BERTScore [59].

### 5.3 Automated Prompt Alignment with Vision-Language Models

A very general-purpose reward function for training a text-to-image model is prompt-image alignment. However, specifying a reward that captures generic prompt alignment is difficult, conventionally requiring large-scale human labeling efforts. We propose using an existing VLM to replace additional human annotation. This design is inspired by recent work on RLAIF [4], in which language models are improved using feedback from themselves.

We use LLaVA [26], a state-of-the-art VLM, to describe an image. The finetuning reward is the BERTScore [59] recall metric, a measure of semantic similarity, using the prompt as the reference and the VLM description as the candidate. Samples that more faithfully include all of the details of the prompt receive higher rewards, to the extent that those visual details are legible to the VLM.

In Figure 2, we show one simple question: “*what is happening in this image?*”. While this captures the general task of prompt-image alignment, in principle any question could be used to specify complex or hard-to-define reward functions for a particular use case. One could even employ a language model to automatically generate candidate questions and evaluate responses based on the prompt. This framework provides a flexible interface where the complexity of the reward function is only limited by the capabilities of the vision and language models involved.

## 6 Experimental Evaluation

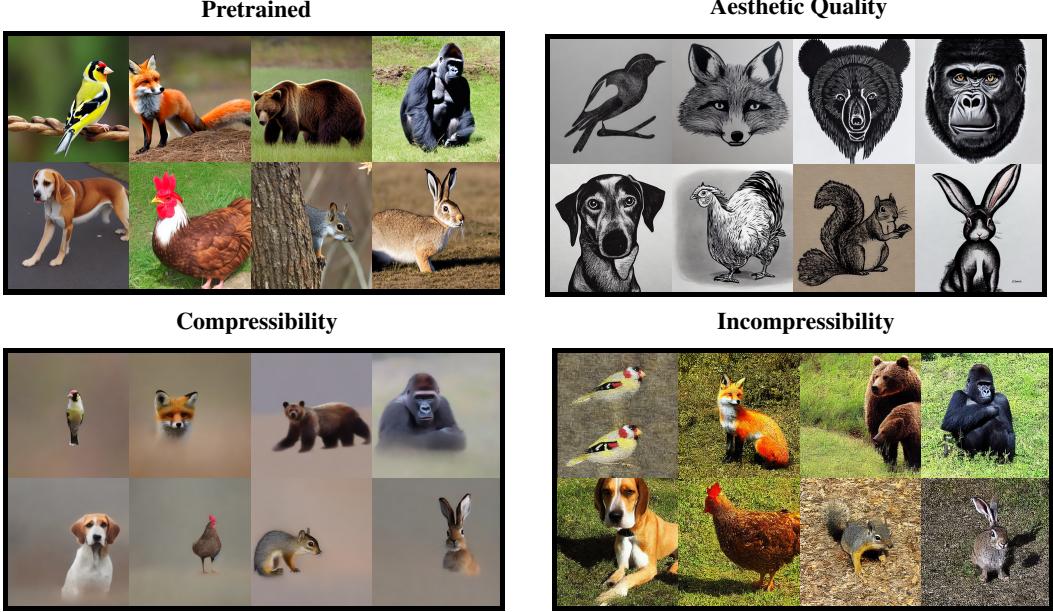
The purpose of our experiments is to evaluate the effectiveness of RL algorithms for finetuning diffusion models to align with a variety of user-specified objectives. After examining the viability of the general approach, we focus on the following questions:

1. How do variants of DDPO compare to RWR and to each other?
2. Can VLMs allow for optimizing rewards that are difficult to specify manually?
3. Do the effects of RL finetuning generalize to prompts not seen during finetuning?
4. Does RL finetuning lead to reward overoptimization?

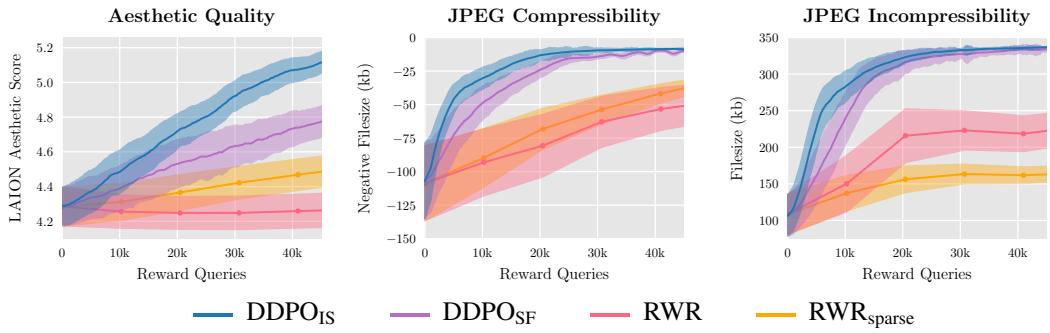
### 6.1 Algorithm Comparisons

We begin by evaluating all methods on the compressibility, incompressibility, and aesthetic quality tasks, as these tasks isolate the effectiveness of the RL approach from considerations relating to automated VLM reward evaluation. We use Stable Diffusion v1.4 [39] as the base model for all experiments. Compressibility and incompressibility prompts are sampled uniformly from all 398 animals in the ImageNet-1000 [9] categories. Aesthetic quality prompts are sampled uniformly from a smaller set of 45 common animals.

As shown qualitatively in Figure 3, DDPO is able to effectively adapt a pretrained model with only the specification of a reward function and without any further data curation. The strategies found to optimize each reward are nontrivial; for example, to maximize LAION-predicted aesthetic quality, DDPO transforms a model that produces naturalistic images into one that produces stylized line drawings. To maximize compressibility, DDPO removes backgrounds and applies a Gaussian blur to



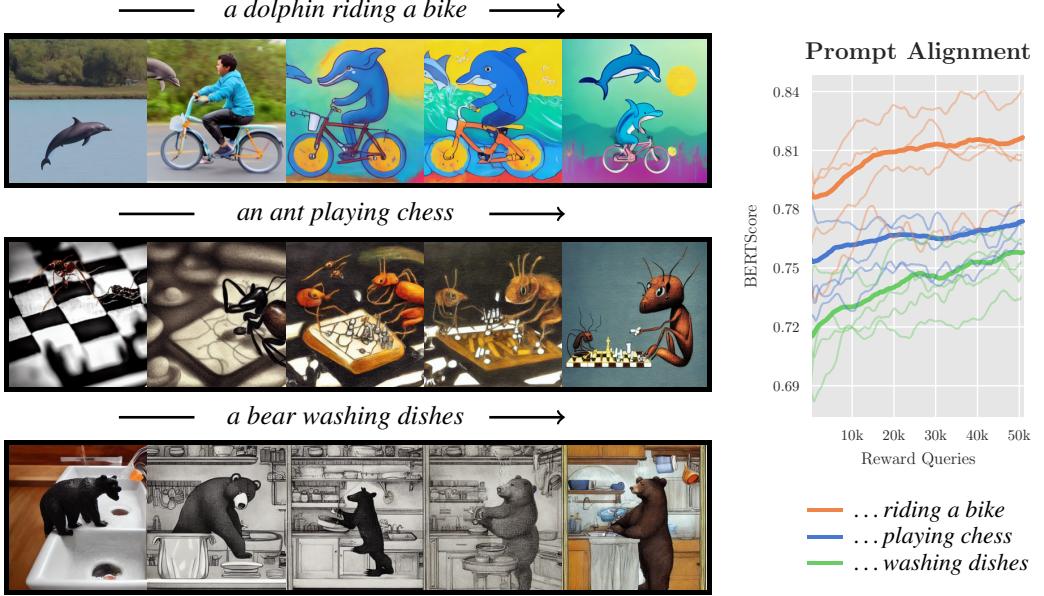
**Figure 3 (DDPO samples)** Qualitative depiction of the effects of RL fine-tuning on different reward functions. DDPO transforms naturalistic images into stylized line drawings to maximize predicted aesthetic quality, removes background content and applies a foreground blur to maximize compressibility, and adds artifacts and high-frequency noise to maximize incompressibility.



**Figure 4 (Finetuning effectiveness)** The relative effectiveness of different RL algorithms on three reward functions. We find that the policy gradient variants, denoted DDPO, are more effective optimizers than both RWR variants.

what remains. To maximize incompressibility, DDPO finds artifacts that are difficult for the JPEG compression algorithm to encode, such as high-frequency noise and sharp edges, and occasionally produces multiple entities. Samples from RWR are provided in Appendix C for comparison.

We provide a quantitative comparison of all methods in Figure 4. We plot the attained reward as a function of the number of queries to the reward function, as reward evaluation becomes the limiting factor in many practical applications. DDPO shows a clear advantage over RWR on all tasks, demonstrating that formulating the denoising process as an MDP and estimating the policy gradient directly is more effective than optimizing a reward-weighted lower bound on likelihood. Within the DDPO class, the importance sampling estimator slightly outperforms the score function estimator, likely due to the increased number of optimization steps. Within the RWR class, the performance of weighting schemes is comparable, making the sparse weighting scheme preferable on these tasks due to its simplicity and reduced resource requirements.



**Figure 5 (Prompt alignment)** (L) Progression of samples for the same prompt and random seed over the course of training. The images become significantly more faithful to the prompt. The samples also adopt a cartoon-like style, which we hypothesize is because the prompts are more likely depicted as illustrations than realistic photographs in the pretraining distribution. (R) Quantitative improvement of prompt alignment. Each thick line is the average score for an activity, while the faint lines show average scores for a few randomly selected individual prompts.

## 6.2 Automated Prompt Alignment

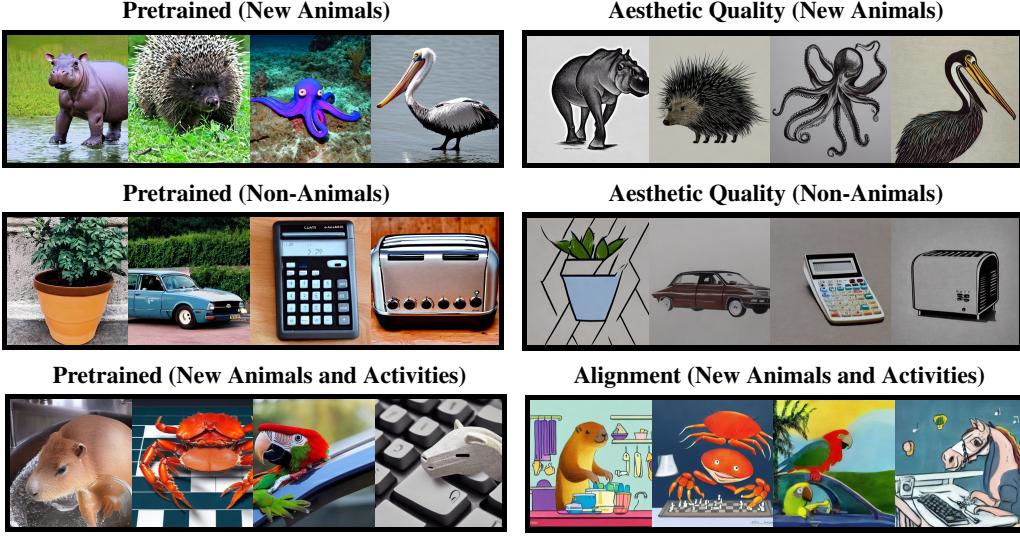
We next evaluate the ability of VLMs, in conjunction with DDPO, to automatically improve the image-prompt alignment of the pretrained model without additional human labels. We focus on DDPO<sub>IS</sub> for this experiment, as we found it to be the most effective algorithm in Section 6.1. The prompts for this task all have the form “*a(n) [animal] [activity]*”, where the animal comes from the same list of 45 common animals used in Section 6.1 and the activity is chosen from a list of 3 activities: “*riding a bike*”, “*playing chess*”, and “*washing dishes*”.

The progression of finetuning is depicted in Figure 5. Qualitatively, the samples come to depict the prompts much more faithfully throughout the course of training. This trend is also reflected quantitatively, though less salient as we found that even small changes in average BERTScore [59] could correspond to large differences in quality. It is important to note that some of the prompts in the finetuning set, such as “*a dolphin riding a bike*”, had zero success rate from the base model; if trained in isolation, this prompt would be unlikely to ever improve because there would be no reward signal. It was only via transfer between prompts that these particular prompts could improve.

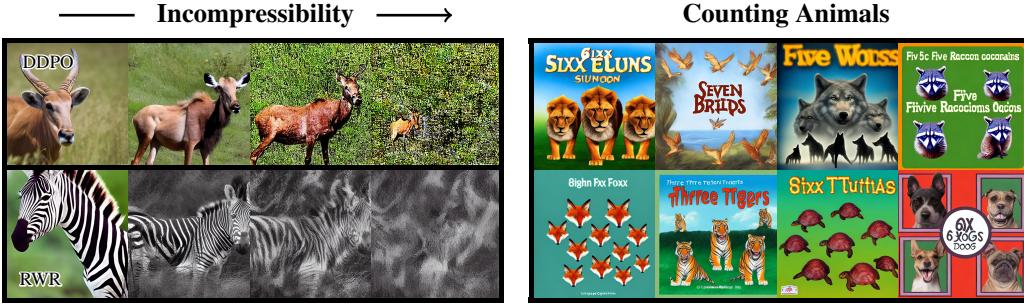
Nearly all of the samples become more cartoon-like or artistic during finetuning. This was not optimized for directly. We hypothesize that this is a function of the pretraining distribution; though it would be extremely rare to see a photorealistic image of a bear washing dishes, it would be much less unusual to see the scene depicted in a children’s book. As a result, in the process of satisfying the content of the prompt, the style of the samples also changes.

## 6.3 Generalization

RL finetuning on large language models has been shown to produce interesting generalization properties; for example, instruction finetuning almost entirely in English has been shown to improve capabilities in other languages [34]. It is difficult to reconcile this phenomenon with our current understanding of generalization; it would *a priori* seem more likely for finetuning to have an effect only on the finetuning prompt set or distribution. In order to investigate the same phenomenon with



**Figure 6 (Generalization)** For aesthetic quality, finetuning on a limit set of 45 animals generalizes to both new animals and non-animal everyday objects. For prompt alignment, finetuning on the same set of animals and only three activities generalizes to both new animals, new activities, and even combinations of the two. The prompts for the bottom row (left to right) are: “*a capybara washing dishes*”, “*a crab playing chess*”, “*a parrot driving a car*”, and “*a horse typing on a keyboard*”. More samples are provided in Appendix C.



**Figure 7 (Reward model overoptimization)** Examples of RL overoptimizing reward functions. **(L)** The diffusion model eventually loses all recognizable semantic content and produces noise when optimizing for incompressibility. **(R)** When optimized for prompts of the form “*n animals*”, the diffusion model exploits the VLM with a typographic attack [14], writing text that is interpreted as the specified number *n* instead of generating the correct number of animals.

diffusion models, Figure 6 shows a set of DDPO-finetuned model samples corresponding to prompts that were not seen during finetuning. In concordance with instruction-following transfer in language modeling, we find that the effects of finetuning do generalize, even with prompt distributions as narrow as 45 animals. We find evidence of generalization to both animals outside of the training distribution and to non-animal everyday objects.

#### 6.4 Overoptimization

Section 6.1 highlights the optimization problem: given a reward function, how well can an RL algorithm maximize that reward? However, finetuning on a reward function, especially a learned one, has been observed to lead to reward overoptimization or exploitation [13] in which the model learns to achieve high reward while moving too far away from the pretraining distribution to be useful.

Our setting is no exception, and we provide two examples of reward exploitation in Figure 7. When optimizing the incompressibility objective, the model eventually stops producing semantically

meaningful content, degenerating into high-frequency noise. Similarly, we observed that VLM reward pipelines are susceptible to typographic attacks [14]. When optimizing for alignment with respect to prompts of the form “*n animals*”, DDPO exploited deficiencies in the VLM by instead generating text loosely resembling the specified number. There is currently no general-purpose method for preventing overoptimization [13]. We highlight this problem as an important area for future work.

## 7 Discussion and Limitations

We presented an RL-based framework for training denoising diffusion models to directly optimize a variety of reward functions. By posing the iterative denoising procedure as a multi-step decision-making problem, we were able to design a class of policy gradient algorithms that are highly effective at training diffusion models. We found that DDPO was an effective optimizer for tasks that are difficult to specify via prompts, such as image compressibility, and difficult to evaluate programmatically, such as semantic alignment with prompts. To provide an automated way to derive rewards, we also proposed a method for using VLMs to provide feedback on the quality of generated images. While our evaluation considers a variety of prompts, the full range of images in our experiments was constrained (*e.g.*, animals performing activities). Future iterations could expand both the questions posed to the VLM, possibly using language models to propose relevant questions based on the prompt, as well as the diversity of the prompt distribution. We also provided preliminary empirical evidence of overoptimization of RL-finetuned diffusion models, but a more rigorous analysis and method to address overoptimization are left as directions for future work. We hope that this work will provide a step toward more targeted training of large generative models, where optimization via RL can produce models that are effective at achieving user-specified goals rather than simply matching an entire data distribution.

**Broader Impacts.** Generative models can be valuable productivity aids, but may also pose harm when used for disinformation, impersonation, or phishing. Our work aims to make diffusion models more useful by enabling them to optimize user-specified objectives. This adaptation has beneficial applications, such as the generation of more understandable educational material, but may also be used maliciously, in ways that we do not outline here. Work on the reliable detection of synthetic content remains important to mitigate such harms from generative models.

## 8 Acknowledgements

This work was partially supported by the Office of Naval Research and computational resource donations from Google via the TPU Research Cloud (TRC). Michael Janner was supported by a fellowship from the Open Philanthropy Project. Yilun Du and Kevin Black were supported by fellowships from the National Science Foundation.

### Code References

We used the following open-source libraries for this work: NumPy [16], JAX [5], Flax [17], optax [2], h5py [8], transformers [54], and diffusers [51].

### References

- [1] Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.
- [2] Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., Kaputowski, S., Keck, T., Kemaev, I., King, M., Kunesch, M., Martens, L., Merzic, H., Mikulik, V., Norman, T., Papamakarios, G., Quan, J., Ring, R., Ruiz, F., Sanchez, A., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stokowiec, W., Wang, L., Zhou, G., and Viola, F. The DeepMind JAX Ecosystem, 2020. URL <http://github.com/deepmind>.
- [3] Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N.,

- Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- [4] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- [5] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- [6] Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. *arXiv preprint arXiv:2303.04137*, 2023.
- [7] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Neural Information Processing Systems*, 2017.
- [8] Collette, A. *Python and HDF5*. O'Reilly, 2013.
- [9] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.
- [10] Dhariwal, P. and Nichol, A. Q. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.
- [11] Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. *arXiv preprint arXiv:2302.11552*, 2023.
- [12] Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.
- [13] Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2210.10760*, 2022.
- [14] Goh, G., †, N. C., †, C. V., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. Multi-modal neurons in artificial neural networks. *Distill*, 2021. <https://distill.pub/2021/multimodal-neurons>.
- [15] Hansen-Estruch, P., Kostrikov, I., Janner, M., Kuba, J. G., and Levine, S. IDQL: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.
- [16] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.
- [17] Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2023. URL <http://github.com/google/flax>.
- [18] Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

- [19] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.
- [20] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., and Salimans, T. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [21] Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.
- [22] Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.
- [23] Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. In *Neural Information Processing Systems*, 2021.
- [24] Knox, W. B. and Stone, P. TAMER: Training an Agent Manually via Evaluative Reinforcement. In *International Conference on Development and Learning*, 2008.
- [25] Lee, K., Liu, H., Ryu, M., Watkins, O., Du, Y., Boutilier, C., Abbeel, P., Ghavamzadeh, M., and Gu, S. S. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.
- [26] Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. 2023.
- [27] Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. *arXiv preprint arXiv:2206.01714*, 2022.
- [28] Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., and McAleese, N. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.
- [29] Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte carlo gradient estimation in machine learning. *The Journal of Machine Learning Research*, 21(1):5183–5244, 2020.
- [30] Nair, A., Dalal, M., Gupta, A., and Levine, S. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.
- [31] Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.
- [32] Nguyen, K., Daumé III, H., and Boyd-Graber, J. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Empirical Methods in Natural Language Processing*, 2017.
- [33] Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 2021.
- [34] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- [35] Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019. URL <https://arxiv.org/abs/1910.00177>.
- [36] Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine learning*, 2007.

- [37] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.
- [38] Ramesh, A., Pavlov, M., Gabriel Goh, S. G., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.
- [39] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.
- [40] Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.
- [41] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.
- [42] Schneuing, A., Du, Y., Charles Harris, A. J., Igashov, I., Du, W., Blundell, T., Lió, P., Gomes, C., Max Welling, M. B., and Correia, B. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [43] Schuhmann, C. Laion aesthetics, Aug 2022. URL <https://laion.ai/blog/laion-aesthetics/>.
- [44] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.
- [45] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [46] Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.
- [47] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.
- [48] Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=St1giarCHLP>.
- [49] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *Neural Information Processing Systems*, 2020.
- [50] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Solla, S., Leen, T., and Müller, K. (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL [https://proceedings.neurips.cc/paper\\_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf).
- [51] von Platen, P., Patil, S., Lozhkov, A., Cuénca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022.
- [52] Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.
- [53] Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pp. 5–32, 1992.

- [54] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [55] Xie, T., Fu, X., Ganea, O.-E., Barzilay, R., and Jaakkola, T. S. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations*, 2021.
- [56] Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., , and Tang, J. GeoDiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021.
- [57] Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.
- [58] Zhang, L. and Agrawala, M. Adding conditional control to text-to-image diffusion models, 2023.
- [59] Zhang, T., Kishore\*, V., Wu, F., Weinberger, K. Q., and Artzi, Y. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020.
- [60] Zhou, L., Du, Y., and Wu, J. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5826–5835, 2021.
- [61] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

## Appendix A Implementation Details

For all experiments, we use Stable Diffusion v1.4 [39] as the base model and finetune only the UNet weights while keeping the text encoder and autoencoder weights frozen.

### A.1 DDPO Implementation

We collect 256 samples per training iteration. For DDPO<sub>SF</sub>, we accumulate gradients across all 256 samples and perform one gradient update. For DDPO<sub>IS</sub>, we split the samples into 4 minibatches and perform 4 gradient updates. Gradients are always accumulated across all denoising timesteps for a single sample. For DDPO<sub>IS</sub>, we use the same clipped surrogate objective as in proximal policy optimization [45], but find that we need to use a very small clip range compared to standard RL tasks. We use a clip range of 1e-4 for all experiments.

### A.2 RWR Implementation

We compute the weights for a training iteration using the entire dataset of samples collected for that training iteration. For  $w_{\text{RWR}}$ , the weights are computed using the softmax function. For  $w_{\text{sparse}}$ , we use a percentile-based threshold, meaning  $C$  is dynamically selected such that the bottom  $p\%$  of a given pool of samples are discarded and the rest are used for training.

### A.3 Reward Normalization

In practice, rewards are rarely used as-is, but instead are normalized to have zero mean and unit variance. Furthermore, this normalization can depend on the current state; in the policy gradient context, this is analogous to a value function baseline [50], and in the RWR context, this is analogous to advantage-weighted regression [35]. In our experiments, we normalize the rewards on a per-context basis. For DDPO, this is implemented as normalization by a running mean and standard deviation that is tracked for each prompt independently. For RWR, this is implemented by computing the softmax over rewards for each prompt independently. For RWR<sub>sparse</sub>, this is implemented by computing the percentile-based threshold  $\bar{C}$  for each prompt independently.

### A.4 JPEG Encoding Code

```
import io
from PIL import Image

def encode_jpeg(x, quality=95):
    """
    x : np array of shape (H, W, 3) and dtype uint8
    """
    img = Image.fromarray(x)
    buffer = io.BytesIO()
    img.save(buffer, 'JPEG', quality=quality)
    jpeg = buffer.getvalue()
    bytes = np.frombuffer(jpeg, dtype=np.uint8)
    return len(bytes) / 1000
```

### A.5 Resource Details

RWR experiments were conducted on a v3-128 TPU pod, and took approximately 4 hours to reach 50k samples. DDPO experiments were conducted on a v4-64 TPU pod, and took approximately 4 hours to reach 50k samples. For the VLM-based reward function, LLaVA inference was conducted on a DGX machine with 8 80Gb A100 GPUs.

## A.6 Full Hyperparameters

		DDPO <sub>IS</sub>	DDPO <sub>SF</sub>	RWR	RWR <sub>sparse</sub>
Diffusion	Sampler	Ancestral	Ancestral	Ancestral	Ancestral
	Denoising steps ( $T$ )	50	50	50	50
	Guidance weight ( $w$ )	5.0	5.0	5.0	5.0
Optimization	Optimizer	AdamW	AdamW	AdamW	AdamW
	Learning rate	1e-5	1e-5	1e-5	1e-5
	Weight decay	1e-4	1e-4	1e-4	1e-4
	$\beta_1$	0.9	0.9	0.9	0.9
	$\beta_2$	0.999	0.999	0.999	0.999
	$\epsilon$	1e-8	1e-8	1e-8	1e-8
RWR	Gradient clip norm	1.0	1.0	1.0	1.0
	Inverse temperature ( $\beta$ )	-	-	0.2	-
	Percentile	-	-	-	0.9
	Batch size	-	-	128	128
	Gradient updates per iteration	-	-	400	400
DDPO	Samples per iteration	-	-	10k	10k
	Batch size	64	256	-	-
	Samples per iteration	256	256	-	-
	Gradient updates per iteration	4	1	-	-
	Clip range	1e-4	-	-	-

## A.7 List of 45 Common Animals

This list was used for experiments with the aesthetic quality reward function and the VLM-based reward function.

cat	dog	horse	monkey	rabbit	zebra	spider	bird	sheep
deer	cow	goat	lion	tiger	bear	raccoon	fox	wolf
lizard	beetle	ant	butterfly	fish	shark	whale	dolphin	squirrel
mouse	rat	snake	turtle	frog	chicken	duck	goose	bee
pig	turkey	fly	llama	camel	bat	gorilla	hedgehog	kangaroo

## Appendix B Additional Design Decisions

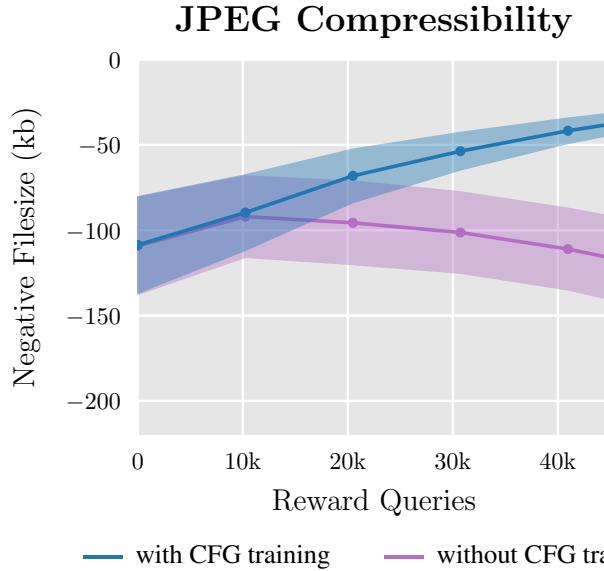
### B.1 CFG Training

Recent text-to-image diffusion models rely critically on *classifier-free guidance* (CFG) [18] to produce perceptually high-quality results. CFG involves jointly training the diffusion model on conditional and unconditional objectives by randomly masking out the context  $c$  during training. The conditional and unconditional predictions are then mixed at sampling time using a guidance weight  $w$ :

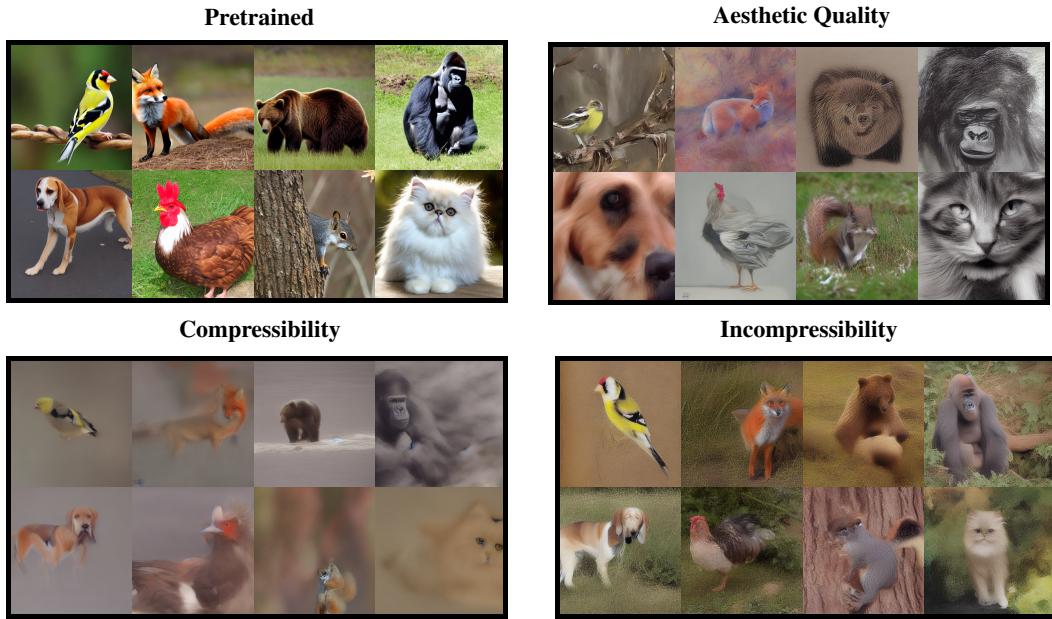
$$\tilde{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}) = w\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) + (1 - w)\epsilon_\theta(\mathbf{x}_t, t) \quad (5)$$

where  $\epsilon_\theta$  is the  $\epsilon$ -prediction parameterization of the diffusion model [19] and  $\tilde{\epsilon}_\theta$  is the guided  $\epsilon$ -prediction that is used to compute the next denoised sample.

For reinforcement learning, it does not make sense to train on the unconditional objective since the reward may depend on the context. However, we found that when only training on the conditional objective, performance rapidly deteriorated after the first round of finetuning. We hypothesized that this is due to the guidance weight becoming miscalibrated each time the model is updated, leading to degraded samples, which in turn impair the next round of finetuning, and so on. Our solution was to choose a fixed guidance weight and use the guided  $\epsilon$ -prediction during training as well as sampling. We call this procedure *CFG training*. Figure 8 shows the effect of CFG training on  $\text{RWR}_{\text{sparse}}$ ; it has no effect after a single round of finetuning, but becomes essential for subsequent rounds.



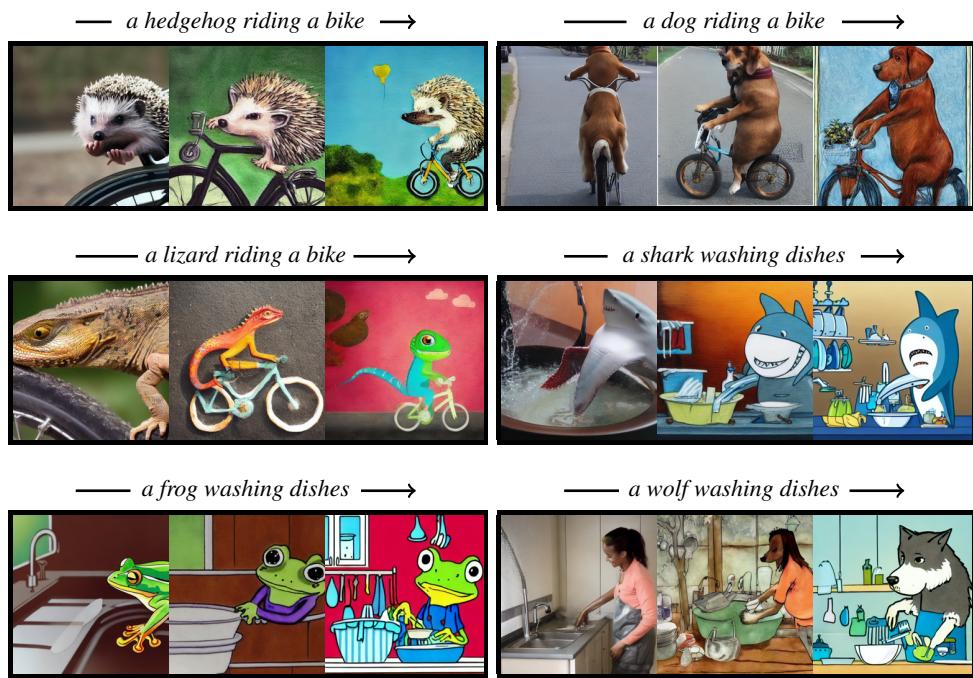
**Figure 8 (CFG training)** We run the  $\text{RWR}_{\text{sparse}}$  algorithm while optimizing only the conditional  $\epsilon$ -prediction (*without CFG training*), and while optimizing the guided  $\epsilon$ -prediction (*with CFG training*). Each point denotes a diffusion model update. We find that CFG training is essential for methods that do more than one round of interleaved sampling and training.



**Figure 9 (RWR samples)**

## Appendix C More Samples

Figure 9 shows qualitative samples from the baseline RWR method. Figure 10 shows more samples on seen prompts from DDPO finetuning with the image-prompt alignment reward function. Figure 11 shows more examples of generalization to unseen animals and everyday objects with the aesthetic quality reward function. Figure 12 shows more examples of generalization to unseen subjects and activities with the image-prompt alignment reward function.



**Figure 10 (More image-prompt alignment samples)**



**Figure 11 (Aesthetic quality generalization)**



**Figure 12 (Image-prompt alignment generalization)**