Training Diffusion Models with Reinforcement Learning

Kevin Black*[1] Michael Janner*[1] Yilun Du[2] Ilya Kostrikov[1] Sergey Levine[1]
[1] University of California, Berkeley [2] Massachusetts Institute of Technology
{kvablack, 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 (Sohl-Dickstein et al., 2015) 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 (Ramesh et al., 2021; Saharia et al., 2022; Ho et al., 2022), drug and material design (Xu et al., 2021; Xie et al., 2021; Schneuing et al., 2022), and continuous control (Janner et al., 2022; Wang et al., 2022; Hansen-Estruch et al., 2023). 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 explicitly concerned 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.

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 (Bai et al., 2022), 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.

*Denotes equal contribution.
Project page: anonymized-ddpo.github.io.

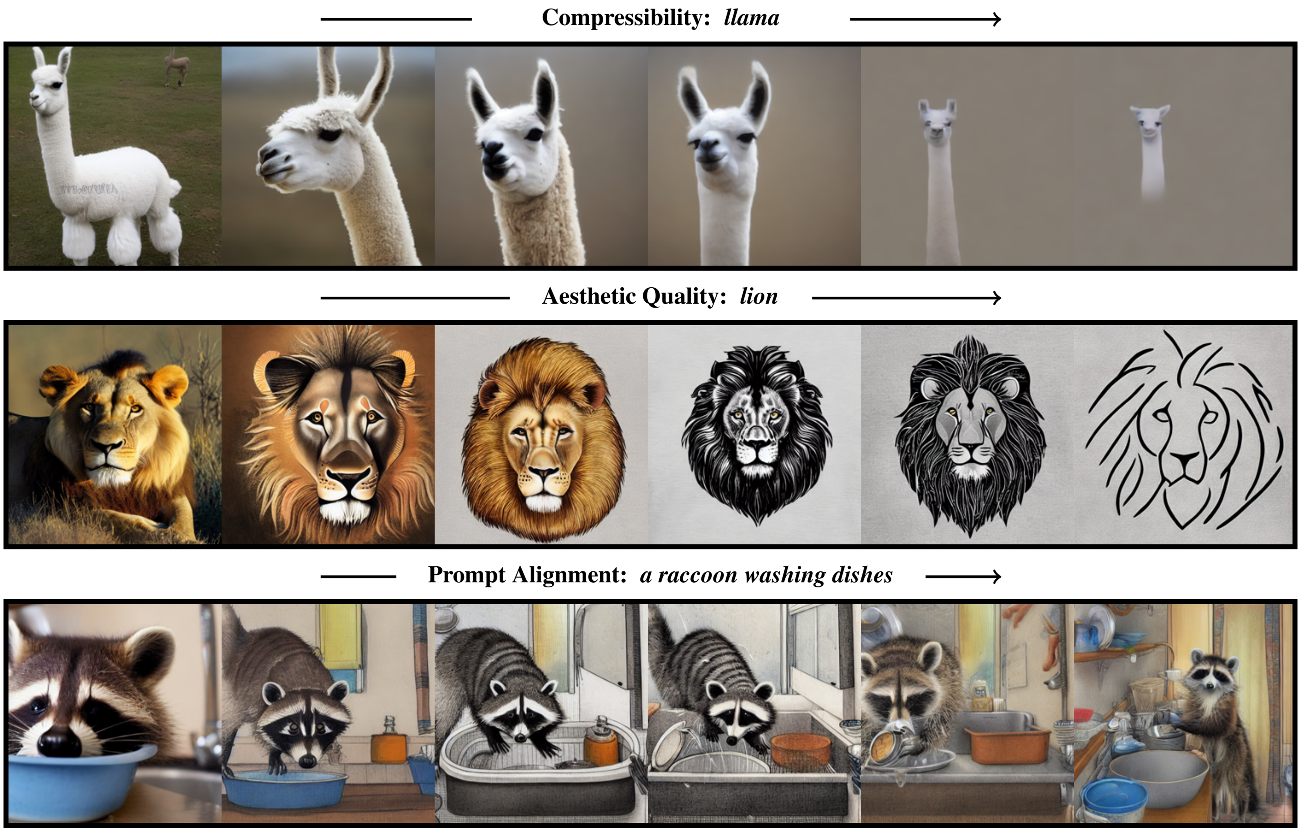
Compressibility: *llama*
Aesthetic Quality: *lion*
Prompt Alignment: *a raccoon washing dishes*

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.

2. 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. We compare reward-weighted regression approaches, denoted RWR, to our proposed policy gradient approaches, denoted DDPO. We evaluate four reward functions: compressibility and incompressibility, as determined by the JPEG compression algorithm; aesthetic quality, as determined by the LAION aesthetic quality predictor (Schuhmann, 2022); and prompt-image alignment, as determined by the LLaVA VLM (Liu et al., 2023). Full details of the algorithms and reward functions are provided in Appendix B and C, respectively. Additional experiments studying zero-shot generlization and reward overoptimization are provided in Appendix D.1 and D.2, respectively.

2.1. 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 (Rombach et al., 2022) as the base model for all experiments. Compressibility and incompressibility prompts are sampled uniformly from all 398 animals in the ImageNet-1000 (Deng et al., 2009) categories. Aesthetic quality prompts are sampled uniformly from a smaller set of 45 common animals.

As shown qualitatively in Figure 2, 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 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 G

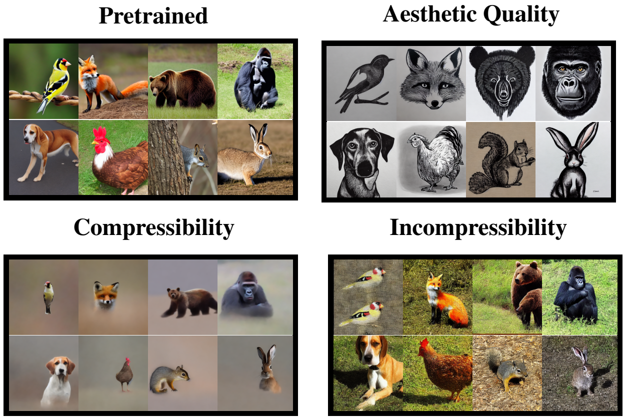

Figure 2 (DDPO samples) Qualitative depiction of the effects of RL fine-tuning on different reward functions. DDPO transforms naturalistic images into stylized line drawings to maximize predicted aesthetic quality, removes background content and applies a foreground blur to maximize compressibility, and adds artifacts and high-frequency noise to maximize incompressibility.

for comparison.

We provide a quantitative comparison of all methods in Figure 3. 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.

2.2. 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$_{\text{IS}}$ for this experiment, as we found it to be the most effective algorithm in Section 2.1. 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 2.1 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 4. Qualitatively, the samples come to depict the prompts much more faithfully throughout the course of training. This trend is also reflected quantitatively, though is less salient as we found that even small changes in average BERTScore (Zhang et al., 2020) 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.

References

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.

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.

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.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2210.10760*, 2022.

Goh, G., †, N. C., †, C. V., Carter, S., Petrov, M., Schubert, L., Radford, A., and Olah, C. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons.

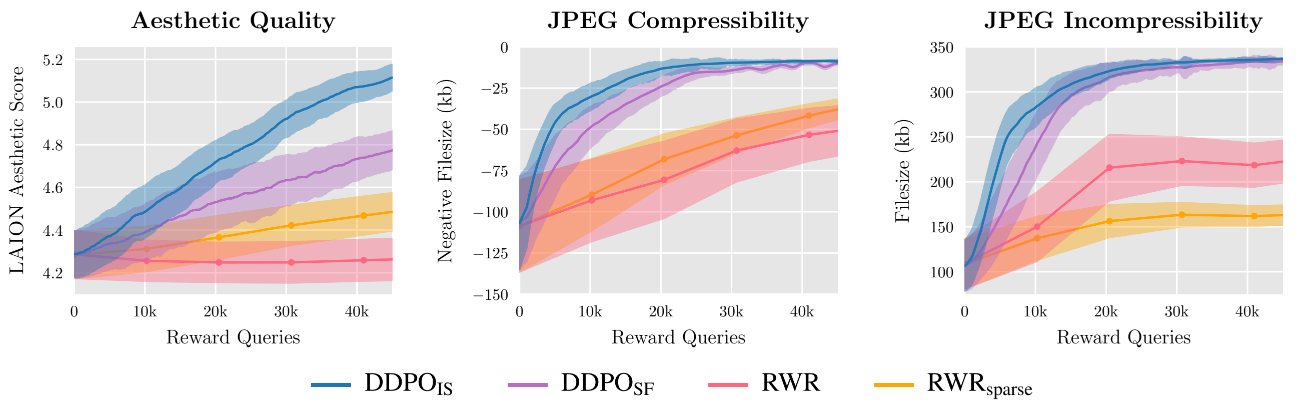

Figure 3 (Finetuning effectiveness) The relative effectiveness of different RL algorithms on three reward functions. We find that the policy gradient variants, denoted DDPO, are more effective optimizers than both RWR variants.

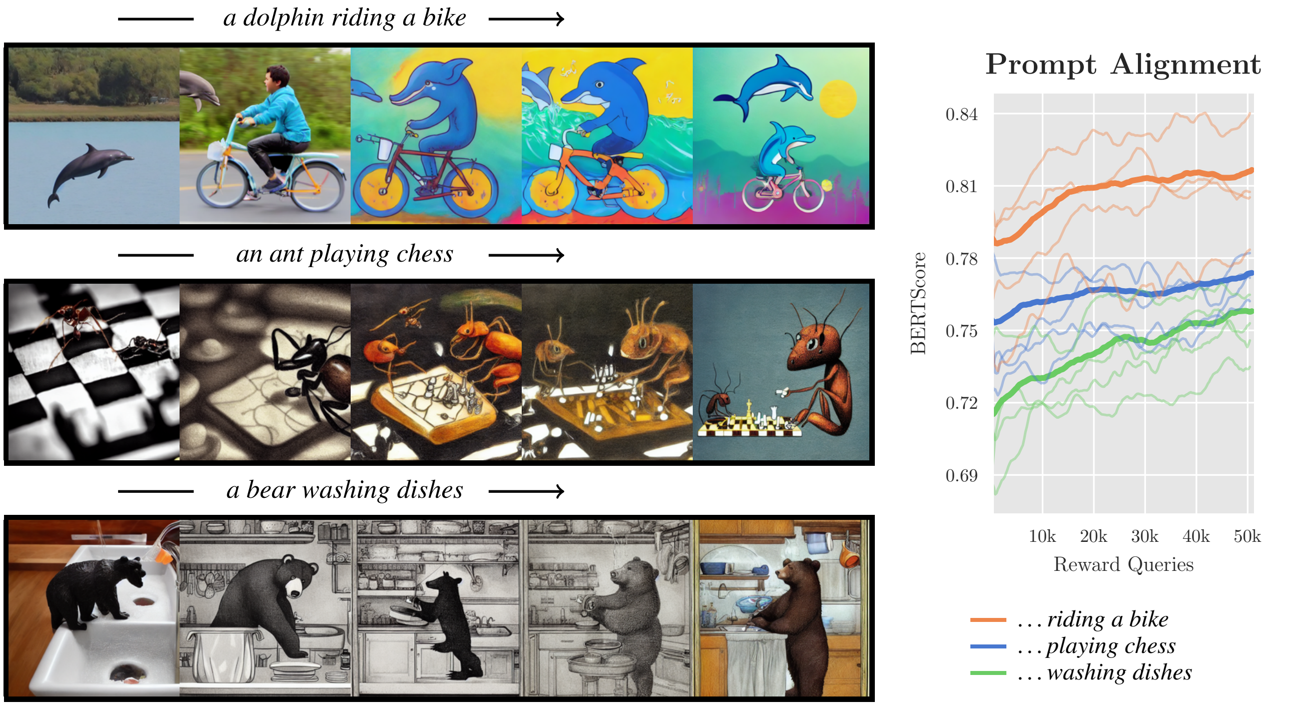

Figure 4 (Prompt alignment) (L) Progression of samples for the same prompt and random seed over the course of training. The images become significantly more faithful to the prompt. The samples also adopt a cartoon-like style, which we hypothesize is because the prompts are more likely depicted as illustrations 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.

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.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

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.

Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

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.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. 2023.

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.

Nair, A., Dalal, M., Gupta, A., and Levine, S. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

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.

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.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine learning*, 2007.

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.

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.

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.

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.

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.

Schuhmann, C. Laion aesthetics, Aug 2022. URL https://laion.ai/blog/laion-aesthetics/.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

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

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

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.

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.

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.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pp. 5–32, 1992.

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.

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.

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.

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.

A. Preliminaries

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

A.1. Diffusion Models

In this work, we consider conditional diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020), 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 $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t, \mathbf{c})$ for all $t \in \{0, 1, \ldots, T\}$ with the following simplified objective:

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

where $\tilde{\boldsymbol{\mu}}$ 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 (Ho et al., 2020).

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 $\{\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_0\}$ ending with a sample $\mathbf{x}_0$. The reverse process depends not only on the predictor $\boldsymbol{\mu}_\theta$ but also the choice of sampler. Most popular samplers (Ho et al., 2020; Song et al., 2021) use an isotropic Gaussian reverse process with a fixed timestep-dependent variance:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c}) = \mathcal{N}(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t, \mathbf{c}\right), \sigma_t^2 \mathbf{I}). \tag{2}$$

A.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, ρ_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 $\tau = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)$. 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:

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

B. Algorithm Details

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.

B.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:

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

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

B.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. (2023) 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) (Peters & Schaal, 2007).

A standard weighting scheme uses exponentiated rewards to ensure nonnegativity,

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

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

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

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:

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

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_{RWR} approximately optimizes $\mathcal{J}_{\text{RL}}(\pi)$ subject to a KL divergence constraint on the policy (Nair et al., 2020). 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.

B.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:

$$\mathbf{s}_t \triangleq (\mathbf{c}, t, \mathbf{x}_t) \qquad \pi(\mathbf{a}_t \mid \mathbf{s}_t) \triangleq p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c}) \qquad P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \triangleq \big(\delta_{\mathbf{c}}, \delta_{t-1}, \delta_{\mathbf{x}_{t-1}}\big)$$

$$\mathbf{a}_t \triangleq \mathbf{x}_{t-1} \qquad \rho_0(\mathbf{s}_0) \triangleq \big(p(\mathbf{c}), \delta_T, \mathcal{N}(\mathbf{0}, \mathbf{I})\big) \qquad R(\mathbf{s}_t, \mathbf{a}_t) \triangleq \begin{cases} r(\mathbf{x}_0, \mathbf{c}) & \text{if } t = 0 \\ 0 & \text{otherwise} \end{cases}$$

in which δ_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 π 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 $\{\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_0\}$ via sampling and updating parameters via gradient ascent on $\mathcal{J}_{\text{DDRL}}$.

The first variant of DDPO, which we call DDPO$_{\text{SF}}$, uses the score function policy gradient estimator, also known as the likelihood ratio method or REINFORCE (Williams, 1992; Mohamed et al., 2020):

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

where the expectation is taken over denoising trajectories generated by the current policy p_θ.

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 (Kakade & Langford, 2002):

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

where θ_{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_θ deviates too far from $p_{\theta_{\text{old}}}$, which can be addressed using trust regions (Schulman et al., 2015) 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 (Schulman et al., 2017). We call this variant DDPO$_{\text{IS}}$.

C. Reward Function Details

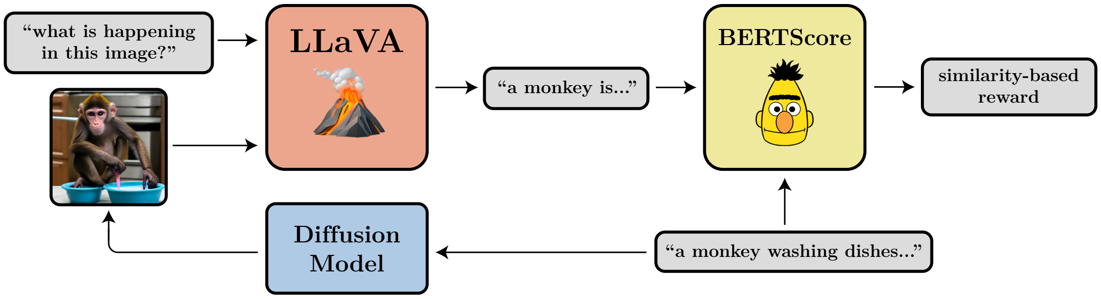

Figure 5 (VLM reward function) Illustration of the VLM-based reward function for prompt-image alignment. LLaVA (Liu et al., 2023) provides a short description of a generated image; the reward is the similarity between this description and the original prompt as measured by BERTScore (Zhang et al., 2020).

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.

C.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.

C.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 (Schuhmann, 2022), which is trained on 176,000 human image ratings. The predictor is implemented as a linear model on top of CLIP embeddings (Radford et al., 2021). 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 (Ouyang et al., 2022; Christiano et al., 2017; Ziegler et al., 2019).

C.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 (Bai et al., 2022), in which language models are improved using feedback from themselves.

We use LLaVA (Liu et al., 2023), a state-of-the-art VLM, to describe an image. The finetuning reward is the BERTScore (Zhang et al., 2020) 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 5, 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.

D. Additional Experiments

D.1. 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 (Ouyang et al., 2022). 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 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.

D.2. Overoptimization

Section 2.1 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 (Gao et al., 2022) 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 (Goh et al., 2021). 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 (Gao et al., 2022). We highlight this problem as an important area for future work.

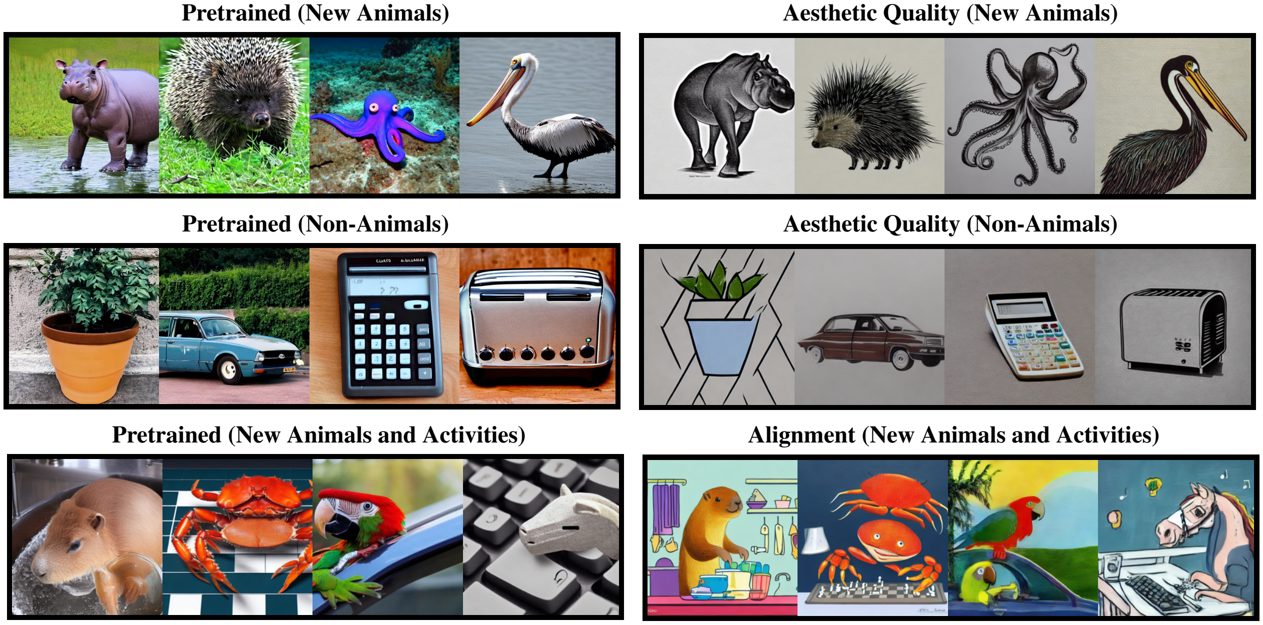
Pretrained (New Animals)
Aesthetic Quality (New Animals)
Pretrained (Non-Animals)
Aesthetic Quality (Non-Animals)
Pretrained (New Animals and Activities)
Alignment (New Animals and Activities)

Figure 6 (Generalization) For aesthetic quality, finetuning on a limited 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 G.

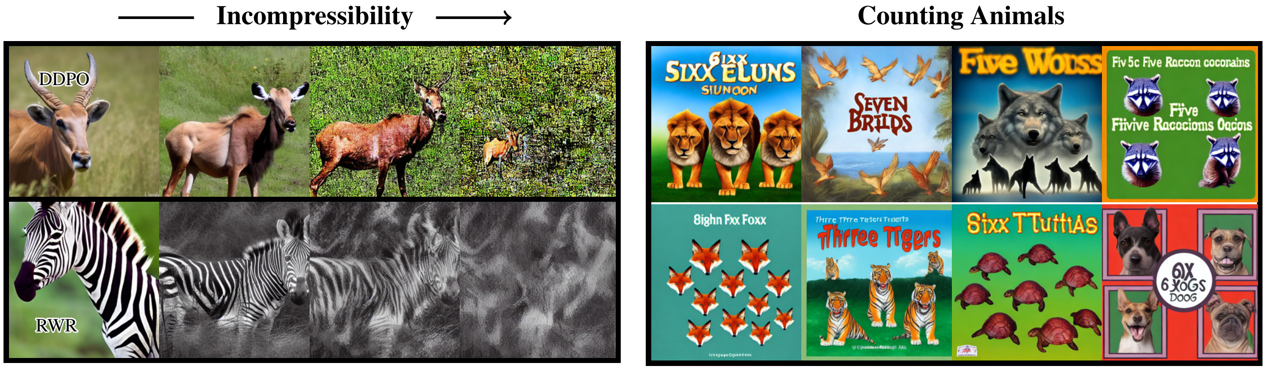

Figure 7 (Reward model overoptimization) Examples of RL overoptimizing reward functions. **(L)** The diffusion model eventually loses all recognizable semantic content and produces noise when optimizing for incompressibility. **(R)** When optimized for prompts of the form "n *animals*", the diffusion model exploits the VLM with a typographic attack (Goh et al., 2021), writing text that is interpreted as the specified number n instead of generating the correct number of animals.

E. Implementation Details

For all experiments, we use Stable Diffusion v1.4 (Rombach et al., 2022) as the base model and finetune only the UNet weights while keeping the text encoder and autoencoder weights frozen.

E.1. DDPO Implementation

We collect 256 samples per training iteration. For DDPO$_{\text{SF}}$, we accumulate gradients across all 256 samples and perform one gradient update. For DDPO$_{\text{IS}}$, 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$_{\text{IS}}$, we use the same clipped surrogate objective as in proximal policy optimization (Schulman et al., 2017), 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.

E.2. RWR Implementation

We compute the weights for a training iteration using the entire dataset of samples collected for that training iteration. For w_{RWR}, the weights are computed using the softmax function. For w_{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.

E.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 (Sutton et al., 1999), and in the RWR context, this is analogous to advantage-weighted regression (Peng et al., 2019). 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$_{\text{sparse}}$, this is implemented by computing the percentile-based threshold C for each prompt independently.

E.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
```

E.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.

E.6. Full Hyperparameters

		DDPO$_{\text{IS}}$	DDPO$_{\text{SF}}$	RWR	RWR$_{\text{sparse}}$
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
	β_1	0.9	0.9	0.9	0.9
	β_2	0.999	0.999	0.999	0.999
	ϵ	1e-8	1e-8	1e-8	1e-8
	Gradient clip norm	1.0	1.0	1.0	1.0
RWR	Inverse temperature (β)	-	-	0.2	-
	Percentile	-	-	-	0.9
	Batch size	-	-	128	128
	Gradient updates per iteration	-	-	400	400
	Samples per iteration	-	-	10k	10k
DDPO	Batch size	64	256	-	-
	Samples per iteration	256	256	-	-
	Gradient updates per iteration	4	1	-	-
	Clip range	1e-4	-	-	-

E.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

F. Additional Design Decisions

F.1. CFG Training

Recent text-to-image diffusion models rely critically on *classifier-free guidance* (CFG) (Ho & Salimans, 2021) to produce perceptually high-quality results. CFG involves jointly training the diffusion model on conditional and unconditional objectives by randomly masking out the context $\mathbf{c}$ during training. The conditional and unconditional predictions are then mixed at sampling time using a guidance weight w:

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

where $\boldsymbol{\epsilon}_\theta$ is the $\boldsymbol{\epsilon}$-prediction parameterization of the diffusion model (Ho et al., 2020) and $\tilde{\boldsymbol{\epsilon}}_\theta$ is the guided $\boldsymbol{\epsilon}$-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 $\boldsymbol{\epsilon}$-prediction during training as well as sampling. We call this procedure *CFG training*. Figure 8 shows the effect of CFG training on RWR$_{\text{sparse}}$; it has no effect after a single round of finetuning, but becomes essential for subsequent rounds.

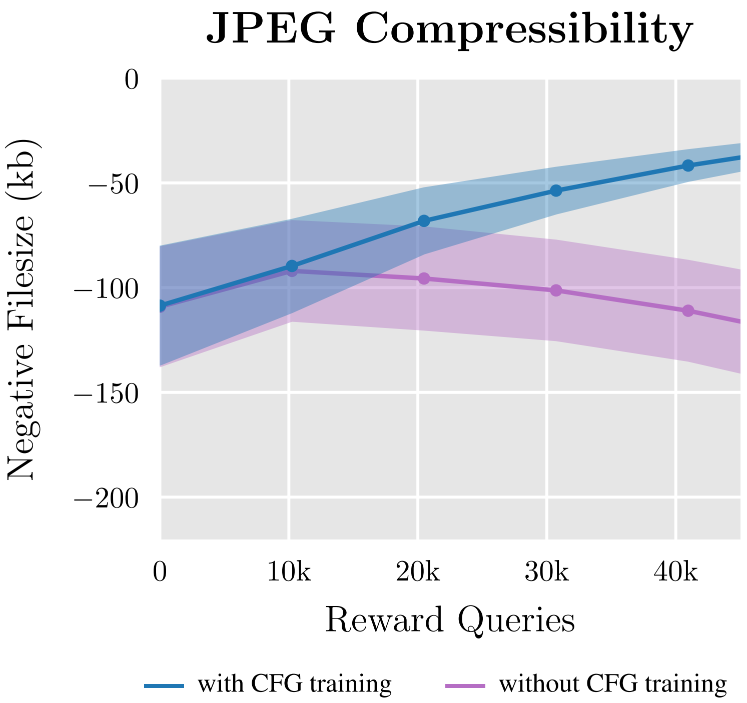

Figure 8 (CFG training) We run the $\text{RWR}_{\text{sparse}}$ algorithm while optimizing only the conditional ϵ-prediction (*without CFG training*), and while optimizing the guided ϵ-prediction (*with CFG training*). Each point denotes a diffusion model update. We find that CFG training is essential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

G. 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.

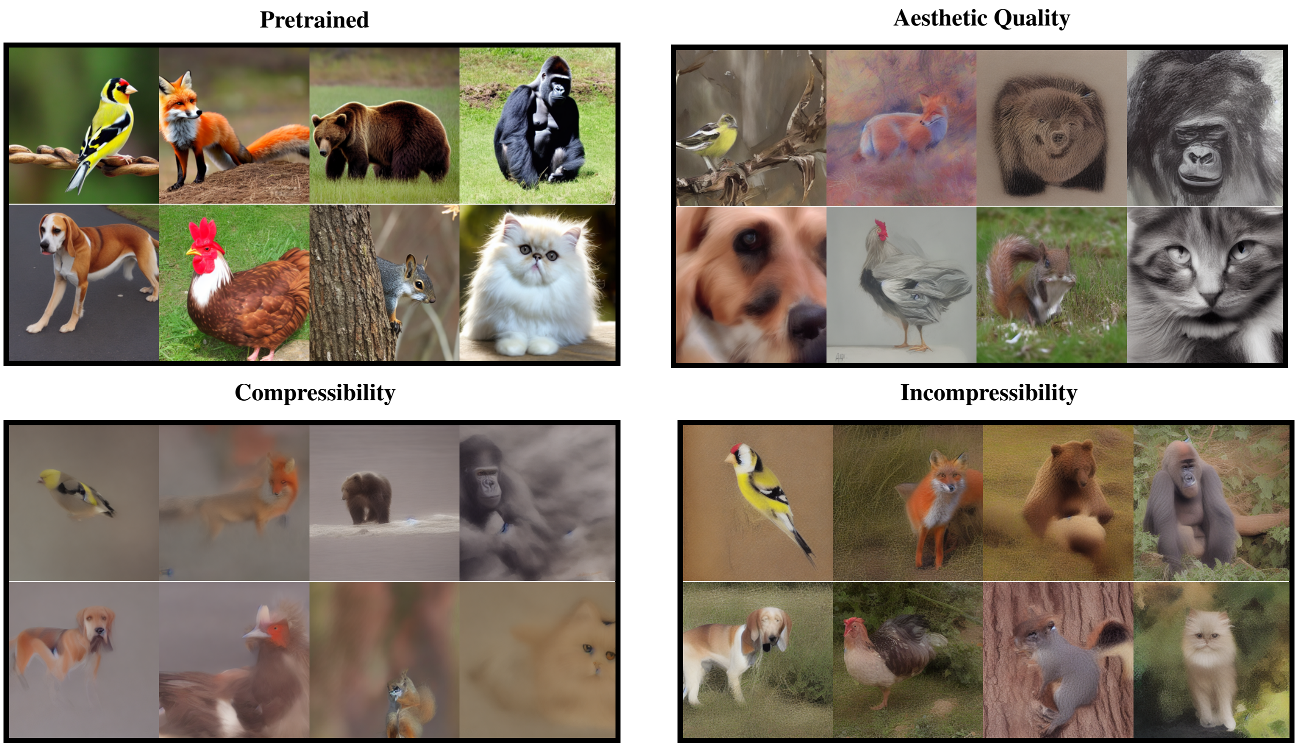

Figure 9 (RWR samples)

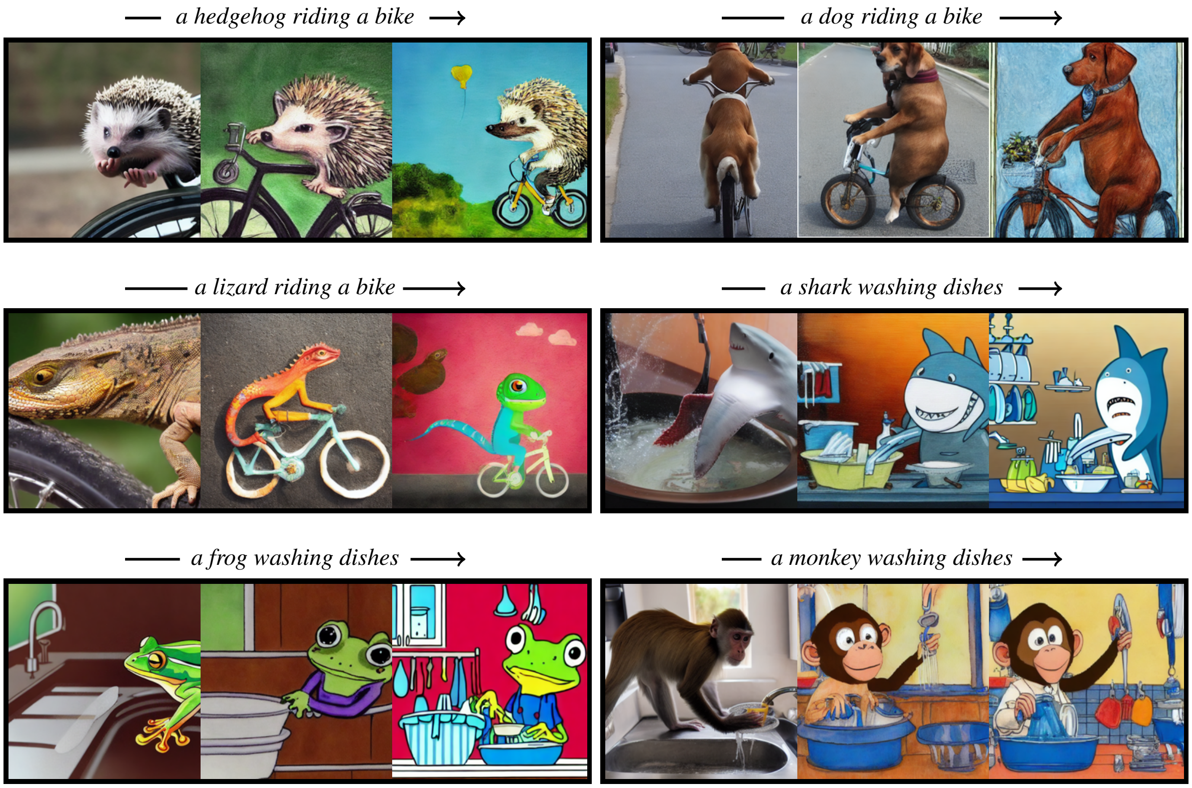

Figure 10 (More image-prompt alignment samples)

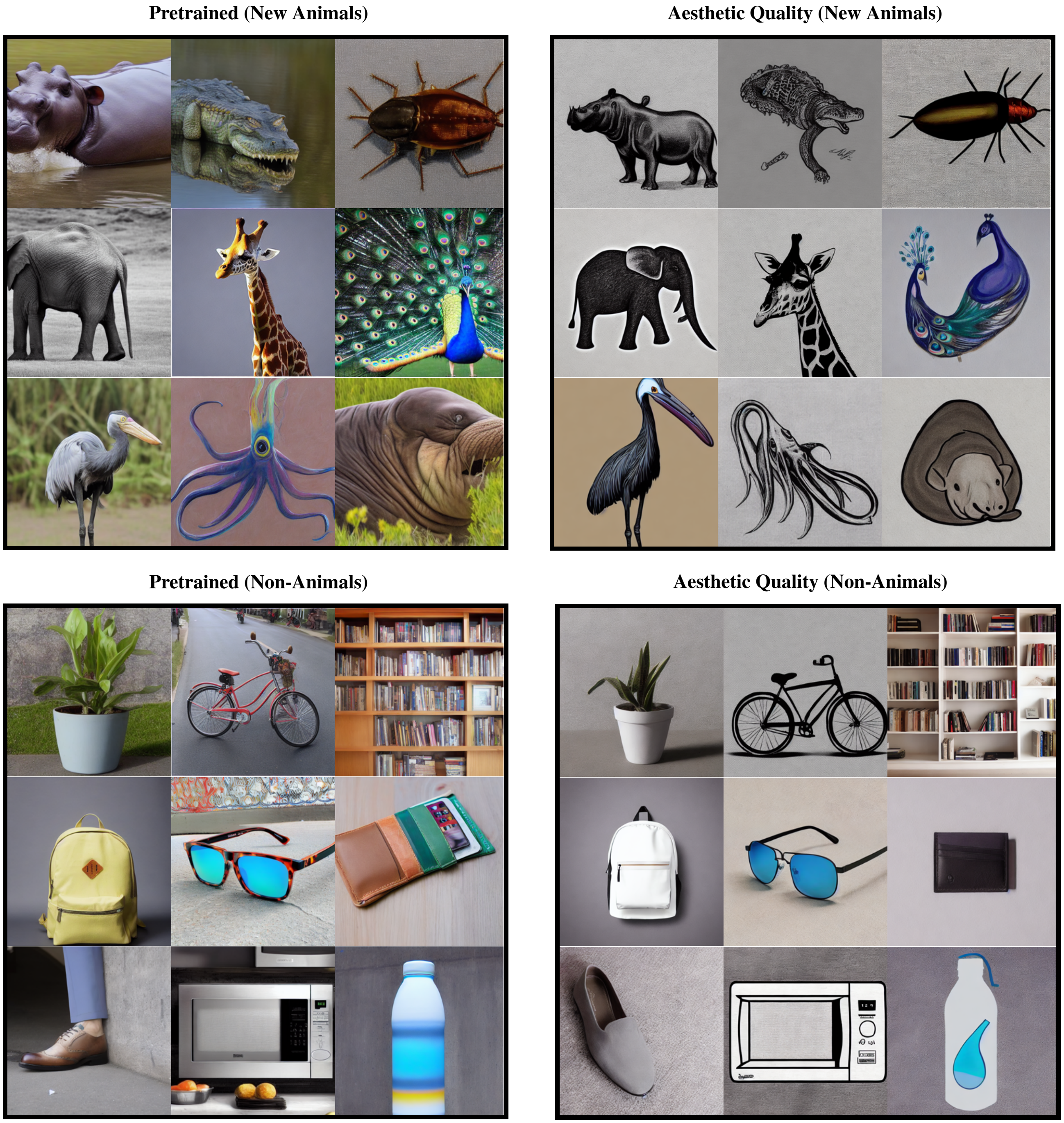
Pretrained (New Animals)
Aesthetic Quality (New Animals)
Pretrained (Non-Animals)
Aesthetic Quality (Non-Animals)

Figure 11 (Aesthetic quality generalization)

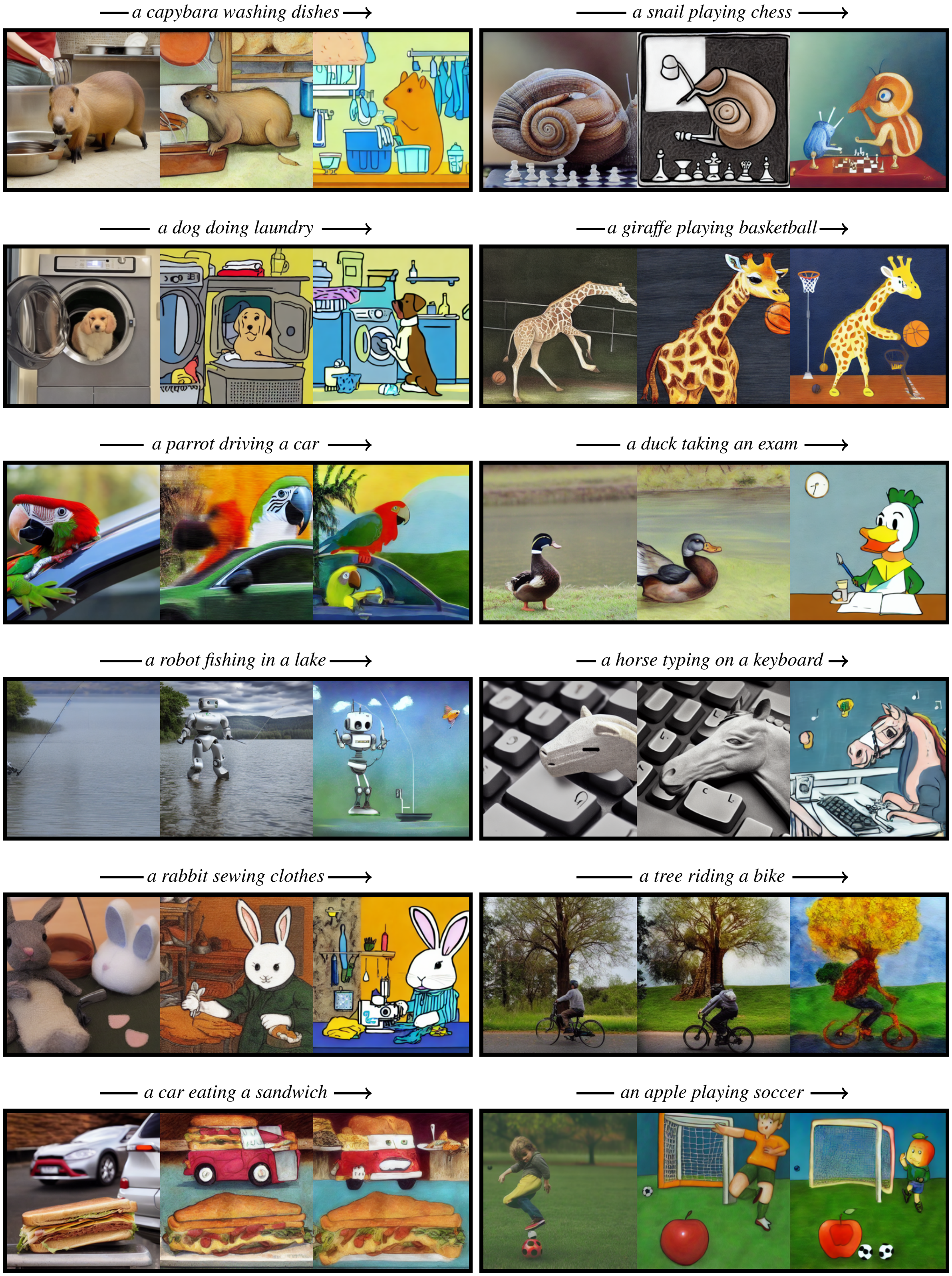

Figure 12 (Image-prompt alignment generalization)