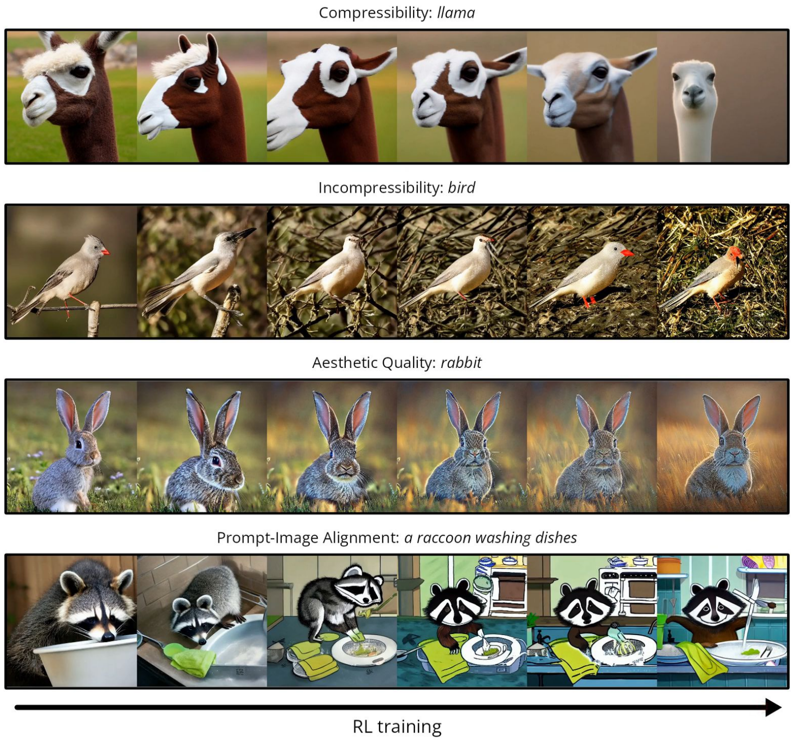
# Training Diffusion Models with Reinforcement Learning

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, Sergey Levine
University of California, Berkeley Massachusetts Institute of Technology

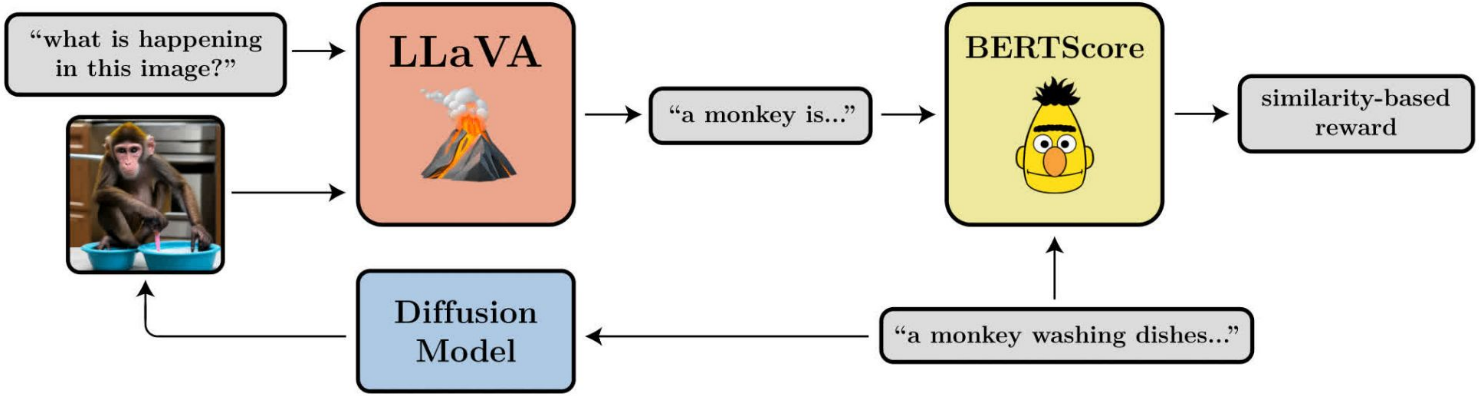
# Motivation

- Diffusion models are trained with an approximation to the log-likelihood objective. However, most use cases of diffusion models are concerned  instead with downstream objectives such as human-perceived image quality.
- This paper investigates reinforcement learning methods for directly optimizing diffusion models for such objectives.
- It describes 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).

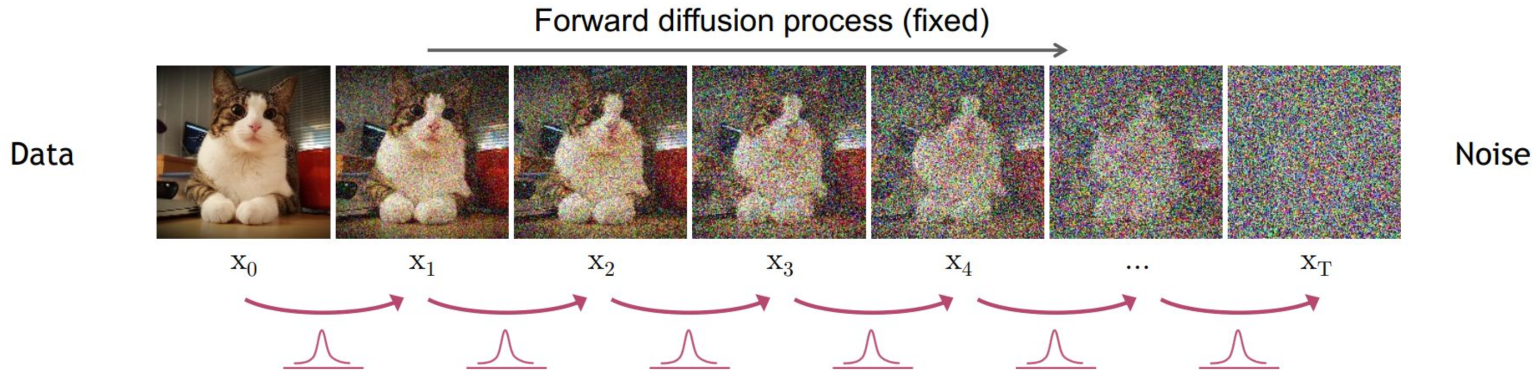


# Approach Summary

- DDPO is used to finetune Stable Diffusion on objectives that are difficult to express via prompting, such as image compressibility, and those derived from human feedback, such as aesthetic quality.
- DDPO can also be used to improve prompt-image alignment without any human annotations using feedback from a vision-language model.



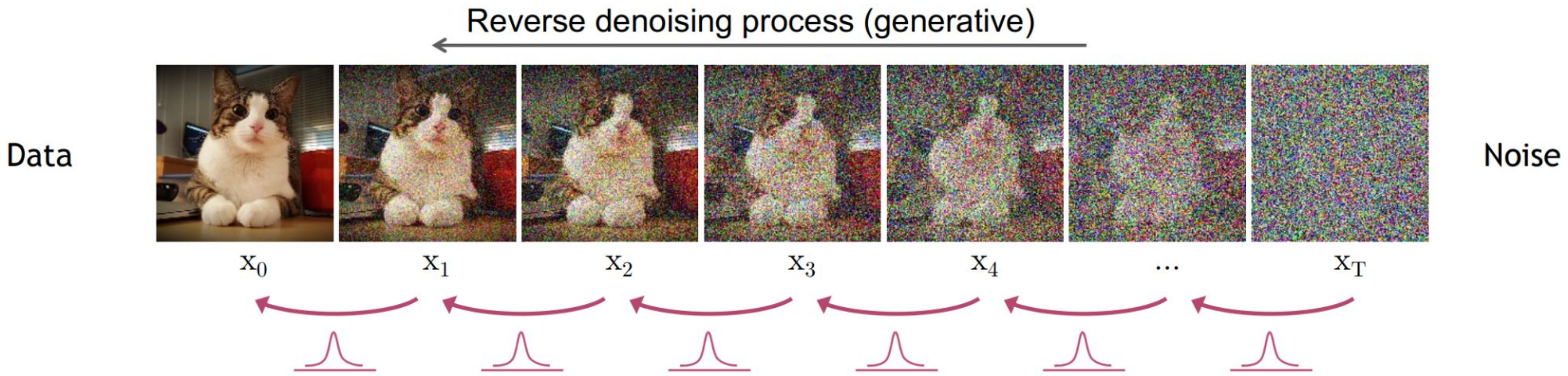
# Background - Diffusion Models




$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

(joint)

# Background - Diffusion Models




$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\mu_\theta(\mathbf{x}_t, t)}_{\substack{\text{Trainable network} \\ \text{(U-net, Denoising Autoencoder)}}}, \sigma_t^2\mathbf{I})$$

$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

# Background - Reinforcement Learning

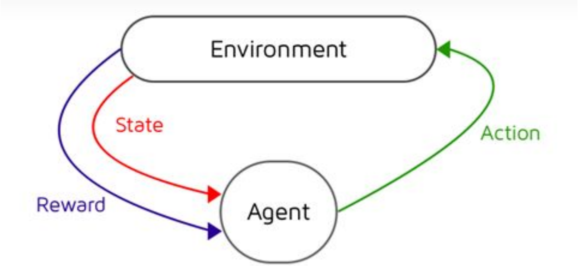
- **Markov Decision Process:** (Ş, Ą, $\mathcal{P}$, $\gamma$, Ŗ),

  where Ş is the state space, Ą is the action space, $\mathcal{P}$ is the transition kernel, $\gamma$ is the discount factor, and Ŗis the reward function.

- **Policy** ($\pi$): A map from state space to action space. May be stochastic.

- **Reward function** - Ŗ(s): Maps each state-action pair to a scalar called reward.

- **Value function** - V(s,a): Total expected return starting from state s and taking a and following the policy $\pi$, discounted by $\gamma$.



# Problem Statement

- The diffusion model induces a sample distribution pθ(x0 | c).
- The denoising diffusion RL objective is to maximize a reward signal r defined on the samples and contexts.

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

# REWARD-WEIGHTED REGRESSION

- 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 that immediately leads to an absorbing termination state. Therefore, maximizing $J_{DDRL}(\theta)$ is equivalent to maximizing the RL objective $J_{RL}(\pi)$ in this MDP.

- The reward is then used as a weighting term in the log-likelihood objective function.

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{z})\sim\mathcal{D}^{\texttt{model}}} \left[ -r_\phi(\mathbf{x}, \mathbf{z}) \log p_\theta(\mathbf{x}|\mathbf{z}) \right]$$

# REWARD-WEIGHTED REGRESSION

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{z})\sim\mathcal{D}^{\texttt{model}}} \left[ -r_\phi(\mathbf{x},\mathbf{z}) \log p_\theta(\mathbf{x}|\mathbf{z}) \right]$$

- Limitations:
  - Diffusion loss does not involve an exact log-likelihood — it is instead derived as a variational bound.
  - It ignores the sequential nature of the denoising process, only using the final sample.
  - $p_\theta(x_0 \mid c)$ is an arbitrarily complicated distribution.

# DENOISING DIFFUSION POLICY OPTIMIZATION

- MDP Formulation:.

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

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

$\delta_y$ = Dirac delta distribution with nonzero density only at y.

- This provides a general framework (DDPO) with an arbitrary reward function, that can be used to optimize arbitrary downstream objectives.

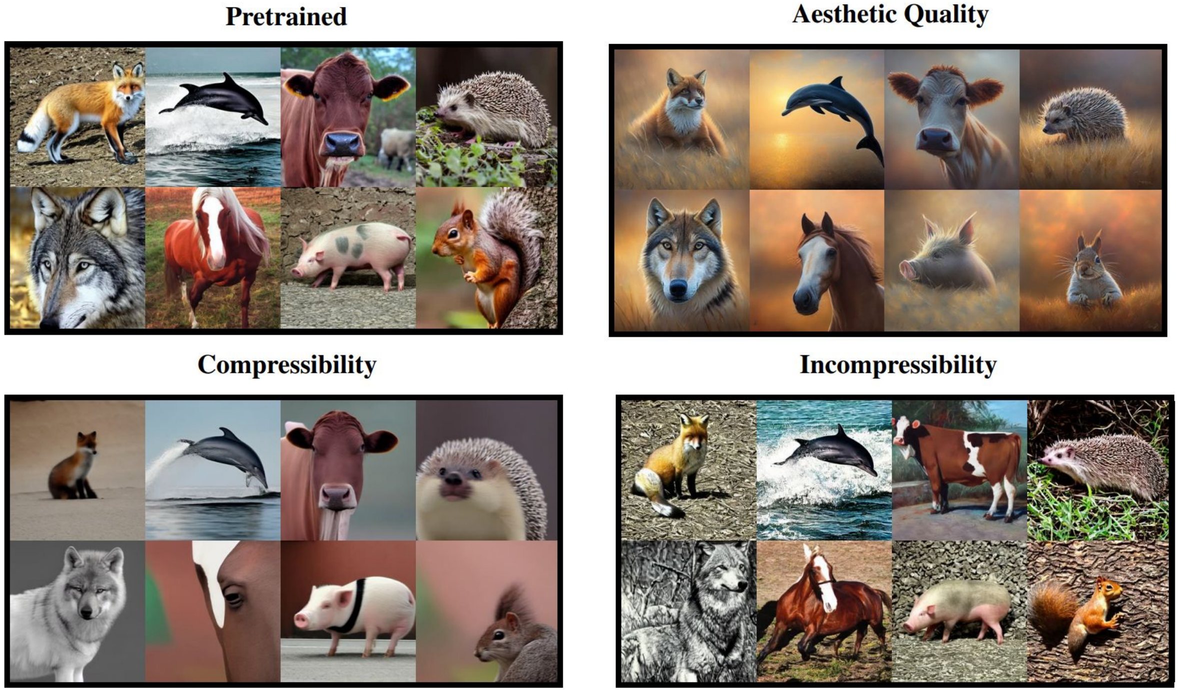
# DENOISING DIFFUSION POLICY OPTIMIZATION

- $\text{DDPO}_{\text{SF}}$ - Score Function Policy Gradient Estimator

$$\nabla_\theta \mathcal{J}_{\text{DDRL}} = \mathbb{E}\left[\sum_{t=0}^{T} \nabla_\theta \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{c})\, r(\mathbf{x}_0, \mathbf{c})\right] \qquad (\text{DDPO}_{\text{SF}})$$

- DDPO alternates collecting denoising trajectories $\{x_T, x_{T-1}, \ldots, x_0\}$ via sampling and updating parameters via gradient descent.

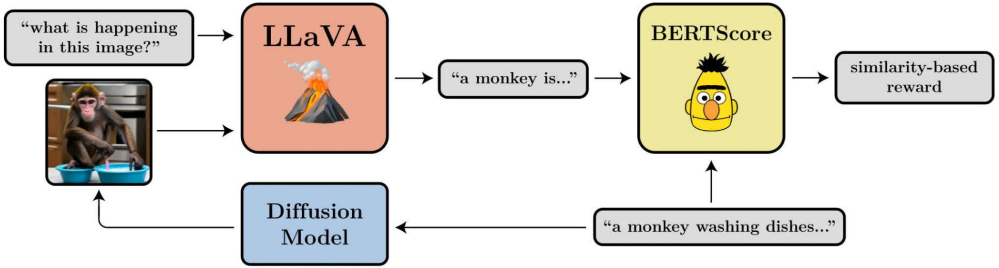
# Results

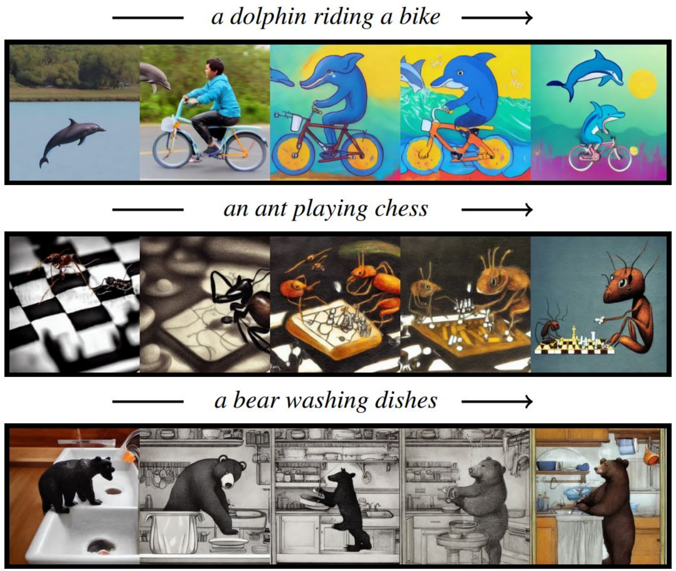


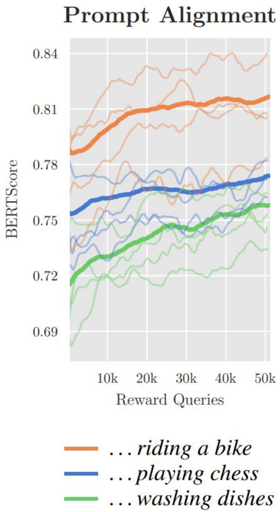

- **Aesthetic Quality**: Reward provided by LAION aesthetics predictor.
- **Compressibility/ Incompressibility**: Reward provided by JPEG compression

# Results

## Automated Prompt Alignment



# Generalization


Finetuning on a limited set of animals generalizes to both new animals and non-animal everyday objects. The prompts for the rightmost two columns are "a capybara washing dishes" and "a duck taking an exam"