GenPO: Generative Diffusion Models Meet On-Policy Reinforcement Learning

Shutong Ding[1,5] Ke Hu[1] Shan Zhong[3] Haoyang Luo[1] Weinan Zhang[2]
Jingya Wang[1,5] Wang Jun[4] Ye Shi[1,5]

[1]ShanghaiTech University [2]Shanghai Jiao Tong University
[3]University of Electronic Science and Technology of China
[4]University College London
[5]MoE Key Laboratory of Intelligent Perception and Human Machine Collaboration

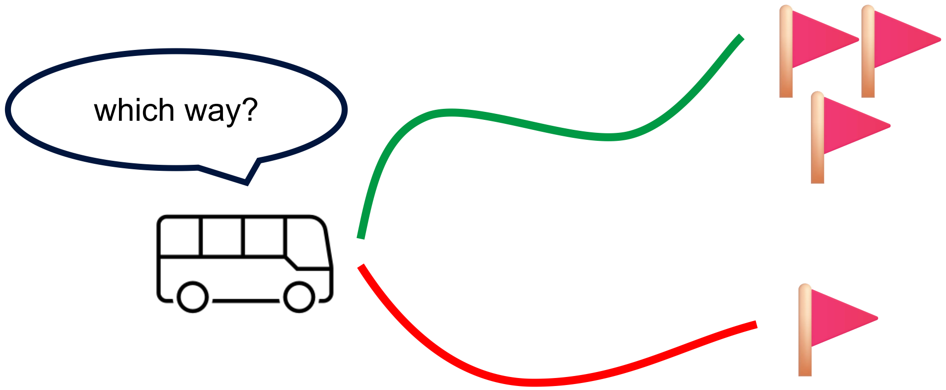
NeurIPS 2025

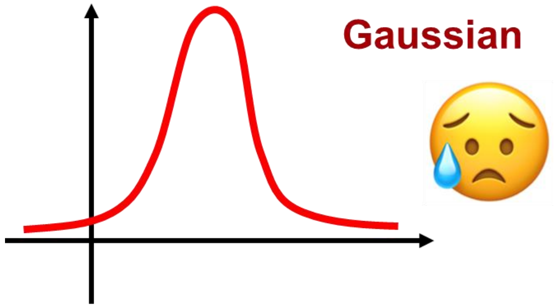
October 7, 2025

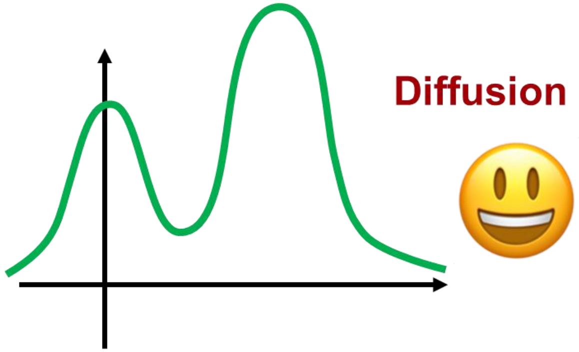
Background: Diffusion in Online RL

1. **Exploration capability** of Gaussian policy or deterministic policy is limited

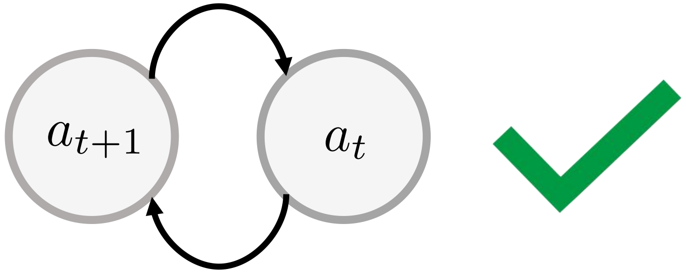
2. **Expressiveness and multimodality** of diffusion avoid policy falling into the local optimality







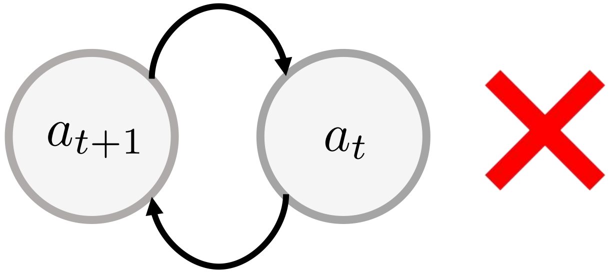
Background: Diffusion in Online RL



Existing diffusion-based RL methods are almost all **off-policy**,
and cannot benefit from the large-scale parallel simulator!

Background: Diffusion in Online RL

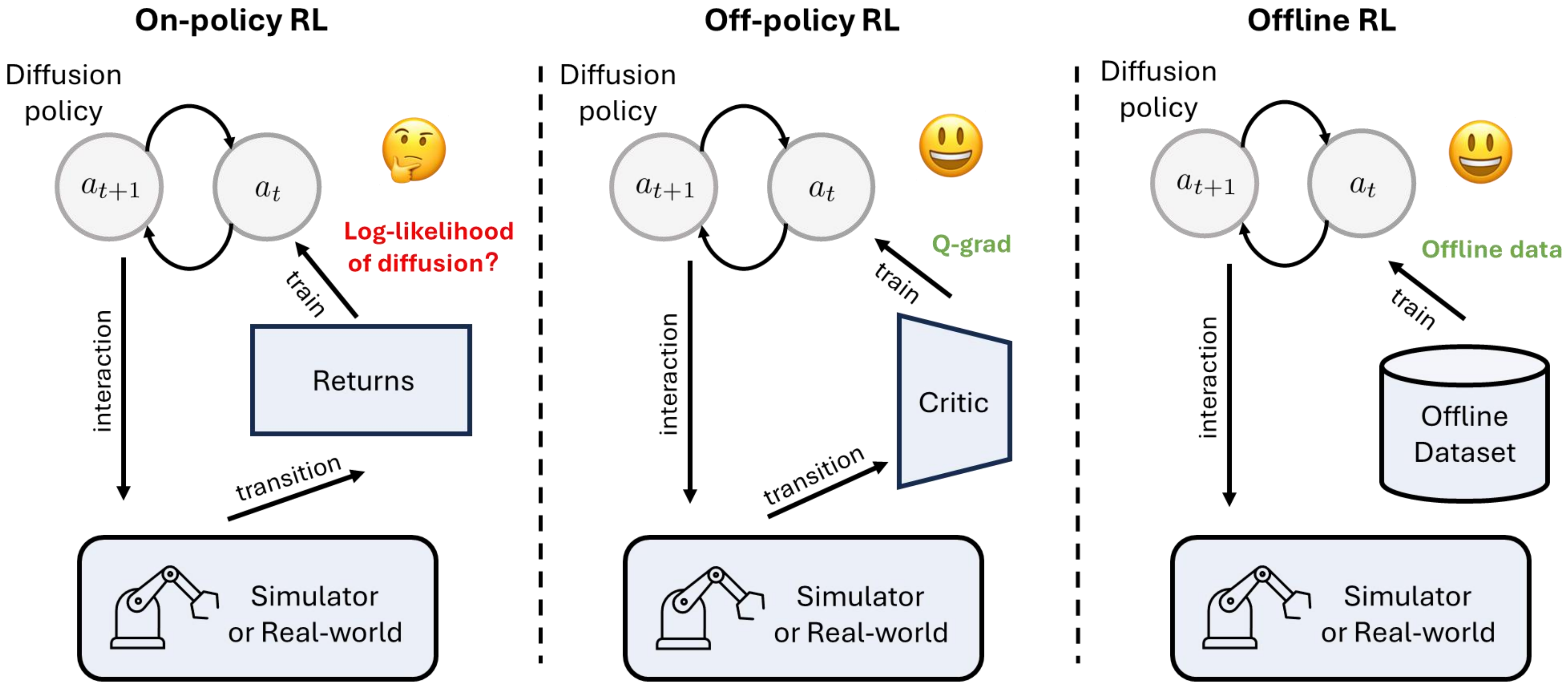
Existing on-policy RL methods (**PPO, TRPO**)



Existing on-policy RL methods cannot benefit from the multimodality and powerful exploration capability of **diffusion model**!

Background: Diffusion in Online RL

上海科技大学
ShanghaiTech University



立志成才报国裕民

Solution: GenPO

How can we train diffusion policy in an on-policy RL paradigm?

Calculate the **log-likelihood** of the diffusion model?

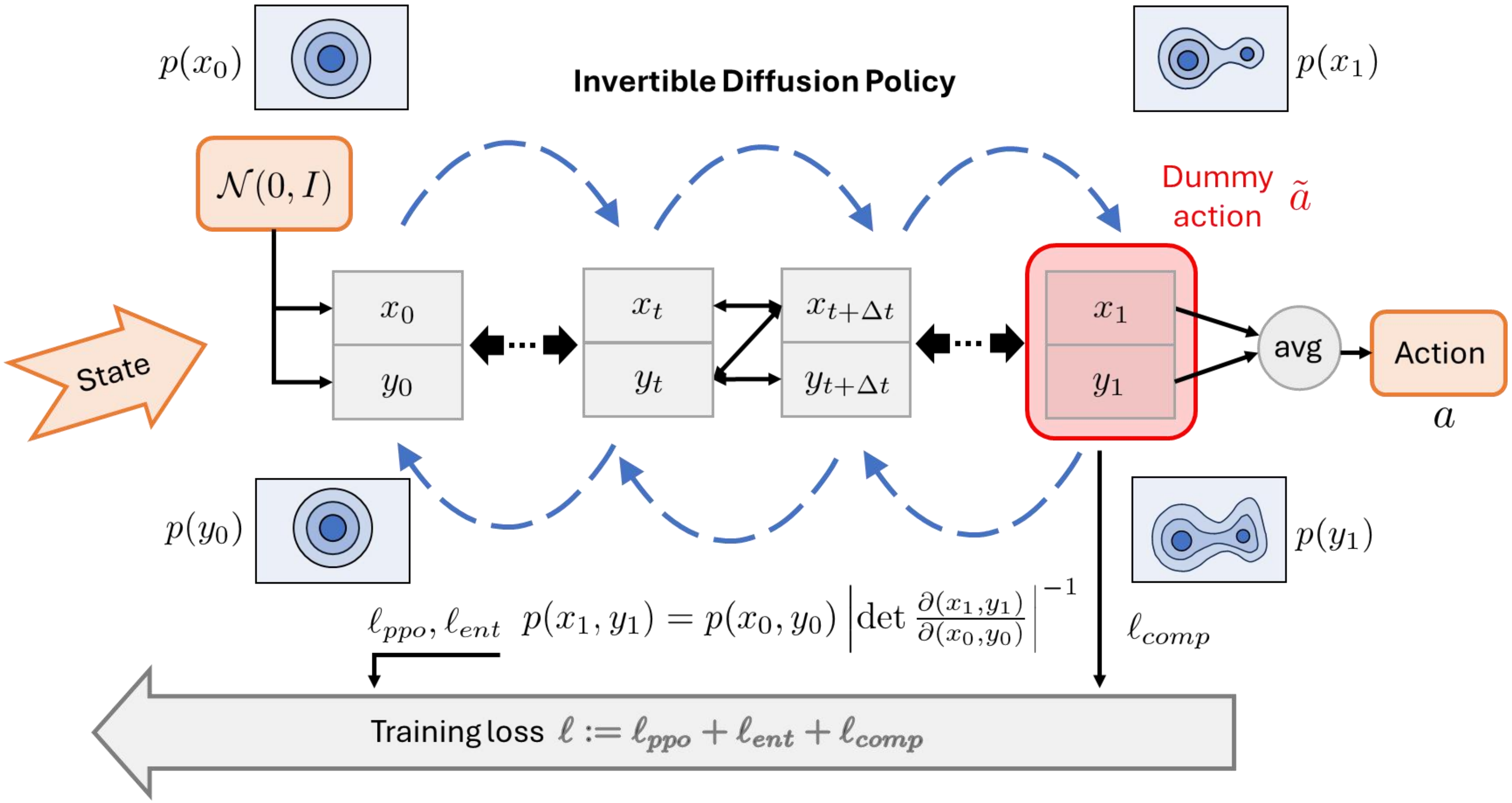
But the log-likelihood of the diffusion model is **not available**.

What if make the denoising procedure **invertible** and calculate the log-likelihood via **change of variables**?

Solution: GenPO

上海科技大学 ShanghaiTech University

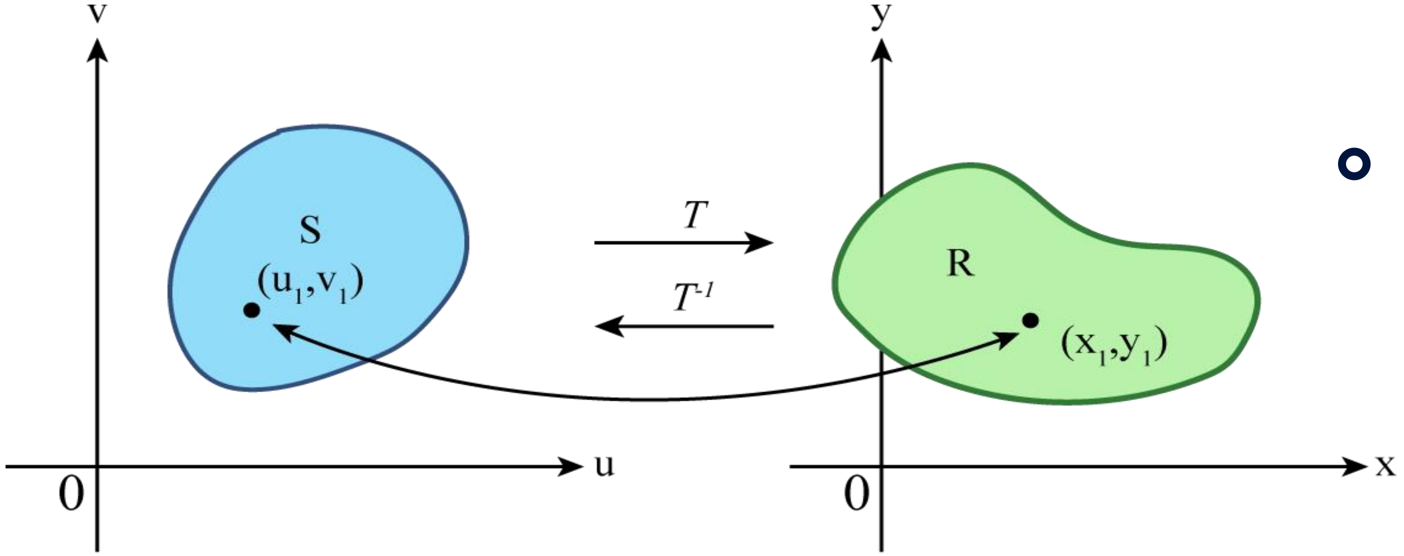
Motivated by these two ideas, we design an invertible diffusion policy and propose GenPO.



Change of Variables

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ is an invertible and smooth mapping. If we have the random variable $X \sim q(x)$ and the random variable $Y = f(X)$ transformed by function f , the distribution of Y is $p(y) = q(x) \left| \det \frac{\partial f}{\partial x} \right|^{-1}$.

Log-likelihood



How to further make the diffusion model invertible?

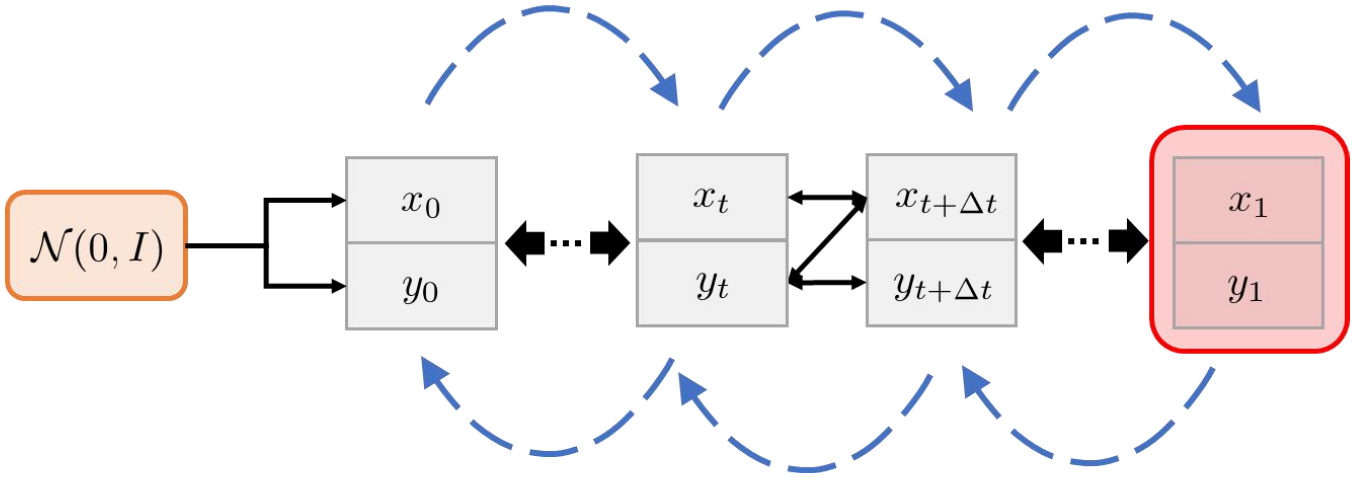
Exact Diffusion Inversion

Classical diffusion models (e.g., DDPM/DDIM) are not invertible due to discretization:

$$\text{Reverse:}\quad x_{t-1} = \sqrt{\alpha_{t-1}}\,\frac{x_t - \sqrt{1-\alpha_t \boldsymbol{\epsilon}_\theta(x_t,t)}}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}}\,\boldsymbol{\epsilon}_\theta(x_t,t)$$

$$\text{Forward:}\quad x_t = \frac{x_{t-1} - b_t \boldsymbol{\epsilon}_\theta(x_t,t)}{a_t} \approx \frac{x_{t-1} - b_t \boldsymbol{\epsilon}_\theta(x_{t-1},t)}{a_t}.$$

Motivated by [1], we design an invertible diffusion model:



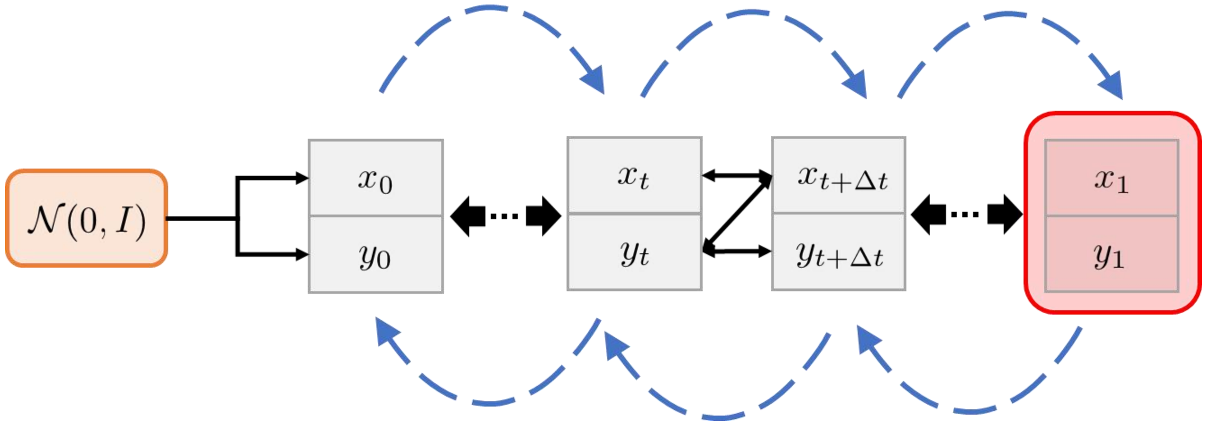
Invertible diffusion with doubled noise vectors

[1] Wallace B, Gokul A, Naik N. Edict: Exact diffusion inversion via coupled transformations[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 22532-22541.

Exact Diffusion Inversion

上海科技大学
ShanghaiTech University

Motivated by [1], we design an invertible diffusion model:



Invertible diffusion with doubled noise vectors

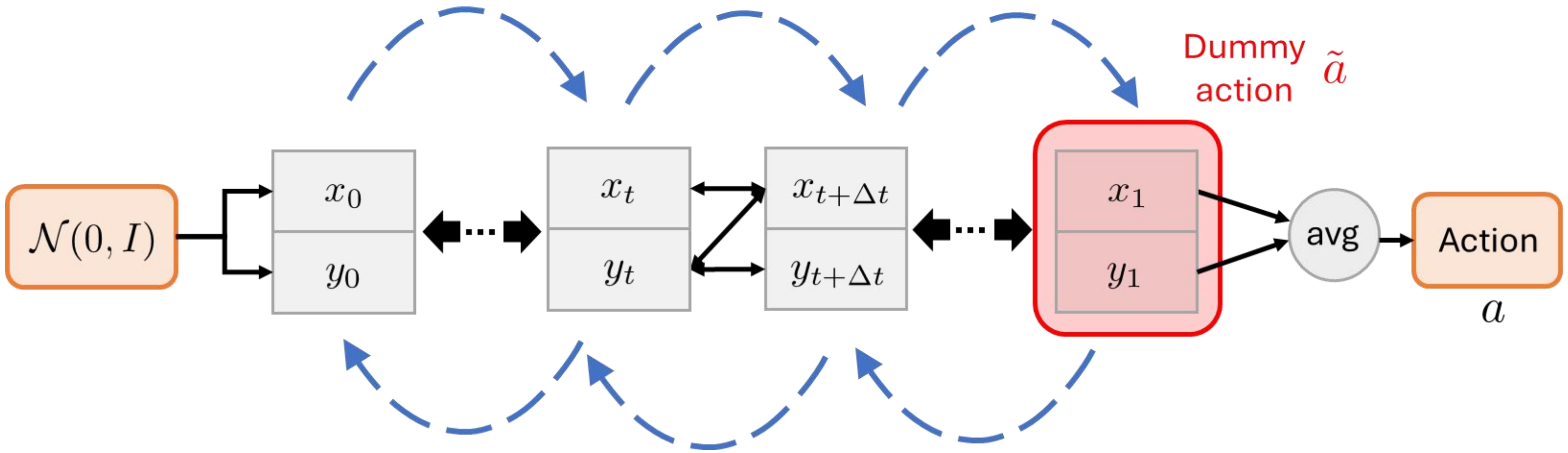
Use x or y as the final action? Or the concatenation of them?

[1] Wallace B, Gokul A, Naik N. Edict: Exact diffusion inversion via coupled transformations[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 22532-22541.

立志成才 报国裕民

Doubled Dummy Action

上海科技大学 ShanghaiTech University



1. Number of actions may be odd
2. Double the action space of the actual problem for optimization
3. Use the average of the two parts of doubled actions as the final action
4. Mixing trick for the consistency of x and y

Unmixing: $\tilde{y}_{t+\Delta t} = \frac{y_{t+\Delta t} - (1-p)x_{t+\Delta t}}{p}$ $\quad \tilde{x}_{t+\Delta t} = \frac{x_{t+\Delta t} - (1-p)\tilde{y}_{t+\Delta t}}{p}$

Forward: $y_t = \tilde{y}_{t+\Delta t} - v_\theta(\tilde{x}_{t+\Delta t}, t)\Delta t,$ $\quad x_t = \tilde{x}_{t+\Delta t} - v_\theta(y_t, t)\Delta t$

Reverse: $\tilde{x}_{t+\Delta t} = x_t + v_\theta(y_t, t)\Delta t,$ $\quad \tilde{y}_{t+\Delta t} = y_t + v_\theta(\tilde{x}_{t+\Delta t}, t)\Delta t$

Mixing: $x_{t+\Delta t} = p \cdot \tilde{x}_{t+\Delta t} + (1-p) \cdot \tilde{y}_{t+\Delta t},$ $\quad y_{t+\Delta t} = p \cdot \tilde{y}_{t+\Delta t} + (1-p) \cdot x_{t+\Delta t}$

立志成才 报国裕民

Compression Loss

$$\mathcal{L}(\theta) := \mathcal{L}^{PPO} + \lambda\mathcal{L}^{ENT} + \nu\mathbb{E}_{x_1, y_1 \sim \pi_\theta}\left[(x_1 - y_1)^2\right].$$

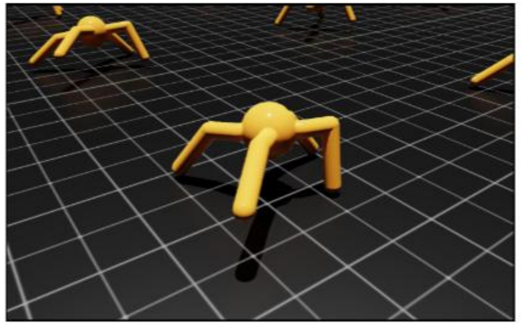
$$\mathcal{L}^{ENT}(\pi_\theta) := \mathbb{E}_{s, \tilde{a} \sim \pi_\theta}\left[\log\left(\pi_\theta(\tilde{a} \mid s)\right)\right].$$

$$\mathcal{L}^{\text{PPO}}(\theta) := \mathbb{E}_{(s_t, \tilde{a}_t) \sim \pi_{\theta_{\text{old}}}}\left[\min\left(\frac{\pi_\theta(\tilde{a}_t \mid s_t)}{\pi_{\theta_{old}}(\tilde{a}_t \mid s_t)}\hat{A}_t,\ \text{clip}(\frac{\pi_\theta(\tilde{a}_t \mid s_t)}{\pi_{\theta_{old}}(\tilde{a}_t \mid s_t)}, 1 - \epsilon, 1 + \epsilon)\hat{A}_t\right)\right].$$

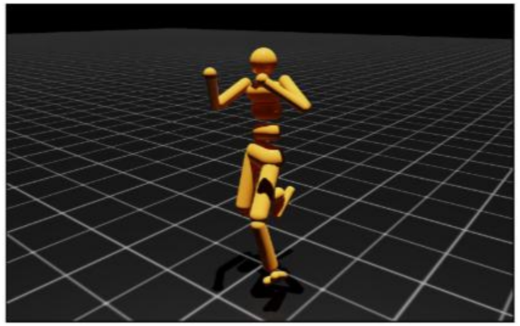
1. Since GenPO can access the log-likelihood of diffusion model, we directly calculate the RL loss and entropy of diffusion policy

2. To avoid **unnecessary exploration** in the doubled action space, we also propose the compression loss to further maintain the consistency of x and y

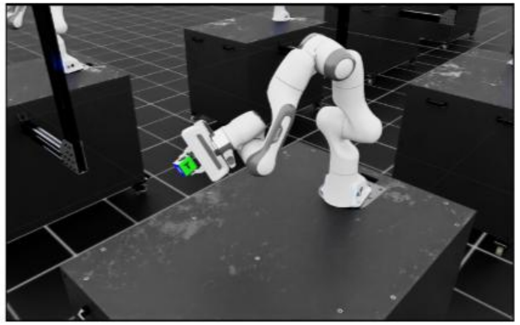
IsaacLab Benchmarks



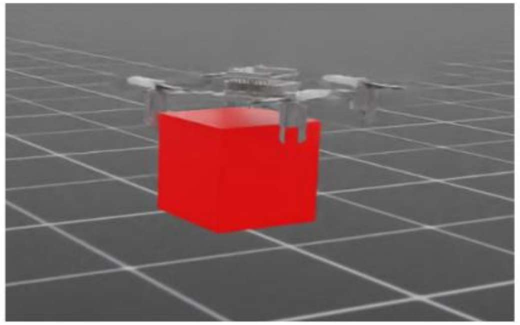
(a) Ant



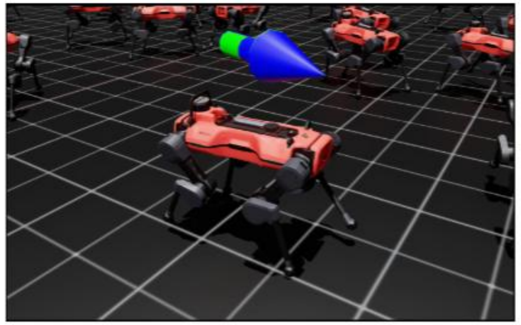
(b) Humanoid



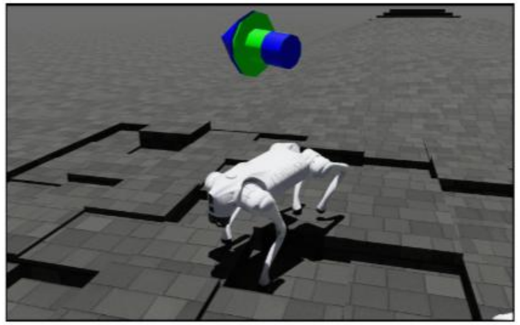
(c) Franka Arm



(d) Quadcopter

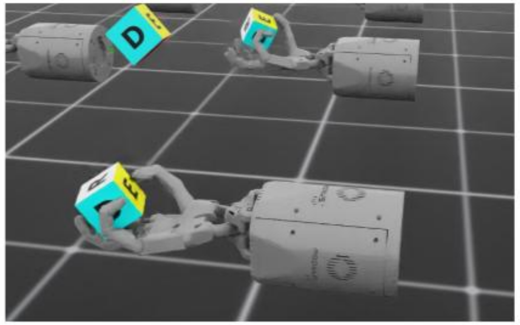


(e) Anymal-D



(f) Unitree-Go2

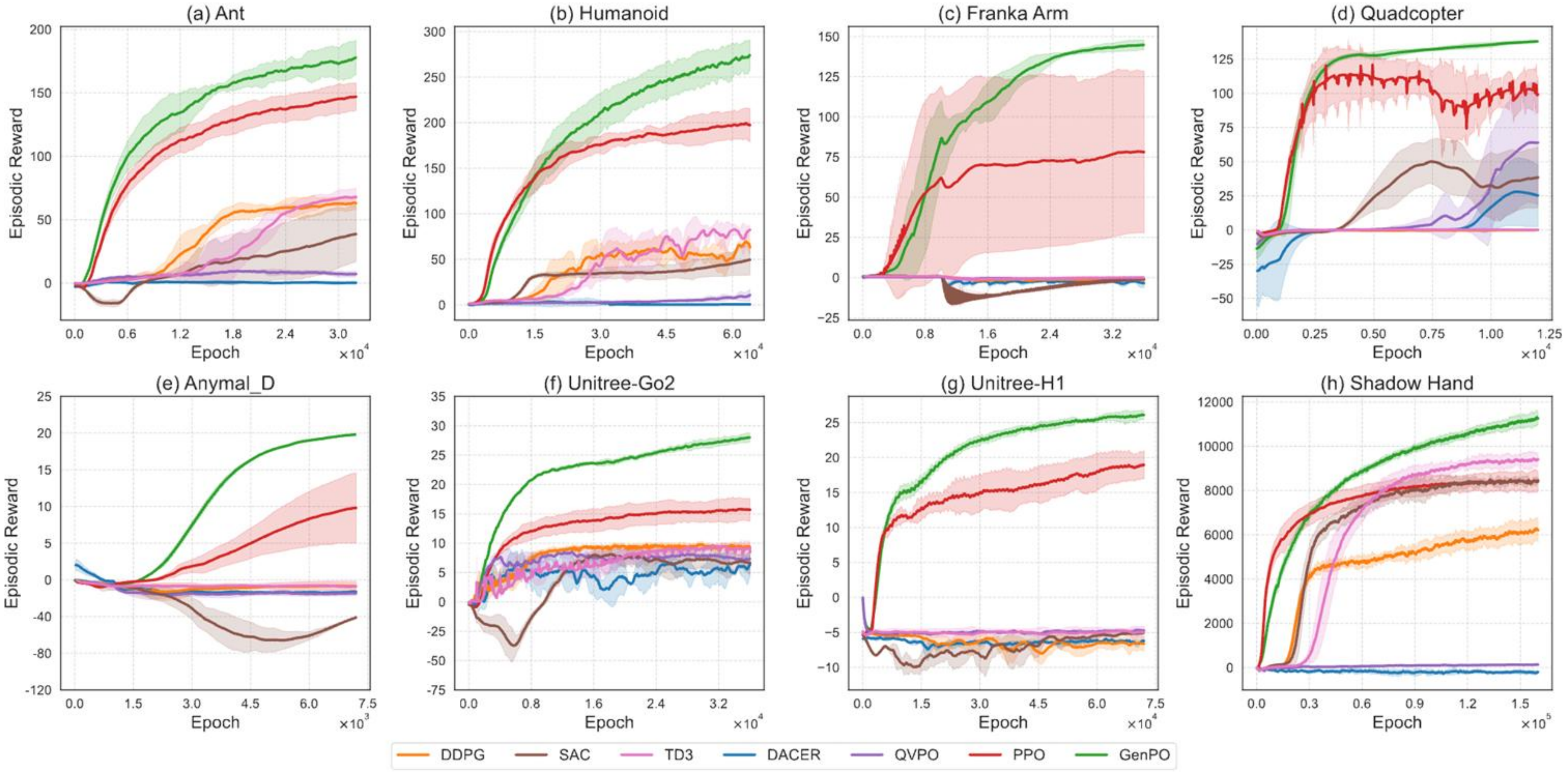
(g) Unitree-H1



(h) Shadow Hand

Results

上海科技大学
ShanghaiTech University



Performance in IsaacLab