# Simple Policy Optimization

Zhengpeng Xie

Chongqing Normal University

April 14, 2024

# Contents

# Reinforcement Learning: A Definition

What is RL? RL is generally defined by a tuple

$$(\mathcal{S}, \mathcal{A}, r, \mathcal{P}, \rho_0, \gamma)$$

where

- $\mathcal{S}$ and $\mathcal{A}$ represent the state space and action space
- $r : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the probability distribution of the state transition function
- $\rho_0 : \mathcal{S} \mapsto \mathbb{R}$ is the initial state distribution
- $\gamma \in [0, 1]$ is the discount factor

# Goal of RL

## Trajectory

An agent observes the current state $s$, takes action $a$, receives a reward $r(s, a)$ from the environment and updates the state until it reaches a terminal state

$$\tau = (s_1, a_1, r_1, \ldots, s_{T-1}, a_{T-1}, r_{T-1}, s_T)$$

- Define discounted return as $R(\tau) = \sum_{i=1}^{T-1} \gamma^{i-1} r_i$
- Our goal is to maximize the expectation of discounted return

$$J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\cdot)} [R(\tau)] = \sum_{\tau} [R(\tau) \cdot p_\theta(\tau)]$$

# Policy Gradient

## Our Goal

$$\max_{\theta} \quad J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\cdot)}\left[R(\tau)\right] = \sum_{\tau}\left[R(\tau) \cdot p_\theta(\tau)\right]$$

- The probability of a trajectory can be written as $p_\theta(\tau) = \rho_0(s_1) \cdot \prod_{t=1}^{T-1} \pi_\theta(a_t|s_t) \cdot \mathcal{P}(s_{t+1}|s_t, a_t)$
- The policy gradient will be

$$\nabla J(\theta) = \sum_{\tau}\left[R(\tau) \cdot \nabla p_\theta(\tau)\right] = \underbrace{\sum_{\tau}\left[R(\tau) \cdot \nabla \log p_\theta(\tau) \cdot p_\theta(\tau)\right]}_{\mathbb{E}_{\tau \sim p_\theta(\cdot)}\left[R(\tau) \cdot \nabla \log p_\theta(\tau)\right]}$$

# Policy Gradient

Then we have

$$\nabla J(\theta) = \mathbb{E}_{\tau \sim p_\theta(\cdot)} \left[ R(\tau) \cdot \nabla \log p_\theta(\tau) \right]$$

$$= \mathbb{E}_{\tau \sim p_\theta(\cdot)} \left\{ R(\tau) \cdot \nabla \log \left[ \rho_0(s_1) \cdot \prod_{t=1}^{T-1} \pi_\theta(a_t|s_t) \cdot \mathcal{P}(s_{t+1}|s_t, a_t) \right] \right\}$$

$$= \mathbb{E}_{\tau \sim p_\theta(\cdot)} \left\{ R(\tau) \cdot \nabla \left[ \log \rho_0(s_1) + \sum_{t=1}^{T-1} \log \pi_\theta(a_t|s_t) + \log \mathcal{P}(s_{t+1}|s_t, a_t) \right] \right\}$$

$$= \mathbb{E}_{\tau \sim p_\theta(\cdot)} \left[ \sum_{t=1}^{T-1} R(\tau) \cdot \nabla \log \pi_\theta(a_t|s_t) \right] \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_n - 1} \left[ R(\tau^n) \cdot \nabla \log \pi_\theta(a_t^n|s_t^n) \right]$$

# Importance Sampling

- We aim to improve sample efficiency by reusing the currently collected trajectories
- According to the definition of expectation, we have

$$\mathbb{E}_{x\sim p(x)} f(x) = \mathbb{E}_{x\sim q(x)} \frac{p(x)}{q(x)} f(x)$$

- Then we have

$$\begin{aligned}\nabla J(\theta) &= \mathbb{E}_{\tau\sim p_\theta(\cdot)} \left[R(\tau)\cdot \nabla \log p_\theta(\tau)\right] \\ &= \mathbb{E}_{\tau\sim p_{\theta_{\text{old}}}(\cdot)} \left[\frac{p_\theta(\tau)}{p_{\theta_{\text{old}}}(\tau)} \cdot R(\tau)\cdot \nabla \log p_\theta(\tau)\right]\end{aligned}$$

# New Policy Gradient and Surrogate Objective

For each data $(s_t, a_t)$, we have

$$\nabla J_{\theta_{\text{old}}}(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[\frac{p_\theta(s_t,a_t)}{p_{\theta_{\text{old}}}(s_t,a_t)} \cdot \overbrace{\hat{A}(s_t,a_t)}^{\approx Q(s_t,a_t)-V(s_t)} \cdot \nabla \log \pi_\theta(a_t|s_t)\right]$$

$$= \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[\frac{p_\theta(s_t)}{p_{\theta_{\text{old}}}(s_t)} \cdot \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot \hat{A}(s_t,a_t) \cdot \nabla \log \pi_\theta(a_t|s_t)\right]$$

$$\approx \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot \hat{A}(s_t,a_t) \cdot \nabla \log \pi_\theta(a_t|s_t)\right] \Longrightarrow$$

$$J_{\theta_{\text{old}}}(\theta) = \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot \hat{A}(s_t,a_t)\right]$$

# Policy Optimization with Trust Region Methods

## Trust Region Policy Optimization (TRPO)

$$\max_{\theta} \ \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}} \left[ \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot \hat{A}(s_t,a_t) \right]$$
$$\text{s.t.} \ \ \mathbb{E}\left[ D_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) \right] \leq \delta$$

- Optimize surrogate objective $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot \hat{A}(s_t,a_t)$
- Constrain the KL divergence $D_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta)$
- Able to achieve monotonic improvement of policy
- A constrained optimization problem

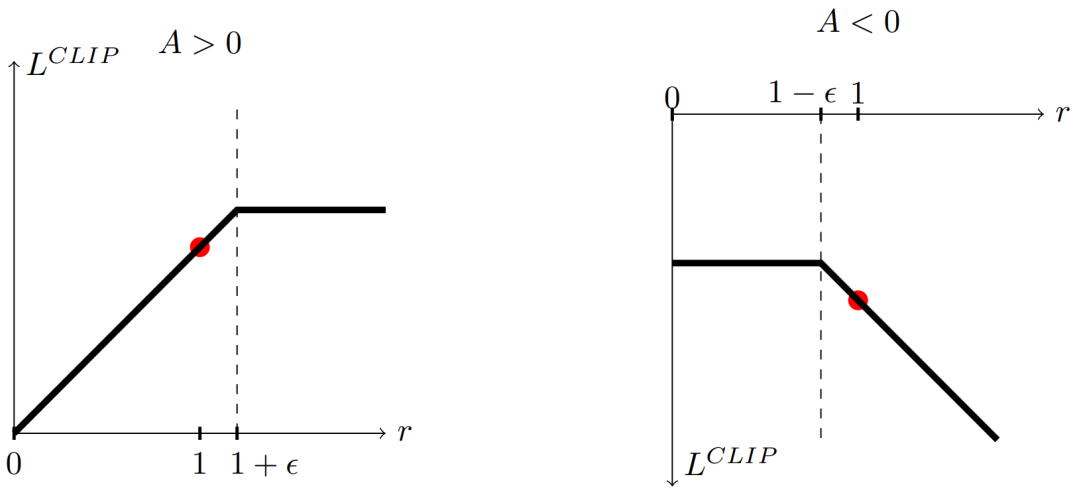
# PPO with Ratio Clipping

## PPO-CLIP

$$\max_{\theta} \mathop{\mathbb{E}}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}} \left\{ \min \left[ \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \cdot \hat{A}(s_t,a_t), \right. \right.$$
$$\left. \left. \text{clip}\left( \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) \cdot \hat{A}(s_t,a_t) \right] \right\}$$

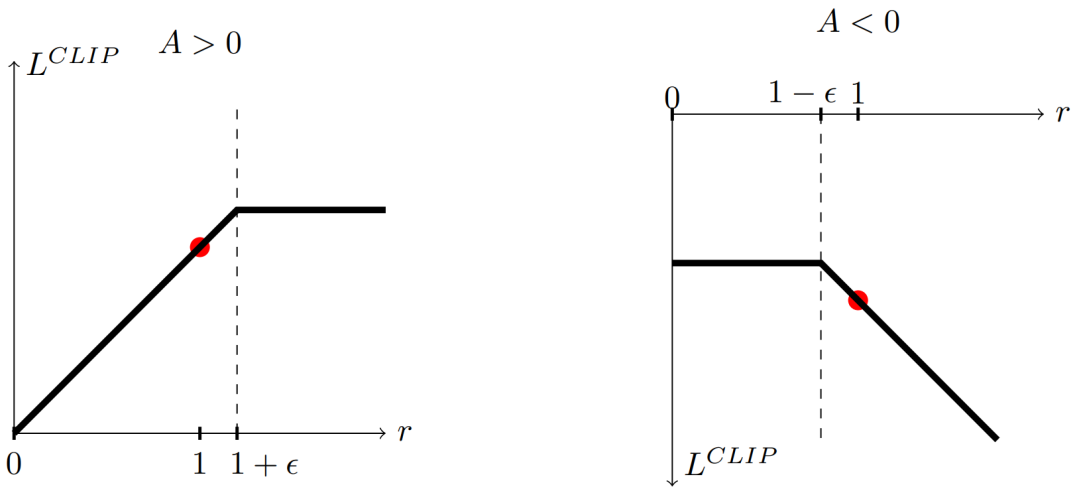
- An unconstrained optimization problem
- Current data can be reused for training

# How does the PPO Work?




- If $\hat{A}(s_t, a_t) > 0$, increase the probability of action $a_t$ until $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} = 1 + \epsilon$
- If $\hat{A}(s_t, a_t) < 0$, decrease the probability of action $a_t$ until $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} = 1 - \epsilon$

# How does the PPO Work?




- So, for each data $(s_t, a_t)$, if $|\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} - 1| \leq \epsilon$, it will propagate gradients during the training process
- Or, if $|\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} - 1| > \epsilon$, the gradient of $(s_t, a_t)$ during the training process will be 0

# Can PPO Really Constrain the KL Divergence?

- The answer is no! See paper "Truly Proximal Policy Optimization"

**Theorem 3.** *Assume that for discrete action space tasks where* $|\mathcal{A}| \geq 3$ *and the policy is* $\pi_\theta(s) = f_\theta^p(s)$, *we have* $\{f_\theta^p(s_t)|\theta \in \mathbb{R}\} = \{p|p \in \mathbb{R}^{+D}, \sum_d^D p^{(d)} = 1\}$; *for continuous action space tasks where the policy is* $\pi_\theta(a|s) = \mathcal{N}(a|f_\theta^\mu(s), f_\theta^\Sigma(s))$, *we have* $\{(f_\theta^\mu(s_t), f_\theta^\Sigma(s_t))|\theta \in \mathbb{R}\} = \{(\mu, \Sigma)|\mu \in \mathbb{R}^D, \Sigma$ *is a symmetric semidefinite* $D \times D$ *matrix*$\}$. *Let* $\Theta = \{\theta|1 - \epsilon \leq r_t(\theta) \leq 1 + \epsilon\}$. *We have* $\sup_{\theta \in \Theta} D_{\mathrm{KL}}^{s_t}(\theta_{\mathrm{old}}, \theta) = +\infty$ *for both discrete and continuous action space tasks.*

# Our goal

## Lower Bound of Monotonic Improvement in TRPO

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - \frac{2\epsilon\gamma}{(1-\gamma)^2} \cdot D_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi})$$

According to the lower bound of monotonic improvement in TRPO, we want to achieve two goals:

- Optimize $\mathfrak{O}_{\theta_{\mathrm{old}}}^{\theta}(s_t, a_t) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)} \cdot \hat{A}(s_t, a_t)$
- Limit the KL divergence $d = \sum_{a \in \mathcal{A}} \pi_{\theta_{\mathrm{old}}}(a|s_t) \cdot \log \frac{\pi_{\theta_{\mathrm{old}}}(a|s_t)}{\pi_\theta(a|s_t)}$

# New Objective

## Simple Policy Optimization

$$\max_{\theta} \mathop{\mathbb{E}}_{(s_t,a_t)\sim\pi_{\theta_{\text{old}}}} \left[ \mathfrak{O}^{\theta}_{\theta_{\text{old}}}(s_t,a_t) \cdot \left( \frac{d_{\text{clip}}}{d} + \sigma - 1 \right) \cdot \sigma \right]$$

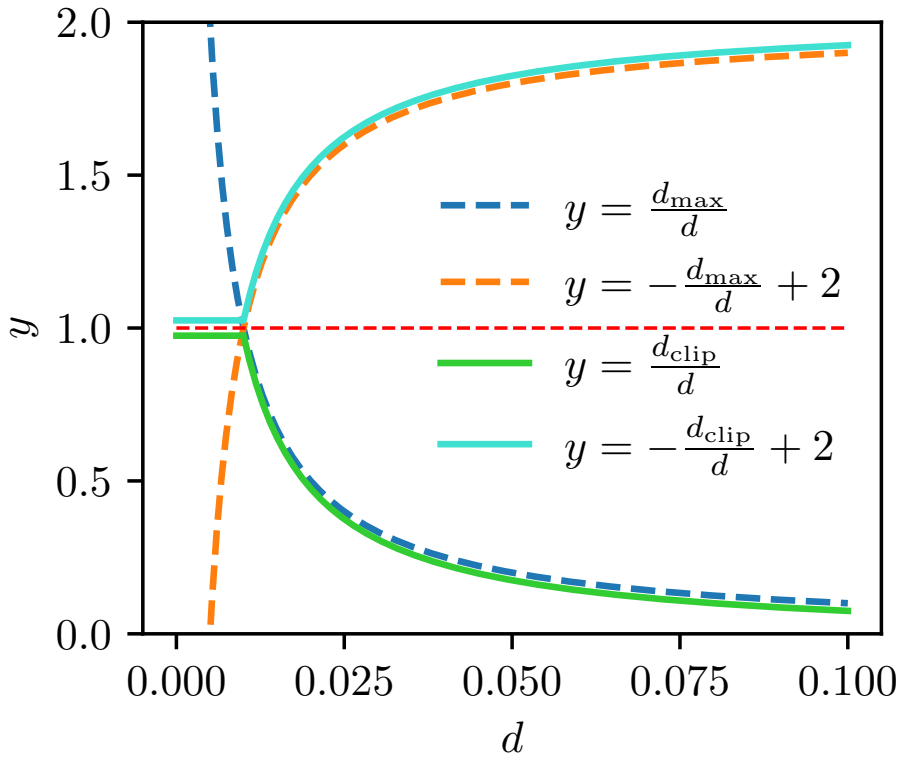
Where

- $d_{\text{clip}} = \text{clip}(d, 0, d_{\max})$, $d_{\max}$ is a hyperparameter
- $\sigma = \text{sign}\left[\hat{A}(s_t, a_t)\right]$
- An unconstrained first-order algorithm

# How Does the SPO Work?

- Let's assume for simplicity that the advantage function is positive ($\sigma = 1$)
- Then the objective will be $\mathfrak{O}^{\theta}_{\theta_{\text{old}}}(s_t, a_t) \cdot \frac{d_{\text{clip}}}{d}$
  - If $d \leq d_{\max}$, it will be the surrogate obective $\mathfrak{O}^{\theta}_{\theta_{\text{old}}}(s_t, a_t)$
  - If $d > d_{\max}$, under this condition, the objective becomes $\frac{\mathfrak{O}^{\theta}_{\theta_{\text{old}}}(s_t, a_t)}{\frac{d}{d_{\max}}}$. At this point, if $d$ continues to increase, the objective function will become very small due to its ratio form
- Therefore, in order to optimize the objective, SPO will secretly optimize the surrogate objective $\mathfrak{O}^{\theta}_{\theta_{\text{old}}}(s_t, a_t)$ within the trust region $d \leq d_{\max}$ as much as possible, and this will improve the policy (results in TRPO)

# How Does the SPO Work?




- However, when $\hat{A}(s_t, a_t)$ is negative ($\sigma = -1$) and $d$ exceeds the threshold, we need to increase the absolute value of objective as a penalty instead of decreasing it

# The Main Differences Between SPO and PPO

For a given data $(s_t, a_t)$,

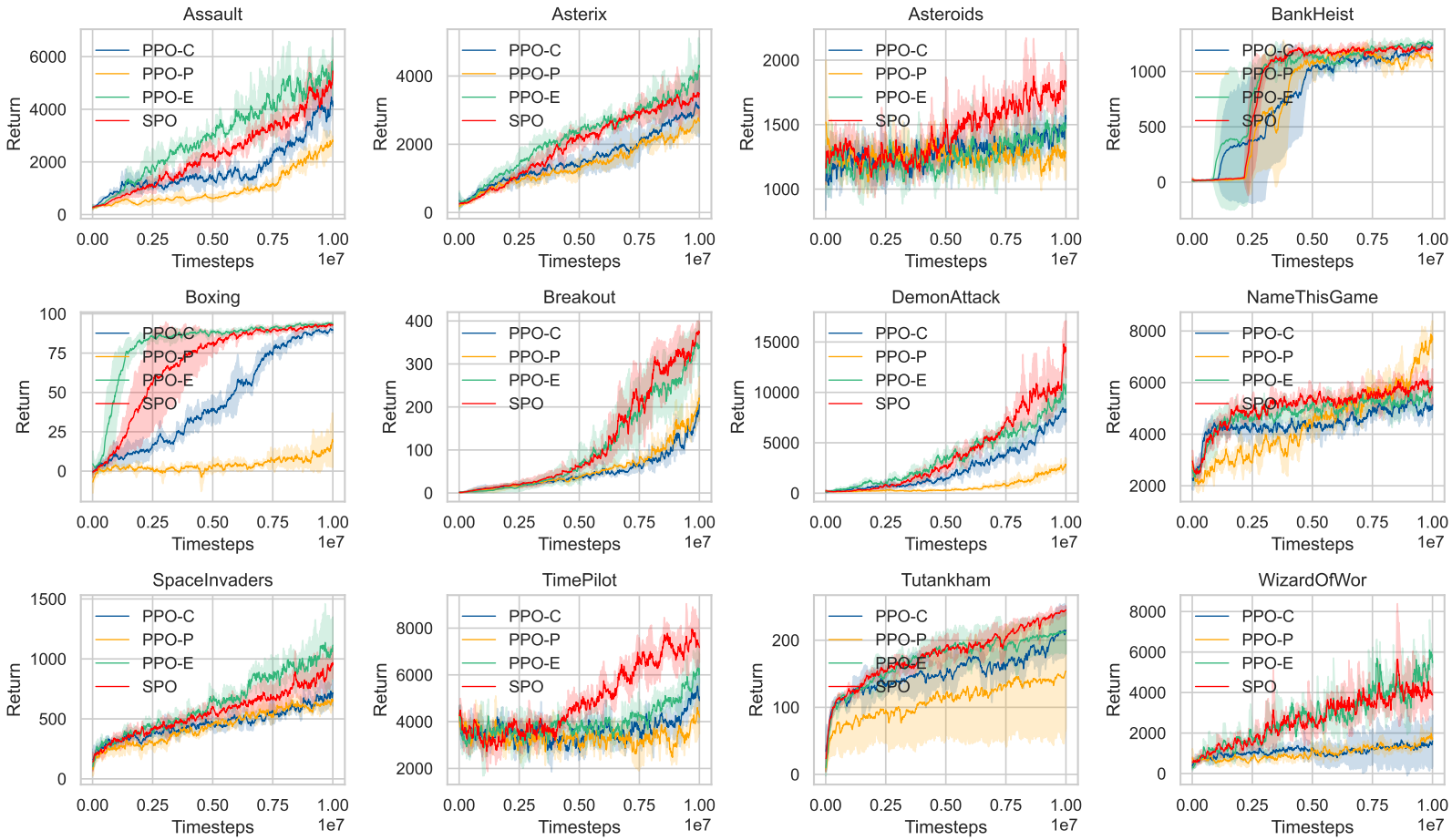
- For PPO, we want to know the probability ratio of action $a_t$ between $\pi_\theta$ and $\pi_{\theta_{\text{old}}}$, not all data will propagate gradients in PPO
- For SPO, we want to know the KL divergence between $\pi_\theta$ and $\pi_{\theta_{\text{old}}}$ under state $s_t$, all data will propagate gradients in SPO

# The Advantages of SPO Compared to PPO

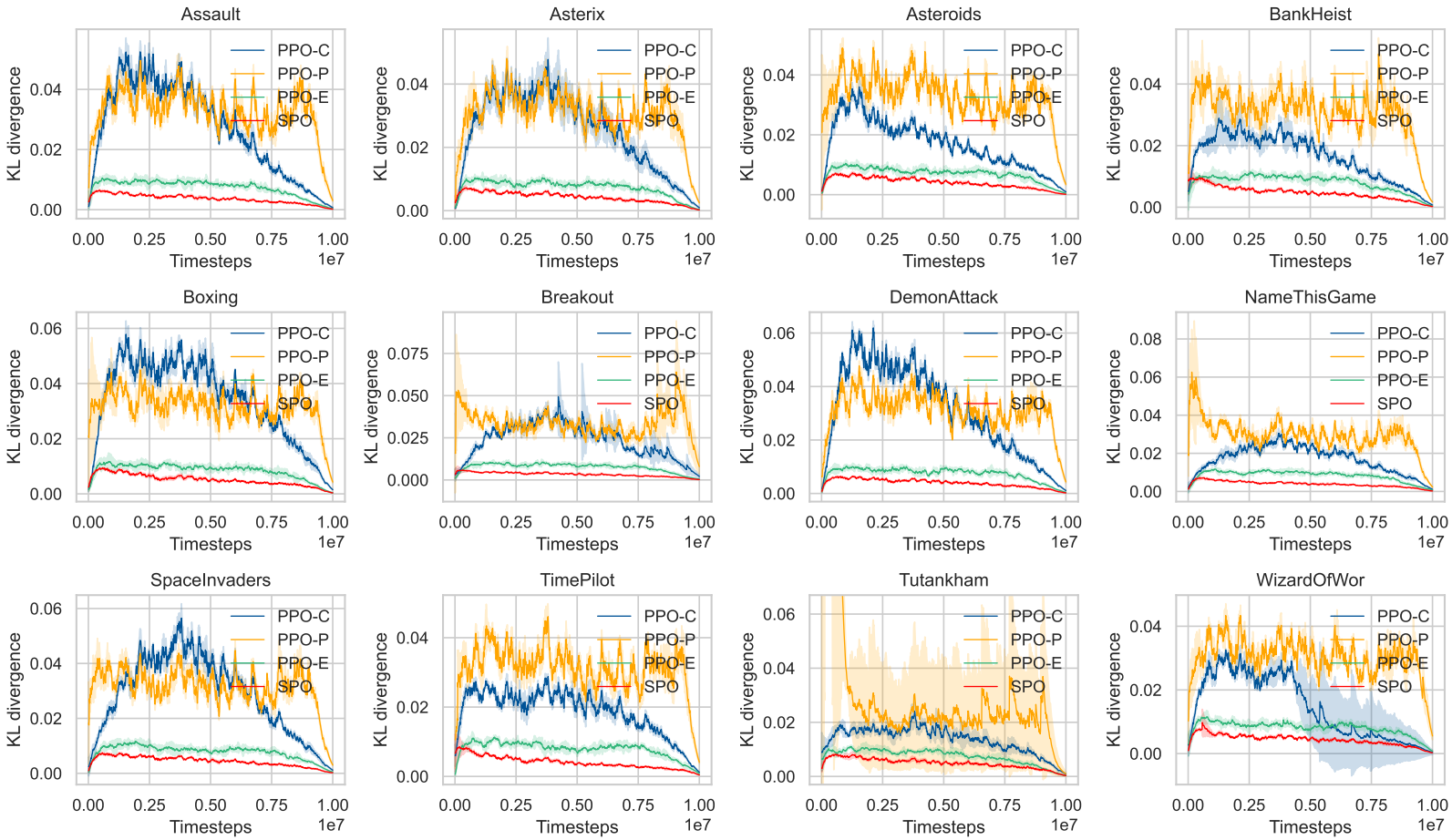
Here are some reasons about why we choose SPO:

- SPO achieves better sample efficiency than PPO (almost)
- The KL divergence between successive policies is controlled in SPO
- SPO is robust to the increase in network depth or complexity, but PPO is not
- SPO maintains the simplicity of an unconstrained first-order algorithm (compared to TRPO)

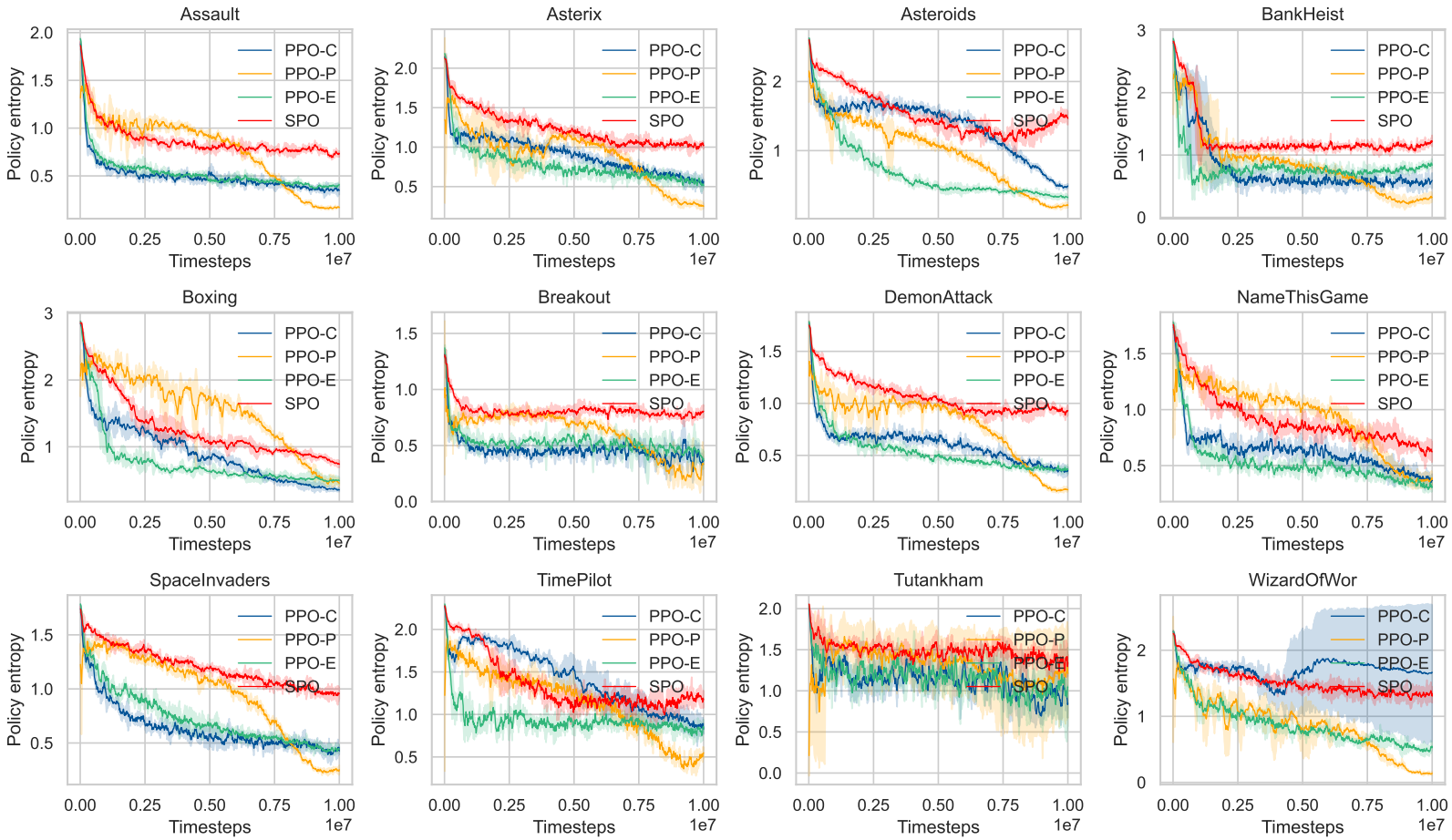
# Main Results



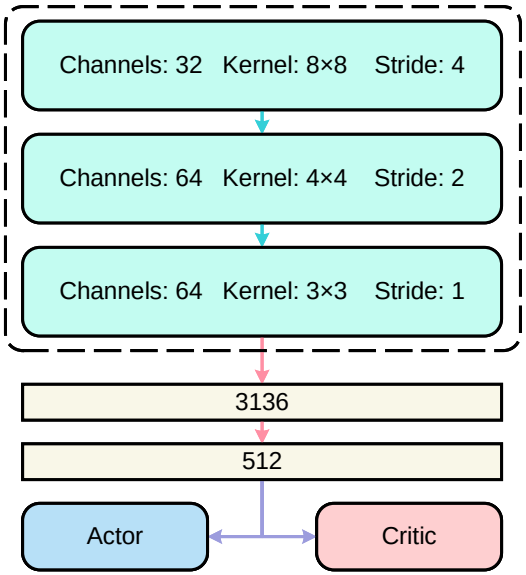
# Main Results



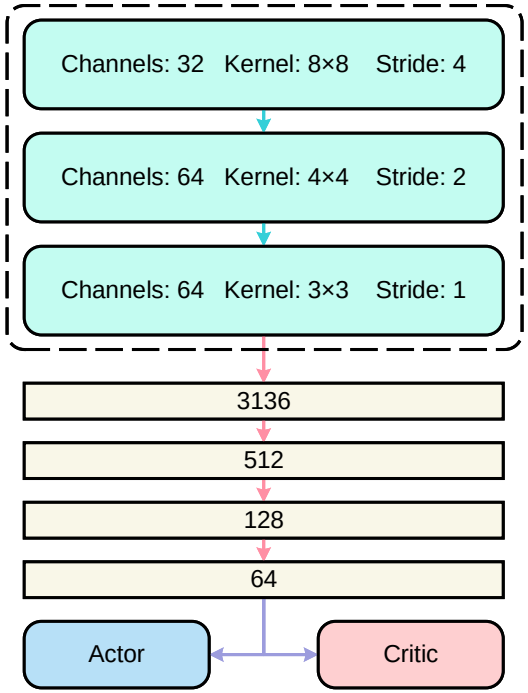
# Main Results



# Network Deepens

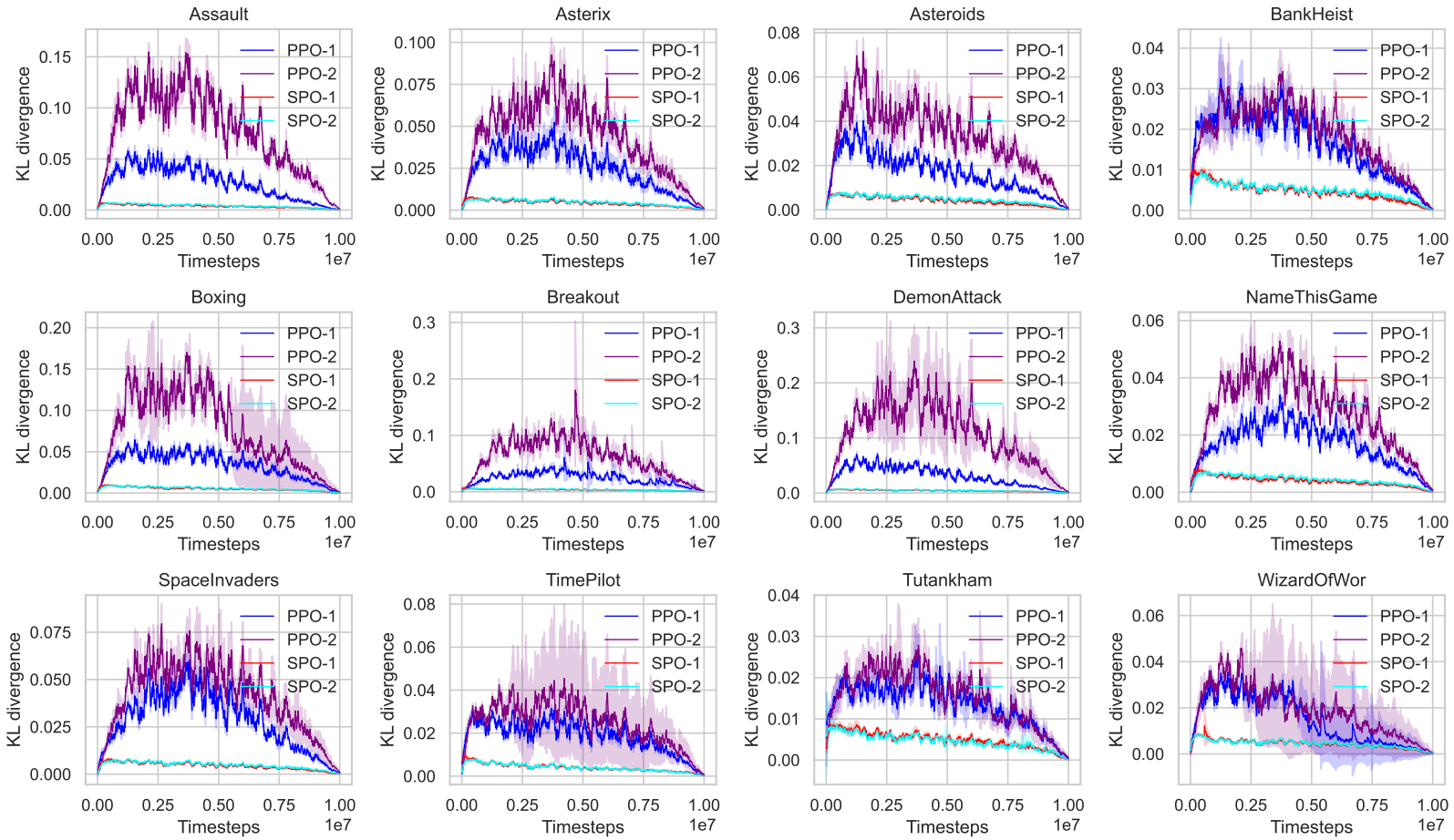



Shallow network

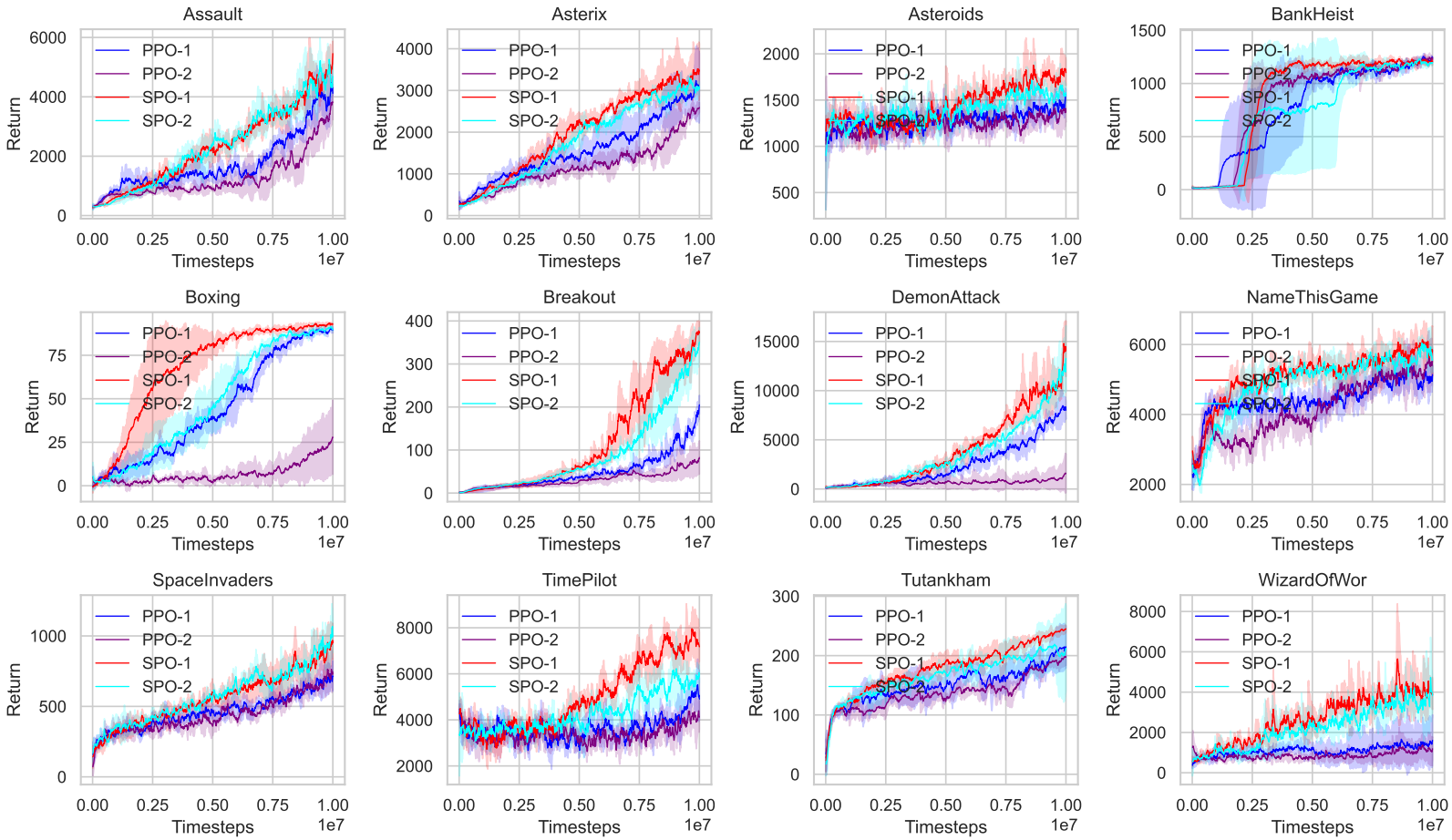



Deeper network

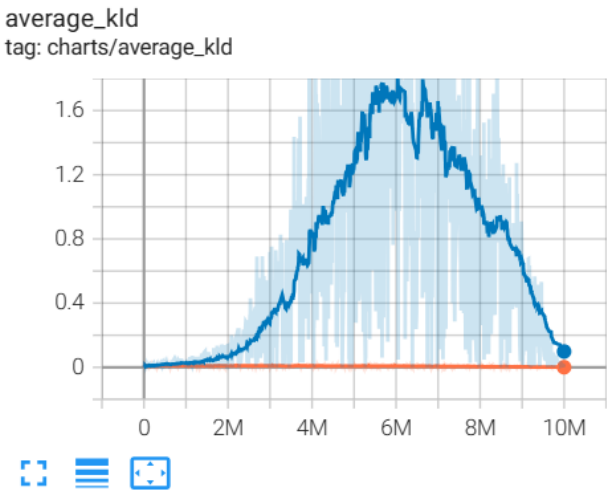
# Network Deepens

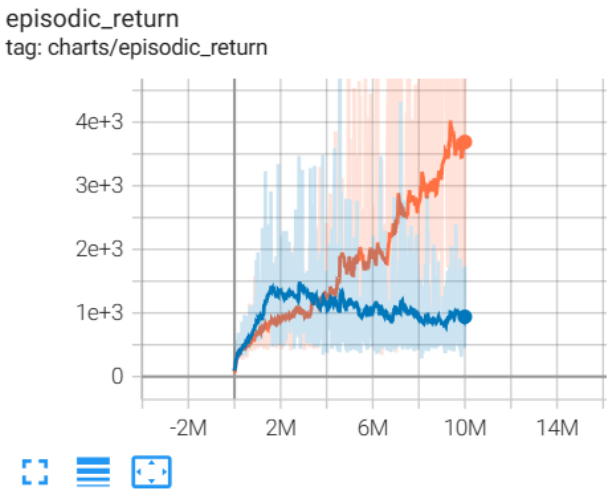


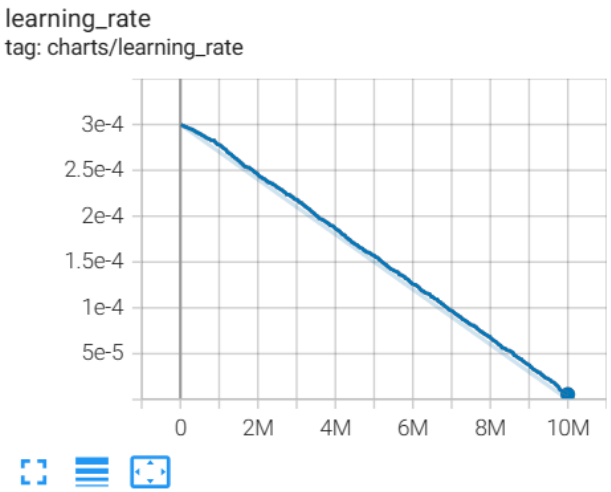
# Network Deepens



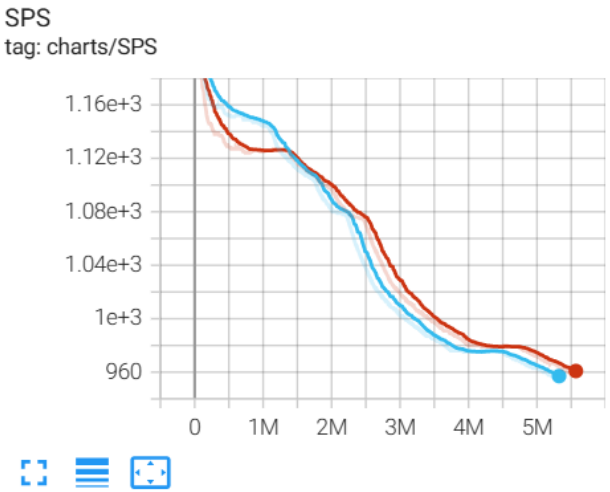
# Scalability of SPO for Continuous Action Spaces

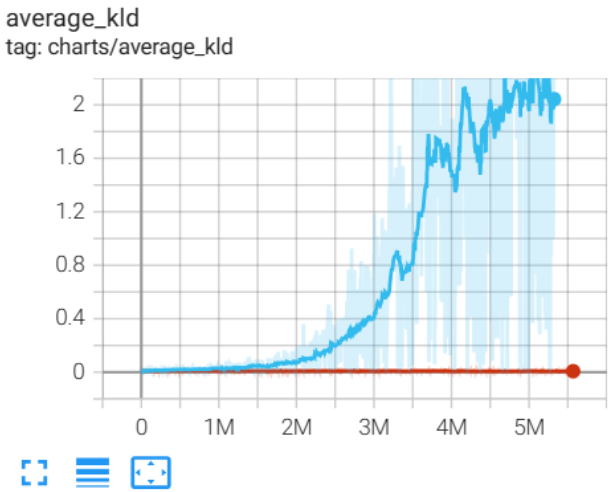



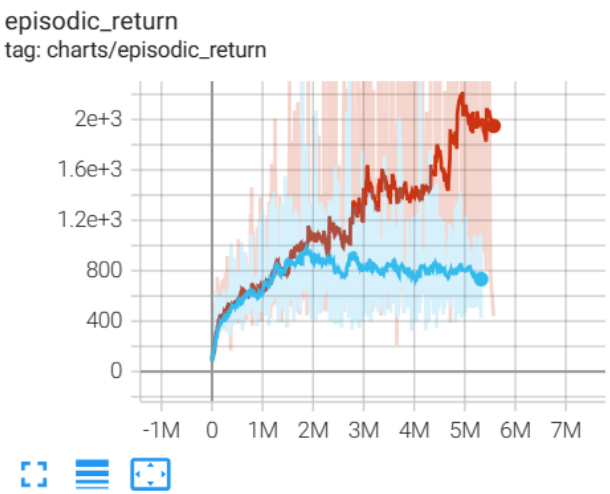



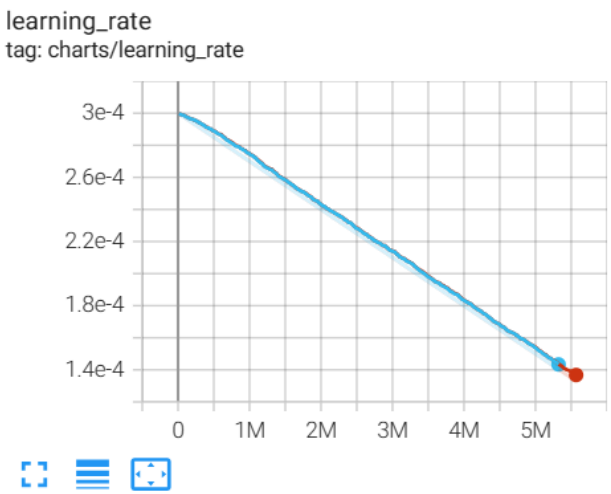



# Scalability of SPO for Continuous Action Spaces



# Can We Use SPO rather than PPO in RLHF?

- In RL Fine-Tuning Phase, the objective is

$$\max_{\theta} \ \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right]-\beta\cdot D_{\mathsf{KL}}\left[\pi_\theta(y|x),\pi_{\mathsf{ref}}(y|x)\right]$$

- So what will happen if we just use

$$\max_{\theta} \ \mathop{\mathbb{E}}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[\mathfrak{D}^{\theta}_{\theta_{\text{old}}}(s_t,a_t)\cdot\left(\frac{d_{\text{clip}}}{d}+\sigma-1\right)\cdot\sigma\right]$$

where $d=D_{\mathsf{KL}}\left[\pi_\theta(y|x),\ \underbrace{\pi_{\mathsf{ref}}(y|x)}_{\text{not }\pi_{\theta_{\text{old}}}(y|x)}\right]$

# More Generally, Can We Use SPO rather than PPO in Alignment?

- Paper "Training Diffusion Models with Reinforcement Learning" use PPO to fine-tune the Diffusion Models
- So what would happen if we directly use SPO as the corresponding algorithm? Given that SPO can be scaled to continuous action spaces