Proximal Policy Optimization with MPC-based policy

Hien Bui, Haoxiang You, Ye Duan

INTRODUCTION

Motivation

- Reinforcement learning (RL), particularly model-free algorithms, has achieved remarkable success in complex locomotion and manipulation tasks in recent years.
- Model-free RL algorithms like **Proximal Policy Optimization (PPO)** (*J Schulman et al. 2017*) often require enormous data, which might be difficult or expensive to obtain in real-world settings.

This work proposes a novel method that is **more data efficient** and yet could yield **similar performance** as PPO.

BACKGROUND

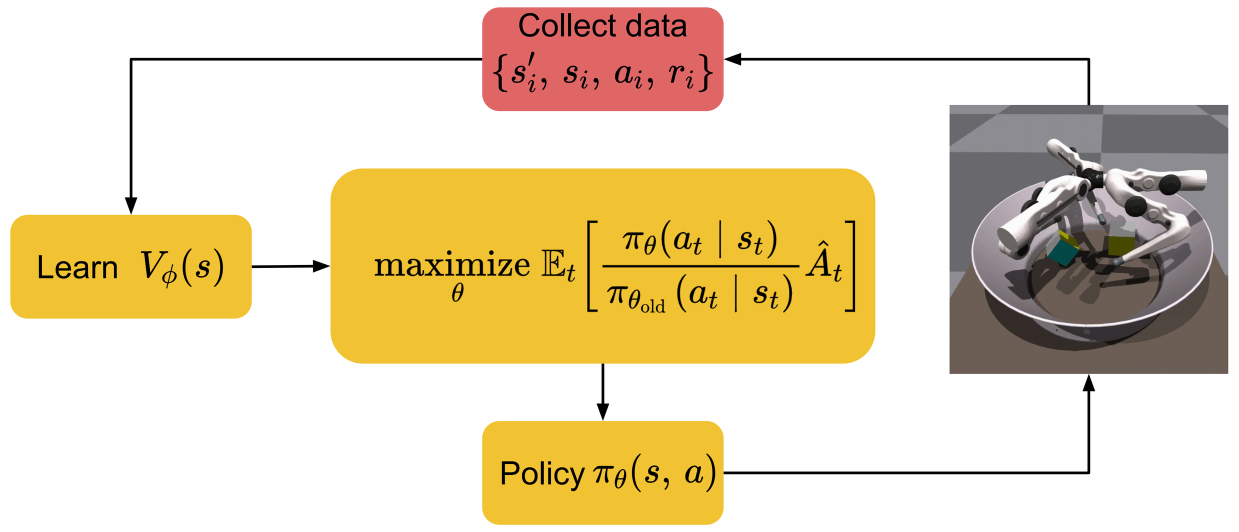
PPO Framework

- PPO is a policy gradient algorithm that concerns: using current data, how can we take the biggest possible improvement step on a policy, without stepping so far that we accidentally cause performance collapse.
- PPO loss function is defined as follows

$$L_\theta^{CLIP} = \mathbb{E}_t\Big[\min\Big(h_t^\theta \hat{A}_t, \text{clip}\big(h_t^\theta, 1-\epsilon, 1+\epsilon\big)\hat{A}_t\Big)\Big]$$

$h_t^\theta = \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$ is the ratio between new and old policy

$\hat{A}_t = r_t + \gamma V_\phi(s') - V_\phi(s)$ is called advantage function which measures how much is a certain action a good or bad decision given a certain state



METHODS

- We formulate the policy of PPO as a MPC problem instead of a neural network.
  - In this MPC problem, we choose a hybrid and piecewise-linear model, **Linear Complementarity System (LCS)**, to represent the simplified dynamics model. LCS is known to sufficiently capture the contact dynamics that frequently arise in manipulation tasks.

$$\pi_\theta(s, a) = \mathcal{N}(\mu_{\theta^-}(s), \sigma^2 I)$$

$$\mu_{\theta^-}(s) = \underset{a}{\text{argmax}} \sum_{k=0}^{H-1} r(s_k, a_k) + V(s_H)$$

$$\begin{aligned}\text{s.t } s_{k+1} &= As_k + Ba_k + C\lambda_k + d \\ 0 \le \lambda_k &\perp Ds_k + Ea_k + F\lambda_k + c \ge 0\end{aligned}$$

  - We need to optimize the following parameters

$$\theta = (\theta^-, \sigma) = (A, B, C, d, D, E, F, c, \sigma)$$

- The gradient of the PPO loss with respect to the above parameters is computed as follows

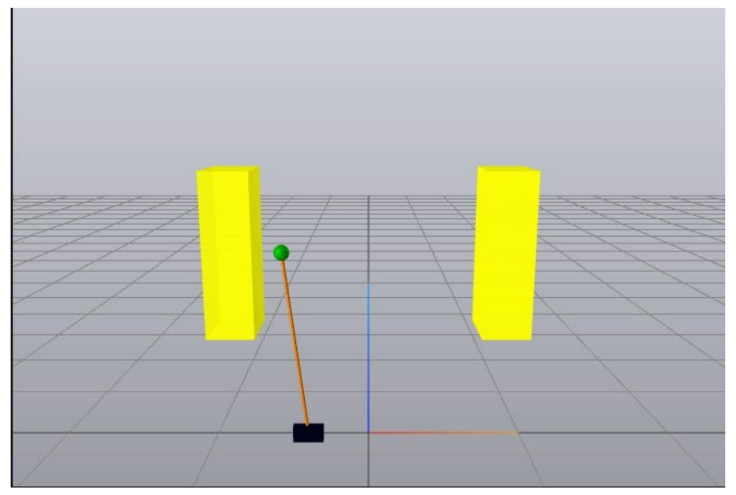
$$\frac{d\,L_\theta^{CLIP}}{d\,\theta} = \frac{1}{|D|\,T}\sum_{\tau\in D}\sum_{t=0}^{T}\begin{cases}\frac{d\,h_t^\theta}{d\,\pi_\theta}\frac{d\,\pi_\theta}{d\,\theta}\hat{A}_t & \text{if } h_t^\theta \ge 1-\epsilon \text{ and } \hat{A}_t < 0\\ 0 & \text{if } h_t^\theta < 1-\epsilon \text{ and } \hat{A}_t < 0\\ \frac{d\,h_t^\theta}{d\,\pi_\theta}\frac{d\,\pi_\theta}{d\,\theta}\hat{A}_t & \text{if } h_t^\theta \le 1+\epsilon \text{ and } \hat{A}_t \ge 0\\ 0 & \text{if } h_t^\theta > 1+\epsilon \text{ and } \hat{A}_t \ge 0\end{cases}$$

  - Computing $\dfrac{d\,\pi_\theta}{d\,\theta}$ requires differentiation through MPC problem.
  - We use differentiation method from previous work *Safe Pontryagin Differentiable Programming (Wanxin Jin et al. 2020)*
- We also add prediction loss into PPO loss to ensure the learned dynamics model in MPC policy could predict meaningful future states.

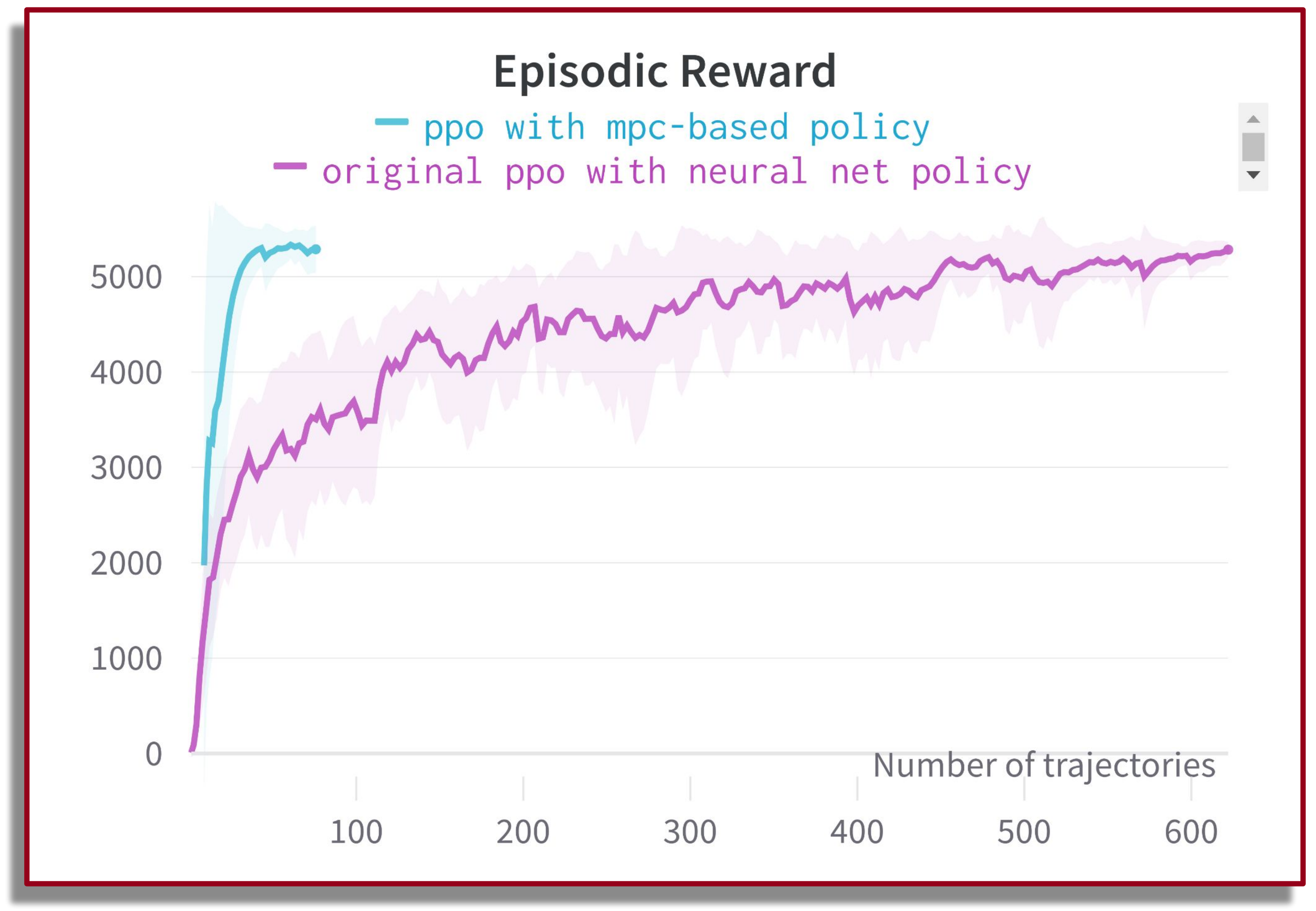
$$L_{\theta^-}^{pred} = -\frac{1}{|\mathcal{D}|T}\sum_{\tau\in\mathcal{D}}\sum_{t=0}^{T}\left\|s_{t+1} - LCS_{\theta^-}(s_t, a_t)\right\|^2$$

RESULTS

- Cartpole with soft walls task: stabilizing cartpole after impacts with walls



- We train and compare the performance of our proposed framework with the original PPO
- For each case, we run 5 experiments with 5 different random seeds.



See our demo here

FUTURE WORK

- We will test the proposed framework on more difficult tasks such as object manipulation with a trifinger robot.
- We will compare the effectiveness of this method to that of other state-of-the-art model-based RL algorithms.