2024951308 Robin Henri Dieudonné

DATA403(00) Student Presentation

(Reinforcement Learning)

Contents

1. Selected environments
2. Selected Algorithm(s)
3. Results
4. Conclusion

1. Selected Environments

- 2 Environments
  - Ant (Lv.2)
  - HumanoidStandup (Lv.3)

- I selected them because it is like watching children grow up, which is fun

2. Selected Algorithm(s)

- PPO for Ant-v4 & HumanoidStandup-v4

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, \ldots$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

Compute rewards-to-go $\hat{R}_t$.

Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.

Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \;\; g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

typically via stochastic gradient ascent with Adam.

Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically via some gradient descent algorithm.

end for

2. Selected Algorithm(s)

- PPO hyperparameters for Ant-v4 & HumanoidStandup-v4

batch_size=64	*vf_coeff=0.5*
n_steps=2048	*max_grad_norm=0.5*
***lr**=3e-4 (default)*	*gae_lambda=0.95*
buffer_size=1e6	*gamma=0.99*
***entropy_coeff**=0.0 (default)*	*clip_range=0.2*

We use a small learning rate with the Adam optimizer as we train for several steps
Parameters that we play with are the learning rate and the entropy coefficient.

2. Selected Algorithm(s)

- TD3/Twin-Delayed DDPG for Ant-v4

Algorithm 1 Twin Delayed DDPG

Input: initial policy parameters θ, Q-function parameters ϕ_1, ϕ_2, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters $\theta_{\text{targ}} \leftarrow \theta$, $\phi_{\text{targ},1} \leftarrow \phi_1$, $\phi_{\text{targ},2} \leftarrow \phi_2$
repeat
  Observe state s and select action $a = \text{clip}(\mu_\theta(s) + \epsilon, a_{Low}, a_{High})$, where $\epsilon \sim \mathcal{N}$
  Execute a in the environment
  Observe next state s', reward r, and done signal d to indicate whether s' is terminal
  Store (s, a, r, s', d) in replay buffer $\mathcal{D}$
  If s' is terminal, reset environment state.
  if it's time to update **then**
    for j in range(however many updates) **do**
      Randomly sample a batch of transitions, $B = \{(s, a, r, s', d)\}$ from $\mathcal{D}$
      Compute target actions

$$a'(s') = \text{clip}\left(\mu_{\theta_{\text{targ}}}(s') + \text{clip}(\epsilon, -c, c), a_{Low}, a_{High}\right), \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

      Compute targets

$$y(r, s', d) = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', a'(s'))$$

      Update Q-functions by one step of gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\phi_i}(s, a) - y(r, s', d)\right)^2 \qquad \text{for } i = 1, 2$$

      if j mod `policy_delay` $= 0$ **then**
        Update policy by one step of gradient ascent using

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_{\phi_1}(s, \mu_\theta(s))$$

        Update target networks with

$$\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho)\phi_i \qquad \text{for } i = 1, 2$$
$$\theta_{\text{targ}} \leftarrow \rho \theta_{\text{targ}} + (1 - \rho)\theta$$

      end if
    end for
  end if
until convergence

3. Results

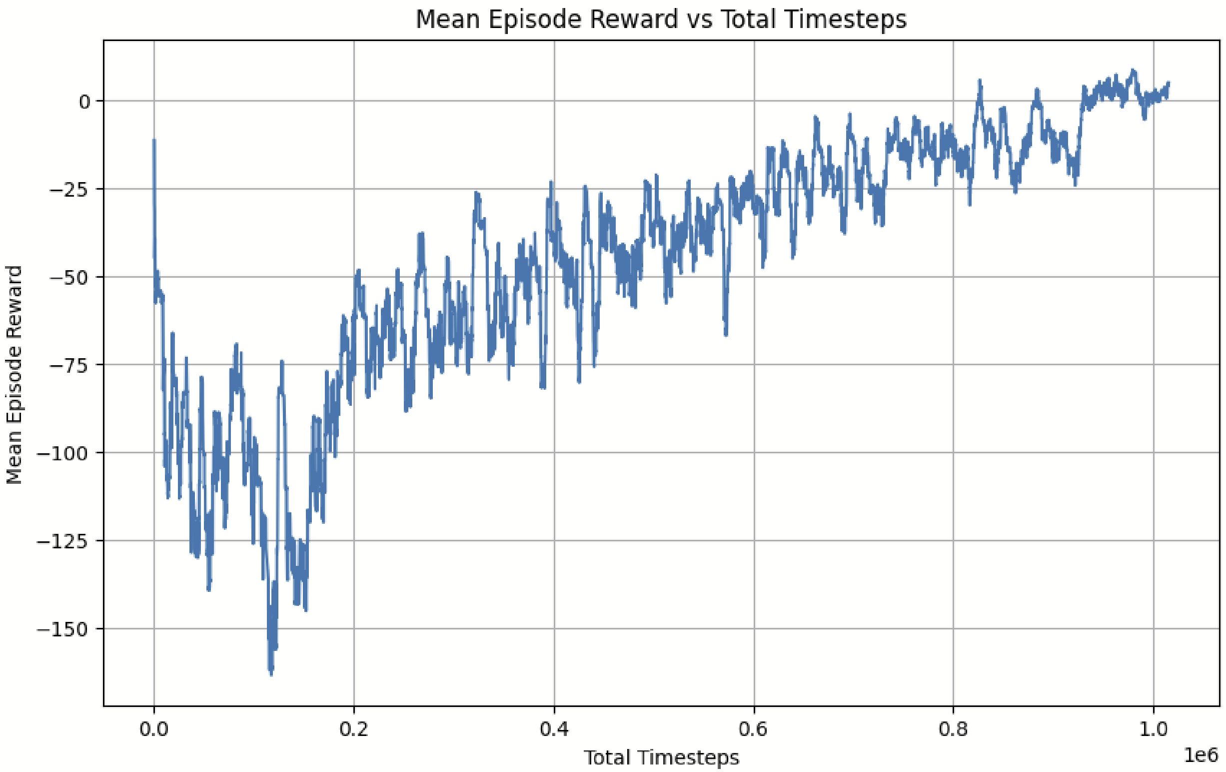
□ Ant-v4

Proximal Policy Optimization (PPO)

ENTROPY = 0.0

LEARNING RATE = 3e-4

FINAL EPISODE MEAN REWARD = 4.69



3. Results

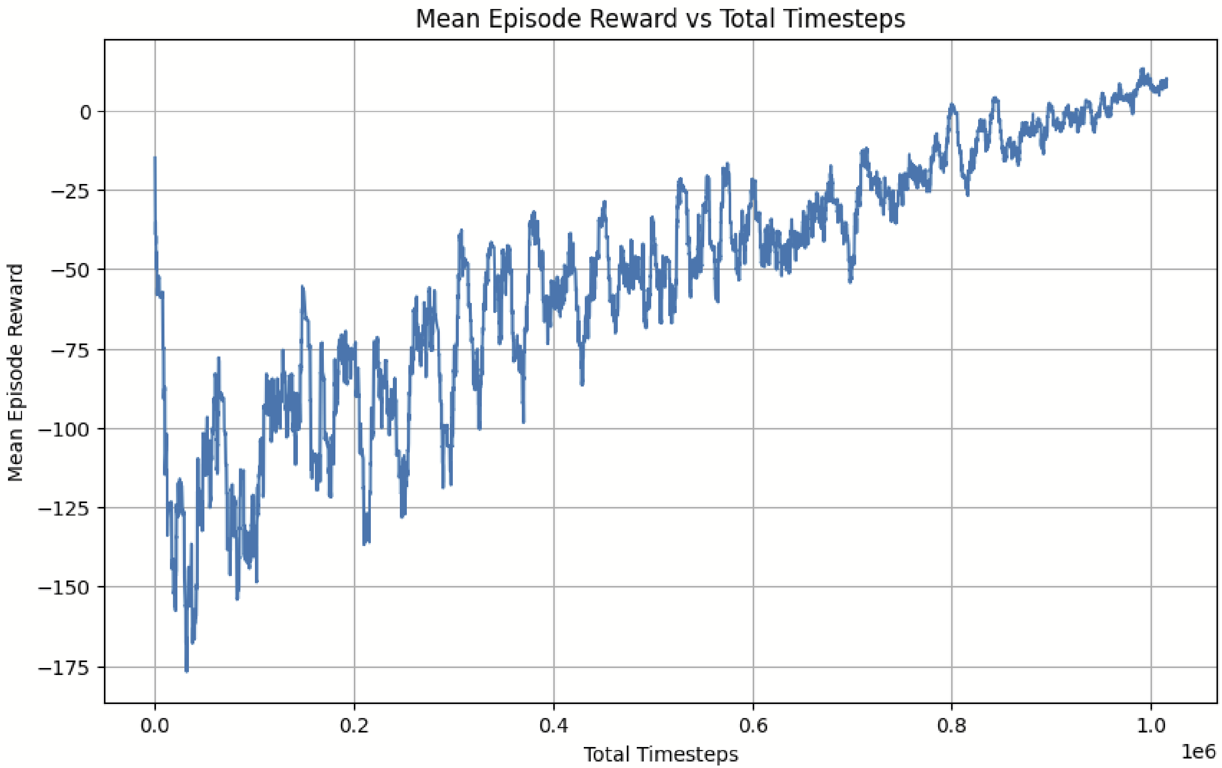
□ Ant-v4

Proximal Policy Optimization (PPO)

ENTROPY = 0.001

LEARNING RATE = 3e-4

FINAL EPISODE MEAN REWARD = 11.9



3. Results

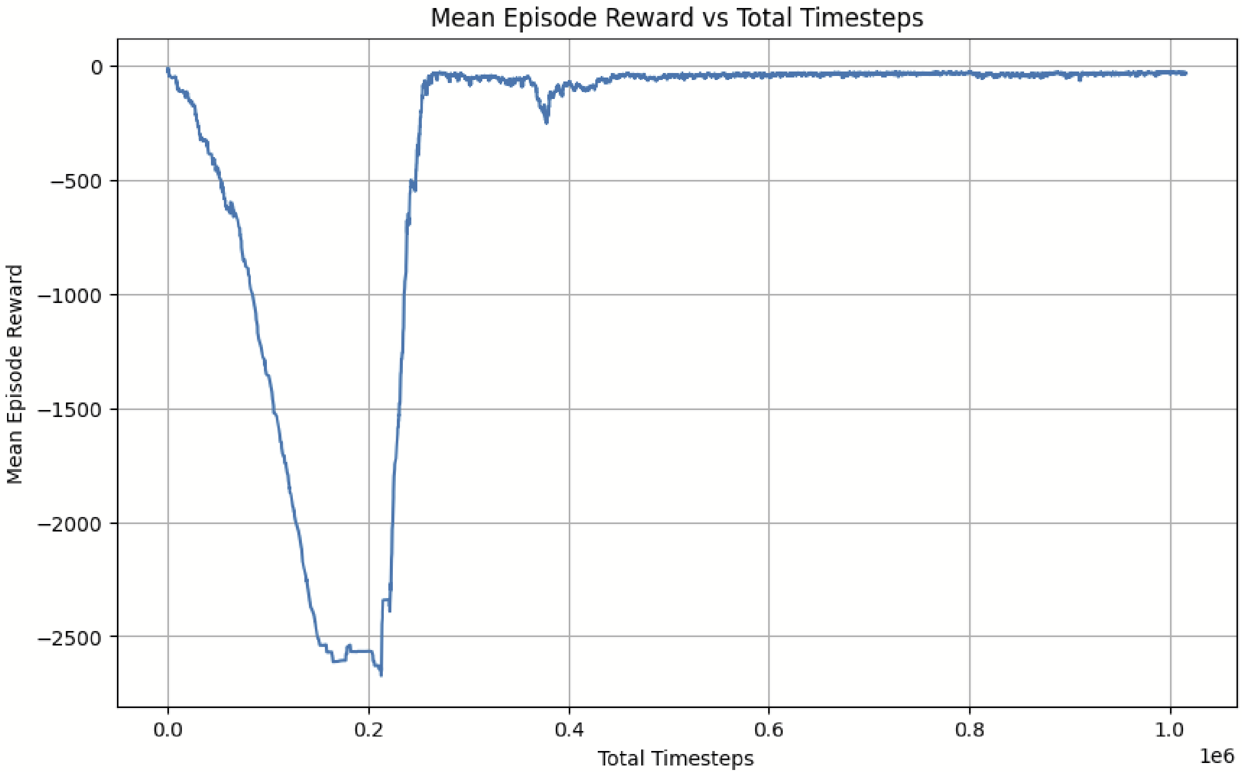
□ Ant-v4

Proximal Policy Optimization (PPO)

ENTROPY = 0.0

LEARNING RATE = 9e-3

FINAL EPISODE MEAN REWARD = -34.1



3. Results

□ Ant-v4

TD3/Twin-Delayed DDPG for Ant-v4

learning_starts=10000

batch_size=100

learning_rate=1e-3

gamma=0.99

tau=0.005

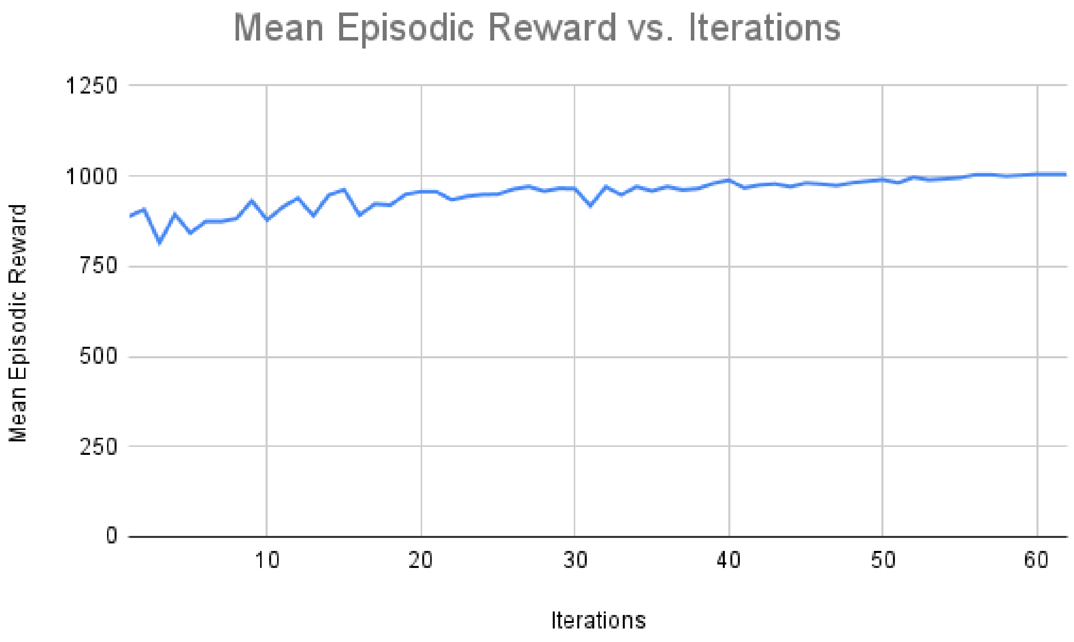
gradient_steps=-1

policy_delay=2

target_policy_noise=0.2

target_noise_clip=0.5

mean reward = 1.004e+03



3. Results

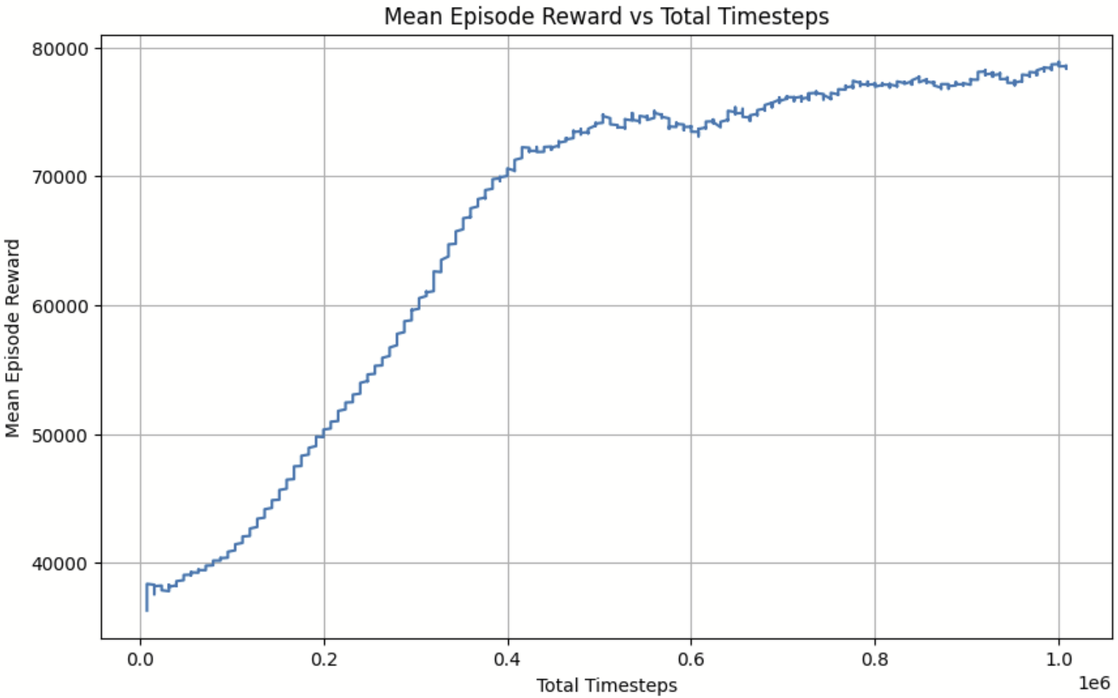
- ☐ Humanoid Standup-v4

Proximal Policy Optimization (PPO)

ENTROPY = 0.0

LEARNING RATE = 3e-4

FINAL EPISODE MEAN REWARD = 7.84e+04



3. Results

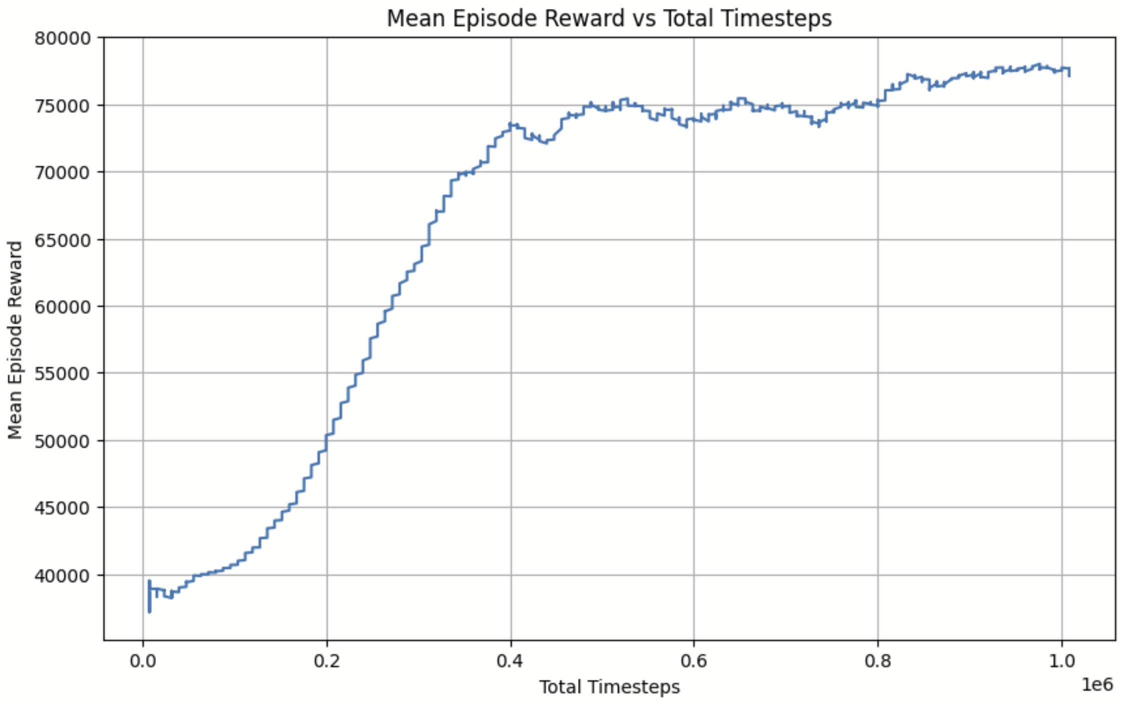
- Humanoid Standup-v4

Proximal Policy Optimization (PPO)

ENTROPY = 0.001

LEARNING RATE = 3e-4

FINAL EPISODE MEAN REWARD = 7.7e+4



3. Results

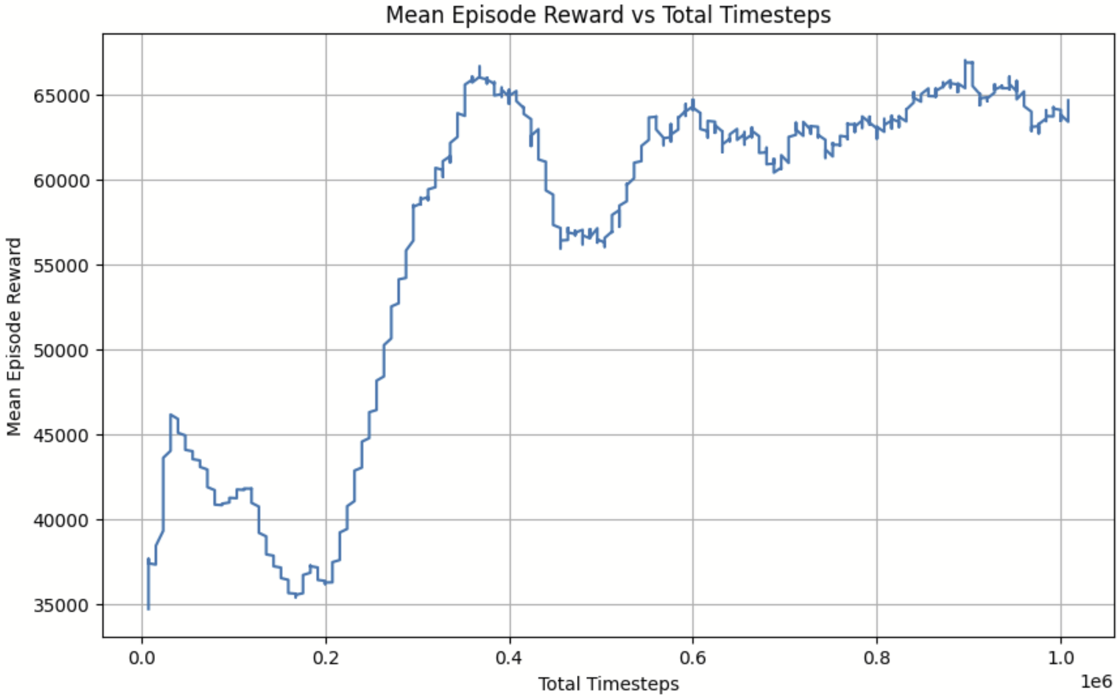
- ☐ Humanoid Standup-v4

Proximal Policy Optimization (PPO)

ENTROPY = 0.0

LEARNING RATE = 9e-3

FINAL EPISODE MEAN REWARD = 6.48e+04



Conclusion

- Results are very sensitive to hyperparameters choices
- Well-balanced entropy coeff => better results
- PPO approx. 4x faster than TD3
- TD3 better than PPO for Ant-V4

Thanks!

- Q & A