Proximal Policy Optimization (PPO) Tutorial

September 21, 2025

1 Introduction

Proximal Policy Optimization (PPO) constrains policy updates by clipping the probability ratio between new and old policies, providing a simple and stable on-policy algorithm. PPO balances exploration and stability without the complexity of trust-region constraints.

2 Theory and Formulas

2.1 Clipped Objective

Given trajectories collected under $\pi_{\theta_{old}}$, PPO maximizes

$$L^{CLIP}(\theta) = \mathbb{E}\big[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\big], \tag{1}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ and $\hat{A}_t$ is an advantage estimate.

2.2 Value and Entropy Losses

The full PPO loss combines policy, value, and entropy terms:

$$L(\theta) = \mathbb{E}\big[L^{CLIP}(\theta) - c_v(V_\theta(s_t) - hatV_t)^2 + c_{\text{ent}}H[\pi_\theta(\cdot \mid s_t)]\big]. \tag{2}$$

Rollouts are typically split into mini-batches, and several epochs of stochastic gradient ascent are performed per batch.

2.3 Advantage Estimation

Generalized Advantage Estimation (GAE) reduces variance:

$$\hat{A}_t = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}, \quad \delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t). \tag{3}$$

In tabular examples shorter horizons suffice, but GAE remains effective with neural networks.

3 Applications and Tips

- **Continuous control**: widely used in robotics and locomotion benchmarks (MuJoCo, Isaac Gym).
- **Large-scale training**: robust under parallel rollout collection and mini-batch updates.
- **Games and simulation**: stable alternative to TRPO with simpler implementation.
- **Best practices**: tune clipping range ϵ, normalize advantages, anneal learning rate, monitor clip fraction and KL divergence, and use value clipping to prevent critic drift.

4 Python Practice

The script `gen_ppo_figures.py` trains a tabular PPO agent on a stochastic grid-world. It logs episode returns and the clip fraction (percentage of samples hitting the clipping boundary) to diagnose policy updates.

Listing 1: Excerpt from gen$_{p}po_{f}igures.py$

```
ratio = np.exp(log_prob_new - log_prob_old)
clipped_ratio = np.clip(ratio, 1 - eps_clip, 1 + eps_clip)
policy_loss = -np.mean(np.minimum(ratio * advantages, clipped_ratio * advantages))
clip_fraction = np.mean((np.abs(ratio - 1.0) > eps_clip/2).astype(float))
```

5 Result

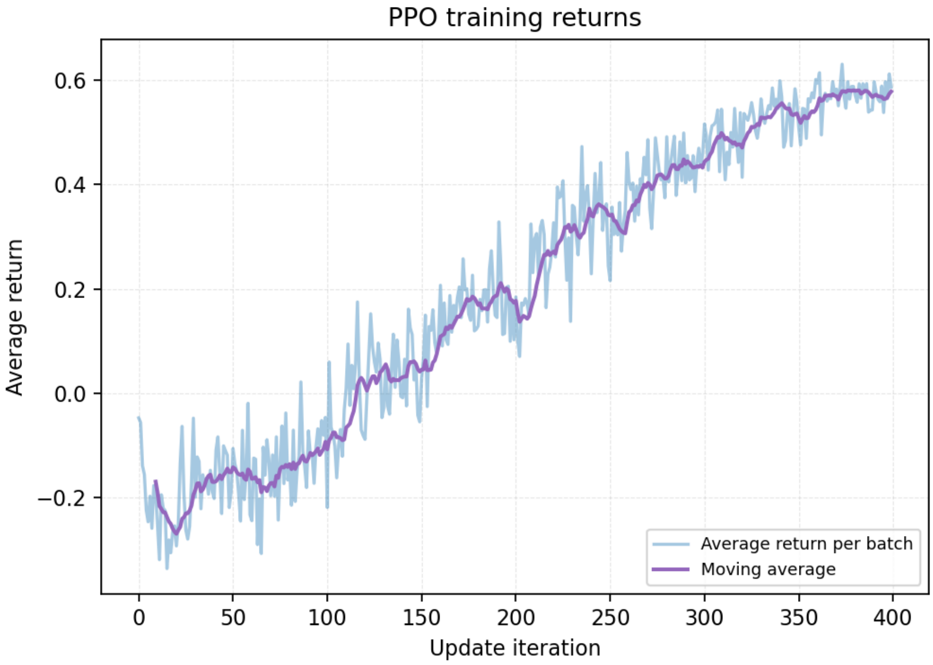

Figure 1: PPO episode returns over training with moving average smoothing

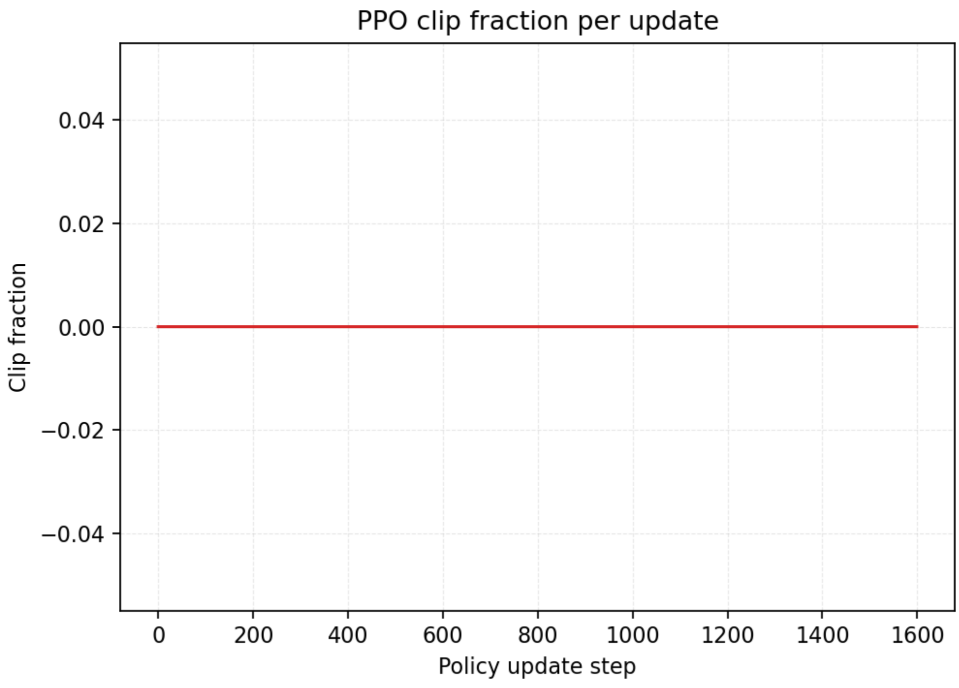

Figure 2: Clip fraction per update, illustrating how often the policy ratio was truncated

6 Summary

PPO performs clipped policy updates to achieve stable on-policy learning with minimal tuning. Advantage normalization, entropy bonuses, and monitoring of clip/ KL statistics help maintain reliable convergence. The grid-world example shows returns improving steadily while clip fractions remain well behaved.