Algorithm Proximal Policy Optimization with Clipped Surrogate Loss

1: **Initialize** policy parameters θ and value function parameters ϕ

For each iteration

2: **Collect trajectories** by running policy π_θ:

a. Record states s_t, actions a_t, rewards r_t, done flags d_t, old log probabilities $\log \pi_{\theta_{\text{old}}}(a_t|s_t)$, and values $V(s_t;\phi)$

3: **Compute advantages** using GAE:

a. Compute temporal-difference residuals:

$$\delta_t = r_t + \gamma V(s_{t+1};\phi) \cdot (1 - d_t) - V(s_t;\phi)$$

b. Compute advantages recursively:

$$A_t = \delta_t + \gamma\lambda(1 - d_t)A_{t+1}$$

4: **Compute returns**:

$$R_t = A_t + V(s_t;\phi)$$

5: **Update policy and value function**:

a. For several epochs, shuffle data and divide into minibatches

b. For each minibatch:

i. Compute new log probabilities $\log \pi_\theta(a_t|s_t)$, entropies $H[\pi_\theta](s_t)$, and values $V(s_t;\phi)$

ii. Calculate probability ratio:

$$r_t(\theta) = \exp\left(\log \pi_\theta(a_t|s_t) - \log \pi_{\theta_{\text{old}}}(a_t|s_t)\right)$$

iii. Compute surrogate loss with clipping:

$$L^{\text{CLIP}} = \text{mean}\left[\min\left(r_t(\theta)A_t,\ \text{clip}\left(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon\right)A_t\right)\right]$$

iv. Compute clipped value estimate:

$$v_{\text{clipped}} = V_{\text{old}}(s_t;\phi) + \text{clip}\left(V(s_t;\phi) - V_{\text{old}}(s_t;\phi),\ -\epsilon,\ \epsilon\right)$$

v. Compute value loss:

$$L^{\text{VF}} = \text{mean}\left[\max\left((V(s_t;\phi) - R_t)^2,\ (v_{\text{clipped}} - R_t)^2\right)\right]$$

vi. Compute entropy bonus:

$$L^{\text{S}} = \text{mean}\left[H[\pi_\theta](s_t)\right]$$

vii. Compute total loss:

$$L = -L^{\text{CLIP}} + c_1 L^{\text{VF}} - c_2 L^{\text{S}}$$

viii. Update parameters θ and ϕ using gradients of L

End For