In [1]:
```
import torch
import numpy as np
import gym
from stable_baselines3 import PPO
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# Create the Pendulum environment
env = gym.make('Pendulum-v1')
```

In [3]:
```
# Define the PPO agent
model = PPO("MlpPolicy", env, verbose=1)
```

```
Using cpu device
Wrapping the env with a `Monitor` wrapper
Wrapping the env in a DummyVecEnv.
```

In [4]:
```
# Train the agent
model.learn(total_timesteps=10000)
```

```
---------------------------------
| rollout/           |          |
|    ep_len_mean     | 200      |
|    ep_rew_mean     | -1.21e+03 |
| time/              |          |
|    fps             | 980      |
|    iterations      | 1        |
|    time_elapsed    | 2        |
|    total_timesteps | 2048     |
---------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 200         |
|    ep_rew_mean          | -1.14e+03   |
| time/                   |             |
|    fps                  | 631         |
|    iterations           | 2           |
|    time_elapsed         | 6           |
|    total_timesteps      | 4096        |
| train/                  |             |
|    approx_kl            | 0.0038027572 |
|    clip_fraction        | 0.0202      |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.41       |
|    explained_variance   | 0.00261     |
|    learning_rate        | 0.0003      |
|    loss                 | 4.16e+03    |
|    n_updates            | 10          |
|    policy_gradient_loss | -0.0043     |
|    std                  | 0.984       |
|    value_loss           | 9.33e+03    |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 200         |
|    ep_rew_mean          | -1.11e+03   |
| time/                   |             |
|    fps                  | 559         |
|    iterations           | 3           |
|    time_elapsed         | 10          |
|    total_timesteps      | 6144        |
| train/                  |             |
|    approx_kl            | 0.0017949551 |
|    clip_fraction        | 0.00781     |
|    clip_range           | 0.2         |
|    entropy_loss         | -1.4        |
|    explained_variance   | 0.153       |
|    learning_rate        | 0.0003      |
|    loss                 | 3.41e+03    |
|    n_updates            | 20          |
|    policy_gradient_loss | -0.00119    |
|    std                  | 0.984       |
|    value_loss           | 6.72e+03    |
-----------------------------------------
-----------------------------------------
| rollout/                |             |
|    ep_len_mean          | 200         |
|    ep_rew_mean          | -1.09e+03   |
| time/                   |             |
|    fps                  | 546         |
|    iterations           | 4           |
```

```
|    time_elapsed         | 15           |
|    total_timesteps      | 8192         |
| train/                  |              |
|    approx_kl            | 0.0027371948 |
|    clip_fraction        | 0.0144       |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.4         |
|    explained_variance   | 0.0658       |
|    learning_rate        | 0.0003       |
|    loss                 | 3.11e+03     |
|    n_updates            | 30           |
|    policy_gradient_loss | -0.00185     |
|    std                  | 0.972        |
|    value_loss           | 6.16e+03     |
-----------------------------------------
-----------------------------------------
| rollout/                |              |
|    ep_len_mean          | 200          |
|    ep_rew_mean          | -1.13e+03    |
| time/                   |              |
|    fps                  | 532          |
|    iterations           | 5            |
|    time_elapsed         | 19           |
|    total_timesteps      | 10240        |
| train/                  |              |
|    approx_kl            | 0.0006635548 |
|    clip_fraction        | 0.00205      |
|    clip_range           | 0.2          |
|    entropy_loss         | -1.4         |
|    explained_variance   | 0.0119       |
|    learning_rate        | 0.0003       |
|    loss                 | 2.52e+03     |
|    n_updates            | 40           |
|    policy_gradient_loss | -0.000545    |
|    std                  | 0.984        |
|    value_loss           | 5.89e+03     |
-----------------------------------------
```

Out[4]: <stable_baselines3.ppo.ppo.PPO at 0x2203f202b30>

In [5]:
```python
# Helper function to evaluate the agent
def evaluate(model, env, n_eval_episodes=10):
    rewards = []
    for _ in range(n_eval_episodes):
        obs, _ = env.reset()  # unpack tuple from Gym >=0.26
        done = False
        episode_reward = 0
        while not done:
            action, _ = model.predict(obs, deterministic=True)
            obs, reward, terminated, truncated, _ = env.step(action)
            done = terminated or truncated
            episode_reward += reward
        rewards.append(episode_reward)
    mean_reward = np.mean(rewards)
    std_reward = np.std(rewards)
    return mean_reward, std_reward
```

In [6]:
```
# Evaluate the agent
mean_reward, std_reward = evaluate(model, env, n_eval_episodes=10)
print(f"Mean reward: {mean_reward:.2f} +/- {std_reward:.2f}")
```

```
Mean reward: -1227.50 +/- 352.13
```

In [ ]:
```

```