

**António Cruz** (140129)**, Cátia Brás** (120093)**, Ricardo Kayseller** (95813)

**Advanced Topics in Deep Learning**

MINI-PROJECT 02

DEEP REINFORCEMENT LEARNING

**DQN and PPO in LUnarLander v3**

**TAAP**

Table of Contents

[1. Introduction 4](#_Toc222739638)

[2. Environment Analysis 4](#_Toc222739639)

[2.1 State Space 4](#_Toc222739640)

[2.2 Action Space 4](#_Toc222739641)

[2.3 Reward Function 5](#_Toc222739642)

[2.4 Termination Rules 5](#_Toc222739643)

[3. Experimental Setup 6](#_Toc222739644)

[3.1 Total Environment Steps 6](#_Toc222739645)

[3.2 Hyperparameter Configurations 6](#_Toc222739646)

[3.3 Random Seeds 7](#_Toc222739647)

[3.4 Evaluation Protocol 7](#_Toc222739648)

[3.5 Best-Model Selection 7](#_Toc222739649)

[3.6 Reproducibility 7](#_Toc222739650)

[4. Evaluation 20](#_Toc222739651)

[4.1 Per-Algorithm Results 21](#_Toc222739652)

[4.2 Cross-Algorithm Comparison 24](#_Toc222739653)

[4.3 Statistical Significance 26](#_Toc222739654)

[4.4 Baseline Comparison 26](#_Toc222739655)

[4.5 Agent Behavior Analysis 27](#_Toc222739656)

[4.6 Sample Efficiency Analysis 28](#_Toc222739657)

[4.7 Training Milestone Timeline 29](#_Toc222739658)

[4.8 Final vs Best Model Comparison 30](#_Toc222739659)

[4.9 Hyperparameter Exploration Summary 31](#_Toc222739660)

[4.10 Visualizations 32](#_Toc222739661)

[5. Experimental Setup 34](#_Toc222739662)

[6. Conceptual Discussion 35](#_Toc222739663)

[6.1 Value-Based vs Policy-Based Learning 35](#_Toc222739664)

[Value-Based Learning 35](#_Toc222739665)

[Policy-Based Learning 35](#_Toc222739666)

[6.2 Overestimation and Instability in Value-Based Algorithms 36](#_Toc222739667)

[Overestimation Bias 36](#_Toc222739668)

[Structural Instability: The Deadly Triad 36](#_Toc222739669)

[6.3 PPO Clipping and Generalised Advantage Estimation (GAE) 36](#_Toc222739670)

[PPO Clipped Objective 36](#_Toc222739671)

[Generalised Advantage Estimation (GAE) 37](#_Toc222739672)

[6.4 Epsilon-Greedy vs Entropy-Driven Exploration 37](#_Toc222739673)

[Epsilon-Greedy Exploration (DQN) 37](#_Toc222739674)

[Entropy-Regularised Exploration (PPO) 38](#_Toc222739675)

[6.5 Conceptual Summary 38](#_Toc222739676)

## 1. Introduction

Deep Reinforcement Learning (DRL) has become a central paradigm for solving complex sequential decision-making problems, particularly those involving continuous state spaces, stochastic dynamics, and delayed rewards. Among the wide range of DRL algorithms, Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) represent two fundamentally different approaches: value-based learning and policy-based learning. Comparing these two algorithms provides valuable insight into how different learning mechanisms behave under the same environment conditions.

In this project, we investigate the performance of DQN and PPO in the LunarLander-v3 environment from Gymnasium. This environment simulates the control of a lunar lander that must descend and touch down smoothly within a designated landing zone. The agent receives an 8-dimensional continuous state vector and must choose among four discrete actions, making the task suitable for both value-based and policy-based methods. The environment includes dense rewards, penalties for fuel consumption, and strong negative rewards for crashes, creating a challenging control problem that requires stability, precision, and efficient exploration.

The goal of this work is to train, evaluate, and compare DQN and PPO under controlled experimental conditions. We analyse their learning curves, sample efficiency, stability across multiple random seeds, and qualitative behaviour through recorded simulations. Additionally, we examine theoretical aspects such as overestimation bias in DQN, the role of clipping and Generalized Advantage Estimation (GAE) in PPO, and the differences between epsilon-greedy and entropy-driven exploration.

By combining quantitative metrics with qualitative observations, this study aims to provide a comprehensive understanding of how DQN and PPO behave in the LunarLander-v3 environment, highlighting their strengths, limitations, and the impact of hyperparameter choices on the final performance.

## 2. Environment Analysis

### 2.1 State Space

The LunarLander-v3 environment provides an 8-dimensional continuous state vector describing the physical configuration of the lander at each timestep: 1. Horizontal position (x) relative to the landing zone 2. Vertical position (y) above the ground 3. Horizontal velocity (vx) 4. Vertical velocity (vy) 5. Lander angle () 6. Angular velocity () 7. Left leg contact indicator (0 or 1) 8. Right leg contact indicator (0 or 1)

These variables allow the agent to infer its location, orientation, stability, and motion, which are essential for controlling the descent.

### 2.2 Action Space

The action space is discrete with four possible actions:

* 0 - Do nothing
* 1 - Fire left thruster (pushes the lander to the right)
* 2 - Fire main thruster (reduces vertical speed)
* 3 - Fire right thruster (pushes the lander to the left)

These actions allow the agent to control both horizontal movement and vertical descent.

### 2.3 Reward Function

The reward function is dense and designed to encourage smooth and stable landings while penalizing unsafe or inefficient behaviour.

**Positive rewards include:** - Moving closer to the landing zone - Reducing horizontal and vertical velocity - Maintaining a stable angle - Each leg touching the ground (+10 each) - Successful landing (+100 to +140)

**Penalties include:** - High velocities - Large tilt angles - Excessive use of the main thruster (fuel cost) - Crashes (around -100)

A score of **200 or higher** typically indicates a successful landing.

### 2.4 Termination Rules

An episode terminates when one of the following occurs: 1. **Successful landing** within the designated zone 2. **Crash** due to excessive speed or unstable angle 3. **Leaving the screen boundaries** 4. **Reaching the maximum number of steps** allowed by the environment (1000 timesteps)

These termination conditions apply equally to both PPO and DQN agents.

Session prefix: dqn\_ppo  
Algorithms: ['dqn', 'ppo']  
Seeds: [42, 123, 3407]  
Wind enabled: False  
Total timesteps per seed: 1,500,000  
Evaluation episodes per seed: 20  
Chart update frequency: every 10 episodes  
Checkpoint frequency: every 100 episodes  
Eval callback frequency: every 25,000 timesteps (20 episodes)  
Solved threshold: 200  
Device: cpu

# 3. Experimental Setup

This section describes the training and evaluation methodology used to compare DQN and PPO in the LunarLander-v3 environment. All experiments were conducted under controlled and reproducible conditions to ensure a fair comparison between the two algorithms.

### 3.1 Total Environment Steps

Both DQN and PPO were trained using a total of **1,500,000 environment steps per seed**. Since the project uses 3 seeds, this results in:

* 3 seeds x 1,500,000 steps = **4,500,000 training steps per algorithm**

Using the same number of interactions ensures that the comparison between DQN and PPO is based on equal sample budgets.

### 3.2 Hyperparameter Configurations

The hyperparameters for each algorithm were selected based on Stable-Baselines3 recommendations and refined through empirical testing. Tables 1 and 2 summarize the final configurations used for training.

| **Hyperparameter** | **Value** |
| --- | --- |
| Policy | MlpPolicy |
| Learning rate | linear\_schedule(6.3e-4) |
| Buffer size | 750,000 |
| Batch size | 128 |
| Gamma (discount factor) | 0.99 |
| Learning starts | 50,000 |
| Exploration strategy | -greedy |
| Exploration fraction | 0.12 |
| Final | 0.1 |
| Target update interval | 250 |
| Train frequency | 4 |
| Gradient steps | 4 |
| Network architecture | 256 x 256 MLP |

**Table 1** - DQN Hyperparameters

| **Hyperparameter** | **Value** |
| --- | --- |
| Policy | MlpPolicy |
| Learning rate | 0.00025 |
| n\_steps | 2048 |
| Batch size | 64 |
| n\_epochs | 10 |
| Gamma (discount factor) | 0.999 |
| GAE | 0.95 |
| Clip range | 0.2 |
| Entropy coefficient | 0.01 |
| Network architecture | 256 x 256 MLP |

**Table 2** - PPO Hyperparameters

These hyperparameters were kept constant across all seeds to isolate the effect of stochasticity in training.

### 3.3 Random Seeds

To evaluate robustness and training stability, both algorithms were trained using **3 different random seeds**: - Seed 1: **42** - Seed 2: **123** - Seed 3: **3407**

Each seed corresponds to a full independent training run of 1,500,000 steps.

### 3.4 Evaluation Protocol

Each trained model was evaluated over **20 deterministic episodes per seed**, for a total of 60 evaluation episodes per algorithm. Deterministic evaluation ensures that the policy is tested without exploration noise, providing a fair assessment of the learned behaviour.

### 3.5 Best-Model Selection

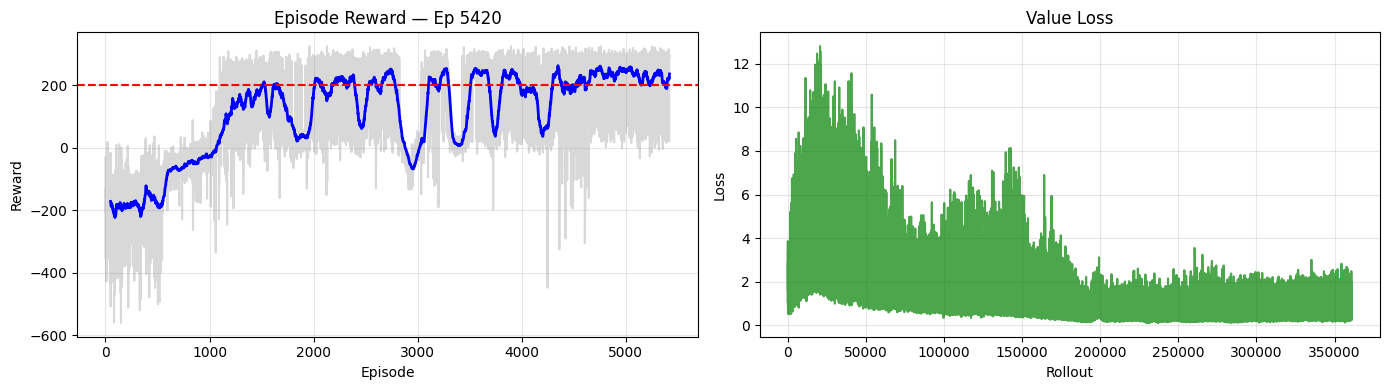
During training, an evaluation callback assessed the model every 25,000 environment steps using 20 deterministic episodes. The best model was selected based on a combined metric: **mean reward - standard deviation**. This metric favours models that are both high-performing and consistent, rather than models that achieve high mean reward with large variance.

A two-tier selection gate ensures that once any model crosses a mean reward of 200, only solved models can replace the current best.

### 3.6 Reproducibility

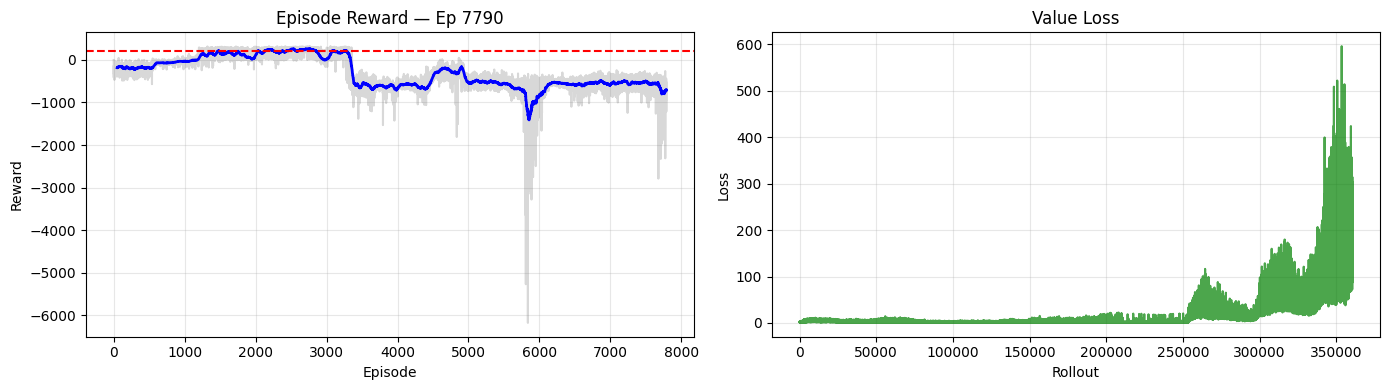
All experiments used fixed random seeds for Python, NumPy, PyTorch, and the Gymnasium environment. Deterministic algorithms were enabled in PyTorch to minimize non-determinism from GPU operations.

Python: 3.12.3  
PyTorch: 2.10.0+cu130  
Device: cpu  
CUDA: 13.0  
  
  
Observation space: Box([ -2.5 -2.5 -10. -10. -6.2831855 -10.  
 -0. -0. ], [ 2.5 2.5 10. 10. 6.2831855 10.  
 1. 1. ], (8,), float32)  
Action space: Discrete(4)  
Initial observation: [-0.00170641 1.4186276 -0.17285168 0.3425566 0.00198406 0.03915349  
 0. 0. ]  
  
  
  
============================================================  
DQN | Seed 42  
============================================================  
  
  
  
  
Output()  
  
  
  
'Episode 5420 | Last 50 Ep — Mean: 236.2 | Std: 95.5 | Min: 15.8 | Max: 315.9 | Success: 82%'



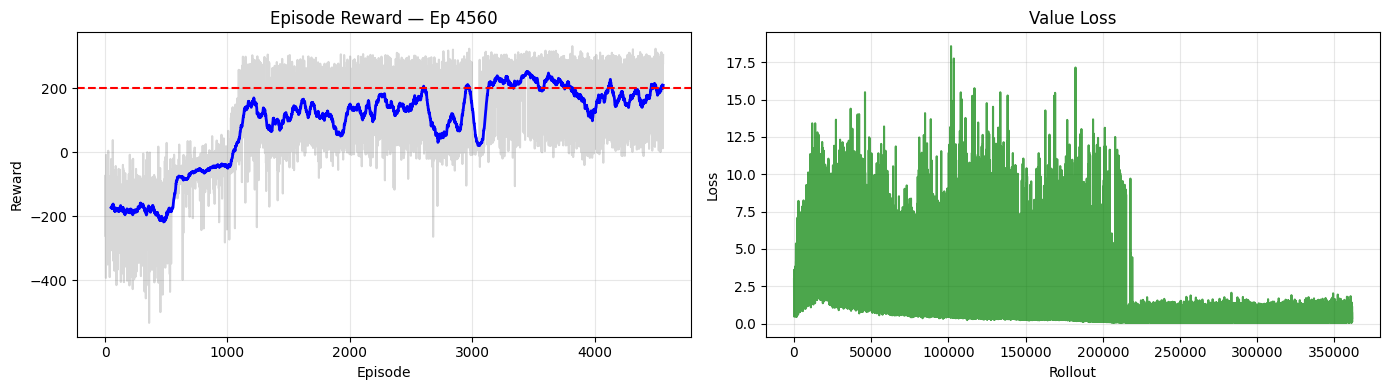
'[Checkpoint] Episode 5400 saved'  
  
  
  
'Eval @ 1500000 steps | Reward: 276.50 +/- 16.66 [SOLVED] | Success: 100% | Score (mean-std): 259.85 | Best: 266.35'

Output()  
  
  
  
'Episode 7790 | Last 50 Ep — Mean: -708.5 | Std: 349.2 | Min: -2311.8 | Max: -262.3 | Success: 0%'  
  
  
  
Training time: 47.6 min (2856 s)  
Final model: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/dqn\_ppo\_dqn\_42  
Best model: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/best\_model  
Checkpoints: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/checkpoints  
Best model stats: Reward: 283.74 +/- 17.39 | Success: 100% | Score (mean-std): 266.35 | @ 1,425,000 steps  
  
============================================================  
DQN | Seed 123  
============================================================



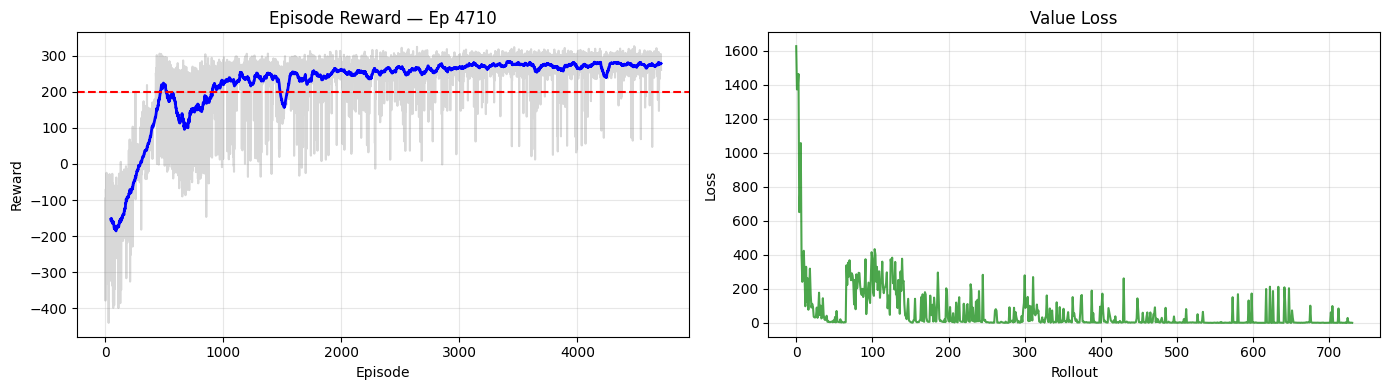
'[Checkpoint] Episode 7700 saved'  
  
  
  
'Eval @ 1500000 steps | Reward: -926.29 +/- 368.67 | Success: 0% | Score (mean-std): -1294.96 | Best: 259.82'

Output()  
  
  
  
'Episode 4560 | Last 50 Ep — Mean: 209.7 | Std: 112.3 | Min: 0.6 | Max: 313.2 | Success: 70%'  
  
  
  
Training time: 47.0 min (2818 s)  
Final model: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/dqn\_ppo\_dqn\_123  
Best model: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/best\_model  
Checkpoints: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/checkpoints  
Best model stats: Reward: 275.61 +/- 15.80 | Success: 100% | Score (mean-std): 259.82 | @ 875,000 steps  
  
============================================================  
DQN | Seed 3407  
============================================================



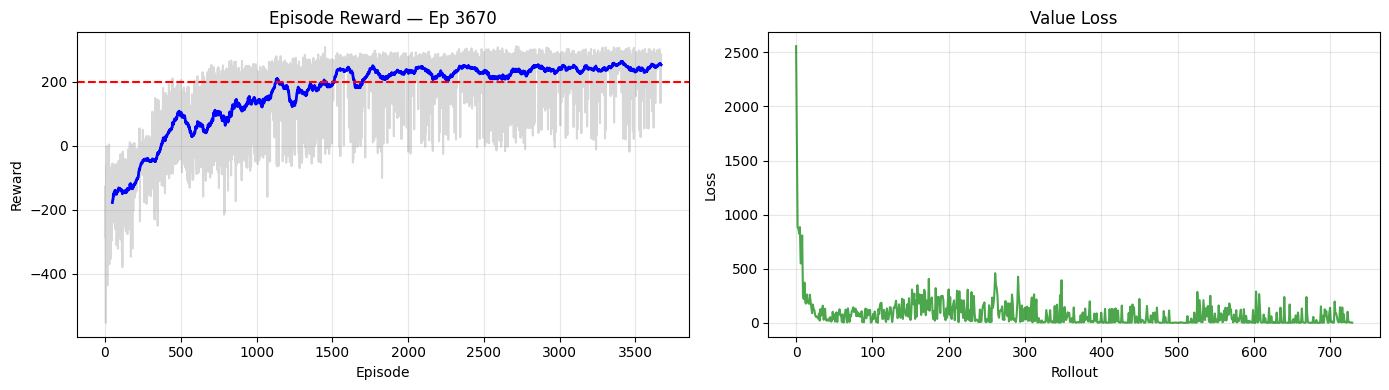
'[Checkpoint] Episode 4500 saved'  
  
  
  
'Eval @ 1500000 steps | Reward: 250.47 +/- 82.52 [SOLVED] | Success: 85% | Score (mean-std): 167.95 | Best: 243.87'

Output()  
  
  
  
Training time: 47.3 min (2839 s)  
Final model: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/dqn\_ppo\_dqn\_3407  
Best model: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/best\_model  
Checkpoints: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/checkpoints  
Best model stats: Reward: 261.53 +/- 17.66 | Success: 100% | Score (mean-std): 243.87 | @ 1,300,000 steps  
  
DQN: All 3 seeds trained.  
  
============================================================  
PPO | Seed 42  
============================================================  
  
  
  
  
'Episode 4710 | Last 50 Ep — Mean: 278.3 | Std: 31.3 | Min: 146.7 | Max: 320.5 | Success: 96%'



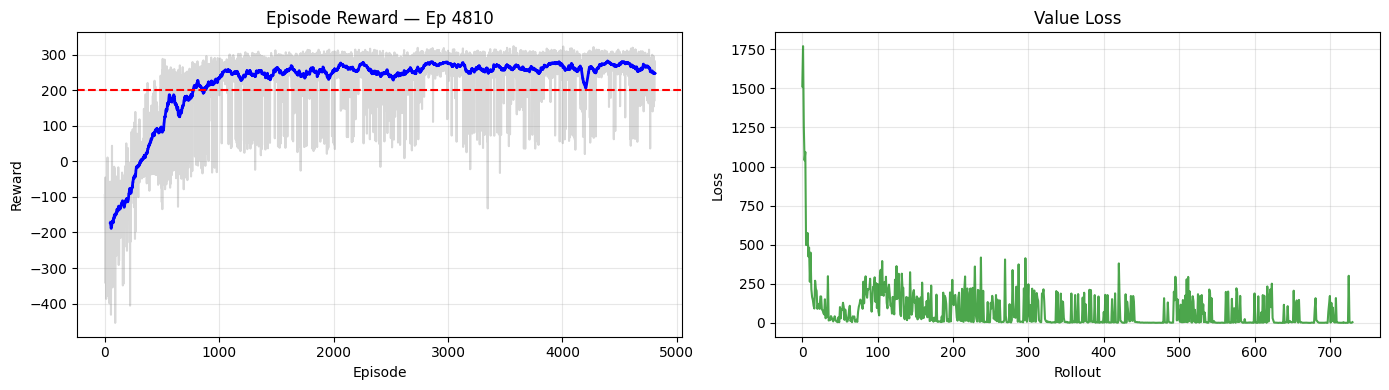
'[Checkpoint] Episode 4700 saved'  
  
  
  
'Eval @ 1500000 steps | Reward: 289.50 +/- 21.48 [SOLVED] | Success: 100% | Score (mean-std): 268.02 | Best: 266.36 >> New best model!'

Output()  
  
  
  
Training time: 20.2 min (1211 s)  
Final model: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/dqn\_ppo\_ppo\_42  
Best model: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/best\_model  
Checkpoints: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/checkpoints  
Best model stats: Reward: 289.50 +/- 21.48 | Success: 100% | Score (mean-std): 268.02 | @ 1,500,000 steps  
  
============================================================  
PPO | Seed 123  
============================================================  
  
  
  
  
'Episode 3670 | Last 50 Ep — Mean: 252.4 | Std: 46.9 | Min: 56.7 | Max: 301.1 | Success: 90%'



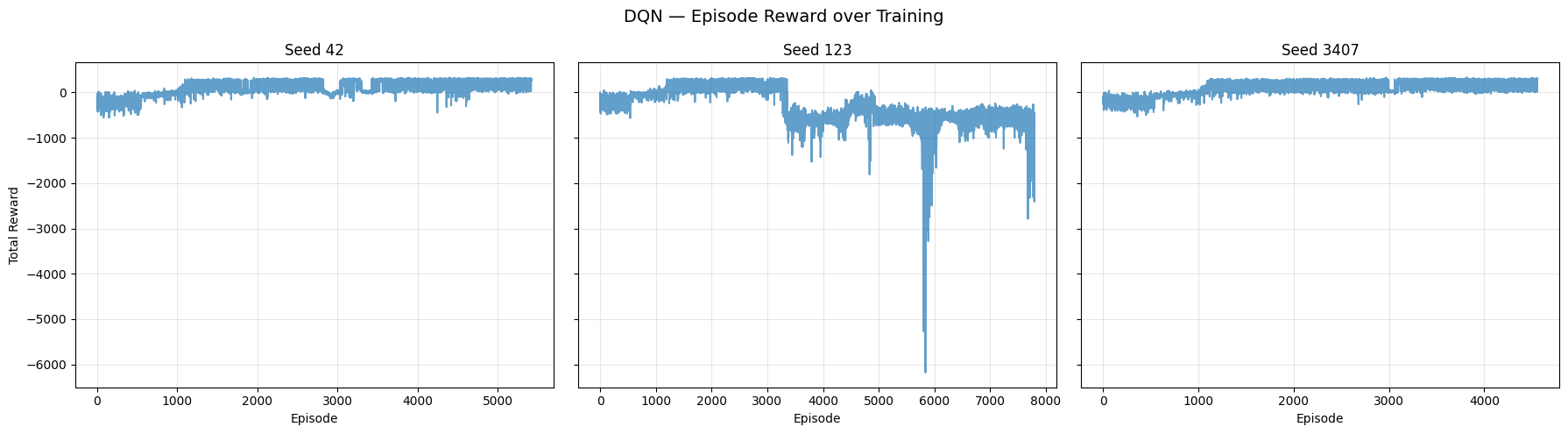
'[Checkpoint] Episode 3600 saved'  
  
  
  
'Eval @ 1500000 steps | Reward: 257.63 +/- 35.30 [SOLVED] | Success: 95% | Score (mean-std): 222.33 | Best: 249.47'

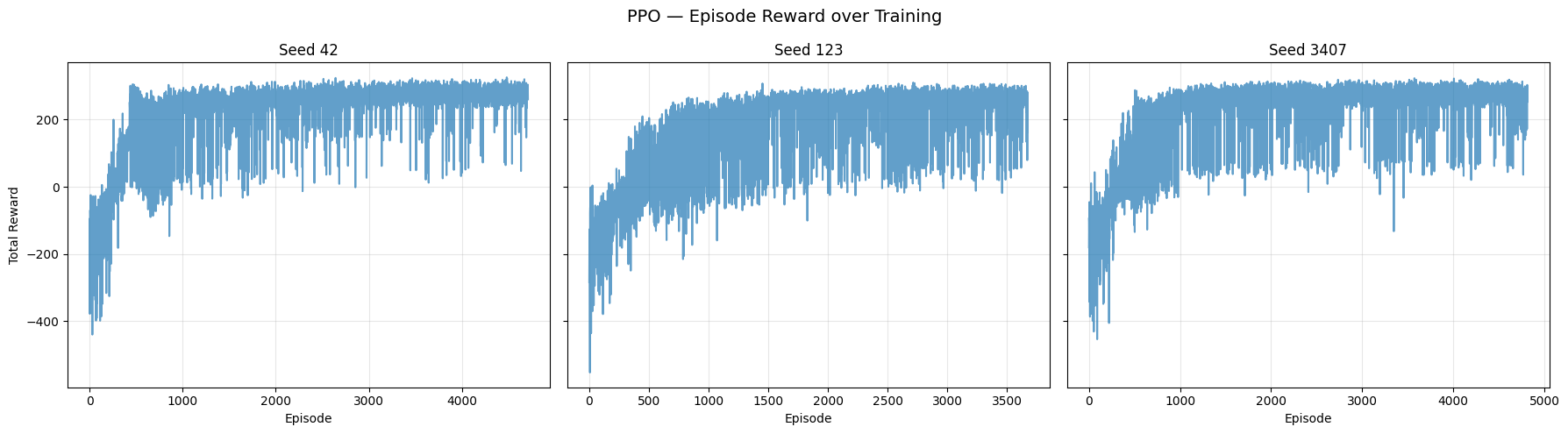
Output()  
  
  
  
Training time: 23.0 min (1380 s)  
Final model: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/dqn\_ppo\_ppo\_123  
Best model: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/best\_model  
Checkpoints: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/checkpoints  
Best model stats: Reward: 267.61 +/- 18.14 | Success: 100% | Score (mean-std): 249.47 | @ 1,300,000 steps  
  
============================================================  
PPO | Seed 3407  
============================================================  
  
  
  
  
'Episode 4810 | Last 50 Ep — Mean: 247.0 | Std: 54.3 | Min: 36.0 | Max: 299.4 | Success: 86%'

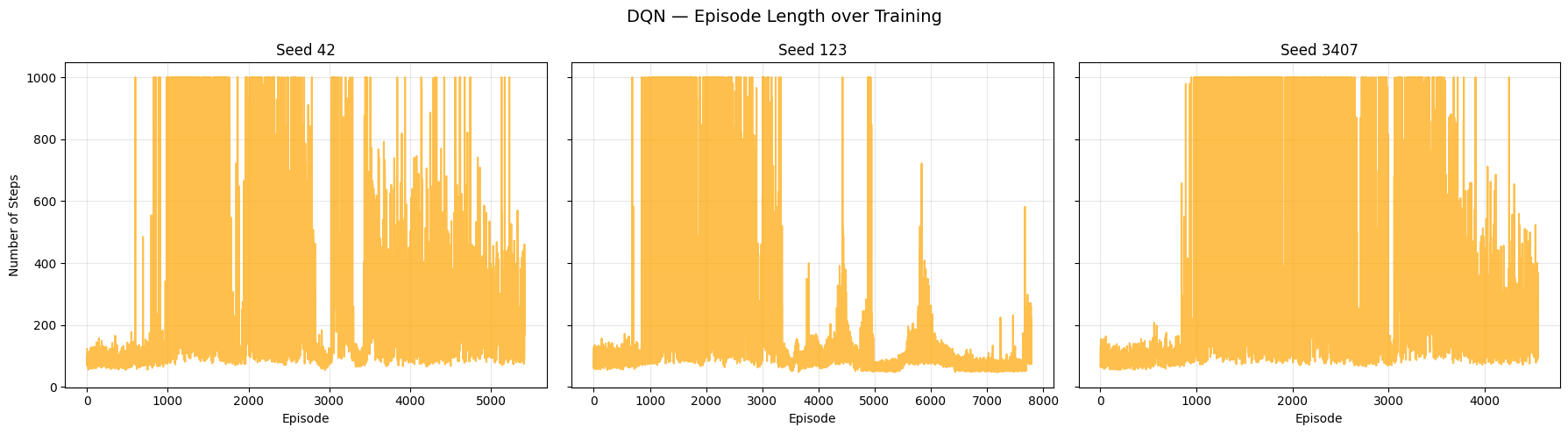


'[Checkpoint] Episode 4800 saved'  
  
  
  
'Eval @ 1500000 steps | Reward: 265.17 +/- 52.61 [SOLVED] | Success: 95% | Score (mean-std): 212.56 | Best: 263.95'

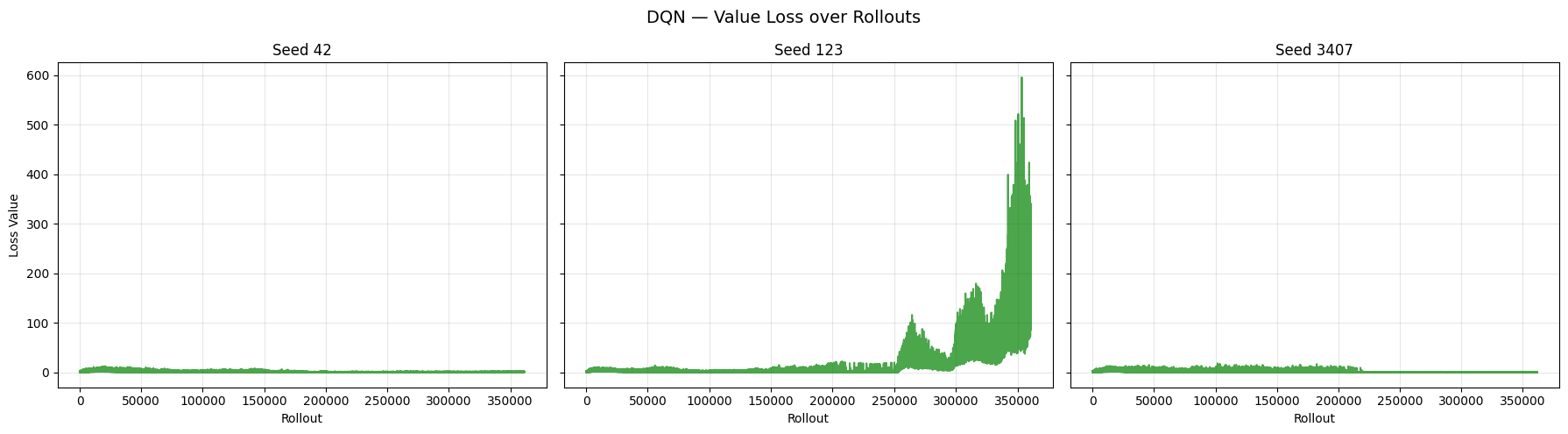
Training time: 20.8 min (1248 s)  
Final model: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/dqn\_ppo\_ppo\_3407  
Best model: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/best\_model  
Checkpoints: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/checkpoints  
Best model stats: Reward: 280.95 +/- 17.01 | Success: 100% | Score (mean-std): 263.95 | @ 1,375,000 steps  
  
PPO: All 3 seeds trained.  
  
============================================================  
BEST MODEL SUMMARY (all algorithms x seeds)  
============================================================  
  
Algorithm Seed Mean Reward Std Reward Success Score (mean-std) @ Timestep  
 DQN 42 283.74 17.39 100% 266.35 1,425,000  
 DQN 123 275.61 15.80 100% 259.82 875,000  
 DQN 3407 261.53 17.66 100% 243.87 1,300,000  
 PPO 42 289.50 21.48 100% 268.02 1,500,000  
 PPO 123 267.61 18.14 100% 249.47 1,300,000  
 PPO 3407 280.95 17.01 100% 263.95 1,375,000  
  
All training complete.  
  
  
Saved DQN seed 42: 5424 episodes -> /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/training\_log.npz  
Saved DQN seed 123: 7796 episodes -> /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/training\_log.npz  
Saved DQN seed 3407: 4560 episodes -> /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/training\_log.npz  
Saved PPO seed 42: 4710 episodes -> /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/training\_log.npz  
Saved PPO seed 123: 3675 episodes -> /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/training\_log.npz  
Saved PPO seed 3407: 4813 episodes -> /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/training\_log.npz  
  
  
\*\*\* TRAINING TIME SUMMARY \*\*\*  
Timesteps per seed: 1,500,000 | Device: cpu  
  
Algorithm Seed Time (s) Time (min)  
 DQN 42 2856 47.6  
 DQN 123 2818 47.0  
 DQN 3407 2839 47.3  
 PPO 42 1211 20.2  
 PPO 123 1380 23.0  
 PPO 3407 1248 20.8  
 DQN Mean 2838 47.3  
 PPO Mean 1280 21.3  
  
  
\*\*\* GRADIENT UPDATES SUMMARY \*\*\*  
Algorithm Actual (from training) Analytical (computed) Total Env Steps Ratio (updates/steps)  
 DQN 1,449,996 1,450,000 1,500,000 0.967  
 PPO 7,320 234,240 1,500,000 0.005  
  
DQN: train\_freq=4, gradient\_steps=4 -> 4 gradient updates per training call, called every 4 env steps  
PPO: n\_steps=2048, n\_epochs=10, batch\_size=64 -> 320 updates per rollout

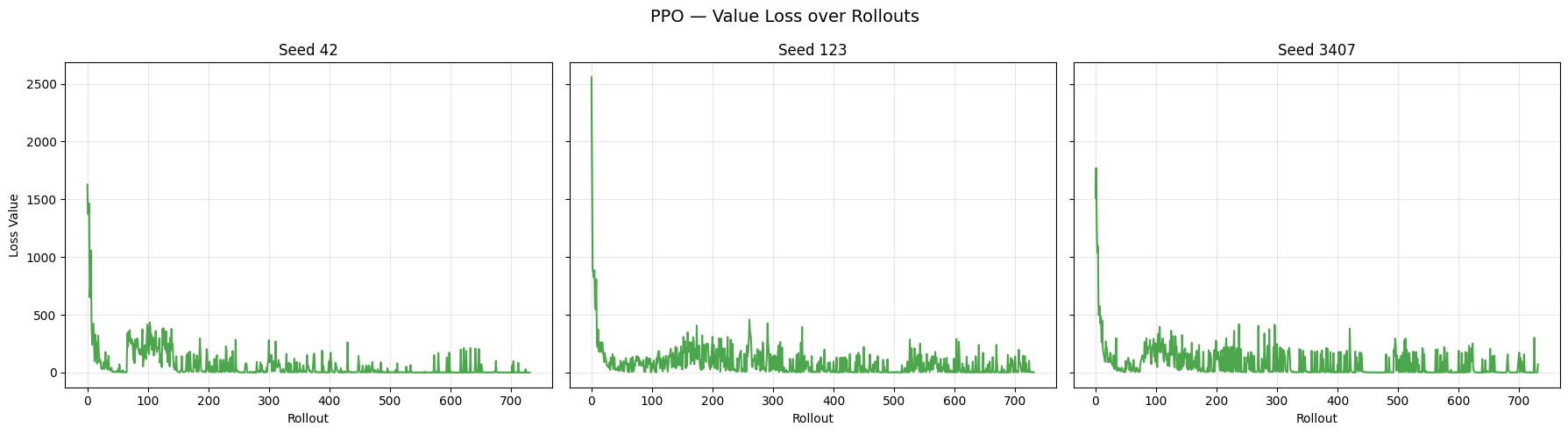


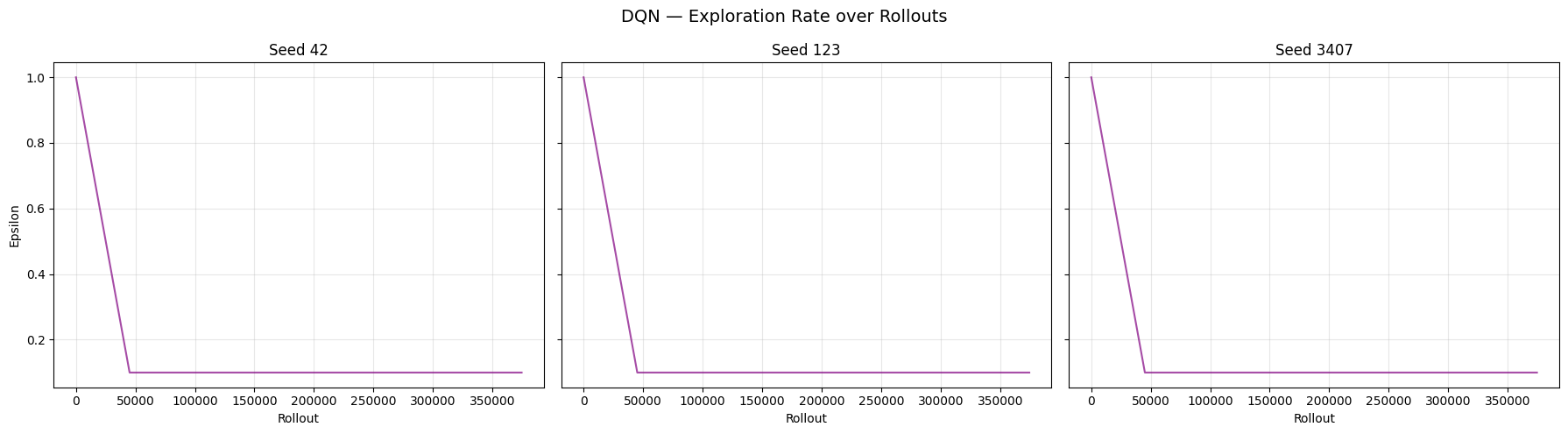


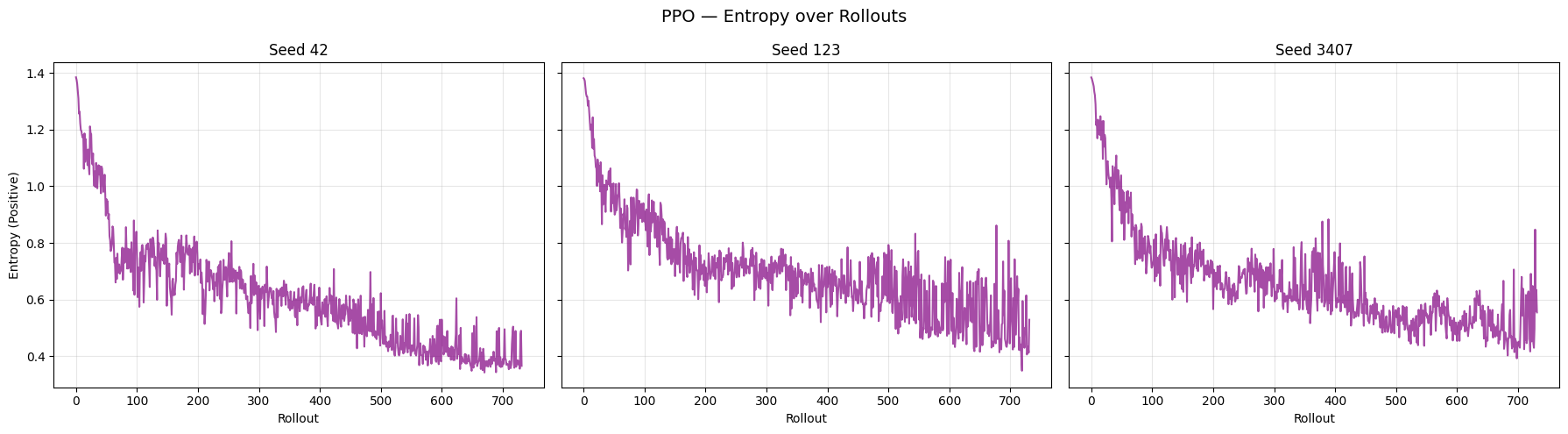


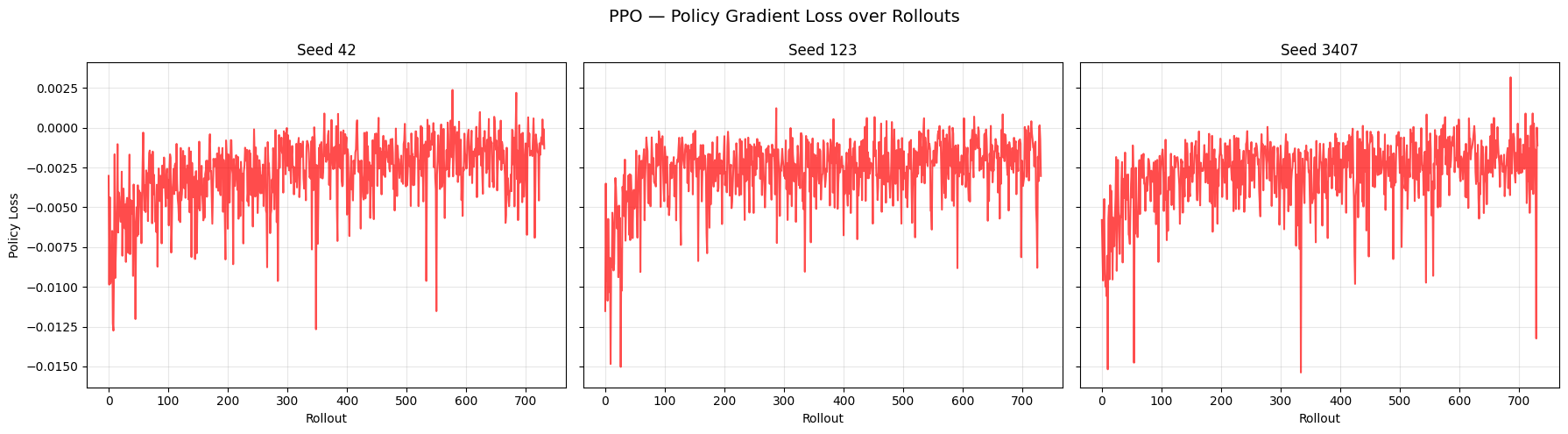


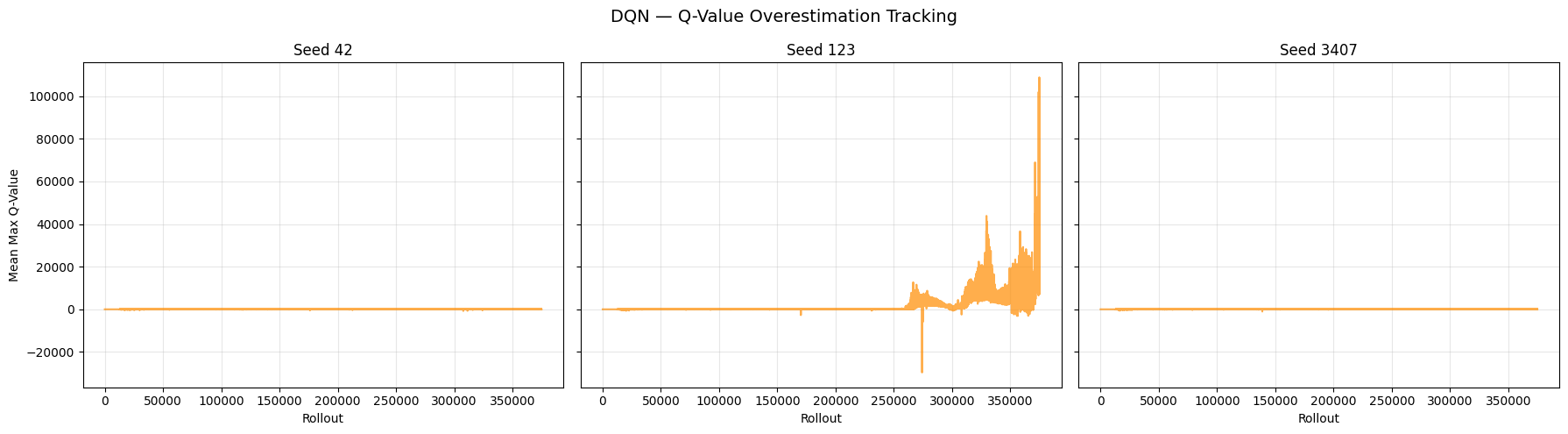




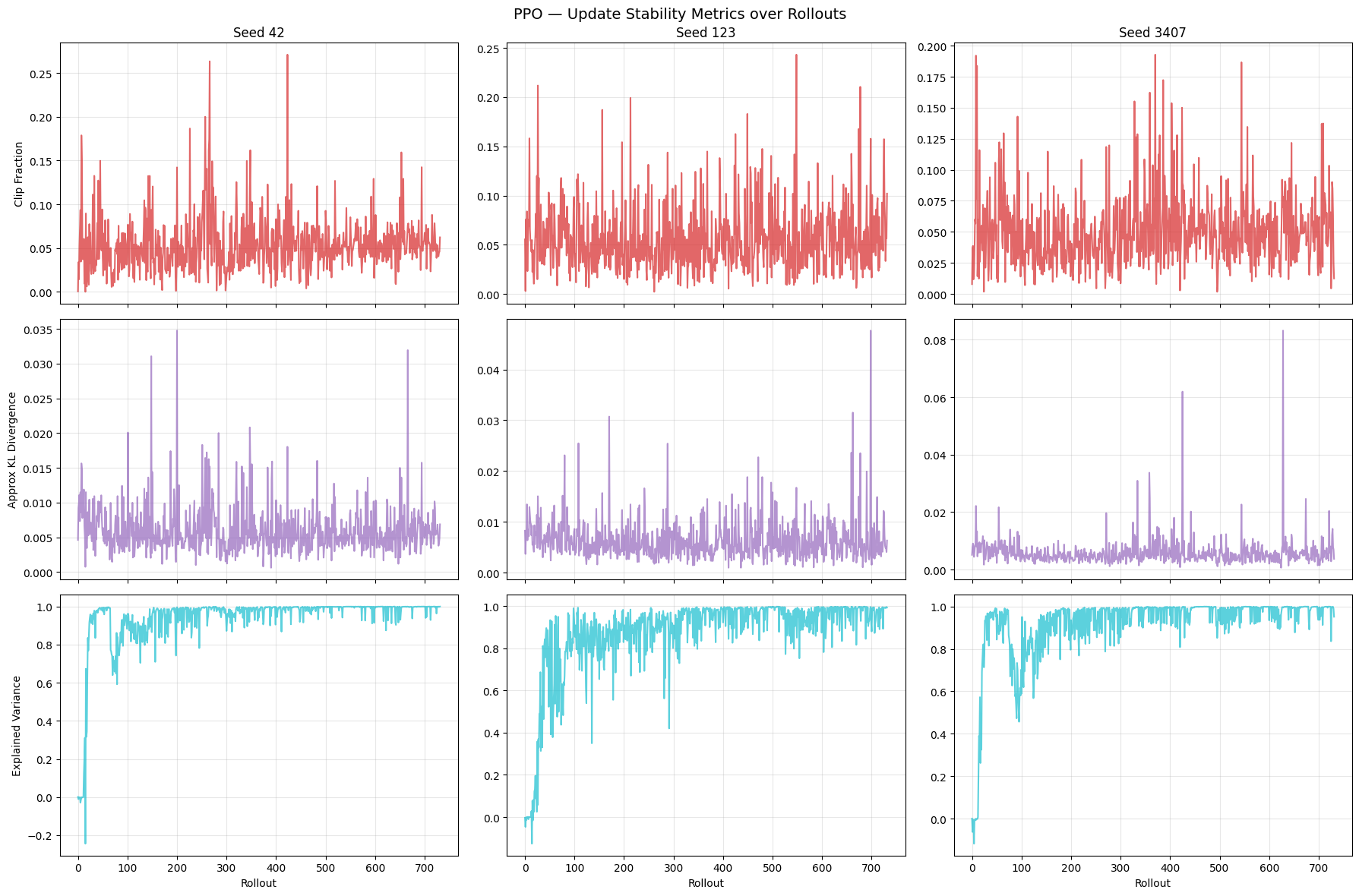




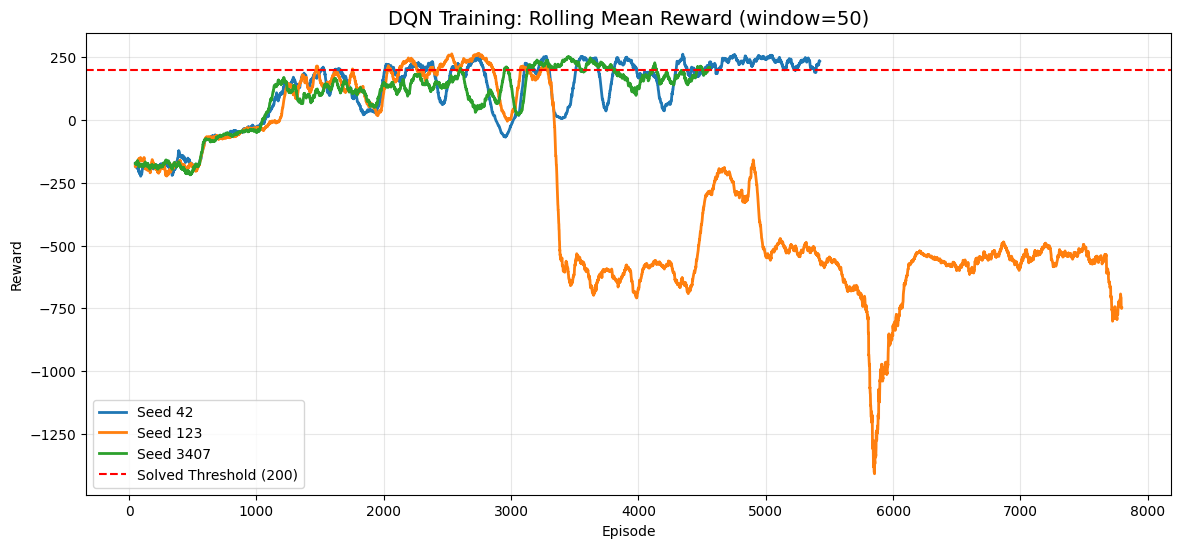


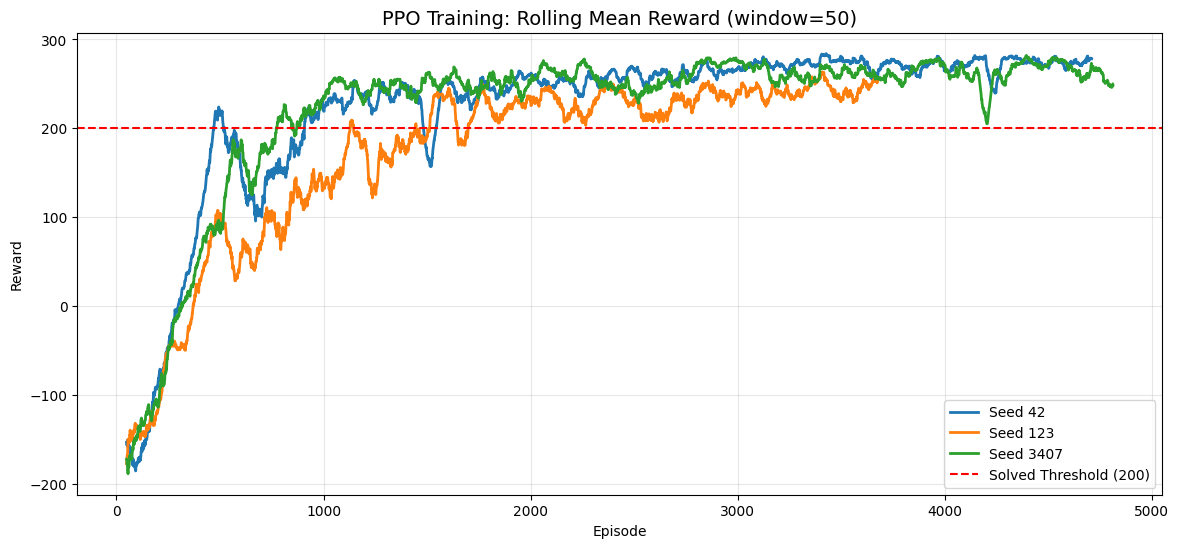


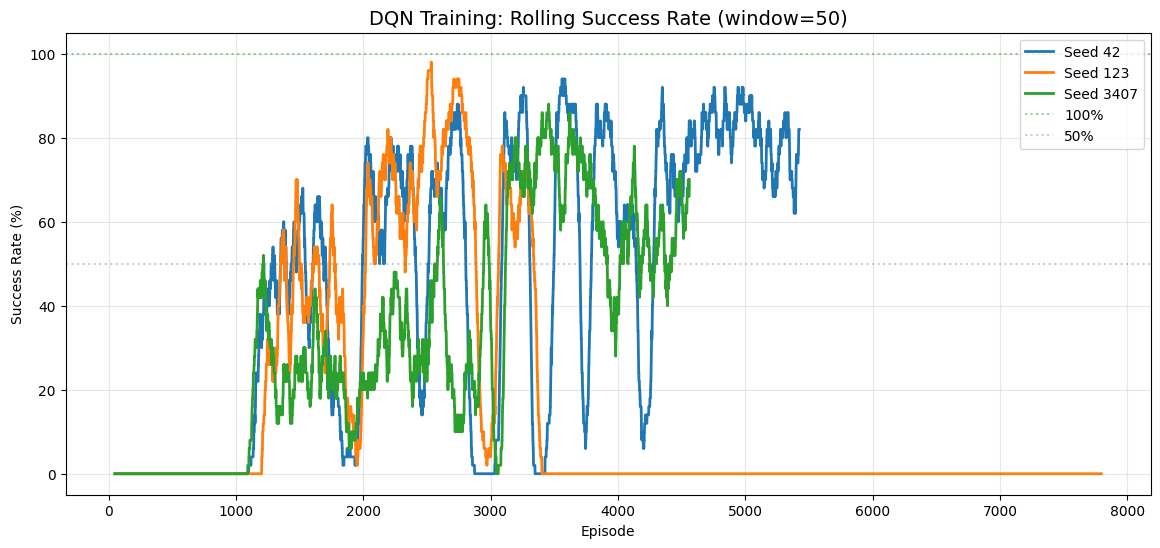
Note: Steadily rising Q-values that diverge from actual returns indicate overestimation.  
A stable or slowly growing curve suggests the target network is controlling overestimation.

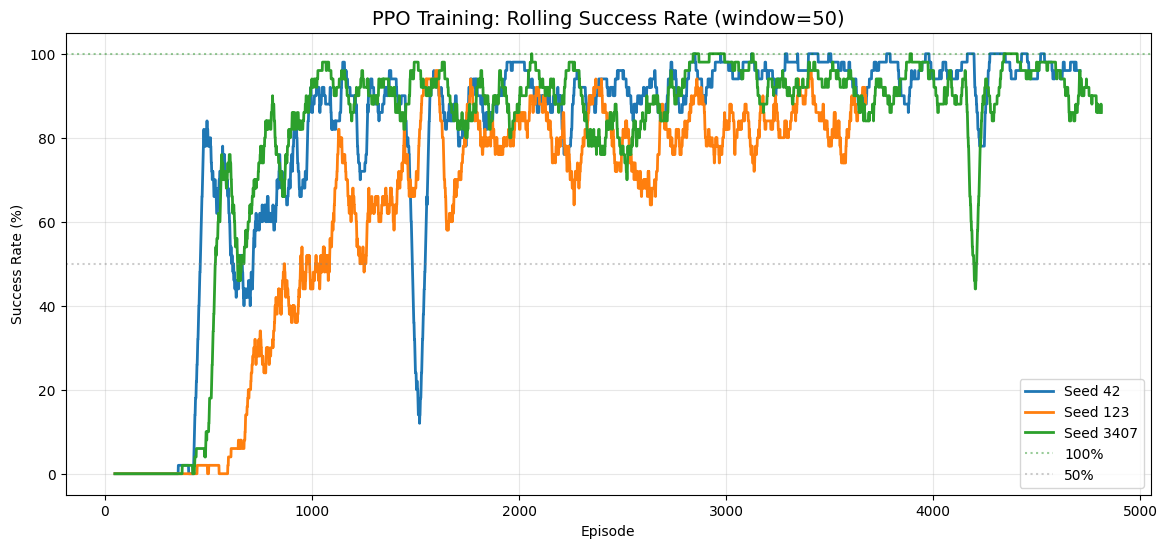


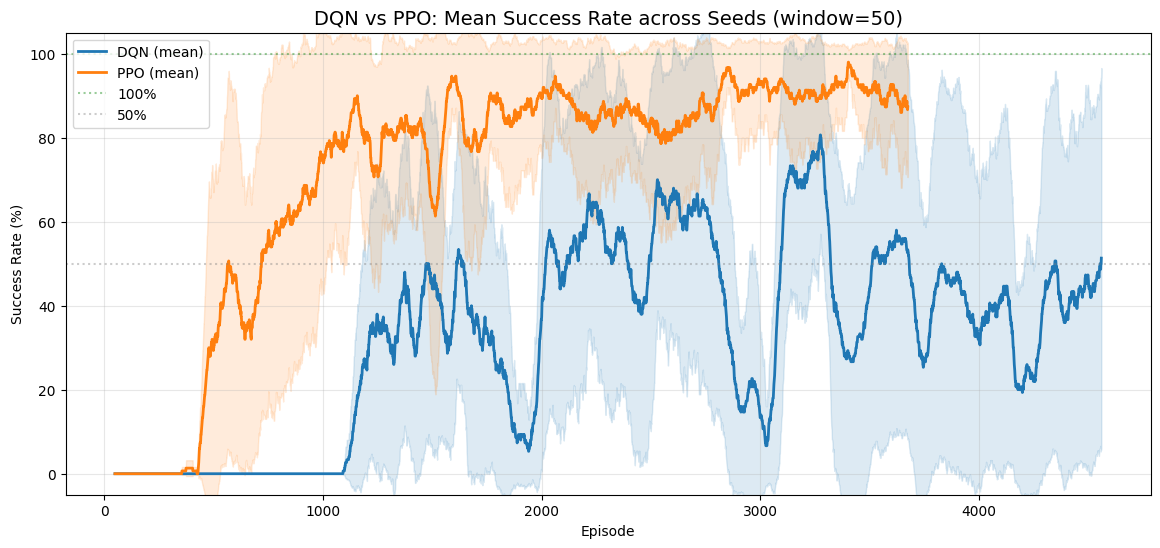
Clip Fraction: Fraction of policy updates clipped by PPO. High values suggest the policy is changing too fast.  
 Approx KL Divergence: KL divergence between old and new policy. Spikes indicate large policy shifts.  
 Explained Variance: How well the value function predicts returns. 1.0 = perfect, 0 = no better than mean.

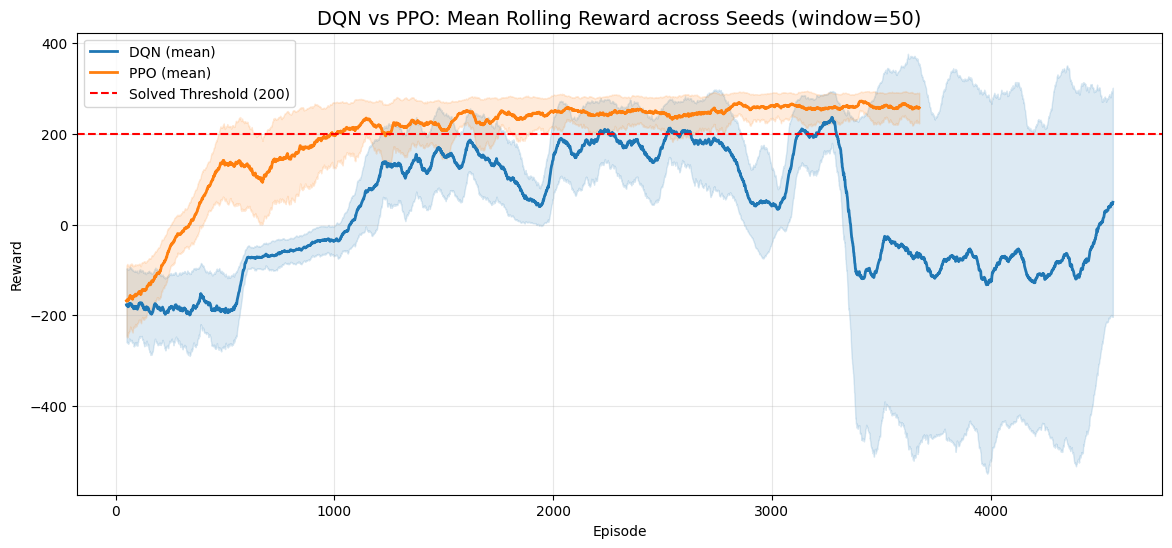












# 4. Evaluation

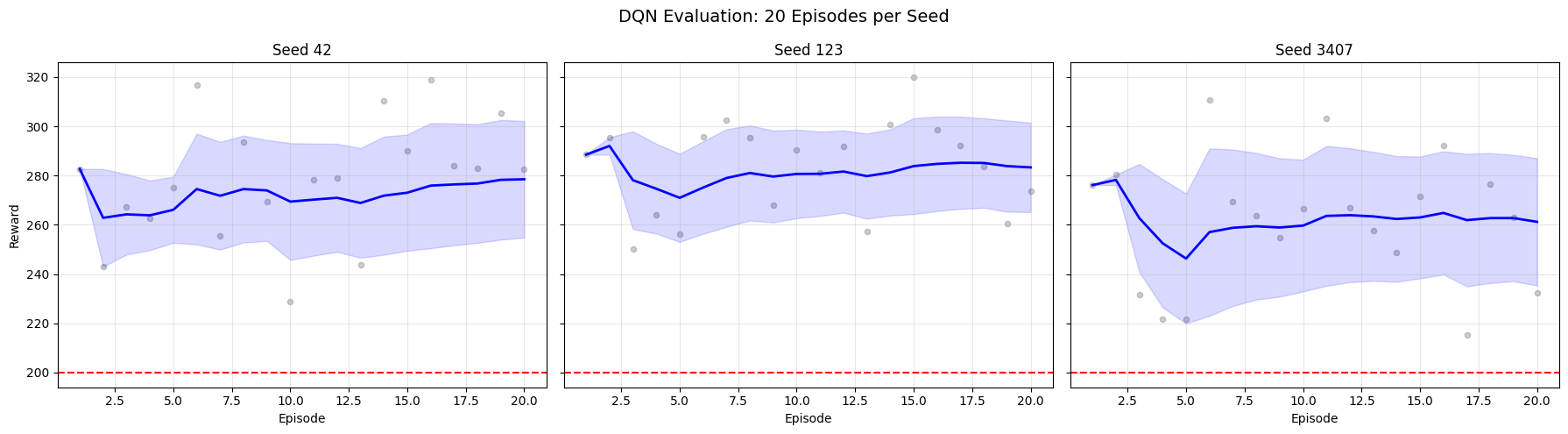
The following tables summarize the evaluation performance of each algorithm across all seeds. Each model was evaluated over 20 deterministic episodes. The overall row aggregates all episodes across seeds.

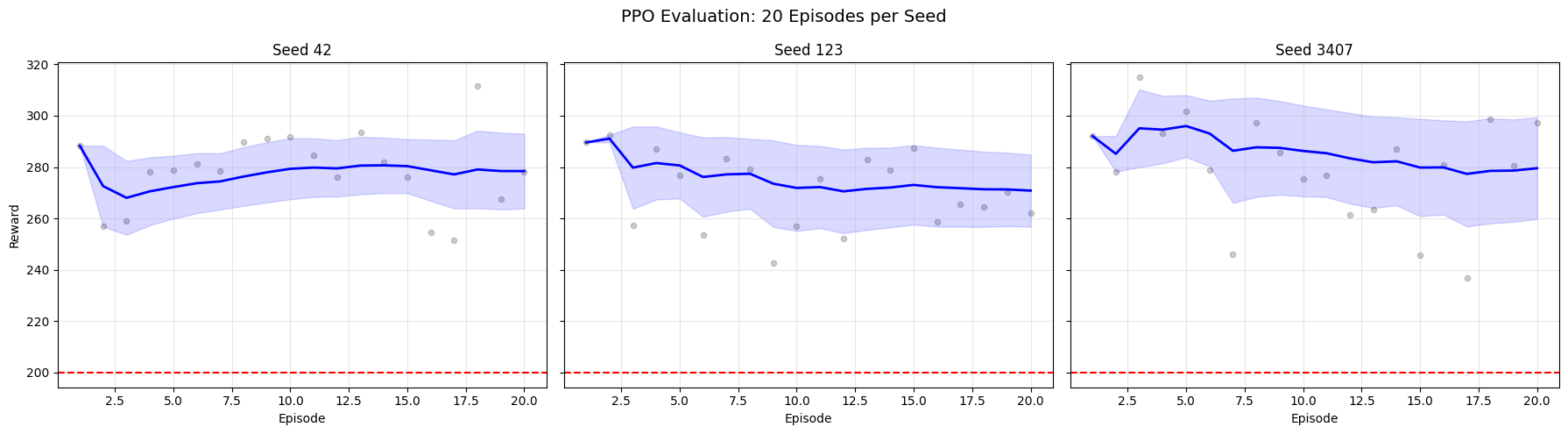
Session: dqn\_ppo  
Algorithms: ['dqn', 'ppo']  
Seeds: [42, 123, 3407]  
  
Discovered best models:  
 DQN seed 42: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/best\_model.zip  
 DQN seed 123: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/best\_model.zip  
 DQN seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/best\_model.zip  
 PPO seed 42: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/best\_model.zip  
 PPO seed 123: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/best\_model.zip  
 PPO seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/best\_model.zip  
  
Discovered eval logs:  
 DQN seed 42: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/eval\_log/evaluations.npz  
 DQN seed 123: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/eval\_log/evaluations.npz  
 DQN seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/eval\_log/evaluations.npz  
 PPO seed 42: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/eval\_log/evaluations.npz  
 PPO seed 123: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/eval\_log/evaluations.npz  
 PPO seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/eval\_log/evaluations.npz  
  
Discovered training logs:  
 DQN seed 42: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/training\_log.npz  
 DQN seed 123: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/training\_log.npz  
 DQN seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/training\_log.npz  
 PPO seed 42: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/training\_log.npz  
 PPO seed 123: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/training\_log.npz  
 PPO seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/training\_log.npz  
  
  
Loading and evaluating DQN seed 42 (best model)...  
Loading and evaluating DQN seed 123 (best model)...  
Loading and evaluating DQN seed 3407 (best model)...  
DQN: evaluation complete.  
  
Loading and evaluating PPO seed 42 (best model)...  
Loading and evaluating PPO seed 123 (best model)...  
Loading and evaluating PPO seed 3407 (best model)...  
PPO: evaluation complete.  
  
All evaluations complete.

## 4.1 Per-Algorithm Results

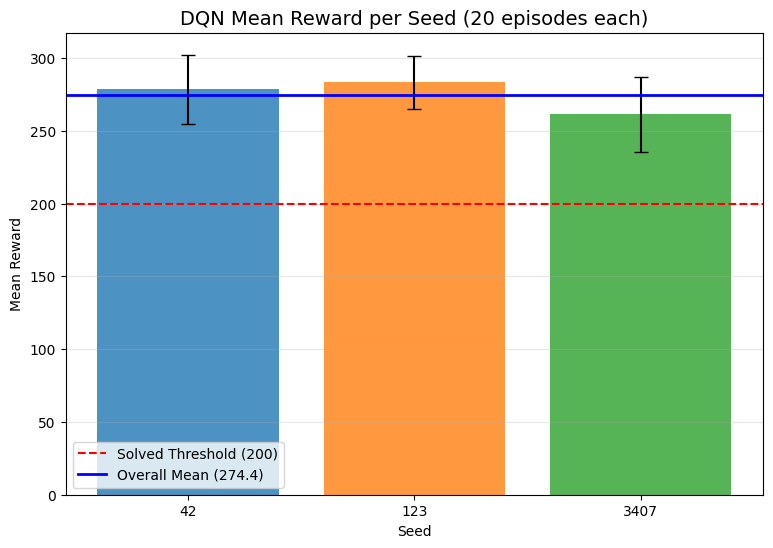
\*\*\* DQN MULTI-SEED EVALUATION SUMMARY \*\*\*  
Episodes per seed: 20 | Total: 60  
 Seed Mean Reward Std Dev Min Reward Max Reward Success Rate  
 42 278.51 23.70 228.67 318.97 100.0%  
 123 283.34 18.19 250.33 319.93 100.0%  
 3407 261.22 25.88 215.37 310.74 100.0%  
Overall 274.36 24.72 215.37 319.93 100.0%  
  
\*\*\* PPO MULTI-SEED EVALUATION SUMMARY \*\*\*  
Episodes per seed: 20 | Total: 60  
 Seed Mean Reward Std Dev Min Reward Max Reward Success Rate  
 42 278.43 14.56 251.45 311.70 100.0%  
 123 270.83 14.06 242.53 292.55 100.0%  
 3407 279.56 19.83 236.91 314.95 100.0%  
Overall 276.27 16.81 236.91 314.95 100.0%

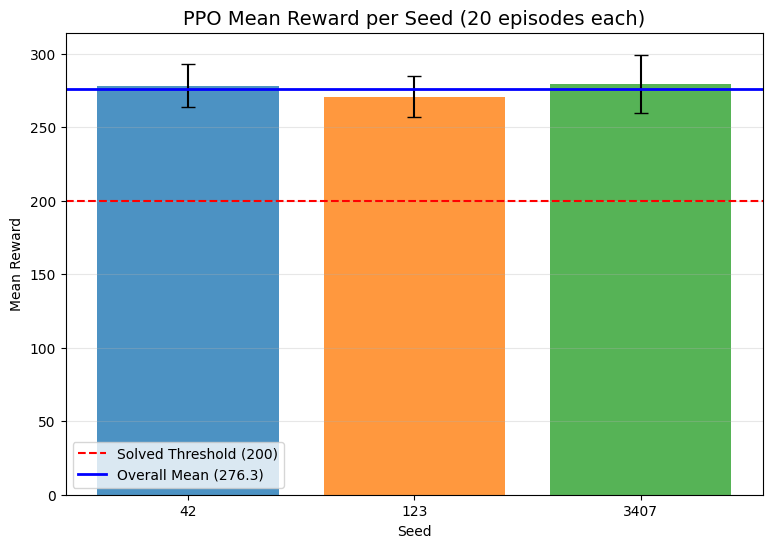
These convergence plots show the running mean reward across the 20 evaluation episodes for each seed. The shaded region represents one standard deviation. A flat, high running mean indicates consistent performance.

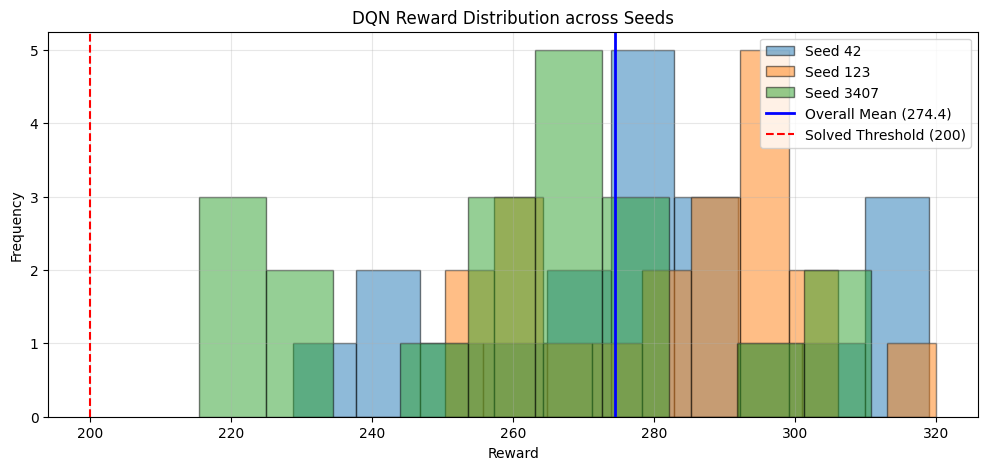


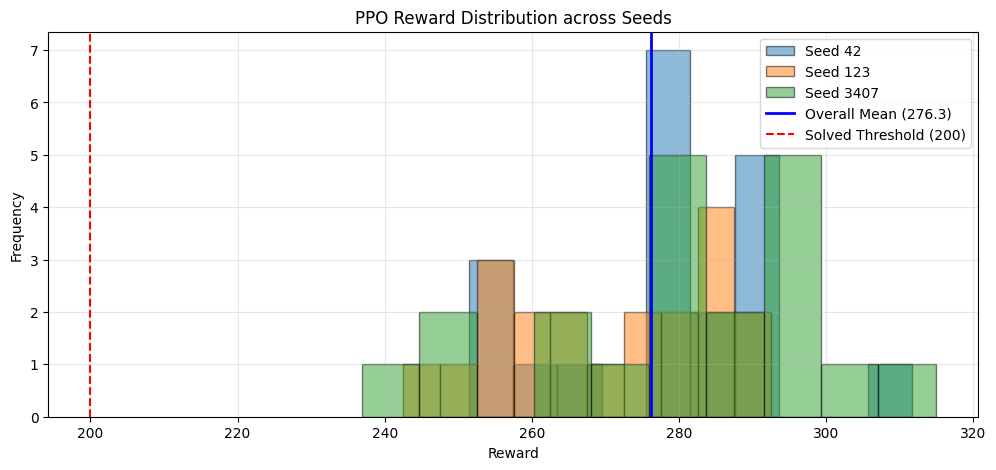


These convergence plots show the running mean reward across the 20 evaluation episodes for each seed. The shaded region represents one standard deviation. A flat, high running mean indicates consistent performance.



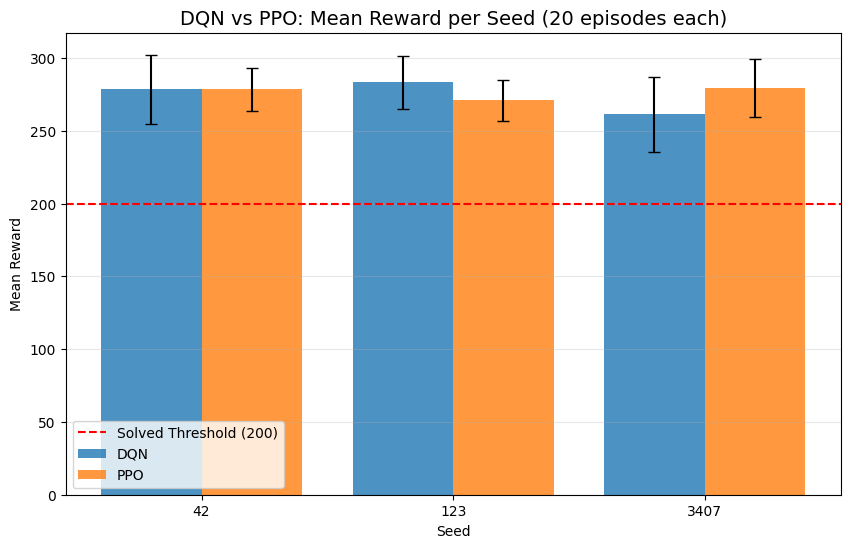


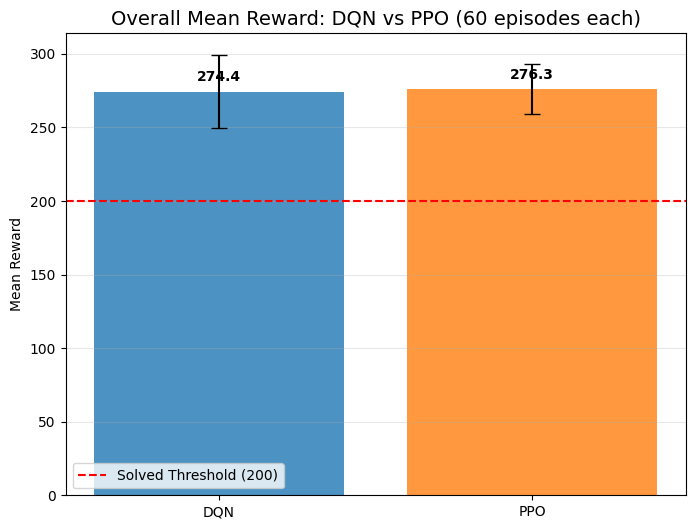


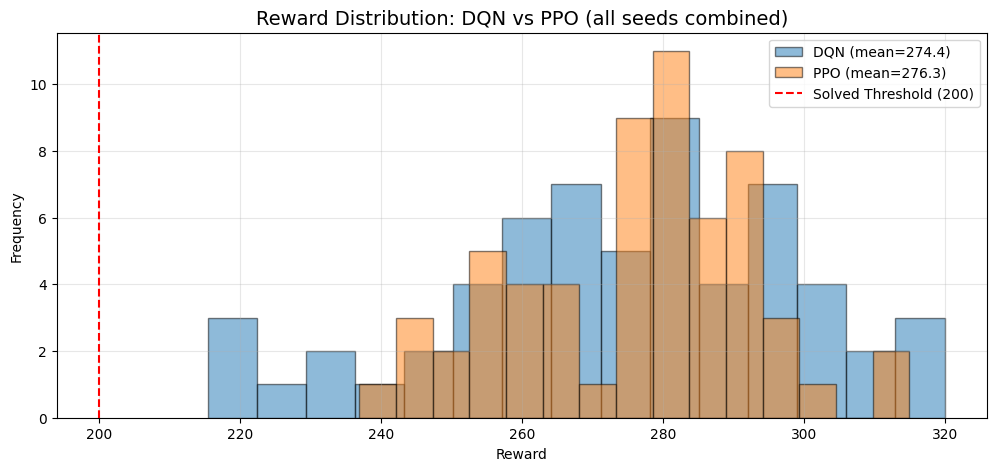


## 4.2 Cross-Algorithm Comparison

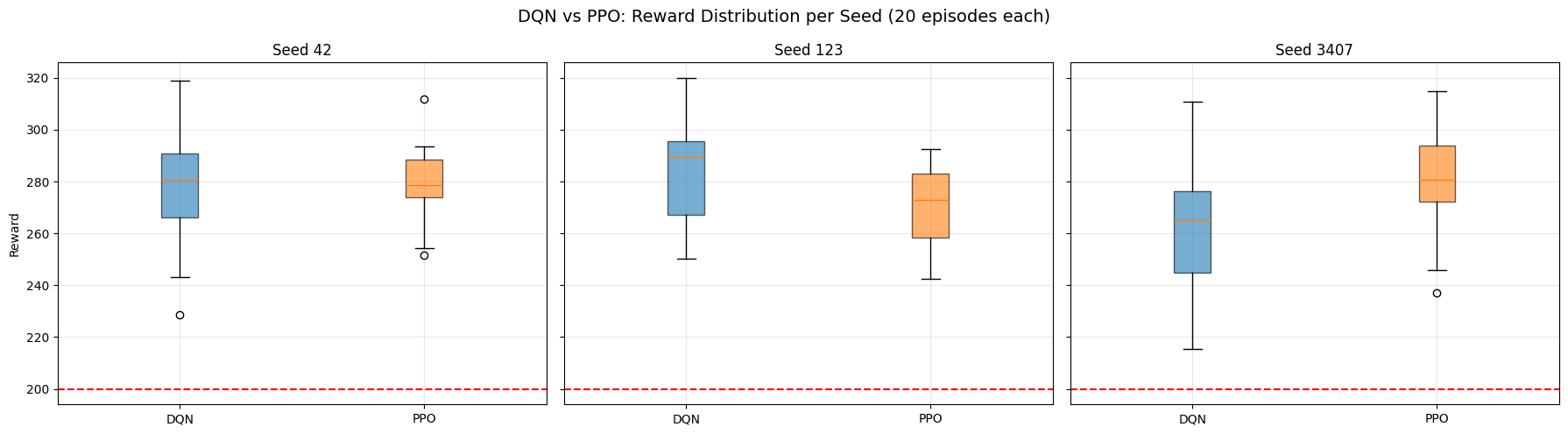
\*\*\* CROSS-ALGORITHM EVALUATION SUMMARY \*\*\*  
Seeds: [42, 123, 3407] | Episodes per seed: 20  
Total episodes per algorithm: 60  
  
Algorithm Mean Reward Std Dev Min Reward Max Reward Success Rate  
 DQN 274.36 24.72 215.37 319.93 100.0%  
 PPO 276.27 16.81 236.91 314.95 100.0%







/tmp/ipykernel\_6515/3208474440.py:9: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick\_labels' since Matplotlib 3.9; support for the old name will be dropped in 3.11.  
 bp = ax.boxplot(data, labels=[a.upper() for a in algo\_names], patch\_artist=True)  
/tmp/ipykernel\_6515/3208474440.py:9: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick\_labels' since Matplotlib 3.9; support for the old name will be dropped in 3.11.  
 bp = ax.boxplot(data, labels=[a.upper() for a in algo\_names], patch\_artist=True)  
/tmp/ipykernel\_6515/3208474440.py:9: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick\_labels' since Matplotlib 3.9; support for the old name will be dropped in 3.11.  
 bp = ax.boxplot(data, labels=[a.upper() for a in algo\_names], patch\_artist=True)



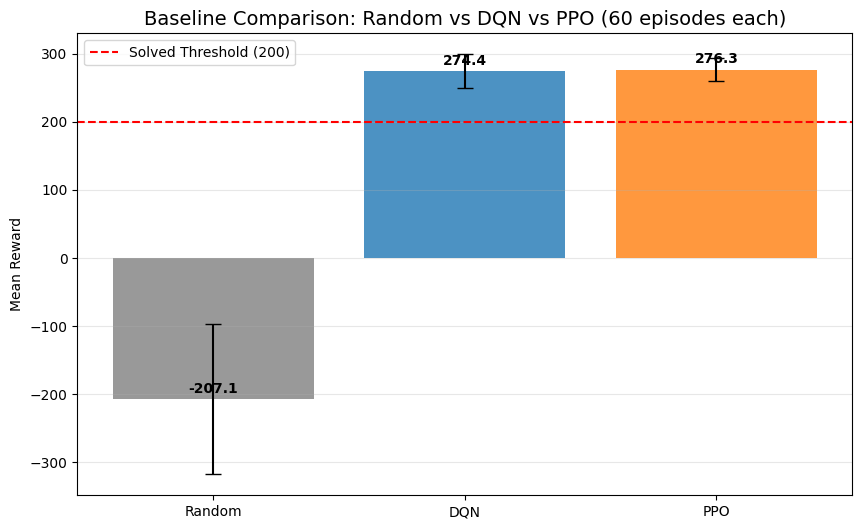
## 4.3 Statistical Significance

Two statistical tests assess whether the observed performance difference between DQN and PPO is significant. The Mann-Whitney U test compares reward distributions (non-parametric, no normality assumption). The Chi-squared test compares success rates as a proportion.

\*\*\* STATISTICAL SIGNIFICANCE TESTS \*\*\*  
Sample size per algorithm: 60 episodes (20 episodes x 3 seeds)  
  
 Metric DQN Value PPO Value Test Statistic p-value Significant (p<0.05)  
 Mean Reward 274.36 276.27 Mann-Whitney U 1771.0 0.8811 No  
Success Rate (>=200) 100.0% 100.0% Chi-squared (skipped: zero row/col) 0.00 1.0000 No

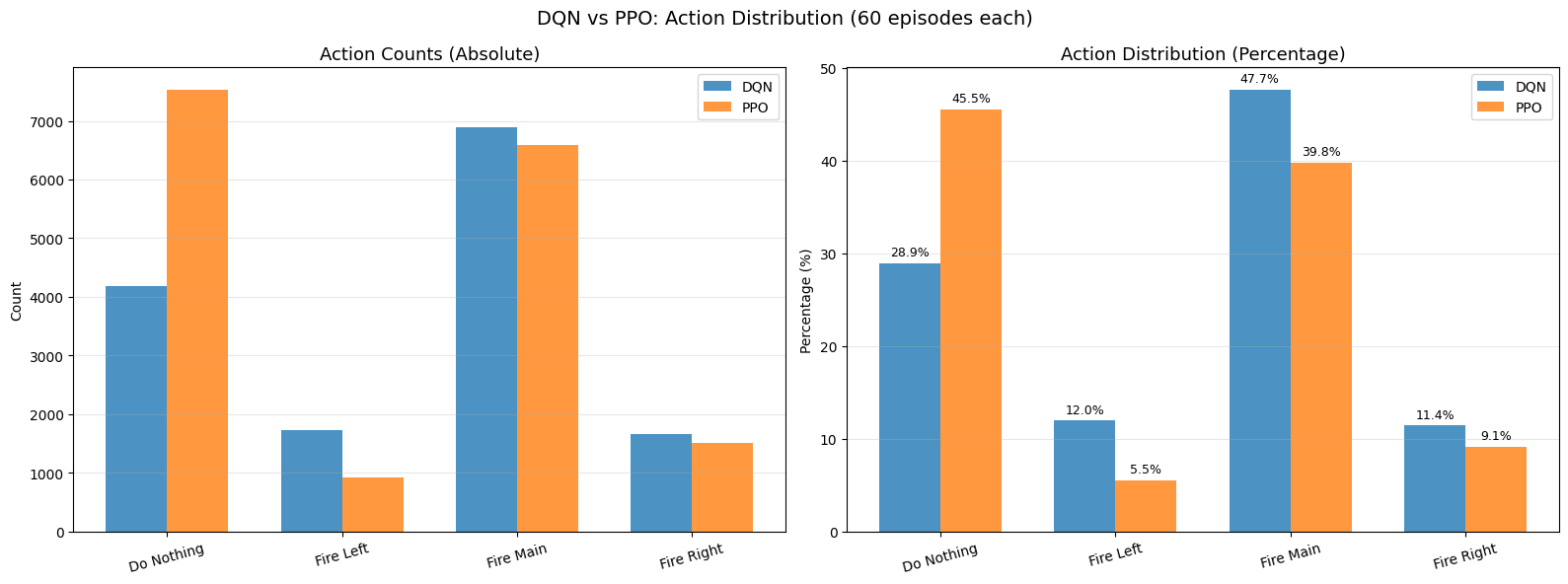
## 4.4 Baseline Comparison

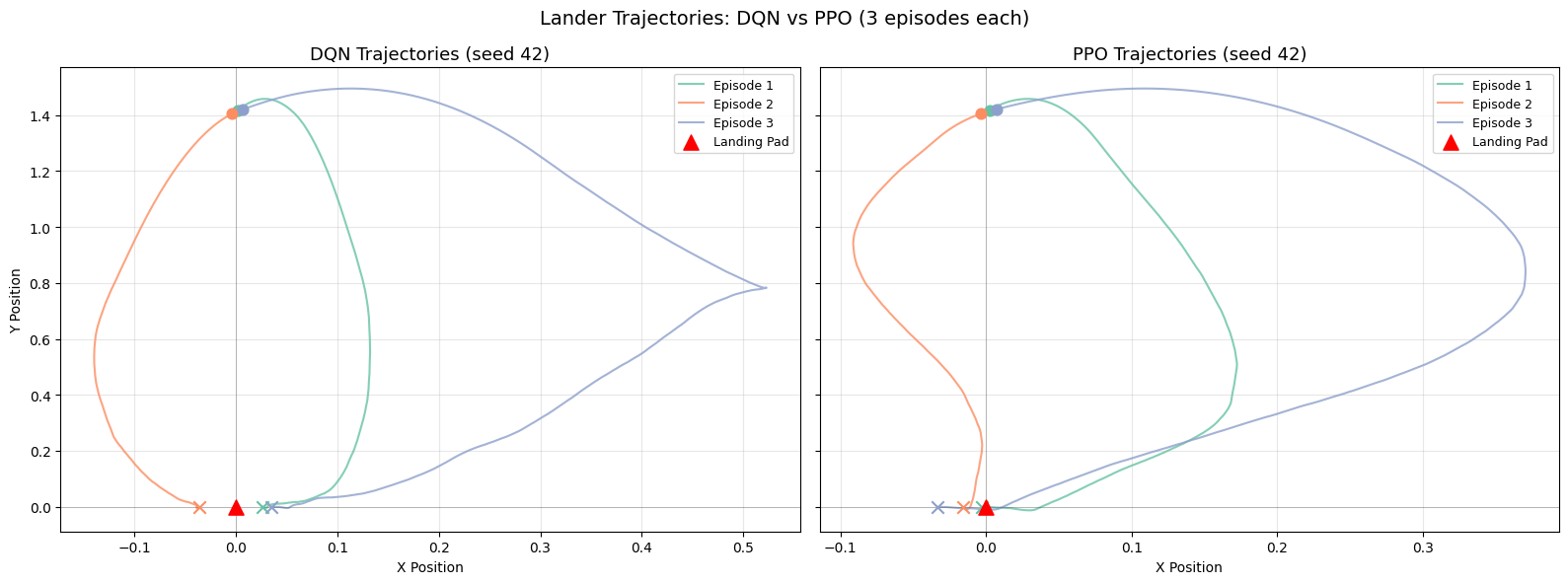
Running random agent with seed 42...  
Running random agent with seed 123...  
Running random agent with seed 3407...  
Random baseline evaluation complete.  
  
  
\*\*\* BASELINE COMPARISON \*\*\*  
 Agent Mean Reward Std Dev Min Max Success Rate  
Random -207.06 110.45 -416.10 15.45 0.0%  
 DQN 274.36 24.72 215.37 319.93 100.0%  
 PPO 276.27 16.81 236.91 314.95 100.0%



## 4.5 Agent Behavior Analysis

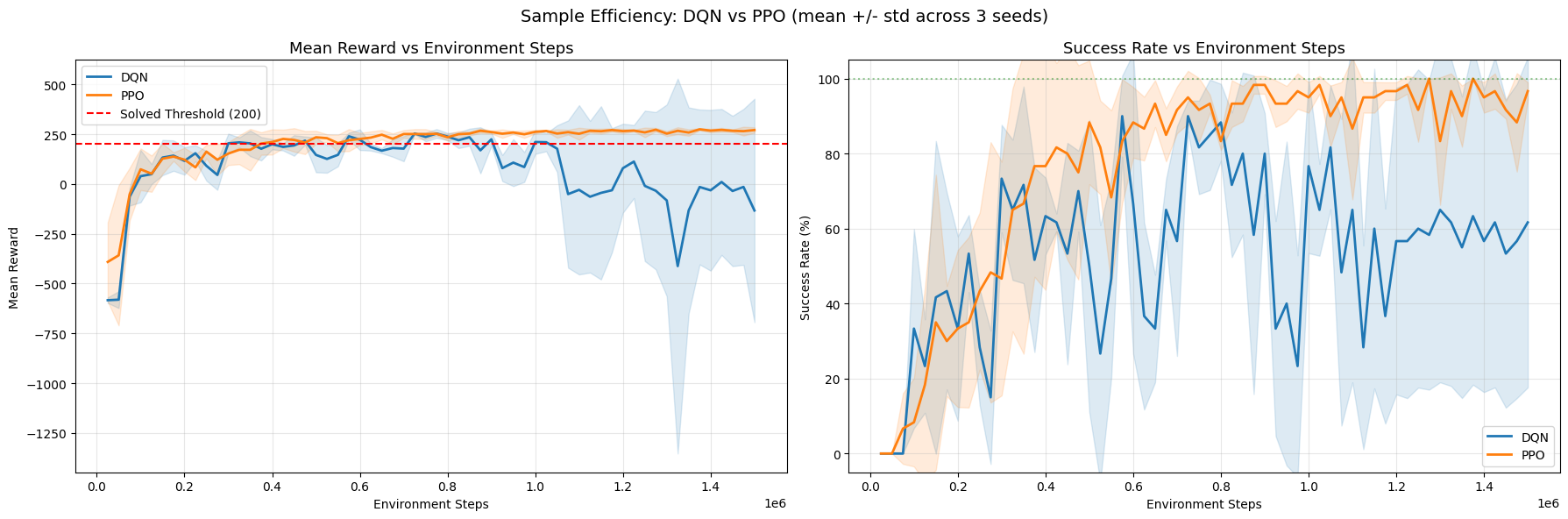
DQN: 14,451 total actions collected across 60 episodes  
PPO: 16,544 total actions collected across 60 episodes  
  
Behavior data collection complete.







## 4.6 Sample Efficiency Analysis



\*\*\* SAMPLE EFFICIENCY: FIRST SOLVED TIMESTEP \*\*\*  
Solved = mean evaluation reward >= 200  
Evaluated every 25,000 timesteps with 20 episodes  
  
Algorithm Seed First Solved at Step  
 DQN 42 300,000  
 DQN 123 300,000  
 DQN 3407 150,000  
 DQN Mean 250,000  
 PPO 42 150,000  
 PPO 123 525,000  
 PPO 3407 325,000  
 PPO Mean 333,333

## 4.7 Training Milestone Timeline

This chart overlays the training reward curves with vertical milestone markers showing when each algorithm reached key performance thresholds. The background curves show the rolling mean reward (window=100 episodes) plotted against cumulative environment steps, reconstructed from the per-episode training logs. Three milestones are marked per algorithm:

* Best model (solid line): the checkpoint selected by the combined metric (mean - std) during training
* First solved (dashed line): first point where the rolling 100-episode mean reward >= 200
* First all-pass (dotted line): first point where all 100 consecutive episodes scored >= 200
* Discovered training logs: DQN seed 42: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_02\_53\_32/training\_log.npz DQN seed 123: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_03\_41\_09/training\_log.npz DQN seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/dqn/2026-02-23\_04\_28\_07/training\_log.npz PPO seed 42: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_15\_27/training\_log.npz PPO seed 123: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_35\_37/training\_log.npz PPO seed 3407: /home/logus/env/iscte/taap\_p2/report/../models/ppo/2026-02-23\_05\_58\_38/training\_log.npz



**Figure 14** - Training milestone timeline showing rolling mean reward curves (per-seed, light lines) with vertical markers for best model selection, first solved (rolling 100-episode mean >= 200), and first all-pass (all 100 consecutive episodes >= 200) milestones.

\*\*\* TRAINING MILESTONES PER SEED \*\*\*  
Rolling window: 100 episodes  
Solved threshold: mean reward >= 200  
All-pass threshold: every episode in window >= 200  
  
Algorithm Seed First Solved (rolling mean >= 200) First All-Pass (all 100 >= 200)  
 DQN 42 553,533 Not reached  
 DQN 123 614,569 Not reached  
 DQN 3407 1,085,079 Not reached  
 PPO 42 166,110 1,407,532  
 PPO 123 662,791 Not reached  
 PPO 3407 289,977 966,886

## 4.8 Final vs Best Model Comparison

Evaluating DQN seed 42 (final model)...  
Evaluating DQN seed 123 (final model)...  
Evaluating DQN seed 3407 (final model)...  
Evaluating PPO seed 42 (final model)...  
Evaluating PPO seed 123 (final model)...  
Evaluating PPO seed 3407 (final model)...  
  
Final model evaluation complete.  
  
  
\*\*\* BEST MODEL vs FINAL MODEL \*\*\*  
Episodes per evaluation: 20  
  
Algorithm Seed Best Mean Best Success Final Mean Final Success Delta Mean  
 DQN 42 278.51 100% 266.82 95% +11.69  
 DQN 123 283.34 100% -935.49 0% +1218.83  
 DQN 3407 261.22 100% 286.92 100% -25.70  
 PPO 42 278.43 100% 276.39 95% +2.04  
 PPO 123 270.83 100% 247.72 85% +23.11  
 PPO 3407 279.56 100% 267.67 95% +11.88  
  
Positive Delta Mean = best model selection improved over end-of-training snapshot.

## 4.9 Hyperparameter Exploration Summary

\*\*\* DQN Final Hyperparameters \*\*\*  
 Parameter Value  
 policy MlpPolicy  
 learning\_rate <function linear\_schedule.<locals>.func at 0x75da6d15cc20>  
 learning\_starts 50000  
 buffer\_size 750000  
 batch\_size 128  
 gamma 0.99  
 exploration\_fraction 0.12  
 exploration\_final\_eps 0.1  
target\_update\_interval 250  
 train\_freq 4  
 gradient\_steps 4  
 policy\_kwargs {'net\_arch': [256, 256]}  
 device cpu  
 total\_timesteps 1,500,000  
 device cpu  
 policy MlpPolicy  
  
\*\*\* PPO Final Hyperparameters \*\*\*  
 Parameter Value  
 learning\_rate 0.00025  
 n\_steps 2048  
 batch\_size 64  
 n\_epochs 10  
 gamma 0.999  
 gae\_lambda 0.95  
 ent\_coef 0.01  
 clip\_range 0.2  
total\_timesteps 1,500,000  
 device cpu  
 policy MlpPolicy  
  
  
  
\*\*\* DQN HYPERPARAMETER EXPLORATION \*\*\*  
  
 Config Key Changes Success Rate Mean Reward Notes  
 A (baseline) lr=6.3e-4, buffer=100k, eps\_final=0.01, explore=0.5, soft updates (tau=0.005) 84.0% 245.47 Slow exploration decay, soft target updates  
B (Zoo-style) lr=linear(6.3e-4), buffer=1M, eps\_final=0.1, explore=0.12, hard updates (250) 96.0% 263.02 Short exploration, high final epsilon, hard target updates  
 C (final) lr=linear(6.3e-4), buffer=750k, eps\_final=0.1, explore=0.12, hard updates (250), learning\_starts=50k TBD (lab011) TBD (lab011) Added learning\_starts buffer fill, gradient\_steps=4 explicit  
  
\*\*\* PPO HYPERPARAMETER EXPLORATION \*\*\*  
  
 Config Key Changes Success Rate Mean Reward Notes  
A (final) lr=2.5e-4, gamma=0.999, n\_steps=2048, batch=64, n\_epochs=10, ent\_coef=0.01 98.0% 269.21 Standard config, single environment  
  
Note: Update Config C results and add any additional configs tested during lab011 runs.

## 4.10 Visualizations

Generating GIF for DQN seed 42 (best model)...  
 Saved: /home/logus/env/iscte/taap\_p2/report/outputs\_dqn/dqn\_seed42.gif  
  
  
  
<IPython.core.display.Image object>  
  
  
Generating GIF for DQN seed 123 (best model)...  
 Saved: /home/logus/env/iscte/taap\_p2/report/outputs\_dqn/dqn\_seed123.gif  
  
  
  
<IPython.core.display.Image object>  
  
  
Generating GIF for DQN seed 3407 (best model)...  
 Saved: /home/logus/env/iscte/taap\_p2/report/outputs\_dqn/dqn\_seed3407.gif  
  
  
  
<IPython.core.display.Image object>  
  
  
Generating GIF for PPO seed 42 (best model)...  
 Saved: /home/logus/env/iscte/taap\_p2/report/outputs\_ppo/ppo\_seed42.gif  
  
  
  
<IPython.core.display.Image object>  
  
  
Generating GIF for PPO seed 123 (best model)...  
 Saved: /home/logus/env/iscte/taap\_p2/report/outputs\_ppo/ppo\_seed123.gif  
  
  
  
<IPython.core.display.Image object>  
  
  
Generating GIF for PPO seed 3407 (best model)...  
 Saved: /home/logus/env/iscte/taap\_p2/report/outputs\_ppo/ppo\_seed3407.gif  
  
  
  
<IPython.core.display.Image object>

# 5. Experimental Setup

Environment: LunarLander-v3  
Observation space: Box([ -2.5 -2.5 -10. -10. -6.2831855 -10.  
 -0. -0. ], [ 2.5 2.5 10. 10. 6.2831855 10.  
 1. 1. ], (8,), float32)  
Action space: Discrete(4)  
Wind enabled: False  
  
Sample observation: [ 0.00229702 1.4181306 0.2326471 0.3204666 -0.00265488 -0.05269805  
 0. 0. ]  
Observation labels: [x, y, vx, vy, angle, angular\_vel, left\_leg, right\_leg]  
  
  
Python: 3.12.3  
PyTorch: 2.10.0+cu130  
Stable-Baselines3: 2.7.1  
Gymnasium: 1.2.3  
NumPy: 2.4.2  
Device: cpu  
CUDA: 13.0

# 6. Conceptual Discussion

This section presents a theoretical discussion of the core conceptual differences between value-based and policy-based reinforcement learning methods, the sources of instability in deep RL, and the exploration mechanisms used in DQN and PPO.

## 6.1 Value-Based vs Policy-Based Learning

Reinforcement Learning algorithms can be categorised according to what is parameterised and optimised.

### Value-Based Learning

Value-based methods approximate a value function, typically the action-value function:

The optimal policy is derived implicitly:

Deep Q-Networks (DQN) approximate and update parameters using the Bellman optimality equation:

This introduces **bootstrapping**, since the target depends on the model’s own predictions.

Key characteristics:

* Off-policy learning
* Bootstrapping
* Policy extracted via maximisation

However, the use of the max operator combined with function approximation introduces instability.

### Policy-Based Learning

Policy-based methods parameterise the policy directly:

and optimise the expected return:

Using the policy gradient theorem:

These methods:

* Learn the policy directly
* Avoid explicit maximisation over value estimates
* Naturally handle continuous action spaces

Because they do not rely on a max operator over Q-values, they tend to be structurally more stable in deep settings.

## 6.2 Overestimation and Instability in Value-Based Algorithms

### Overestimation Bias

Overestimation arises from the maximisation step . If Q-value estimates contain zero-mean noise:

Thus, even unbiased estimators lead to positively biased maximum estimates.

In DQN, the same network selects the action (argmax) and evaluates the action. This coupling amplifies overestimation bias.

### Structural Instability: The Deadly Triad

Deep value-based RL combines:

1. Function approximation
2. Bootstrapping
3. Off-policy learning

This combination is known as the **deadly triad**, which may cause divergence or unstable oscillations.

PPO avoids these structural issues because:

* It does not use a max over Q-values
* It performs on-policy updates
* Value estimates are auxiliary rather than directly driving action selection

## 6.3 PPO Clipping and Generalised Advantage Estimation (GAE)

### PPO Clipped Objective

PPO constrains policy updates through the clipped surrogate objective:

where:

This objective:

* Approximates a trust-region method
* Prevents excessively large updates
* Improves training stability

### Generalised Advantage Estimation (GAE)

The advantage function is defined as:

GAE computes:

where the temporal-difference residual is:

The parameter controls the bias-variance trade-off:

* : lower variance, higher bias
* : lower bias, higher variance

GAE reduces variance in policy gradient updates while maintaining acceptable bias, improving sample efficiency and stability.

## 6.4 Epsilon-Greedy vs Entropy-Driven Exploration

### Epsilon-Greedy Exploration (DQN)

The behaviour policy is defined as:

Limitations:

* Uniform and uninformed exploration
* Independent of uncertainty
* Becomes fully greedy as

Exploration is externally imposed rather than integrated into optimisation.

### Entropy-Regularised Exploration (PPO)

PPO includes an entropy bonus in the objective:

where entropy is defined as:

This mechanism:

* Encourages stochastic policies early in training
* Gradually reduces randomness
* Couples exploration with optimisation

Unlike epsilon-greedy, exploration is intrinsic to the learning objective.

## 6.5 Conceptual Summary

In summary, value-based methods optimise an approximation of the value function and derive the policy implicitly through action maximisation. This structural reliance on the max operator and bootstrapping introduces overestimation bias and makes deep value-based methods particularly sensitive to instability when combined with function approximation and off-policy learning. In contrast, policy-based methods directly parameterise and optimise the policy, avoiding explicit maximisation over value estimates and thereby reducing structural sources of instability. PPO further improves stability through its clipped surrogate objective, which constrains policy updates, and through Generalised Advantage Estimation, which manages the bias-variance trade-off in gradient estimates. Regarding exploration, epsilon-greedy strategies impose external and uninformed randomness that vanishes as epsilon decays, whereas entropy-regularised exploration integrates stochasticity directly into the optimisation objective, allowing exploration to adapt naturally as learning progresses. Overall, these theoretical differences explain why PPO typically exhibits smoother and more stable learning dynamics compared to DQN in complex environments.