NOVA IMS Information Management School

Mestrado em Data Science and Advanced Analytics, com especialização em Data Science

Reinforcement Learning Project

Abstract

This study presents a comprehensive evaluation of reinforcement learning algorithms across two distinct environments with different state and action space characteristics. We implemented and compared multiple RL approaches on ALE/Bowling, a discrete-action visual environment, and PandaReach-v3, a continuous control robotic manipulation task. For the Bowling environment, we evaluated Q-Learning with discretized observations, Deep Q-Network (DQN) with convolutional neural networks, and Proximal Policy Optimization (PPO) with actor-critic architecture. Results demonstrate that PPO significantly outperformed other methods, achieving stable rewards of 55–60 with minimal variance, while Q-Learning and DQN showed volatile behavior and lower performance (22–35 average rewards). For the PandaReach-v3 environment, we compared Soft Actor-Critic (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithms, both with and without Hindsight Experience Replay (HER) using three goal selection strategies (Final, Future, Random). Contrary to expectations, all algorithm variants achieved poor performance with approximately 10% success rates, and HER provided no significant improvements over baseline methods. The study reveals the effectiveness of policy gradient methods for visual environments while highlighting the challenges of sparse reward robotic manipulation tasks. Our findings suggest that successful goal-conditioned robotic learning may require extended training periods, more sophisticated exploration strategies, or alternative algorithmic approaches beyond current state-of-the-art methods.

Eduardo Mendes, 20240850
Jorge Cordeiro, 20240594
Marta Boavida, 20240519
Sofia Gomes, 20240848

Spring Semester 2024–2025

Contents

1 Introduction

In this project, we developed two reinforcement learning (RL) agents to solve two distinct environments not covered in class, as required. We selected the following environments: ALE/Bowling [2], a classic Atari environment with visual (image-based) observations and discrete action space and Panda-Gym (Reach Task) [3], a robotic control environment featuring continuous state and action spaces.

These environments represent different challenges: the first focuses on visual perception and discrete decision-making, while the second requires precise motor control in a continuous 3D space.

This report outlines the steps taken to build our solution, supported by visualizations and analysis. The full project code is available at: https://github.com/Dumendes10/RL24_25.

2 Methodology - Bowling

The Bowling environment, part of the Atari Learning Environment (ALE), is a discrete-action game in which the agent must learn to knock down pins by strategically controlling a bowling ball. The environment presents the agent with visual input in the form of RGB image frames with a resolution of 210 by 160 pixels. The available actions are discrete and limited to simple commands, such as moving left or right and throwing the ball. This setting poses a significant challenge due to the sparsity of rewards and the delayed nature of the feedback, making it difficult for the agent to associate specific actions with successful outcomes. The main objective is to maximize the cumulative reward by discovering and refining an effective bowling strategy through trial and error.

To address the Bowling environment, we implemented a general-purpose agent architecture designed to support multiple reinforcement learning algorithms. This agent was defined in a separate module and served as a reusable interface for training across different learning strategies. Our overall setup was structured around a central script that handled environment initialization and training execution, allowing for consistent experimentation across algorithms.

We applied three distinct reinforcement learning methods to train agents in this environment: Q-Learning, Deep Q-Network (DQN), and Proximal Policy Optimization (PPO). Each algorithm was implemented in its own dedicated training script, allowing for a modular and consistent experimental setup. The Q-Learning implementation required discretizing the observation space to make tabular learning applicable to the environment. In contrast, DQN used a convolutional neural network to directly process the high-dimensional visual input provided by the game frames. PPO, a policy-gradient-based method, employed an actor-critic architecture, with separate networks for policy and value estimation. It aimed to achieve stable policy updates through the use of clipped surrogate objectives.

This multi-algorithm approach enabled us to compare the performance, learning

dynamics, and suitability of each method for solving the same visual, sparse-reward environment.

3 Methodology - PandaReach-v3 Environment

3.1 Environment Description

The `PandaReach-v3` environment from the `panda-gym` package simulates a 7-DOF Franka Emika Panda robotic arm performing goal-conditioned reaching tasks. It presents a challenging continuous control problem, representative of real-world robotic manipulation scenarios.

3.1.1 State Space Characteristics:

- **Observation Space**: Multi-dict space containing:
  - `observation`: 25-dimensional continuous vector including joint positions, velocities, gripper state, and object position
  - `achieved_goal`: 3-dimensional vector representing current end-effector position
  - `desired_goal`: 3-dimensional vector representing target position
- **State Representation**: The 25-dimensional observation vector provides comprehensive robot configuration and environment state information

3.1.2 Action Space:

- **Dimensionality**: 4-dimensional continuous action space with range $[-1, 1]$
- **Action Mapping**: Actions are mapped to joint velocity commands for robot actuators
- **Constraints**: Bounded continuous control with smooth transitions for stable operation

3.1.3 Reward Structure:

- **Reward Type**: Sparse rewards with -1 per timestep until goal achievement (0 upon success)
- **Success Criterion**: Task successful when Euclidean distance between `achieved_goal` and `desired_goal` is less than 0.05 units
- **Episode Rewards**: Range from -50 (timeout) to 0 (immediate success)
- **Challenge**: Sparse rewards provide minimal feedback during exploration

3.1.4 Episode Characteristics:

- **Episode Length**: Maximum 50 timesteps per episode
- **Reset Conditions**: Episodes terminate upon task completion or timeout
- **Goal Sampling**: Goals randomly sampled within robot's reachable workspace
- **Initial Conditions**: Robot starts from random initial configuration

3.2 Discretization Techniques

This study deliberately maintains continuous state and action spaces without discretization. This choice was made because: (1) real robotic systems operate in continuous spaces, making this more representative of deployment scenarios; (2) robotic manipulation requires fine-grained control precision that discretization would eliminate; (3) SAC and TD3 are designed for continuous control and perform optimally in continuous spaces; (4) continuous actions enable smooth robot movements.

The 25-dimensional state space would require an intractably large discrete representation, and discretization would fundamentally change the problem characteristics while reducing real-world applicability.

3.3 Algorithm Selection and Rationale

3.3.1 Primary Algorithms: SAC and TD3

Soft Actor-Critic (SAC): SAC was selected due to its advantages for robotic learning:

Technical Characteristics:

- **Off-policy Actor-Critic**: Enables sample-efficient learning from stored experiences
- **Maximum Entropy Framework**: Encourages exploration through entropy regularization
- **Automatic Entropy Tuning**: Dynamically adjusts exploration-exploitation balance
- **Twin Q-Networks**: Reduces overestimation bias in value function learning

Implementation Details:

- Network Architecture: 256-unit hidden layers for actor and critic networks
- Learning Rate: 3×10^{-4} with Adam optimizer
- Replay Buffer: Stores up to 200,000 transitions
- Target Networks: Soft updates with $\tau = 0.005$

Twin Delayed Deep Deterministic Policy Gradient (TD3): TD3 was chosen to provide comparative analysis with SAC:

Technical Characteristics:

- **Deterministic Policy**: Learns deterministic state-to-action mapping
- **Twin Q-Networks**: Uses two critic networks to reduce overestimation bias
- **Delayed Policy Updates**: Updates policy less frequently than critics for stability
- **Target Policy Smoothing**: Adds noise to target actions for robustness

Implementation Details:

- Network Architecture: 256-unit hidden layers for actor and critic networks
- Exploration Noise: Gaussian noise with $\sigma = 0.1$ during training
- Policy Update Frequency: Every 2 iterations (delayed updates)
- Target Policy Smoothing: Noise with clipping for robust computation

3.3.2 Hindsight Experience Replay (HER)

HER was integrated with both algorithms to address sparse reward challenges in goal-conditioned tasks [1]. The mechanism retrospectively treats failed episodes as successful for alternative goals, creating multiple training samples per episode and an implicit curriculum from easier to harder goals.

Goal Selection Strategies:

1. **Final Strategy**: Uses episode's final achieved state as alternative goal, guaranteeing achievable relabeled transitions
2. **Future Strategy**: Selects $k = 4$ random future states as alternative goals, balancing diversity with achievability
3. **Random Strategy**: Randomly selects $k = 4$ states from entire episode, maximizing goal diversity

Technical Implementation: HER samples are added to the standard replay buffer alongside original experiences. Distance-based rewards are computed for alternative goals (0 if distance < 0.05, -1 otherwise) using 3D Cartesian coordinates for goal representation.

3.4 Experimental Design Rationale

The experimental design enables systematic comparison across base algorithm performance (SAC vs TD3), HER effectiveness, goal selection strategies, and convergence characteristics. Random search hyperparameter optimization was employed rather than fixed parameters to ensure fair comparison, with carefully selected ranges based on literature recommendations and results based on best-performing configurations for each algorithm.

This methodology provides a comprehensive comparison of state-of-the-art algorithms for goal-conditioned robotic manipulation, focusing on the impact of different goal selection strategies in sparse reward environments.

4 Evaluation & Visualizations - Bowling

To evaluate the performance of each agent, we conducted both quantitative and visual analyses of the training process. Figure 1 presents a comparison of the cumulative rewards per episode for all three algorithms: Q-Learning, DQN, and PPO. This overview allows us to observe global performance trends and convergence behavior. This plot clearly demonstrates that PPO achieved the highest and most stable performance, maintaining rewards consistently in the 55–60 range across most episodes. The curve is smooth and relatively flat, indicating that the agent converged quickly and maintained a strong, reliable policy. In contrast, Q-Learning and DQN showed more volatile behavior, with frequent fluctuations in episode rewards and slower convergence.

Additionally, Figure 2 shows a histogram of the rewards achieved in the final 50 episodes of training, providing insight into the consistency and reliability of each agent's policy once training had stabilized. The PPO agent produced a narrow and concentrated distribution of high rewards (above 55) in the final episodes. In contrast, Q-Learning and DQN exhibited wider distributions with significantly lower average scores. This reinforces the conclusion that PPO not only achieved the best average performance but also did so with high consistency.

Looking at the individual training curves, Figure 3 illustrates the reward progression for the Q-Learning agent. While it showed some signs of improvement, the learning curve was highly unstable, and the moving average stayed between 22 and 30 throughout most of training. This suggests that tabular Q-Learning, even with discretization, struggled to deal with the high-dimensional input of the Bowling environment.

In the Figure 4, which presents the reward per episode during DQN training, shows a more gradual improvement. The moving average increased steadily and eventually stabilized around 30–35. This result highlights the benefits of using convolutional networks for function approximation in pixel-based environments, though the convergence speed and final performance were still inferior to PPO.

Figure 5 shows the training performance of the PPO agent. Its learning curve is the most stable of the three, with minimal variance and high final rewards. The

agent converged early in training and maintained optimal performance, confirming that PPO was the most effective algorithm for this task.

In summary, PPO outperformed both DQN and Q-Learning in this environment, showing faster convergence, higher reward stability, and greater reliability. DQN was moderately effective and showed potential for further improvement, while Q-Learning proved inadequate for the complexity of visual input in ALE/Bowling, even when discretized.

5 Evaluation & Visualizations - Panda

5.1 Performance Metrics

5.1.1 Success Rate Analysis

Figure 6 and Figure 7 reveal that all algorithm variants achieve similarly poor final success rates of approximately 10%. While temporary performance peaks of up to 30% occur during training for some variants, these improvements are not sustained, and all algorithms converge to minimal task performance.

The success rate curves show high volatility across all variants, with frequent drops to 0% success, indicating unstable learning and inability to maintain consistent performance.

5.1.2 Episode Reward and Length Analysis

Episode reward patterns in both figures demonstrate high variance across all algorithms, with rewards fluctuating between -30 and -50 without clear learning progression. No algorithm shows sustained improvement in reward acquisition over the training period.

Episode length patterns remain consistently around $40 - 50$ steps for all variants, suggesting that most episodes reach the time limit without task completion, regardless of the algorithm or HER strategy employed.

5.1.3 Convergence Analysis

Only TD3+HER(Final) demonstrates measurable convergence speed (approximately 13 evaluation steps), but this convergence leads to the same poor final performance ($\sim$ 10% success) as other algorithms. Most algorithm variants show no meaningful convergence within the training budget.

5.2 Comparative Analysis of Algorithms

5.2.1 HER Strategy Analysis

Contrary to literature expectations[1], HER does not provide significant performance improvements in our experimental setup:

- **No HER advantage**: HER variants perform similarly to baseline algorithms

- **Goal selection irrelevant**: Final, Future, and Random strategies yield equivalent poor results
- **Strategy failure**: No goal selection strategy enables sustained learning

5.2.2 Algorithm Characteristics

SAC vs TD3 Comparison:

- **Similar final performance**: Both achieve $\sim 10\%$ success rates
- **TD3 convergence**: Shows faster initial convergence but to poor performance levels
- **High volatility**: Both algorithms demonstrate unstable learning curves

5.3 Limitations and Challenges

5.3.1 Task Difficulty

The PandaReach-v3 environment proves extremely challenging for all tested RL approaches, with no algorithm achieving meaningful task performance above 10% success rate. This suggests either:

- The task complexity exceeds current algorithm capabilities
- Insufficient training time for meaningful learning
- Suboptimal hyperparameter selection across all variants

5.3.2 HER Implementation Challenges

The failure of HER to provide expected improvements indicates potential issues with:

- **Goal representation**: The 3D Cartesian goal representation may be insufficient
- **Reward structure**: The sparse binary reward may not provide adequate learning signals even with HER
- **Implementation details**: Technical aspects of HER integration may require refinement

5.3.3 Experimental Limitations

Limited Random Search Trials: Only 2 trials per algorithm variant were conducted during random search, which is insufficient for robust hyperparameter optimization. This limited exploration of the hyperparameter space may have prevented discovery of optimal configurations that could enable successful learning.

Hyperparameter Sensitivity: High variance across the limited trials suggests that optimal hyperparameters may not have been discovered through the constrained random search, potentially masking true algorithm capabilities. More extensive hyperparameter exploration with additional trials would be necessary to draw definitive conclusions about algorithm performance.

Sample Efficiency: The sparse reward structure combined with high-dimensional continuous control may require significantly more training data than provided.

5.3.4 Methodological Considerations

Environment Complexity: The PandaReach-v3 task may represent a complexity threshold that current methods cannot readily overcome without additional techniques such as:

- Curriculum learning approaches
- Demonstration-based initialization
- Advanced exploration strategies
- Reward shaping techniques

Evaluation Metrics: The focus on success rate may not capture incremental learning progress that could indicate algorithm potential with extended training.

6 Conclusion

This study provided a comprehensive evaluation of reinforcement learning algorithms across two fundamentally different environments, offering insights into algorithm performance characteristics and task-specific challenges.

6.1 Key Findings

6.1.1 ALE/Bowling Environment

Our evaluation of discrete-action visual learning demonstrated clear performance hierarchies among the tested algorithms. PPO emerged as the superior method, achieving stable and high performance (55-60 reward range) with minimal variance and rapid convergence. This success can be attributed to PPO's policy gradient approach with clipped surrogate objectives, which proved well-suited for the sparse reward structure and visual complexity of the Bowling environment.

DQN showed moderate effectiveness with gradual improvement and convergence around 30-35 average rewards, highlighting the benefits of convolutional neural networks for processing pixel-based observations compared to tabular methods. However, its performance remained substantially below PPO levels.

Q-Learning, despite discretization efforts, proved inadequate for the high-dimensional visual input, achieving only 22-30 average rewards with high instability. This demonstrates the limitations of tabular methods in complex visual environments, even when state space reduction techniques are applied.

6.1.2 PandaReach-v3 Environment

The continuous control robotic manipulation task revealed significant challenges for current state-of-the-art RL methods. All tested algorithms—SAC, TD3, and their HER variants—achieved disappointingly uniform poor performance with approximately 10% success rates. This outcome was unexpected, particularly for HER, which is specifically designed to address sparse reward goal-conditioned learning problems.

The failure of HER to provide anticipated improvements suggests several possibilities: (1) the task complexity may exceed current algorithmic capabilities within the given training constraints, (2) implementation details may require further refinement, or (3) the specific characteristics of PandaReach-v3 may present unique challenges not adequately addressed by standard HER approaches.

Notably, while temporary performance peaks of up to 30% occurred during training for some variants (particularly SAC+HER(Final) and TD3+HER(Final)), these improvements were not sustained, indicating unstable learning dynamics and inability to maintain consistent performance.

6.2 Methodological Insights

The contrasting results between environments highlight the importance of algorithm-environment matching. Policy gradient methods (PPO) excelled in the visual discrete-action domain, while actor-critic methods designed for continuous control (SAC, TD3) struggled with the sparse reward robotic task despite theoretical suitability.

Our experimental design revealed methodological limitations, particularly the constraint of only 2 trials per algorithm variant during random search. This limited hyperparameter exploration may have prevented discovery of optimal configurations, especially for the challenging robotic manipulation task where extensive tuning is often critical for success.

6.3 Limitations and Future Work

Several limitations constrain the generalizability of our findings. The poor performance across all algorithms in the PandaReach-v3 environment suggests that 80,000 training timesteps may be insufficient for this task complexity, or that current methods require significant enhancement for reliable sparse reward learning.

Future research directions should include: (1) extended training periods with more comprehensive hyperparameter optimization, (2) investigation of curriculum learning approaches for gradual task complexity increase, (3) exploration of demonstration-based initialization methods, (4) development of more sophisticated goal selection strategies beyond the basic HER approaches tested, and (5) integration of advanced exploration techniques specifically designed for sparse reward environments.

6.4 Broader Implications

This study contributes to the growing understanding of RL algorithm performance across diverse task domains. The success of PPO in visual environments reinforces the value of policy gradient methods for complex observation spaces, while the challenges encountered in robotic manipulation highlight the continued difficulty of sparse reward learning despite algorithmic advances.

The uniform poor performance in PandaReach-v3 across all tested approaches suggests that goal-conditioned robotic manipulation remains a challenging frontier for current RL methods. This finding has important implications for practical robotics applications, indicating that successful deployment may require hybrid approaches combining RL with other techniques such as classical control, imitation learning, or explicit curriculum design.

6.5 Final Remarks

Our comparative analysis demonstrates that algorithm selection must be carefully tailored to environment characteristics, with no universal solution across different RL problem domains. While significant progress has been made in visual learning tasks, continuous control in sparse reward robotic environments continues to present substantial challenges requiring continued research and development.

References

[1] Marcin Andrychowicz et al. "Hindsight Experience Replay". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5048–5058. URL: https://papers.nips.cc/paper/7090-hindsight-experience-replay.

[2] The Farama Foundation. *ALE/Bowling Environment - Gymnasium Documentation*. Accessed: 2025-06-14. 2024. URL: https://ale.farama.org/environments/bowling/.

[3] Quentin Gallouedec. *Panda-Gym: A set of robotic environments using PyBullet and OpenAI Gym*. https://github.com/qgallouedec/panda-gym/. Accessed: 2025-06-14. 2022.

A Annexes

Figure 1: Comparison of all algorithms

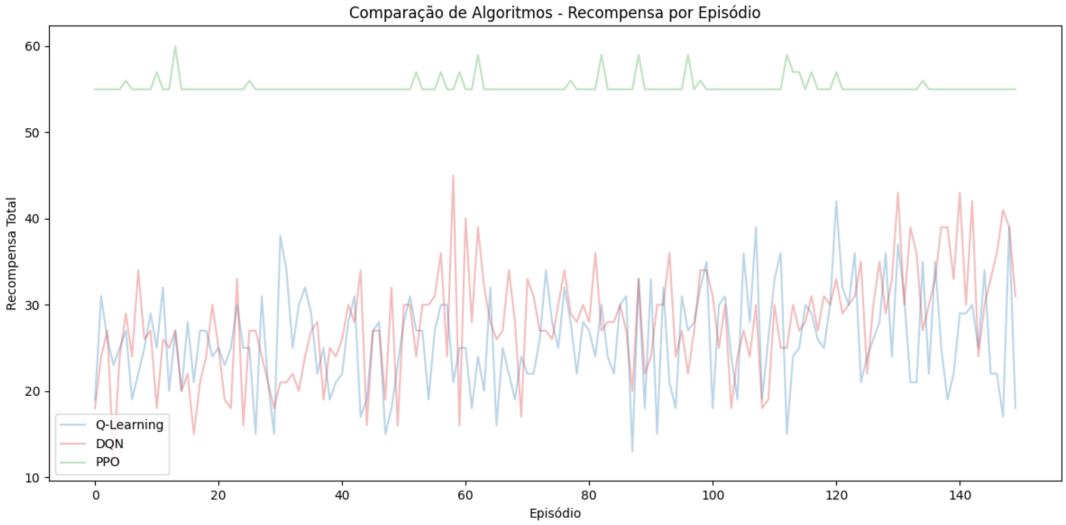

Figure 2: Histogram of all algorithms

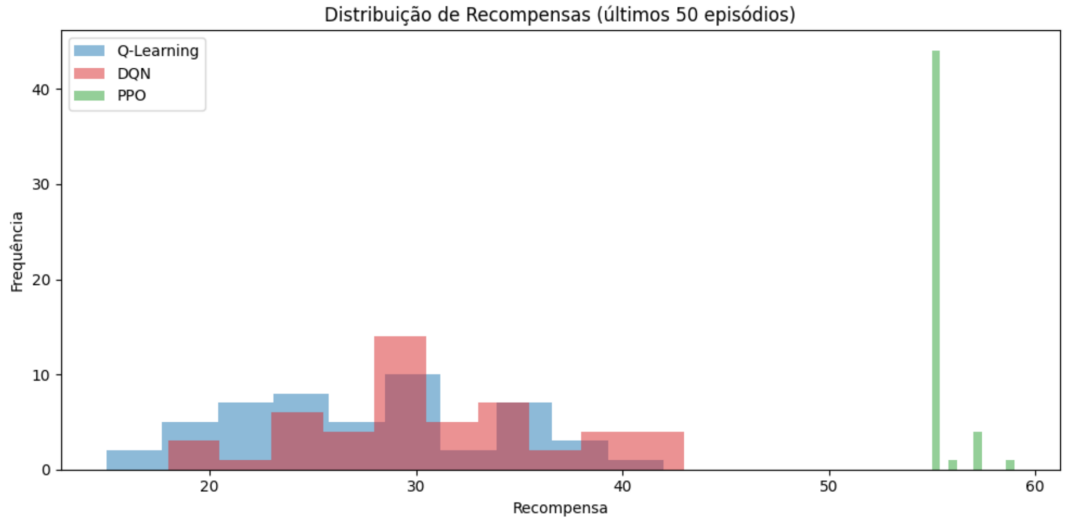

Figure 3: Train of Q_learning

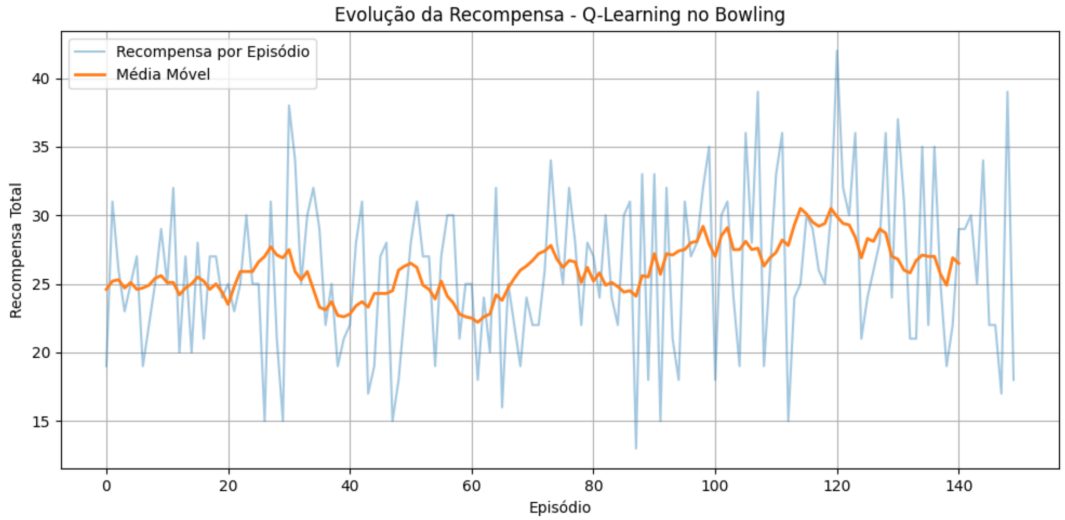

Figure 4: Train of DQN

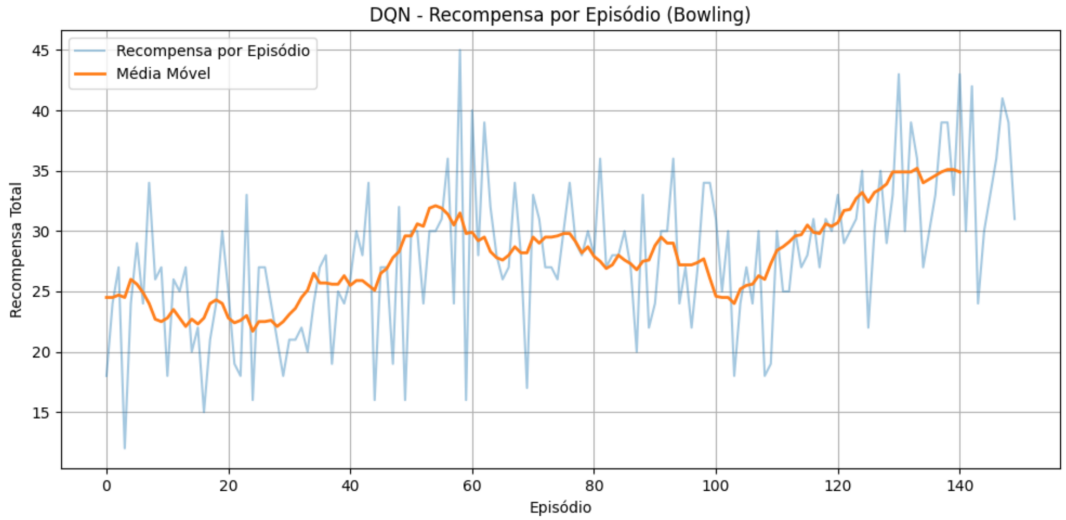

Figure 5: Train of PPO

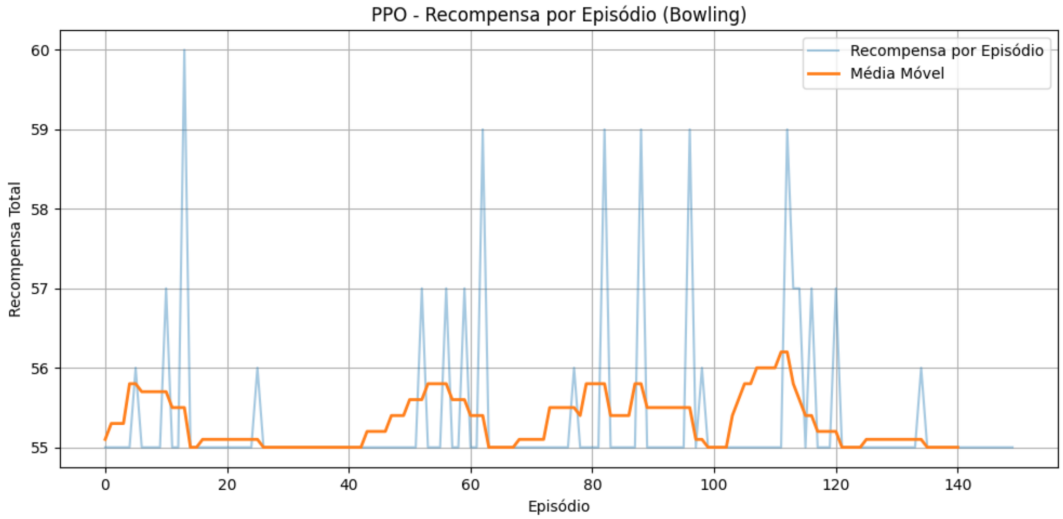

Figure 6: Train of SAC

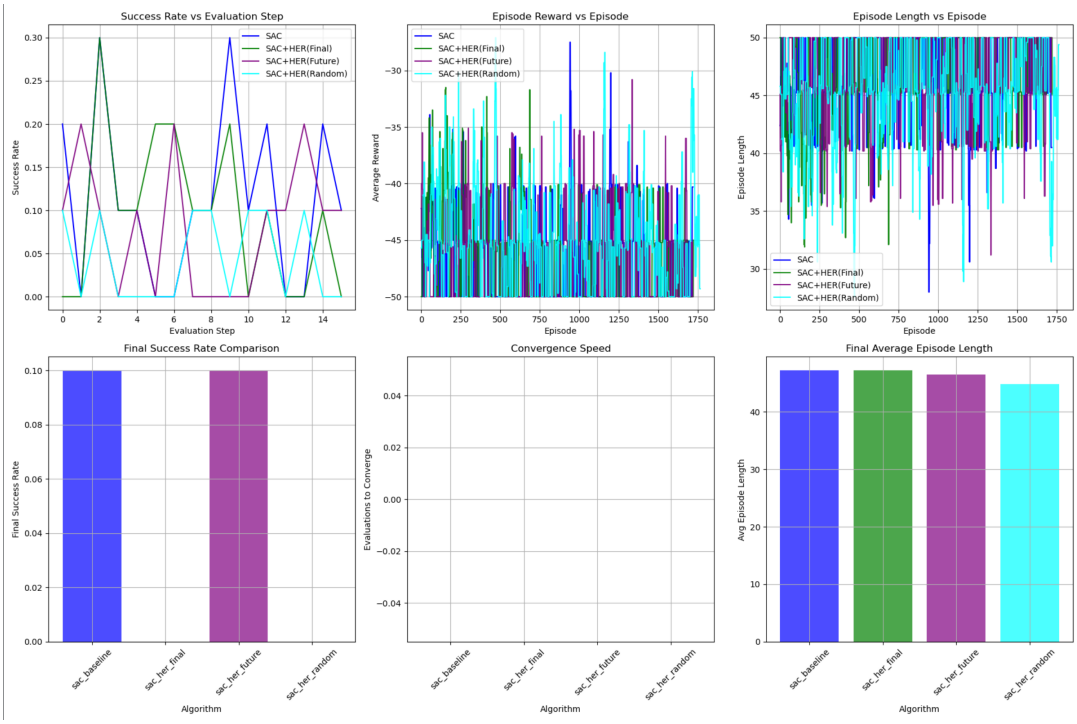

Figure 7: Train of TD3