RL-Course 2025/26: Final Project Report

abcdef: Julian Jurcevic

February 16, 2026

1 Introduction

Reinforcement Learning (RL) enables agents to learn decision-making strategies through interaction with an environment. At each time step, the agent observes the current state, selects an action, and receives a scalar reward. The objective is to maximize the expected cumulative reward over time. Continuous control tasks are particularly challenging due to high-dimensional action spaces and instability in value estimation.

In this project, we consider a competitive air hockey environment implemented in Gymnasium [2] and based on a physics simulation using Box2D [1]. Two agents compete in a zero-sum setting. Each player controls translation, rotation, and shooting through continuous actions. The observation space consists of positional and velocity information for both players and the puck. Episodes terminate when a goal is scored or when a time limit is reached.

The agent is trained by maximizing the reward signal provided by the environment. The reward includes goal outcomes and additional shaping components. Performance is evaluated separately using the win rate against predefined weak and strong opponents.

The goals of this work are i) implement the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm for continuous control in the hockey environment.

2 Method

2.1 Twin Delayed Deep Deterministic Policy Gradients

We use Twin Delayed Deep Deterministic Policy Gradients (TD3) [3], an off-policy actor–critic algorithm for continuous control.

TD3 employs two critics, Q_{ϕ_1} and Q_{ϕ_2}, and computes the Bellman target via clipped double Q-learning:

$$y = r + \gamma(1 - d) \min_{i=1,2} Q_{\phi'_i}(s', \tilde{a}').$$

Taking the minimum reduces overestimation bias.

Target policy smoothing perturbs the target action:

$$\tilde{a}' = \text{clip}\big(\pi_{\theta'}(s') + \epsilon, -c, c\big), \quad \epsilon \sim \mathcal{N}(0, \sigma^2),$$

which regularizes the critic by smoothing sharp value peaks.

Critics minimize the Bellman error

$$\mathcal{L}(\phi_i) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}\big[(Q_{\phi_i}(s, a) - y)^2\big].$$

The actor is updated less frequently (delayed updates) and optimized to maximize $Q_{\phi_1}(s, \pi_\theta(s))$. Target networks are updated via Polyak averaging:

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi'.$$

2.2 Replay and Exploration

Transitions (s, a, r, s', d) are stored in a replay buffer to enable off-policy learning.
We consider both uniform and prioritized sampling, where transitions are drawn proportional to their TD error. Importance weighting is applied to reduce sampling bias.
Exploration is performed in action space:

$$a = \pi_\theta(s) + \epsilon.$$

We evaluate Gaussian, Ornstein–Uhlenbeck, and temporally correlated pink noise. An initial random phase populates the replay buffer.

2.3 Noise Annealing

To improve stability, exploration noise is annealed over training:

$$\sigma_t = \max\left(\sigma_0(1 - \tfrac{t}{T}), \sigma_{\min}\right).$$

This enables broad exploration early and increased exploitation later.

2.4 Self-Play and Opponent Scheduling

Training is conducted in a two-player setting. We combine fixed scripted opponents (weak and strong) with self-play.
Policy snapshots are stored periodically in a finite pool and sampled as opponents during training. Opponent probabilities are scheduled over time: early training emphasizes weak opponents, while later stages increase strong and self-play opponents.
This curriculum improves robustness and reduces overfitting to fixed behaviors.

2.5 Network Architecture

Actor and critics are fully connected networks with two hidden layers of 256 units and $\tanh$ activations. The actor outputs actions in $[-1, 1]$, and critics receive concatenated state–action pairs.

3 Experimental Results

References

[1] E. Catto. Box2d physics engine documentation. https://box2d.org/documentation/md_simulation.html, 2023. Accessed: 2026-02-15.

[2] Farama Foundation. Gymnasium: A standard api for reinforcement learning environments. https://github.com/Farama-Foundation/Gymnasium, 2023. Accessed: 2026-02-15.

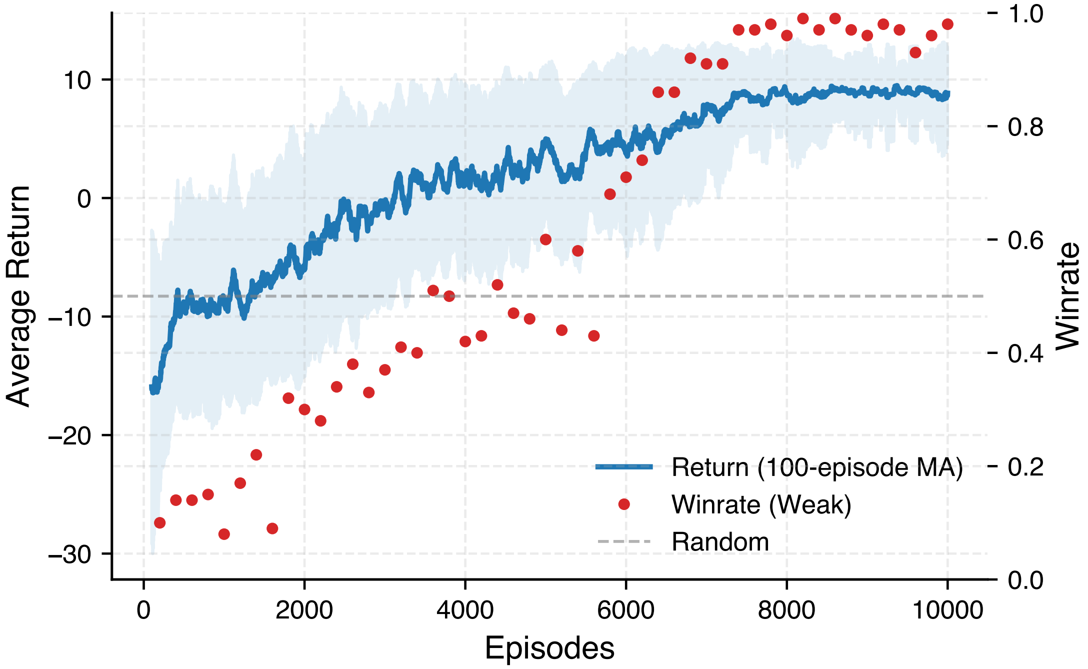

Figure 1: Training performance against the weak opponent. The blue curve shows the 100-episode moving average return with one standard deviation shading. Red markers denote evaluation win rate measured every 200 episodes. The dashed line indicates random performance.

[3] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.