RL-Course 2025/26: Final Project Report

alphabet: Julian Jurcevic

February 18, 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 sampled proportional to their TD error

$$\delta = Q_\phi(s, a) - y,$$

measuring the discrepancy between the current estimate and the Bellman target. Importance weighting corrects the induced sampling bias.
Exploration is performed in action space:

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

Here, π_θ denotes the deterministic actor with parameters θ. 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

3.1 Experimental Setup

We conduct three experiments: (i) curriculum training for the final competition agent, (ii) exploration noise comparison, and (iii) self-play and prioritized replay analysis.
For controlled comparisons, we use the Stage II setup and vary only the component under study.

Stage	Phase	Strong	Weak	Self-Play
I (10k)	All	0.00	1.00	0.00
II (25k)	Early	0.55	0.45	0.00
	Mid	0.45	0.45	0.10
	Late	0.50	0.40	0.10
III (12k)	Early	0.30	0.70	0.00
	Mid	0.60	0.30	0.10
	Late	0.35	0.35	0.30

Table 1: Piecewise opponent scheduling across curriculum stages.

Noise Type	WR Weak (%)	WR Strong (%)	Return Weak	Return Strong
Gaussian	92.50 ± 4.48	81.00 ± 0.47	8.22 ± 0.80	5.69 ± 0.10
Ornstein–Uhlenbeck	$\mathbf{94.67 \pm 0.58}$	$\mathbf{89.00 \pm 2.65}$	$\mathbf{8.56 \pm 0.13}$	$\mathbf{7.06 \pm 0.46}$
Pink	92.56 ± 4.30	86.11 ± 2.84	8.17 ± 0.57	6.40 ± 0.34
Uniform	91.22 ± 2.84	80.22 ± 8.39	8.10 ± 0.55	5.51 ± 1.51

Table 2: Exploration noise ablation (mean $\pm$ std across seeds).

During training, evaluation is performed every 200 episodes with 100 games against the weak and 100 against the strong opponent. Plots show these evaluations for a representative single-seed run.
For Tables 2 and 3, the best checkpoint of each run (highest evaluation win rate) is re-evaluated separately. Results report mean and standard deviation over three seeds.

3.2 Curriculum Training

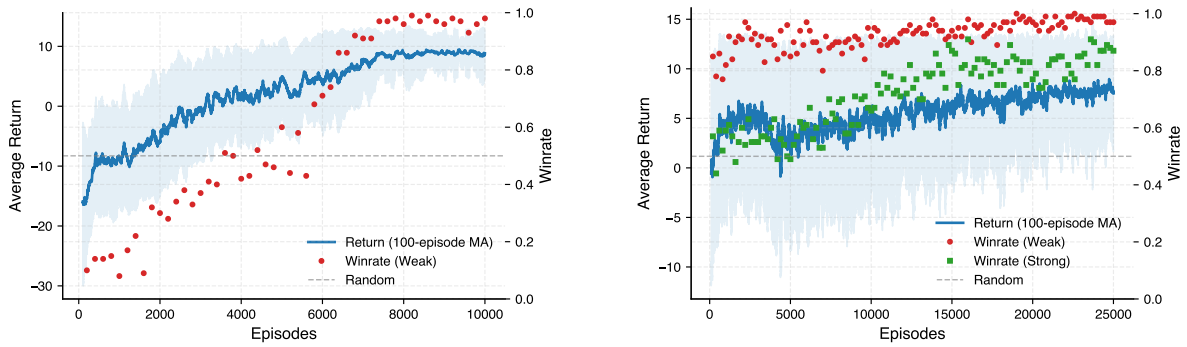
Curriculum training is used to obtain a versatile competition agent that performs robustly against weak, strong, and previously unseen opponents.
Training proceeds in three phases (Table 1). Stage I trains exclusively against the weak opponent to establish reliable puck control. Stage II introduces a mixture of weak and strong opponents with occasional self-play. Stage III further increases exposure to strong and self-play opponents to improve generalization.
Figure 1 shows that weak-only training quickly improves performance but does not transfer to the strong opponent. The curriculum progressively improves performance against stronger opponents while retaining high win rates against the weak baseline. Stage III yields the strongest overall competition-ready policy (Figure 2).

3.3 Exploration Noise Comparison

To analyze the impact of exploration noise, we compare Gaussian, Ornstein-Uhlenbeck, pink, and uniform noise under identical training conditions. All variants use the Stage II curriculum and identical hyperparameters, and only the exploration process differs.

3.4 Effect of Self-Play and Prioritized Replay

Table 3 shows that self-play does not improve performance against the fixed weak and strong opponents. However, the degradation remains moderate and performance stays within a comparable range. Since the



(a) Stage I: Weak-only training

(b) Stage II: Curriculum training

Figure 1: Training progression during initial and curriculum stages. Blue: 100-episode moving average return. Red/green: evaluation win rate.

Variant	WR Weak (%)	WR Strong (%)	Return Weak	Return Strong
No PER, No Self-Play	**93.07 ± 3.75**	**78.27 ± 3.07**	**8.33 ± 0.66**	**5.00 ± 0.70**
No PER, Self-Play	90.73 ± 5.90	72.60 ± 7.63	7.62 ± 1.26	4.06 ± 1.56
PER, No Self-Play	75.80 ± 9.18	66.07 ± 4.69	4.22 ± 2.00	1.99 ± 1.04
PER, Self-Play	78.27 ± 2.23	65.33 ± 5.14	4.71 ± 0.54	1.78 ± 1.01

Table 3: Final evaluation results (mean ± std over three random seeds).

final tournament also includes matches against previously unseen student agents, self-play is retained. It is expected to improve robustness and reduce overfitting to the fixed scripted opponents.

PER was implemented to focus updates on transitions with high TD error. However, in our setting it consistently reduced stability and final performance. We attribute this to increased variance and overfitting to rare high-error transitions in the competitive environment. Therefore, PER is not used for the final competition agent.

Rerun 10k weak with weak and strong evaluation!

Rerun Experiment using Stage 2!

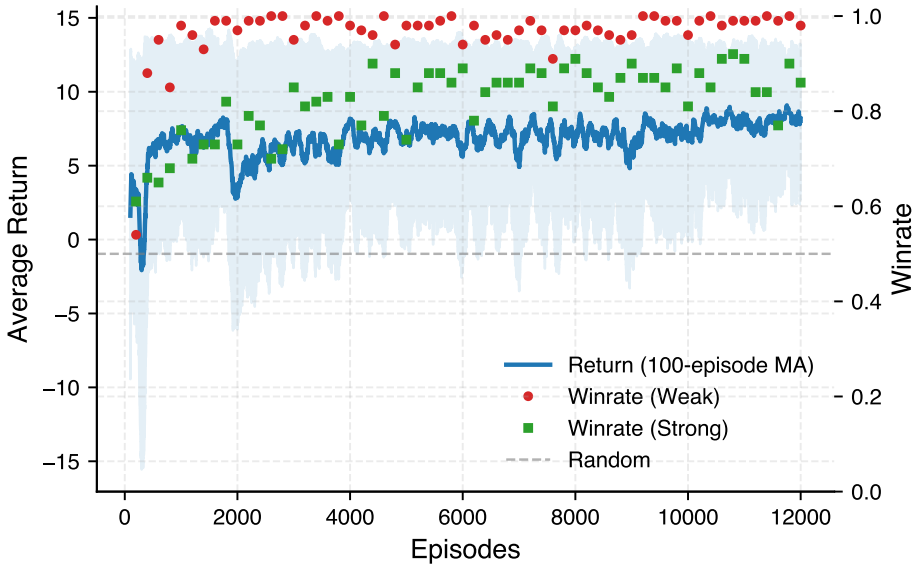


Figure 2: Stage III: Final curriculum refinement.

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.

[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.