RL-Course 2025/26: Final Project Report

alphabet-td3: Julian Jurcevic

February 20, 2026

1 Introduction

Two-player reinforcement learning is challenging due to non-stationarity: as the opponent improves, the learning target changes. In continuous-control air hockey, this difficulty is amplified by fast dynamics, sparse terminal rewards, and the need to combine positioning, shooting, and defense within a single policy. Policies that overfit to one opponent often fail to generalize, which is critical in competitive settings.

We study a two-player continuous air hockey environment implemented using the Gymnasium API [4]. In this setting, two agents control paddles via four continuous actuators: translation in x and y, rotation, and shooting. The 18-dimensional state encodes positions, velocities, and angles of both players and the puck. Episodes end after a goal or 250 steps. Terminal rewards are ± 10, supplemented by shaped rewards for puck proximity and direction.

We apply Twin Delayed Deep Deterministic Policy Gradient (TD3) [5], an off-policy actor–critic method for continuous control. We make the following contributions:

- A empirical comparison of four exploration noise processes (Gaussian, Ornstein–Uhlenbeck, Pink, and Uniform).
- A three-stage curriculum strategy that combines scripted opponents with self-play to mitigate non-stationarity and improve generalization.
- An ablation study of prioritized replay and self-play, analyzing their impact on stability, robustness, and benchmark performance.

The primary objective is to develop a competitive agent for the final tournament. The full source code and training setup are publicly available [6].

2 Method

2.1 Twin Delayed Deep Deterministic Policy Gradients

We use TD3, 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

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

The actor is optimized via the deterministic policy gradient objective

$$\mathcal{L}_{actor}(\theta) = -\mathbb{E}_{s\sim\mathcal{D}}\left[Q_{\phi_1}(s, \pi_\theta(s))\right]$$

where gradients are taken with respect to the actor parameters θ. The actor is updated less frequently than the critics (delayed policy updates).
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 [8], where transitions are sampled proportional to their TD error

$$\delta_i = Q_{\phi_i}(s,a) - y, \quad i \in \{1,2\}$$

measuring the discrepancy between the current estimate and the Bellman target. In the TD3 setting, priorities are computed from the mean absolute TD error across both critics. Sampling bias is partially corrected via importance weighting.
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 [7], and pink noise [3]. 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 [1].
Policy snapshots are stored every k episodes in a fixed-size pool of size N_{pool}. When the pool exceeds N_{pool}, the oldest snapshot is discarded. Opponents are sampled proportionally to a difficulty score updated after each game: the score increases by factor 1.2 after a loss and decays by factor 0.95 after a win. This biases training toward opponents the agent currently struggles against. We set k=150 and N_{pool}=25 for Stage II and III.
Opponent probabilities are scheduled over time: early training emphasizes weak opponents, while later stages increase strong and self-play opponents. The scheduling proportions were determined empirically by monitoring win rates against both opponents during training. Stage transitions were triggered when weak win rates plateaued above 85%, indicating readiness for stronger opposition. The exact proportions reflect manual tuning rather than systematic optimization, which is a limitation of this work.

Parameter	**Value**
Discount factor γ	0.99
Actor / Critic LR	$2 \cdot 10^{-4}$
Target update τ	0.005
Policy update frequency	2
Batch size	256
Replay buffer size	300k
Target noise (scale / clip)	0.2 / 0.3
Exploration noise	OU (default; best in ablation)
Hidden units	2 layers × 256

Table 1: Key hyperparameters of the final TD3 agent.

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. Unless stated otherwise, all experiments share the same TD3 configuration summarized in Table 1.

During training, evaluation is performed every 200 episodes with 100 games against the weak and 100 against the strong opponent. Model selection during training uses $\min(\text{WR}_{\text{weak}}, \text{WR}_{\text{strong}})$ as score. This enforces robustness by penalizing policies that overfit to a single opponent. Plots show these evaluations for a representative single-seed run.

For Tables 3 and 4, the best checkpoint of each run (highest evaluation win rate) is re-evaluated separately. Results report mean and standard deviation over three seeds. Due to the limited number of seeds, observed differences should be interpreted as empirical trends rather than statistically verified effects.

3.2 Curriculum Training

Curriculum training [2] is used to obtain a versatile competition agent that performs robustly against weak, strong, and previously unseen opponents.

Training proceeds in three phases (Table 2). 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 1a 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 1c).

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 2: 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	**94.67 ± 0.58**	**89.00 ± 2.65**	**8.56 ± 0.13**	**7.06 ± 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 3: Exploration noise ablation (mean ± std across seeds).

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. Ornstein–Uhlenbeck (OU) noise outperformed all alternatives, particularly against the strong opponent (89% vs. 81% for Gaussian). A plausible explanation is that the temporal correlation of OU noise produces smoother action trajectories. In a fast-paced physics environment, uncorrelated perturbations can disrupt otherwise stable motor sequences. Pink noise, which also introduces temporal correlation, performed second-best, consistent with this hypothesis. Uniform noise showed the highest variance across seeds, suggesting unstable exploration behavior.

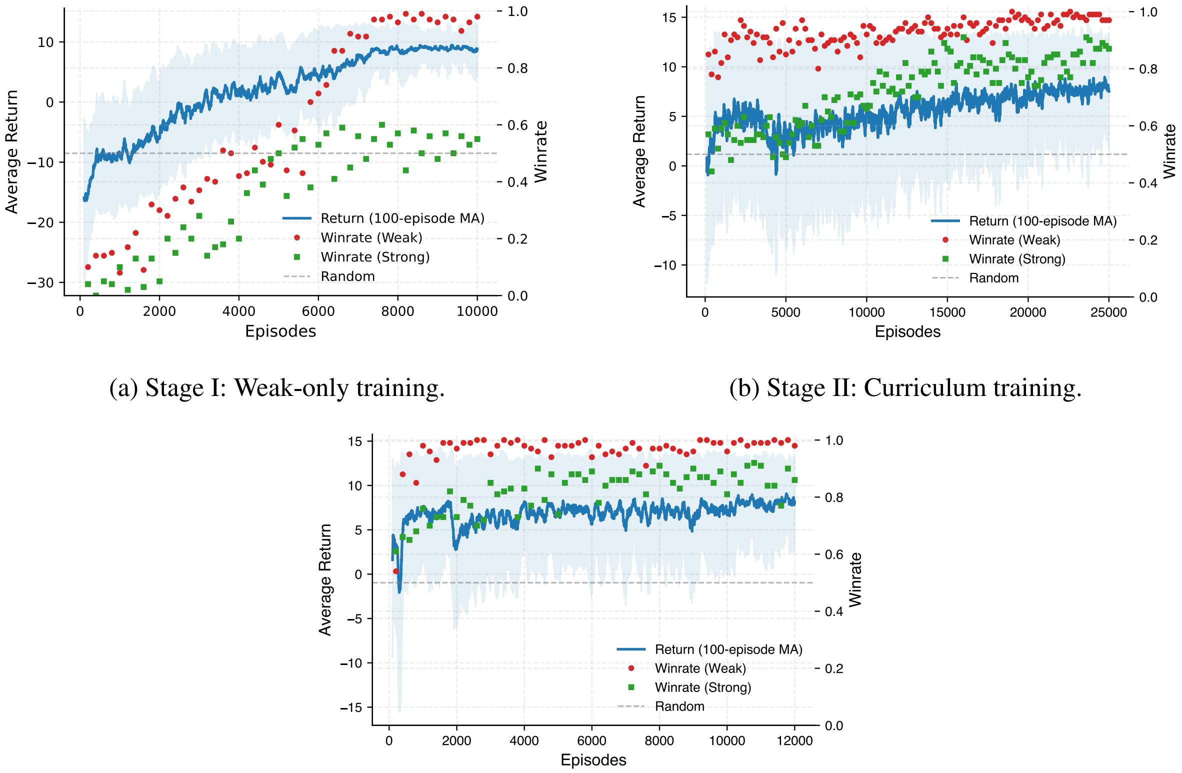
3.4 Effect of Self-Play and Prioritized Replay

Self-play slightly reduced performance against fixed scripted opponents (Table 4), as the policy adapts to past versions of itself rather than benchmark bots. Difficulty-weighted sampling limits exploitation of weak snapshots, but pool diversity decreases as policies converge, explaining diminishing returns. Population-based training could mitigate this. Despite lower benchmark scores, self-play is retained to improve robustness against unseen tournament opponents.

PER was implemented to focus updates on transitions with high TD error. However, in our setting it consistently reduced stability and final performance. This may be due to increased update variance and sensitivity to rare high-error transitions in the competitive setting. Therefore, PER is not used for the final competition agent.

3.5 Limitations

All training plots reflect single-seed runs due to computational constraints, limiting the interpretability of curve-level comparisons. The three-seed evaluation in Tables 3 and 4 provides more reliable estimates, but statistically significant conclusions would require additional seeds. Future work could explore adaptive opponent scheduling and population-based self-play to further improve generalization.



(a) Stage I: Weak-only training.

(b) Stage II: Curriculum training.

(c) Stage III: Final curriculum refinement.

Figure 1: Training progression across curriculum stages. Blue: return (100-episode MA). Red/green: 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 4: Final evaluation results (mean ± std over three random seeds).

4 Discussion and Conclusion

Curriculum learning was the most impactful design choice. Without it, the agent overfits to weak-opponent strategies (Figure 1a). Staged scheduling resolves this while retaining base competence.

OU noise yielded the clearest gain (+8% against strong, Table 3), likely due to temporally correlated exploration producing smoother actions suited to fast physics dynamics. PER consistently hurt performance (Table 4), possibly because non-stationary opponents amplify the variance of priority-based sampling. Self-play reduced benchmark scores but is retained for tournament robustness; pool diversity remains a limitation as snapshots converge over training.

Overall, TD3 with curriculum learning, OU noise, and self-play produces a stable, competitive agent for the air hockey environment.

References

[1] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations (ICLR)*, 2018.

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 41–48. ACM, 2009.

[3] O. Eberhard, J. Hollenstein, C. Pinneri, and G. Martius. Pink noise is all you need: Colored exploration for deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

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

[5] 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.

[6] J. Jurcevic. air-hockey-td3: Td3 with curriculum learning and self-play. `https://github.com/julilili42/air-hockey-td3`, 2026. GitHub repository.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[8] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.