RL-Course 2025/26: Final Project Report

Ansel Cheung, Jannik Mänzer, Niklas Abraham

February 5, 2026

Abstract

This report presents our application and comparative study of modern Reinforcement Learning (RL) algorithms—specifically, Twin Delayed Deep Deterministic Policy Gradient (TD3), Soft Actor-Critic (SAC), and the model based TD-MPC2—within the challenging Laser Hockey environment. We discuss the environment's design, its state and action spaces, and the unique characteristics making it a compelling testbed for RL research. Our work details the methodological approaches taken with these algorithms, summarizes key experimental findings, and reflects on lessons learned in this multi agent, continuous control setting.

1 Introduction

1.1 Environment Overview

The Laser Hockey environment [2] is a custom reinforcement learning benchmark built on the Gymnasium Python API and powered by the Box2D physics engine. In this multi agent setting, two agents each control a hockey stick and compete to score goals by striking a puck into the opponent's net. The environment is designed such that both agents receive identical but mirrored observations at each timestep, which eliminates the need for side specific learning strategies.

At each timestep, the agent receives an 18 dimensional continuous state vector that captures the complete game state. This observation includes the position x_1, y_1 and orientation θ_1 of both players relative to the center of the field, along with their linear $v_{x,1}, v_{y,1}$ and angular ω_1 velocities. The state also contains the puck's position x_p, y_p and velocity $v_{x,p}, v_{y,p}$, as well as time remaining for each player's puck possession $t_{\text{puck},1}, t_{\text{puck},2}$ in keep mode, which ranges from 0 to 15 steps. Player 1 refers to the agent currently being controlled, while Player 2 represents the opponent. The observation structure ensures that when the viewpoint switches during training, the indices are automatically mirrored so that each agent always perceives itself as Player 1, maintaining a consistent learning perspective.

The action space consists of a 4 dimensional continuous vector that controls the agent's stick. The first two components specify forces applied for stick movement in the x and y directions, while the third component controls the torque applied to adjust the stick's angle. The fourth component is a thresholded scalar that determines whether to release the puck when the agent is in possession. Reward feedback in this environment is sparse and goal oriented. An agent receives a reward of +10 for scoring a goaland 0 for a draw. Additionally, there is a small reward signal for maintaining proximity to the puck, which helps guide exploration during early learning stages. Each episode begins with all entities positioned at the center of the field and terminates when a goal is scored or a timeout occurs. The environment supports both scripted opponents, such as the BasicOpponent, and learning agents as adversaries, making it a versatile and challenging benchmark for continuous control and multi agent reinforcement learning research.

2 Method

2.1 TD-MPC 2 - Niklas Abraham

TD-MPC2 [1] is a model-based reinforcement learning algorithm that learns a world model to predict future states and rewards, and uses this model to select actions through planning. The core idea is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ in a Markov Decision Process with an infinite horizon. The policy is constructed to maximize the expected discounted return. In TD-MPC2, this is achieved by learning a world model and selecting actions by planning with the learned models.
For planning, TD-MPC2 employs the Model Predictive Control (MPC) framework, in which actions are optimized based on planning over action sequences of a finite horizon H:

$$\pi(s_t) = \arg\max_{a_1,\ldots,a_H} \mathbb{E}_\pi \left[\sum_{\tau=0}^{H} \gamma^\tau r(s_{t+\tau}, a_{t+\tau}) \right]. \tag{1}$$

The return of each trajectory is estimated by simulating action sequences through the learned world model. However, this approach often leads to only locally optimal policies. To address this limitation, TD-MPC2 additionally utilizes a value function to guide the planning process and improve the policy toward a more globally optimal solution.
Rather than predicting raw future observation states, TD-MPC2 learns to predict a maximally useful latent representation for accurately estimating the outcomes of action sequences. The algorithm is composed of five distinct neural network components that interact in a coordinated manner, as illustrated in Figure 1.

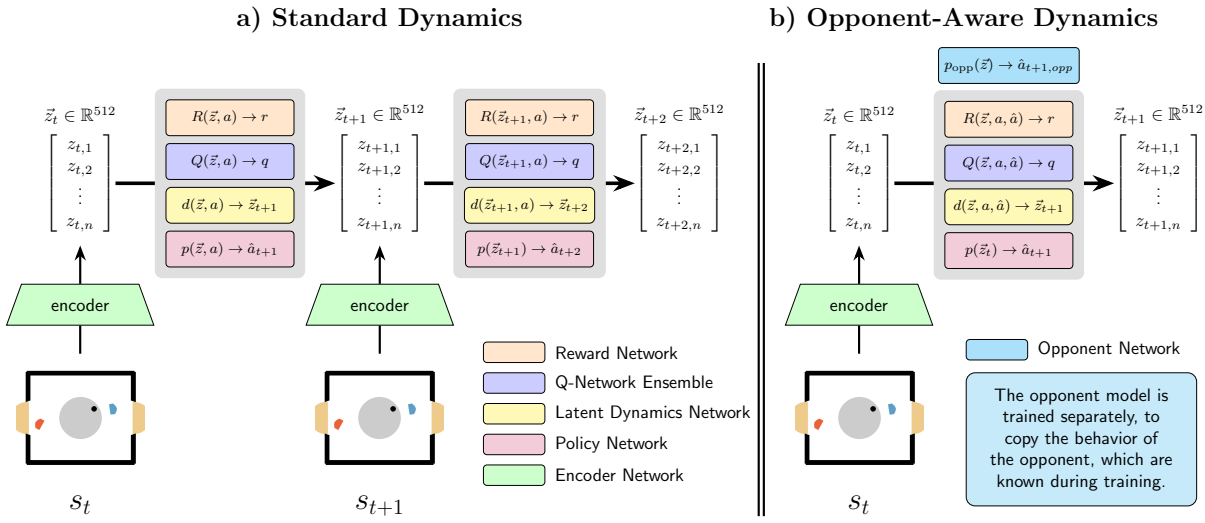

Figure 1: On the left in a) TD-MPC2 agent architecture and their individual components. b) shows the modified architecture with an additional opponent network to model the behavior of the adversarial agent in the Laser Hockey environment.

- **Encoder:** Maps the observed state s to a 512-dimensional latent vector $\vec{z} = h(s)$.
- **Latent Dynamics:** Predicts the next latent $\vec{z}_{t+1}$ from current latent and action: $\vec{z}_{t+1} = d(\vec{z}_t, a)$.
- **Reward Head:** Estimates reward r for a given $(\vec{z}, a)$ pair: $r = R(\vec{z}, a)$.

- **Termination Head:** Predicts early episode end, e.g., when a goal is imminent.
- **Q-Network Ensemble:** An ensemble (5 networks) of Q-functions estimating value $q = Q(\vec{z}, a)$. The minimum of two sampled networks reduces value overestimation.
- **Policy Network:** Guides action selection in planning: $p(\vec{z}, a) \to \hat{a}$.

2.1.1 Architecture and Training

All network components are multi-layer perceptrons (MLPs) with Mish activations. As in [1], the latent representation $\vec{z}$ is projected into L-dimensional simplices via a softmax to stabilize training and enforce sparsity.
Training uses an experience replay buffer $\mathcal{B}$ with full episode trajectories. Model parameters are optimized over sampled subsequences of length $H+1$ from $\mathcal{B}$ by minimizing a joint loss for dynamics, reward, and value prediction:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})_{t=0}^{H} \sim \mathcal{B}} \left[\sum_{t=0}^{H} \lambda^t \left(\|\vec{z}_{t+1} - \text{sg}(h(s_{t+1}))\|_2^2 + \text{CE}(\hat{r}_t, r_t) + \text{CE}(\hat{q}_t, q_t) \right) \right], \quad (2)$$

where $\text{sg}(\cdot)$ is stop-gradient, $\vec{z}_{t+1} = d(\vec{z}_t, a_t)$ is the predicted next latent, $\hat{r}_t = R(\vec{z}_t, a_t)$, $\hat{q}_t = Q(\vec{z}_t, a_t)$, and λ is a temporal discount factor. The Q-value target is $q_t = r_t + \gamma \bar{Q}(\vec{z}_{t+1}, p(\vec{z}_{t+1}, a_{t+1}))$, using an EMA of Q-net parameters ($\bar{Q}$) for stability. Following TD-MPC2, reward and value predictions are regressed in a log-transformed space with cross-entropy loss and soft targets.
The policy p is optimized according to a maximum entropy RL objective:

$$\mathcal{L}_p(\theta) = \mathbb{E}_{(s_t, a_t)_{t=0}^{H} \sim \mathcal{B}} \left[\sum_{t=0}^{H} \lambda^t \left(\alpha Q(\vec{z}_t, p(\vec{z}_t, a_t)) - \beta \mathcal{H}(p(\cdot|\vec{z}_t)) \right) \right], \quad (3)$$

where $\vec{z}_{t+1} = d(\vec{z}_t, a_t)$ with $\vec{z}_0 = h(s_0)$, and $\mathcal{H}(p(\cdot|\vec{z}_t))$ is the policy entropy. Hyperparameters α and β balance value maximization and entropy, preventing premature collapse to deterministic policies.

2.1.2 Planning with MPPI

For local planning, TD-MPC2 leverages Model Predictive Path Integral (MPPI) control [3], sampling action sequences with guidance from the policy network. At each step, it estimates $\mu^*, \sigma^* \in \mathbb{R}^{H \times m}$, the mean and standard deviation of a multivariate Gaussian that maximizes expected return:

$$\mu^*, \sigma^* = \arg\max_{\mu, \sigma} \mathbb{E}_{a_{t:t+H} \sim \mathcal{N}(\mu, \sigma^2)} \left[\gamma^H Q(\vec{z}_{t+H}, a_{t+H}) + \sum_{\tau=t}^{H} \gamma^\tau R(\vec{z}_\tau, a_\tau) \right]. \quad (4)$$

This is optimized by iteratively sampling actions from $\mathcal{N}(\mu, \sigma^2)$, evaluating their returns, and updating μ and σ based on weighted top samples. The termination model predicts early ends in sampled rollouts. To speed up convergence, a fraction of samples comes from the policy p, and μ, σ are initialized from the previous step.

3 Results

References

[1] N. Hansen, H. Su, and X. Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024.

[2] G. Martius. Hockey environment. https://github.com/martius-lab/hockey-env, 2023.

[3] G. Williams, A. Aldrich, and E. Theodorou. Model predictive path integral control using covariance variable importance sampling, 2015.