Adaptive Tactical Decision Making in Ice Hockey: Integrating Multi-agent Reinforcement Learning Framework With Advanced Computer Vision Techniques

Boyang Zhang, University of Turku, Finland

The European Conference on Education 2025
Official Conference Proceedings

Abstract

This study proposes an integrated framework that combines with Multi Agent Reinforcement Learning (MARL) with advanced computer vision techniques, including pose recognition via MMaction2, object detection with Yolov11, and multi-object tracking using ByteTrack. The McGill Hockey Player Tracking Dataset (MHPTD) is used to analysis the ice hockey team sport. In building the framework with MARL, each player is modeled as an autonomous agent whose observation space encompasses self-position, puck state, and the spatial locations of teammates and opponents. By incorporating inputs from the MHPTD dataset, the framework dynamically adapts strategic behaviors, like defending, attacking. When simulating realistic ice hockey scenarios, the framework can be used for strategy optimization using the group object detection, play behavior modeling by pose recognition and positional data tracking, advanced AI opponents' strategic analysis in future games. Reward function is setup to encourage agents to move towards the puck and getting rewards when shooting to the opponents' door. Multi-agent setup can simulate full team sport. The framework can enhance the simulation of complex player interactions, bridge the gap between MARL simulation and real-world fix-field team sports, provide insight in coaching and sport education.

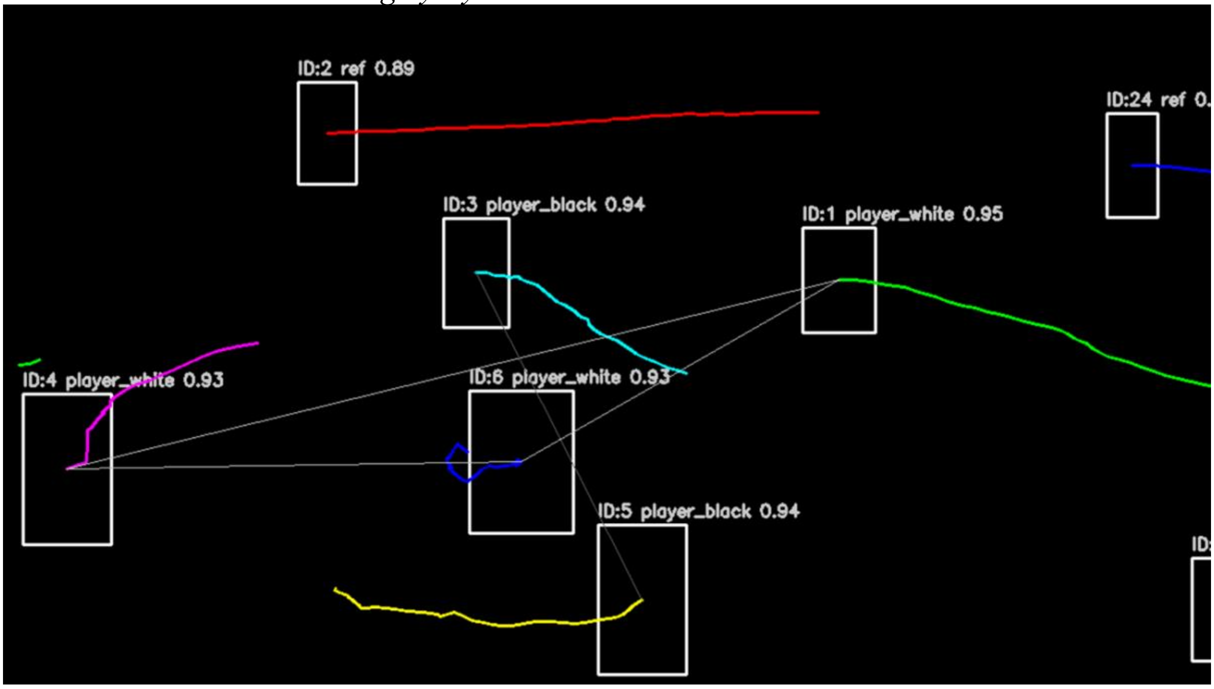
Keywords: multi agent reinforcement learning, object detection, pose detection

Introduction

With the development of science and technology, the application of artificial intelligence and machine learning in sport analytics have opened new opportunities for optimizing team dynamics, tactical strategies and individual player evaluation in sport education (Weber et al., 2022; B. Zhang, 2024). This research presents an integrated framework that utilizes Multi Agent Reinforcement Learning (MARL) with computer vision techniques including Yolov11 for object detection, MMAction2 for pose recognition, and ByteTrack for multi-object tracking. The framework applied to the McGill Hockey Player Tracking Dataset (MHPTD) in order to create an environment for simulation and testing the MARL.

The center of this research lies in agent-based modeling of players; each individual player can be conceptualized as a learning agent who is able to perceive and interact with other agents (Albrecht et al., 2024). Each agent is equipped with a space which represents the coordinate's locations, also the distance to the other players including teams and opponents, the puck. Therefore, by employing a Multi Agent Reinforcement Learning (MARL), agents learn to optimize the individual rewards, at the same time coordinate with team structures, reflect hocky strategies, such as power-play setups, transition plays.

Figure 1
Historical Movement Tracking by ByteTrack



In order to build the integrated framework, Yolov11 is utilized for object detection (Jocher & Qiu, 2024) to identify players from the same team and opponent team, the puck (ball), and the referee in each frame. ByteTrack links these frames across time to produce the historical movement tracking, continuous identifying the movement of objects (Y. Zhang et al., 2022). In the next step, MMaction2 is used to perform pose-based action recognition which is designed by MMaction2 Contributors (2020), to identify each individual player's movement including skating, passing, shooting positions. After the vision-based techniques, it enables reinforcement learning agents to perceive status information of the players' location, movement, and behavior.

Theoretical Background

Compare to traditional reinforcement learning, the concept of Multi Agent Reinforcement Learning (MARL) was introduced by Markov games, a generalization of Markov Decision Processes to multi agent settings (Littman, 1994), building upon this foundation, a structured theoretical overview of MARL by categorizing algorithms across Markov and extensive-form game frameworks, and analyzing task structure (cooperative, competitive, mixed) was introduced (K. Zhang et al., 2021) with decentralized learning over networks, mean-field approximations, convergence behavior of policy-gradient methods, and the influence of game structure on theoretical guarantees and algorithm design. Moreover, modern MARL theories has built connections with methodologies by covering theoretical models (e.g., Markov games, coordination graphs), learning paradigms (independent, centralized, decentralized, and partially observable settings), algorithmic developments (value-based, policy-gradient, actor-critic, and communication-based methods), and practical applications ranging from robotics and autonomous systems to complex multi-agent simulations (Albrecht et al., 2024). Practically, the usage of MARL supports full team simulations and facilities automated strategy optimization and opponent behavior analysis.

In MARL, the Multi Agent Deep Deterministic Policy Gradient (MADDPG) can address the non-stationarity problem, it adopts a centralized training with decentralized execution framework. Each agent i maintains an actor $\mu_i(s_i|\theta_i)$ that maps local observations to actions, and a critic $Q_i(s, a_1, ..., a_1|\varphi_i)$ that uses the global state and joint actions during training (Lowe et al., 2017). Transitions (s, a, r, s′) are stored in a shared replay buffer. The critic is updated by minimizing the loss:

$$L(\varphi_i) = E[(r_i + \gamma\, Q_i'(s', \mu_1'(s_1'), ..., \mu_n'(s_n')) - Q_i(s, a))^2] \tag{1}$$

and the actor is updated using the policy gradient:

$$\nabla\theta_i J = E[\nabla\theta_i \mu_i(s_i)\, \nabla_{ai}\, Q_i(s, a) \mid a_i = \mu_i(s_i)] \tag{2}$$

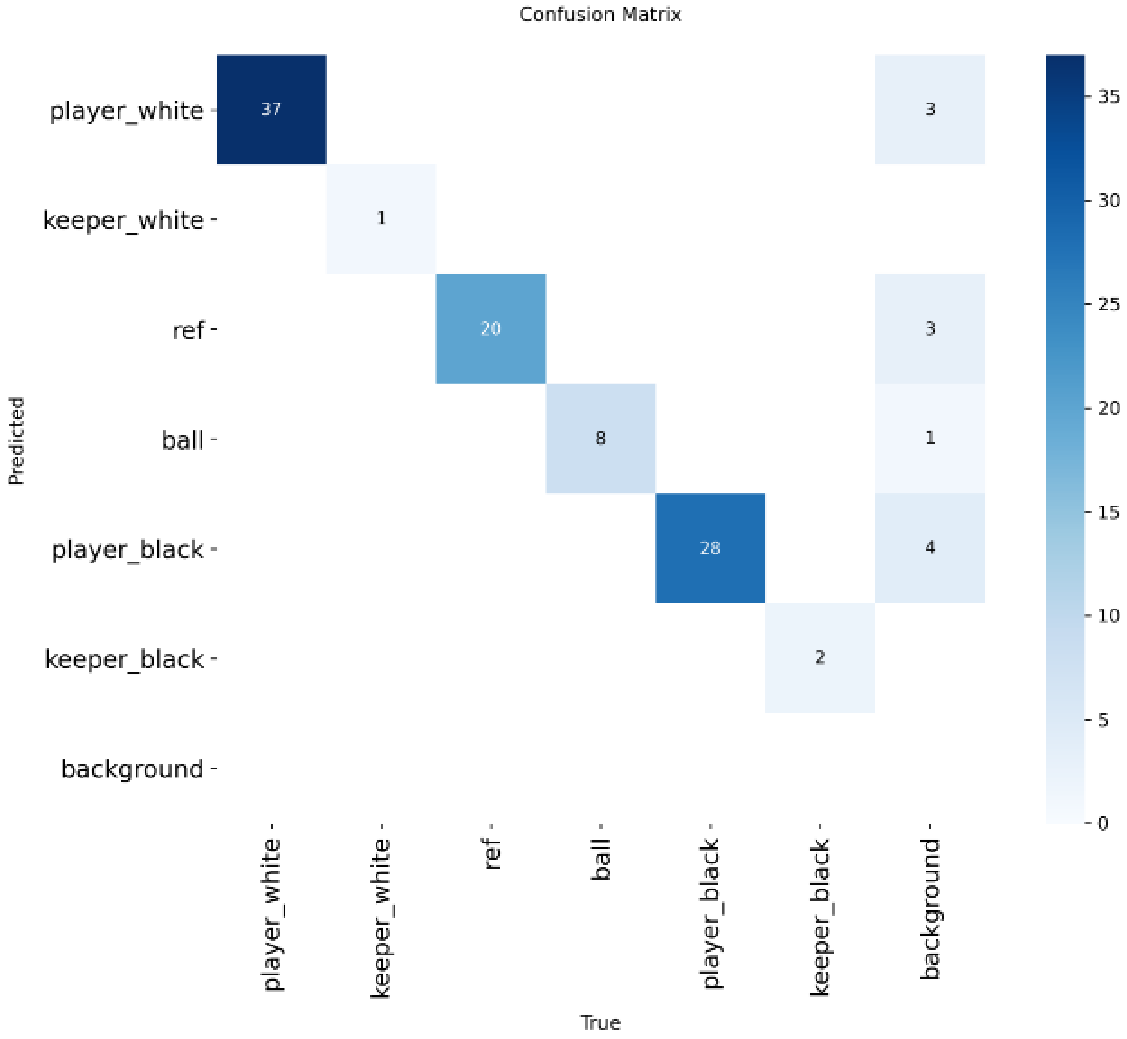
(Lowe et al., 2017)

To simulate ice hockey with MARL, we can design the model with 8-agents where each player (4 per team: 3 players + 1 goalie) observes local state information like position (by ByteTrack), puck location (by Yolov11), and nearby players. Because of reinforcement learning, the reward function combines sparse rewards for goals with dense rewards for puck possession, positioning, score, and team coordination. On the other hand, the punishment can be loss of puck possession, loss of score.

Methodology

In this chapter, the research methodology utilized for vision techniques and MARL simulation are presented in detail. First, the McGill Hockey Player Tracking dataset (MHPTD) is selected to make the analysis (Zhao et al., 2020). The dataset contains video clips of NHL gameplay, six classes are designed to perform the object detection in yolo format with player_color1, goalie_color1, player_color2, goalie_color2, referee, and puck. In the traditional ice hockey game, there are 12 players, because of the camera field of view limitations, each frame contains partial view of the rink. With simplified and focused environment, we plan to design the 4 vs 4 game play to reduce complexity and accelerate training time in the reinforcement learning.

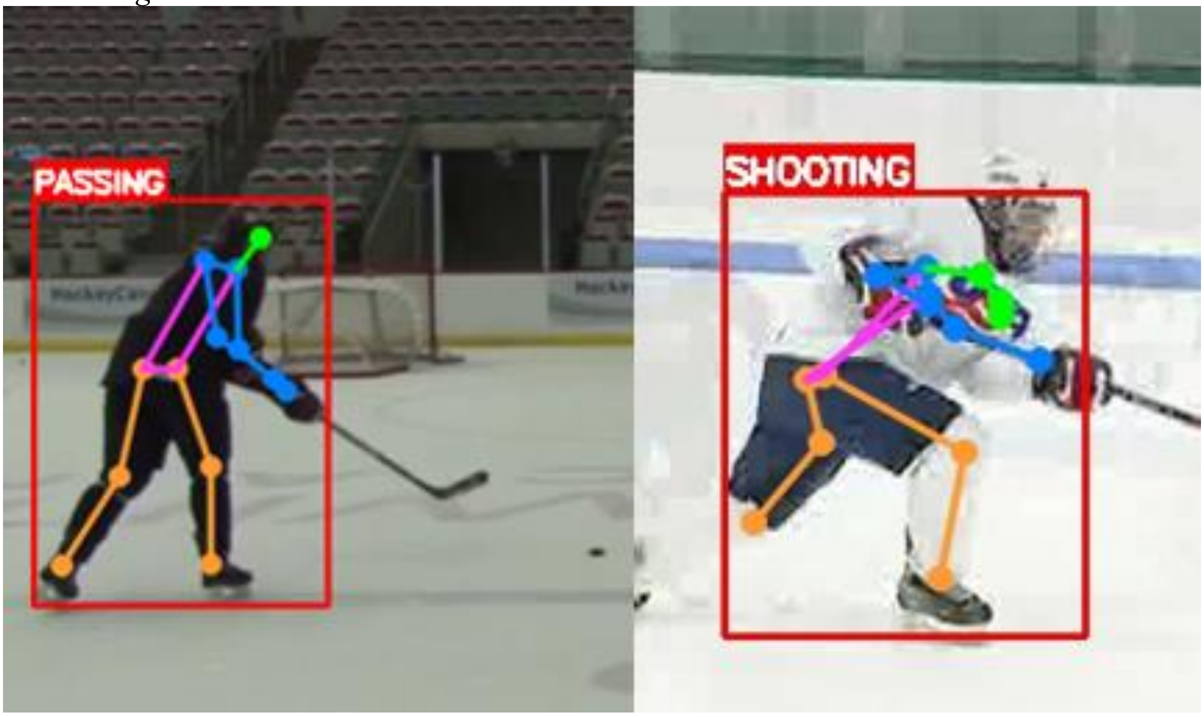
Figure 2
Yolov11 Confusion Matrix in Training



In the Yolov11 training, data was partitioned into training (80%), validation (10%), and testing (10%) sets to enable model generalization assessment. The evaluation was conducted using metrics including mean Average Precision (mAP) and class-wise confusion matrix analysis. From Figure 2, the confusion matrix of ice hockey class training revealed strong classification performance on dominant classes like player_white and player_black. This pipeline demonstrates YOLOv11's capability for real-time multi-object detection in dynamic ice hockey environments.

After object detection by Yolov11, Byte Track is utilized to track players across frames. ByteTrack links those detections across time, by following each player as they move around the rink. Even when players overlap, change directions quickly, or partially disappear, ByteTrack can keep tracking of players, by matching their position, and confidence scores. As a typical example in Figure 1, the historical movement tracking shows the changing locations of each object.

Figure 3
Pose Recognition With MMAction2



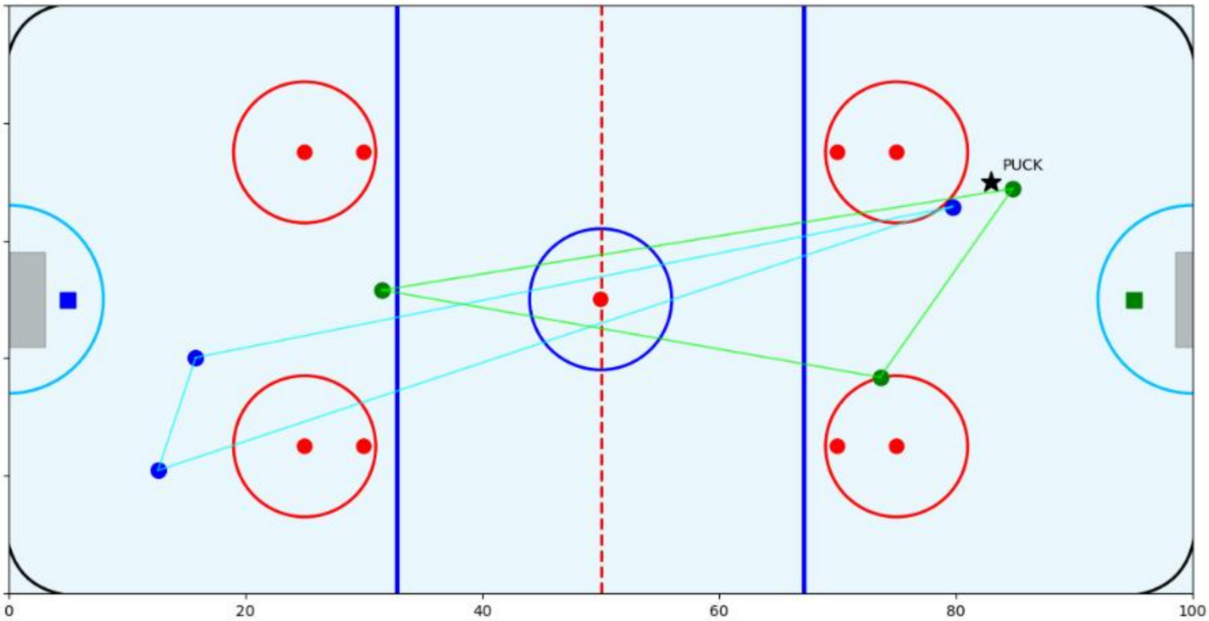
In the next step, we use MMAction 2 to recognize player actions like passing, shooting, and skating. In each frame, pose estimation is conducted to analyze body movements of hockey players, such as in Figure 3, when a player raises their stick to shoot or stretches out to make a pass. This is especially useful in ice hockey, where actions happen quickly. With MMAction2, we can automatically detect key plays like shots on goal, successful passes, or even defensive actions. This makes it possible to break down the game into meaningful collection of movements, which can then be used for strategy review, performance statistically analytics, or even highlights for market values.

After the pose recognition, the next step is to simulate ice hockey game using Multi Agent Reinforcement Learning. In the simulation, each agent (player or goalie) acts independently based on the detected positions from the Yolov11. In the detection of Yolov11, it provides the position of objects in a frame by bounding boxes and class probabilities. Therefore, the center of the objects is defined as the (x_i, y_i) coordinates of the midpoint of the bounding box which represents the estimated self-location. In the perspective of ice hockey game, there are agents of TeamA, TeamB, and the puck location. Based on the results of MMAction2, the agent actions are defined for each frame, like skating, passing, shooting; or the moving direction which can be detected by ByteTrack. It provides the velocity vector of each agent as (V_x, V_y) for direction or movements. In designing the reinforcement learning, the reward function is programmed to incentivized agents (not puck) behaviors through action-based feedback. Agents receive a reward of +0.5 for successfully passing the puck to a teammate, and +1.0 for executing a shot, encouraging cooperative play and offensive engagement. When a team scores a goal, all players on that team receive +5.0, while players on the opposing team are penalized –2.0, reinforcing coordinated strategy and defensive responsibility. After the above preparation, we can begin to train MADDPG agents in vision grounded hockey simulation.

Findings and Discussions

In the MARL simulation, it demonstrates that reward and penalty structure affect agent behavior. There is a rise in meaning team dynamics. The puck is moving passively in the simulation, agents learn to pursue the puck, coordinate basic passing, and shoot toward the goal when in procession. The team-level rewards for scoring and penalties for conceding promotes cooperative strategies within each team, as evidenced by frequent passing attempts and spatial clustering around the puck. In MADDPG training, the combined vision inputs (Yolov11, ByteTrack, and MMActions2) serve as the input status to individual agent's policy network which presents the selected actions based on the observation, such as the agent's position, puck position, teammates position, opponents position. It enables training of MADDPG agents to learn in dynamic environments.

Figure 4
MARL Simulation



In Figure 4, the green dots represent TeamA, blue dots are TeamB; the black star is the puck. In the reinforcement learning, they act as agents, each agent's policy network receives a vector of local observations as input. For agent i, the observation vector T_i includes of the agent's own position at time T, the position of the puck, and the positions of teammates and opponents. For example:

$$T_i = [x_i, y_i, x\_puck, y\_puck, x\_teammate1, y\_teammate1, ...] \quad (3)$$

The actor network takes this observation vector and outputs the next timestamp T_i action vector that represents a direction to move at the current timestamp.

$$T_{i\_next} = [\Delta x, \Delta y] \in \mathbb{R}^2 \quad (4)$$

In the above formular, Δx and Δy indicate the movement direction in the horizontal and vertical axes. This output can be directly applied to update the agent's position within the environment for the next T timestamp. After calculating the movement direction, it is time to train the centralized critic network of MADDPG that intakes all actions of all agents, and puck to evaluate how good each joint action is. Past experiences are stored in the replay buffer which can help agents to learn from the training. During Training, each agent acts according to their own observation which shows what we discussed before about the centralized training with decentralized execution framework (Lowe et al., 2017), make it possible for agents to study the team strategies in ice hockey.

Conclusion and Limitations

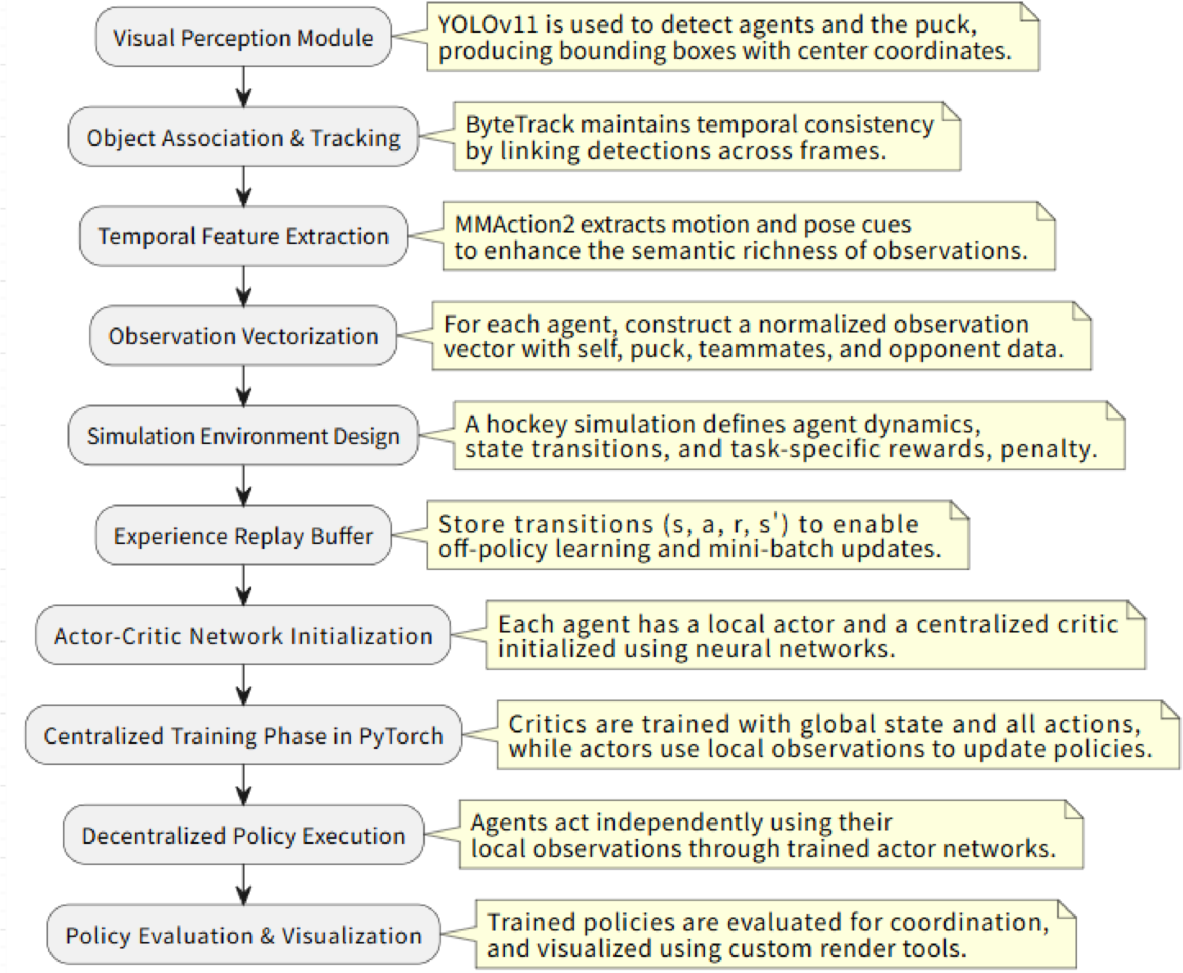
This research developed how to integrate computer vision techniques with MARL which could enable trained agents to simulate in complex and dynamic fix-field multiplayers sport field. As illustrated in Figure 5, the research pipeline begins with applying Yolov11 for object detection to identify players in different teams and the puck, followed by using ByteTrack to track movement of players and the puck. MMAction2 method is used to extract pose of the players. All above factors are vectorized into structured vectors, and feed into reinforcement learning simulations with reward and penalty. Transitions are stored in a replay buffer, which is used to enable off-policy learning. In the simulation of ice hockey, we designed 4 vs 4 agent game. As in MADDPG algorithm, each agent is equipped with an actor critic pair, where centralized critics are trained. This centralized training with decentralized execution framework is implemented in PyTorch, and the learned policies are evaluated through coordinated behavior and visual analytics.

Through the MADDPG training, each agent learns the decision making strategies in two ways, one is the collocation within team members, such as the connection between ball-holder who are going to attack, with other attackers and defenders at the same team; or the connection between defenders with goalie. Each agent can access the others' status including location, historical movement and pose; to make joint strategies within the team, it is easy for coaches to develop the teammate collocation patterns, for example, how to design successful attack strategies. At the same time, issues can be noticed, such as overcrowding and breakdowns. Another is the connection with opponents, agents can learn competitive responses based on the opponents' behavior. Opponents are modeled as part of the system; each agent's critic considers the actions and positions of opponents. It helps agents to understand and interpret tactical countermeasures, such as exploring weak zones, planning successful passing.

In the simulation environment, it helps coaches and educators to replicate realistic game environments and how team response under different strategies. Coaches can test the influence of tactical changes and observe the actions from agents of opponents. In the pedagogical perspective, the simulation provides detailed visualized feedback from historical realistic gameplay. It offers coaches to test different strategies based on the simulation, and observe the responses from the agents.

Figure 5

Research Pipeline



Visual Perception Module
YOLOv11 is used to detect agents and the puck, producing bounding boxes with center coordinates.
Object Association & Tracking
ByteTrack maintains temporal consistency by linking detections across frames.
Temporal Feature Extraction
MMAction2 extracts motion and pose cues to enhance the semantic richness of observations.
Observation Vectorization
For each agent, construct a normalized observation vector with self, puck, teammates, and opponent data.
Simulation Environment Design
A hockey simulation defines agent dynamics, state transitions, and task-specific rewards, penalty.
Experience Replay Buffer
Store transitions (s, a, r, s') to enable off-policy learning and mini-batch updates.
Actor-Critic Network Initialization
Each agent has a local actor and a centralized critic initialized using neural networks.
Centralized Training Phase in PyTorch
Critics are trained with global state and all actions, while actors use local observations to update policies.
Decentralized Policy Execution
Agents act independently using their local observations through trained actor networks.
Policy Evaluation & Visualization
Trained policies are evaluated for coordination, and visualized using custom render tools.

There is one significant limitation of the McGill Hockey Player Tracking Dataset (MHPTD) in this research, it is the restricted field of view from the moving camera. The camera is always tracking the movement of the puck and game; it cannot provide the full-size rink view which can offer view on the entire ice hockey field. As a result, there are observation bias, making it difficult to locate the exact location of objects, in using Yolov11 for object detection, which is the relative location, not the exact location of detection. Moreover, in training of the MADDPG, the relative locations of agent exert negative influence on the accuracy of result. The moving camera also provides challenges on Re-Identification (Re-ID) of objects, especially in the situation where the camera follows the puck or zooms in. Individual objects, such as the players might temporarily disappear from the camera, it can cause ID switching or loss of tracking.

The ideal situation for future research could be utilized using a full-size camera, this kind of camera is able to catch the entire rink view. At the same time, each player can be photographed to offer more details visual information, such as body orientation, pose dynamics, and team number, which can enhance the process of Re-ID. For example, we can train the photographs of Teemu Selänne to develop the personalized player recognition module, enable ByteTrack to track him with body features and pose. When it applies to all players, it will enhance accuracy of object detection and tracking of players.

References

Albrecht, S. V., Christianos, F., & Schäfer, L. (2024). *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press.

Contributors, Mma. (2020). *OpenMMLab's Next Generation Video Understanding Toolbox and Benchmark*. https://github.com/open-mmlab/mmaction2

Jocher, G., & Qiu, J. (2024). *Ultralytics YOLO11* (Version 11.0.0) [Computer software]. https://github.com/ultralytics/ultralytics

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In W. W. Cohen & H. Hirsh (Eds.), *Machine Learning Proceedings 1994* (pp. 157–163). Morgan Kaufmann. https://doi.org/10.1016/B978-1-55860-335-6.50027-1

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Neural Information Processing Systems (NIPS).*

Weber, J., Ernstberger, A., Reinsberger, C., Popp, D., Nerlich, M., Alt, V., & Krutsch, W. (2022). Video analysis of 100 matches in male semi-professional football reveals a heading rate of 5.7 headings per field player and match. *BMC Sports Science, Medicine & Rehabilitation*, *14*(1), 132. https://doi.org/10.1186/s13102-022-00521-2

Zhang, B. (2024). *Unlocking Ice Hockey Prowess: Pose-Centric Analysis With MMaction2, Yolov10, and BoT-Sort for Sports Education*. 239–246. https://doi.org/10.22492/issn.2188-1162.2024.18

Zhang, K., Yang, Z., & Başar, T. (2021). Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. In K. G. Vamvoudakis, Y. Wan, F. L. Lewis, & D. Cansever (Eds.), *Handbook of Reinforcement Learning and Control* (pp. 321–384). Springer International Publishing. https://doi.org/10.1007/978-3-030-60990-0_12

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *Proceedings of the European Conference on Computer Vision (ECCV).*

Zhao, Y., Li, Z., & Chen, K. (2020). A Method for Tracking Hockey Players by Exploiting Multiple Detections and Omni-Scale Appearance Features. *Project Report*.