Eligibility Traces in an Autonomous Soccer Robot with Obstacle Avoidance and Navigation Policy

Seyed Omid Azarkasb[1]✉, Seyed Hossein Khasteh[2]

[1] Visiting professor and Ph.D. Student of Artificial Intelligence and Robotics, K.N. Toosi University of Technology, Tehran, Iran
Seyedomid.azarkasb@email.kntu.ac.ir

[2] Assistant Professor of Artificial Intelligence, K.N. Toosi University of Technology, Tehran, Iran
Khasteh@kntu.ac.ir

Abstract:

As the thematic literature of machine learning suggests, reinforcement learning falls between the two methods of observational learning and non-observational learning. In this method, the learning agent receives a reward or punishment from the environment according to its action. Therefore, the learning agent interacts with the environment through trial and error and learns to choose the optimal action to achieve its goal. In the meantime, the eligibility traces are considered as one of the main mechanisms of reinforcement learning in receiving delayed rewards. In the use of conventional reinforcement learning methods, when the learning agent achieves a goal, only the value function of the last state-action pair changes, but all the trace states are affected based on the eligibility traces. In other words, the delayed rewards are distributed throughout the trace. Like the ant pheromone effect, this method can increase learning speed empirically to some extent. A soccer robot on the field encounters moving obstacles such as balls, rival robots, and home robots, and fixed obstacles such as gates and flags, so its environment is in a very dynamic state. Therefore, due to the dynamic environment, it is an important issue for autonomous soccer robots to avoid obstacles and should be considered in real play. The main idea of this study is to determine the appropriate traces for the robot to move towards the ball and ultimately to score with the approach of avoiding obstacles in simulating a real soccer match. The desired results obtained from a game played indicate a high level of online experience and decision-making power in the face of new situations.

Keywords: Reinforcement Learning, Eligibility Traces, Autonomous Soccer Robot, Obstacle Avoidance, Navigation policy, RoboCup, Small-Size Soccer Robot.

1- Introduction

The exploration of robot behavior in dynamic environments, specifically in the context of soccer playing, reveals a nuanced decision-making process influenced by eligibility traces and obstacle navigation. The essence of the robot's adaptive and strategic maneuvers becomes apparent through a thorough analysis of the obtained paths. This paper aims to enhance the navigation capabilities [1] and

obstacle avoidance [2] of autonomous soccer robots [3]. It employs reinforcement learning techniques, particularly Eligibility Traces, enabling these robots to learn from past experiences and make intelligent decisions in dynamic environments. The primary focus lies in improving the robots' ability to navigate a soccer field effectively while intelligently circumventing obstacles. The innovative approach proposed here integrates eligibility traces into the control system, empowering the robots to adapt and make informed decisions based on their sensory inputs. Ultimately, this research contributes to advancing autonomous robotics, specifically in the realm of soccer-playing robots, by amalgamating reinforcement learning with obstacle avoidance for refined navigation strategies. The necessity of conducting this research lies in its potential to advance the field of robotics, particularly in autonomous soccer-playing robots, based on meticulous scientific reasoning:

1. Enhancing Autonomous Robot Capabilities: This research is crucial as it seeks to augment the capabilities of autonomous robots, specifically in the domain of soccer-playing robots, by integrating eligibility traces. This integration is expected to significantly improve their navigation, obstacle avoidance, and decision-making skills.
2. Adaptation in Dynamic Environments: With the integration of eligibility traces, the robots are anticipated to adapt more effectively to dynamic and complex environments, such as soccer fields, where they encounter diverse obstacles, opponents, and changing scenarios.
3. Refined Learning and Performance: By leveraging past experiences and real-time decision-making, the robots are anticipated to exhibit refined learning strategies over time, leading to more intelligent navigation choices and efficient obstacle avoidance.
4. Foundational Contribution to Robotics: The outcomes of this research have the potential to contribute foundational insights to the broader field of robotics, emphasizing the efficacy of reinforcement learning techniques like eligibility traces in enhancing the adaptability and intelligence of autonomous systems.

The main research question focuses on exploring the efficacy of integrating eligibility traces within the control system of autonomous soccer robots to enhance navigation and obstacle avoidance capabilities. The Research question is “How does the integration of eligibility traces contribute to improving the navigation, obstacle avoidance, and decision-making abilities of autonomous soccer robots in dynamic environments?”. The hypothesis posits that “integrating eligibility traces into the control system of autonomous soccer robots will significantly enhance their adaptive navigation strategies”. By leveraging past experiences and sensory inputs, these robots will demonstrate improved obstacle avoidance, efficient movement towards the ball, and intelligent decision-making while maintaining possession and reaching target positions effectively in dynamic soccer field environments. Here are the steps followed as the methodology for conducting this research:

1. Objective and Hypotheses Definition: Clearly define the overarching goal of the research and the initial hypotheses it aims to investigate.

2. Review and Analysis of Previous Works: Exhaustively reviewing relevant prior works and previous articles in the field under study, including algorithms and methodologies used in soccer robotics.
3. Environmental Modeling: Defining and describing the soccer game environment, restricting it to half the field as a modeling environment for experiments and analyses.
4. Algorithm Determination and Learning Process: Employing eligibility traces and other reinforcement learning algorithms to enhance navigation and obstacle avoidance capabilities.
5. Experimentation and Evaluation: Conducting experiments within the modeled environment and analyzing results to evaluate the robot's performance in navigation and obstacle avoidance.
6. Knowledge Management: Utilizing a knowledge base to store and reuse learned paths in similar environments for future decision-making.
7. Soccer Match Simulation: Simulating a soccer match scenario with robots, estimating positions, and obstacles in a real-world setting.

The research introduces several innovative contributions that advance the field of autonomous robotics in the domain of soccer-playing robots.

1. Field Modeling and Generalization: One significant innovation of this research lies in the modeling of the soccer field environment. By considering only one-half of the field for experiments, the method demonstrates the ability to symmetrically remove half of the field, allowing for the generalization of eligibility traces to the entire field. This modeling approach enhances the scalability and practical applicability of the proposed method. By maintaining soccer's inherent complexities, the Half Field setting focuses the agents on decision-making within specific constraints. It offers uniform interfaces for interacting with both the environment and fellow agents, along with standardized evaluation tools for measuring performance.
2. Eligibility Traces Without Obstacles: Another key innovation of this research is the determination of eligibility traces regardless of obstacles. By initially intelligently ignoring obstacles such as rival robots, team robots, and field obstacles, the method focuses on capturing the fundamental navigation capabilities of the robot. This innovation provides valuable insights into the robot's ability to move towards the ball and score goals, regardless of hindrances, showcasing the robustness and adaptability of the proposed method.
3. Efficient Handling of Obstacles: The research further advances the field by introducing a method for determining eligibility traces considering obstacles with a minimal computational burden. By strategically incorporating obstacles in a later stage of the experiments, the proposed approach minimizes the complexity of obstacle handling, allowing for efficient navigation and obstacle avoidance. This innovation contributes to the practical implementation and real-time decision-making capabilities of the robot in dynamic soccer field environments.
4. Comprehensive Analysis and Visualization: To effectively communicate the findings, this paper offers a comprehensive analysis and visualization of the obtained eligibility traces. By summarizing

and presenting 761 eligibility traces in 211 rows, the research provides a rich dataset that demonstrates the breadth and depth of knowledge derived from the experiments. This innovation enables researchers and practitioners to gain valuable insights into the robot's navigation behavior and decision-making process.

5. Knowledge Base and Reasoning: The inclusion of a knowledge base for storing desired eligibility traces represents another notable innovation. By capturing the reason behind a series of actions and storing the corresponding eligibility traces, the method empowers the robot with the ability to reason and adapt based on previous experiences. This innovation enhances the robot's decision-making power and contributes to the development of more intelligent and context-aware autonomous soccer robots.
6. Real-World Simulation and Mapping: Lastly, the research showcases the practical application and effectiveness of the proposed method through a simulated match and mapping of agents to the real field. This innovation bridges the gap between theory and practice, demonstrating the online experience and high decision-making capabilities of the robot in response to real-world scenarios. The simulated match serves as a testament to the potential impact and practicality of the proposed method in the field of autonomous soccer robots.

The research framework involves a multi-stage approach. Initially, it focuses on determining eligibility traces for robot navigation and goal-scoring without considering obstacles. Subsequently, it extends to incorporating obstacles into the navigation strategy and simplifying the trace representation. Another phase involves establishing a knowledge base to store and apply previously obtained eligibility traces based on the prevailing conditions. Finally, it culminates in a simulation of a soccer match, mapping agents' positions to a real soccer field through a clustering method. Simulators exist in this regard, such as [4]. The RoboCup Soccer Simulator serves as a research and educational tool dedicated to multiagent systems and artificial intelligence studies [5]. It allows two teams of 11 autonomous robotic players in simulation to engage in soccer matches. The rest of this manuscript is prearranged as follows: a brief review of recent related researches and basic concepts is presented in section 2. Section 3 depicts environmental modeling. Section 4 focuses on determining the eligibility traces and analyzing the results. The conclusions are summed up in section 5. Finally, future works and practical suggestions are presented in the section 6. We believe that the integration of eligibility traces into the navigation policy of autonomous soccer robots opens up new possibilities for their application in various real-world scenarios. By overcoming the challenges of dynamic environments and obstacle avoidance, these robots can contribute significantly to the advancement of intelligent robotic systems.

2- Research Background

Reinforcement learning-based systems allow a learner agent to do different operations in different situations, receive different rewards, and to learn actions tailored to each situation based on the total

rewards received [6]. In this regard, the system should have a high online efficiency because the evaluation of the system is often performed simultaneously with the learning process [7]. Making a balance between exploration and exploitation is one of the issues in this field [8]. The discount rate component establishes this balance. The closer the discount rate is to zero, the more the learning agent will be shortsighted in this situation, and the more the impact of recent immediate rewards will be greater than subsequent rewards. The closer this coefficient is to one, the more longsighted our learning agent will be and will consider the impact of further rewards as much as the impact of current rewards. In general, changes in the discount rate can have a huge impact on the agent's learning type. The important point here is that reward should not be in the hands of the agent, but it should be outside its hands. The environment is not necessarily unknown to the agent, and it just should not be controllable [9]. Reinforcement learning teaches the agent what to achieve (not how to achieve it!), and the agent must be able to assess its success rate.

The goal of navigation is for the robot to reach a target point and be able to move towards a safe path with a known destination while avoiding obstacles throughout the path [10]. A mobile robot is an intelligent device that detects its position and moving status based on its environment and the defined targets and then follows a targeted path [11]. The control actions that must be performed by a mobile soccer robot to explore and identify the environment are as follows [12]: avoiding a collision with fixed and moving obstacles, aimless wandering without colliding with fixed obstacles, exploring the surrounding environment by observing objects that are in their field of view and determining their distances, paying attention to changes in the obstacles, achieving conclusions from the environment in the form of identifiable objects and performing tasks by the observed objects, formulating the execution of necessary decisions under the changes in the states of the robot's surrounding environment, deciding on the behavior of objects in the surrounding environment and modifying the final decision based on them. One of the issues that should be solved to design a navigation system for soccer robots in unknown and dynamic environments [13] is the lack or shortage of information about robot behaviors. This lack of knowledge affects various stages such as the robot's locating, routing, and avoiding obstacles [14]. At the same time, relocating agents after making a decision and before taking action requires a delayed strategy. If the other agents turn to the other side just before kicking, the suggested point will no longer be the best possible point. This is the main motivation of current article to focus on eligibility traces. This paper presents a novel approach for determining eligibility traces, solving navigation problems, and avoiding obstacles in the context of soccer-playing robots. The development of autonomous robots capable of performing complex tasks has been a significant area of research in recent years. Among these tasks, navigating in dynamic environments and avoiding obstacles pose significant challenges. One particular application domain where these challenges are prominent is autonomous soccer-playing robots. Path planning stands as a fundamental facet within robotics research, serving as the linchpin for achieving autonomous navigation objectives. This entails enabling a robot to autonomously chart a

seamless, obstacle-free trajectory from its initial point to a specified target position [15]. [16] introduces a fast path planning algorithm. The authors highlight that while its generality is a strength, it can become a weakness in contexts where factors such as processing time or path smoothness are crucial. Consequently, they proposed an ad-hoc heuristic specifically tailored for path planning in non-cluttered dynamic environments. Their approach has undergone testing across a range of artificial and real RoboCup Small Size League scenarios, revealing its ability to discover smoother and shorter paths more efficiently on average. Three main objectives are considered for optimal path planning to the ball in a robot soccer system. The first objective is to minimize the elapsed time. By optimizing the path and reducing the time required to reach the ball, the robot will be able to approach the ball faster and enhance its overall performance in the game. The second objective is to minimize the heading direction error. Using a multi-objective planner, a path is selected for the robot that minimizes the error in heading direction towards the ball. This error represents the robot's accuracy in aligning its direction with the optimal path. By reducing the heading direction error, the robot will move more directly and accurately towards the ball. The third objective is to minimize the posture angle error. The posture angle refers to the deviation of the robot's orientation from the desired angle while moving towards the ball. By selecting a path that minimizes the posture angle error, the robot will move more accurately and consistently with the optimal path [17]. By optimizing the path to the ball to reduce these three objectives, the robot will be able to approach the ball optimally and more accurately, leading to improved performance in real-world soccer matches. [18] emphasizes the significance of bridging the disparity between real-world and virtual soccer matches. As such, the available data, orientation, and decision-making models for both real-world and virtual soccer matches are synthesized and summarized in [18].

2-1- Reinforcement Learning

In dynamic programming, a general problem is divided into some sub-problems, and by finding the solution to the sub-problems, the general answer to the problem is obtained. In this regard, these sub-problems should overlap to some extent and not be completely independent of each other, i.e. the solutions obtained from each of them can be used to solve the other sub-problems. The condition for using dynamic programming techniques is to have Markov property. The Markov decision-making process is a mathematical model with applications in the study of complex systems, and its most important components are the state and how it moves from one state to another. If the probability of being in a new state and receiving a reward depends solely on what state the agent was in in the previous step and what action it chose, and it does not depend on the rest of the trace taken (trace history from the first step up to now), then this problem and the model are called the Markov model. The most important feature of the Markov model is that it has no memory. In the Markov decision process, the policy is a mapping from the history of states and the history of selected actions to the current state.

Finally, the learner agent seeks to maximize its receiving rewards. As soon as a reward is received, as actions are performed and transferring from one state to another is done, the estimates of the states and actions are immediately updated according to the Bellman equations [19]. [20] focuses on modeling soccer as a Markov process. The main contribution of this paper lies in the discretization of the soccer pitch into nine zones. The states of the Markov process are defined based on the zone of the pitch where the ball is located, the team in possession, and the score. By considering these factors, the researchers aim to analyze and estimate the zonal variation of team strengths in soccer. Furthermore, works have been conducted in non-Markovian environments such as [21]. This study emphasizes the potential application of Exploitation-oriented Learning (XoL) in non-Markovian multi-agent soccer environments, highlighting its rationality and effectiveness through computer simulations. Specifically, the focus is on the Penalty Avoiding Rational Policy Making algorithm (PARP), which is an XoL method designed to learn a policy that avoids penalties. To enhance its performance and adaptability to uncertain scenarios, an improved version of PARP, referred to as Improved PARP, has been developed, incorporating memory-saving mechanisms and strategies for handling uncertainties. In contrast, there is the Monte Carlo method, in which updating is done only at the end of the episodes (learning period). In other words, there is no need to know the environment model, and the estimates that become eternal at the end of an episode are not dependent on the previous estimates. Experience is nothing but a set of states, performing actions in these states, and receiving rewards corresponding to performing the actions. A sequence of a certain length is generated from experience. Experience is the same episode consisting of values and policies. The approximation considered for one state is separate from the approximations for the other states and there is no so-called bootstrap and averaging is done for observations. In the Monte Carlo method, although it is still possible to learn without having an environment model since this learning is based on different observations and gaining successive experiences in the environment, in most cases the learning speed will be slower than the speed in the dynamic programming method. Therefore, the Monte Carlo method can be selected when there is no model of the environment, while the learning agent seeks to achieve the optimal policy [22]. Meanwhile, the use of the Monte Carlo tree can greatly improve the work result [23].

In the meantime, it is no exaggeration to say that the main idea in the whole of reinforcement learning is the idea of temporary difference [6]. The agent learns from the experiences it is currently gaining and does not need to be fully experienced, so this method has the benefits of both dynamic programming and the Monte Carlo method. Similar to the Monte Carlo method, this method learns directly from the raw experiences of the environment, and similar to dynamic programming, it expresses its approximation of the value function of a state or state-action pair based on the approximations of the value functions of other existing states or state-action pairs; in other words, it bootstraps as well. The temporal difference method has the shallowest and lowest level backup diagram [17]. In this method, a completely increasing trend of updating at each stage can be seen. The agent can learn before it knows

the result and also without knowing the result. In terms of convergence speed, neither of these methods is superior to the other, and both converge at infinity. Intuitively, the temporal difference method usually converges faster and at a smaller value than infinity, and also the two methods do not necessarily converge to the same value. In the group of temporary difference methods, the State–Action–Reward–State–Action (SARSA), Q-Learning, and Eligibility Traces are introduced, which are described in the following.

2-2- Eligibility Traces

The SARSA method was introduced in 1994 and is one of the basic methods of a temporary difference that operates based on behavioral policy, i.e. the trace taken by the agent is the same as the updating trace [8]. The agent in the current state selects an action, goes to a new state, and receives a reward, and then selects the next action. This method is one of the on-policy methods and based on current estimates, it always acts in a greedy or quasi-greedy manner. It first updates the value functions of the state-action pairs, then the policy, and so on, step by step [24].

In the Q-Learning method, the main idea is that one policy is being implemented in the environment, and the agent, independently of it, improves and optimizes another policy. This method is one of the off-policy methods and considers the difference between the evaluated policy and the implemented policy in the learning process. In other words, the agent is working with a soft policy while at the same time learning and optimizing another greedy policy. Unlike the SASRA method, in this method, the observed state-action value is not updated based on the next state-action value, but updating is done based on the action with the highest value in the next state. The challenge here is that updating the value function can depend on the action that is selected in the next state i.e. when the agent has not yet acquired sufficient knowledge in that regard, in other words, in the experiences it has gained, the agent has not seen all the selectable actions until the moment and has only seen some of them [25]. In general, the SASRA algorithm exhibits faster convergence, whereas the Q-Learning algorithm demonstrates superior overall performance. Nonetheless, the SARSA algorithm is susceptible to getting trapped in local minima, while Q-Learning requires a longer learning period.

Eligibility traces are one of the main mechanisms of reinforcement learning in receiving delayed rewards [26], which can be generalized to both SASRA and Q-Learning algorithms [8]. In the eligibility traces, there are two states of observing the effect. Theoretically, the eligibility traces are a bridge between the Monte Carlo and temporal difference methods and have a forward-looking approach from one perspective and a backward-looking approach from another [17].

Eligibility traces have memory and can store actions and states. The idea of a forward-looking approach is suitable for mathematical studies and is not very effective for practical work because it does not have the $r_t + 1$, $r_t + 2$, etc. factors, in other words, it does not have future knowledge. To do this, it uses a

backward-looking approach. Backward, initially, is the combination of the SASRA algorithm and Q-Learning [27]. Rather than examining an action selection strategy, as demonstrated in [27], we can explore the integration of Q-Learning and the SASRA algorithm, known as backward Q-Learning, which can be incorporated into both the SASRA algorithm and Q-Learning. The backward Q-Learning algorithm directly adjusts the Q-values, subsequently indirectly influencing the action selection policy. Consequently, these proposed RL algorithms can potentially accelerate learning and enhance the final performance. It can be easily inferred that the backward-looking approach is the same as the off-line forward-looking one. On the other hand, the advantage of being offline is that it is more stable because changing and updating the value function at any time is not good and creates uncertainty, possibly leading to machine failure. With this approach, the agent no longer manipulates the control in the middle of each episode. In the temporary difference method, the agent can update itself in each step (the current state) by performing an action and observing the next state, but in the Monte Carlo method, the agent can update itself only when it reaches the goal. In the eligibility traces, the agent can itself choose the number of updating steps. Of course, the number of steps is an important factor. If too many steps are taken, some states of bad traces will be rewarded positively as well, which is not desirable. If a small number of steps is considered, only those close to the goal will receive a positive effect. Therefore, the optimal number of steps varies according to the problem. The general idea is that instead of relying on prior knowledge, the reality that the agent wants to see is taken into account. The downside to this approach is that the agent has to wait to get the next one. But this will increase the accuracy. In other words, by doing so, the agent is making a compromise between the time it takes to reach the goal and the degree of accuracy. In the online updating, the agent updates its value function as soon as it sees the state, while in the offline updating, the agent collects all the updates it has received during the intended state visit and applies them entirely at the end of the episode. In normal methods, when one reaches the goal, only the value function of the last state-action pair changes, but all the trace states are affected based on the eligibility traces[17], as shown in Figure 1. In other words, the reward is distributed with delay throughout the trace. Just as the ant pheromone effect, this method can increase learning speed empirically to some extent.

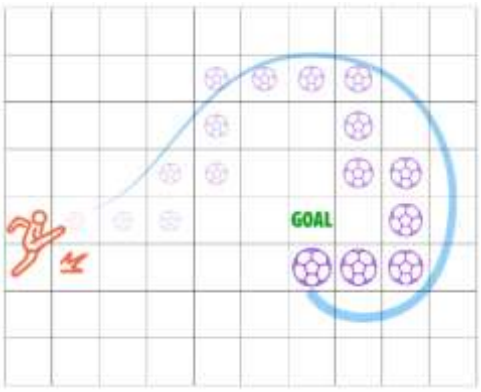

Figure 1. Visualizing the effects of states on eligibility traces.

2-3- Related Work Done

The inception of the RoboCup competition dates back to 1997, establishing itself as the most longstanding league within the RoboCup initiative [28]. In this regard, the two-dimensional RoboCup soccer simulation is designed as a basis for successful international soccer matches and research challenges in the field of robotics and artificial intelligence. To gain a comprehensive and advanced understanding of the game, soccer-playing agents need to be equipped with the ability to learn and acquire fundamental low-level skills. These skills can then be combined and utilized to imitate the expertise of seasoned players. In a study by [29], reinforcement learning is employed to acquire the foundational skills of intercepting a moving ball. The absence of mutual cooperation unquestionably leads to failure. In this regard, [30] introduces an adaptive Q-Learning approach that facilitates multiagent cooperation and coordination. This method aims to construct a self-learning cooperative strategy for robot soccer systems and addresses local strategy selection, while [31] improves the decision-making of robots by utilizing an innovative joint action policy called Consensus Action Policy (CAP) to prevent actions that commonly lead to team failures. The CAP policy records past failures caused by joint actions, computes a cooperation tendency, and integrates it with each agent's Q-value and Nash bargaining solution to determine joint actions. This tendency promotes cooperative actions while avoiding those that may lead to team failures. One notable study in this area is the work on heuristic-based Q-Learning soccer players, which presents a new reinforcement learning approach for RoboCup Simulation [32]. This approach, known as Heuristic Accelerated Q-Learning (HAQL), incorporates heuristics to expedite the learning process of the well-known Q-Learning algorithm. By adopting and adapting this methodology, authors aim to enhance the capabilities and performance of their soccer robots in the context of research. Accordingly, [33] first describes the features of the RoboCup simulation system and then presents a new Q-Learning-based strategy by introducing the limitations of Q-Learning. [34] presents the application of the Q-Batch update-rule to learn robotic soccer controllers within the domain of batch reinforcement learning. This approach proposed operates under the assumption that agents engage with the environment through episodes comprised of interlinked trajectories. The update-rule, employing trajectory rollouts, expedites the distribution of rewards back to the initial states. This strategy incurs a marginal rise in computational cost, involving an additional maximization process, compared to Q-Learning. Nonetheless, it offers advantages such as data reuse and the ability to conduct off-policy backups. [35] introduces three decentralized reinforcement learning (DRL) methods of robot behaviors: DRL-Independent, DRL Cooperative-Adaptive (CA), and DRL-Lenient. These methodologies are applied across four distinct robotic challenges, demonstrating that DRL techniques exhibit enhanced performance and faster learning compared to centralized approaches. Notably, the DRL-Lenient and DRL-CA models show superior efficacy in solving these complex tasks, hinting at the potential of decentralized approaches in intricate scenarios where centralized methods face limitations. [36] introduces an open-access expansive

framework for exploring reinforcement learning and sim-to-real applications in robot soccer. The authors advocate for a simulated environment conducive to training both continuous and discrete control policies governing the comprehensive behavior of soccer agents. Additionally, they propose a sim-to-real approach that relies on domain adaptation to transfer learned policies onto real-world robots. Remarkably, this method outperformed human-designed policies in the 2019 Latin American Robotics Competition (LARC), securing a commendable 4th place. Penalty kicks are one of the most challenging issues in soccer robot games because if the rival turns to the other side just before kicking the ball, the suggested point for the goalkeeper to move will no longer be the best possible point. In [37], the authors specifically address the inclusion of the shooting angle reward as a means to enhance the scoring rate within the framework. They highlight the significance of incorporating this additional reward component based on the underlying basic reward function. Considering this fact, hard coding and other learning methods are no longer suitable for this purpose and a delayed reward learning algorithm is useful to solve this problem. Therefore, [38] has used the Q-Learning method to control the goalkeeper during the penalty shootout. To achieve better results as well as to avoid getting caught in a local optimal trap, in addition to the usual reward function, a new reward function based on the value of each period has been applied in this method, so that the value of the reward function increases based on the number of episodes. Despite the advantage of escaping the local optimum, this approach will likely destabilize the model. To avoid this problem, in [38] the reduction coefficient has been used to reduce the value of the learning rate during the episodes. The results show that changing reward performance based on courses makes the learning process faster and more efficient. [39] has provided an effective strategy for goalkeepers in midfield. The proposed solution consists of the SARSA algorithm with eligibility traces and tile coding to discretize the state variables, which according to the reported experiments has led to better results compared to the random decision. In particular, the tile coding technique is a function approximation method that is used to prevent the exponential growth of state space. Hence, it also prevents the curse of dimensionality [40]. The curse of dimensionality arises when the variables involved in the problem increase, leading to a too-large state space and hence making it difficult to solve the problem. [41] utilizes partial eligibility traces to find initial cases in case-based learning to solve the problem of RoboCup Soccer Keepaway. This approach updates recently visited state-action values. The methodology involves the Problem Solver process running SARSA(λ) until a successful episode is achieved, storing a non-optimal case as a new instance. It's termed "partial SARSA(λ)" as SARSA(λ) isn't executed until convergence. [42] explores the cooperation mode in soccer robot games by introducing an improved SARSA algorithm. The authors begin by presenting the agent model used in RoboCup2D and then compare plan design and scheme design methodologies implemented with the SARSA algorithm. To enhance learning, heuristic information is incorporated, allowing for value sharing among participants and reinforcement learning. A comprehensive analysis is conducted to assess the feasibility of the SARSA algorithm within the context of RoboCup2D application, and experimental results demonstrate its effectiveness in enhancing the offensive and defensive capabilities of the team.

One of the most important goals of the teams participating in the RoboCup league is the ability to increase the number of shoots. The reason is that superiority over the opponent requires a powerful and precise shoot. At the same time, scoring a goal is the maximum reward [43]. he methods introduced for shooting so far are mostly based on Inverse Kinematics (IK) and point analysis. Occasionally, opponent modeling is performed [44]. In order to enhance learning efficiency directly, the opponent modeling Q(λ) algorithm, which integrates fictitious play in game theory and eligibility trace in reinforcement learning, is utilized in [45]. The assumption of these methods is that the positions of the robot and the ball is fixed. However, this is not always the shooting case. In [46], a shooting strategy is presented for situations where the robot is walking. Here, a curved path is designed to move the robot towards the ball so that the robot will eventually have an optimal position to shoot. Hence, robot movement parameters such as speed and angle are more precisely adjusted by the Q-Learning algorithm. [47] utilized an enhanced reinforcement learning approach, particularly the Hierarchical Movement Grouped Deep-Q-Network (HMG-DQN) algorithm, to effectively train robots in critical skills such as ball approach and ball shooting within the context of RoboCup soccer games of the Middle-Size League. [48] introduces a novel approach that utilizes Q-Learning for solving the problem of mobile robot path planning in unknown and dynamic environments. Unlike previous studies that primarily focused on static environments, this research tackles the more complex challenge of dynamic environments. The key innovation lies in the definition of a new state space that effectively reduces the size of the Q-table. By limiting the number of states, the training process for the intelligent agent becomes more manageable, facilitating faster convergence and navigation. This approach enables the mobile robot to make reliable and efficient decisions in real-time, even in the face of a continuously changing environment. The action valuation of on-ball and off-ball soccer players is different [49]. In practice, no complete knowledge of the environment is available and there is only a series of experiences. Experiences can be gained online or be simulated. Online experiences are the real ones, which are also incomplete. In the simulation, there are no complete experiences too, but in return, a model of the environment is created from incomplete experiments, and then in this model, several experiences are gained to extract more knowledge from the environment and obtain better results. As a result, the research gap here pertains to the absence of comprehensive methods that effectively integrate eligibility searches and robust knowledge bases in optimizing the performance of robots, specifically in dynamically changing environments such as sports competitions. The problem definition lies in the inadequacy of existing approaches to adequately address the complexities of navigation, obstacle avoidance, and intelligent decision-making in such dynamic settings. Table 1 summarizes the related work done. It's important to note that this section reviewed several valuable references relevant to the paper's topic. However, to offer a more closed comprehensive perspective, additional limitations have been applied in Table 1, specifically focusing solely on studies associated with Reinforcement Learning (RL), Eligibility Traces (ET), and RoboCup (RC). Therefore, the listing of all references in this table has been avoided.

Table 1: A review of relevant literature addressing limitations such as Reinforcement Learning (RL), Eligibility Traces (ET), and RoboCup (RC)

Ref.	Year	RL	ET	RC	Description
[29]	2004	✓	✓	✓	Explores how player agents in the RoboCup Soccer Simulation can learn and acquire low-level skills through reinforcement learning. These skills are then combined to emulate the expertise of experienced players.
[30]	2004	✓	✓	✓	Introduces an adaptive Q-Learning approach that facilitates multiagent self-learning cooperation and coordination for robot soccer systems.
[24]	2005	✓	✓	✓	Application of episodic SMDP SARSA(λ) with linear Tile Coding function approximation and variable Lambda to learning higher-level decisions in a keep-away subtask of RoboCup soccer.
[32]	2007	✓	✓	✓	Used heuristics to speed up the Q-Learning in the RoboCup 2D Simulator.
[4]	2008	✓	✓	✓	Investigates the impact of action selection strategies on temporal difference learning for optimal control, proposing a modified SARSA(λ) algorithm with simulated annealing and demonstrating enhanced convergence in a soccer simulation environment.
[33]	2008	✓	✓	✓	Making a description of the characteristics of the RoboCup simulation system, proposed a new strategy based on Q-Learning proposed and compared with traditional Q-Learning.
[17]	2009	×	×	✓	Focuses on the educational dimension of mobile robotics with a stabilized robot soccer system, presenting a multi-objective population-based incremental learning (MOPBIL) algorithm for fuzzy path planning. Seeks to optimize the path to the ball by minimizing elapsed time, heading direction, and posture angle errors in the robot soccer system, demonstrating effectiveness through simulation.
[40]	2011	✓	✓	✓	The underlying mechanism of eligibility traces is implemented in terms of on-policy and off-policy procedures, as well as accumulating traces and replacing traces.
[48]	2011	✓	✓	×	Presents a novel approach for mobile robot path planning in unknown dynamic environments utilizing Q-Learning, while simultaneously addressing dynamic challenges through a state space redefinition, resulting in a reduction of the Q-table size and improved navigation algorithm speed.
[21]	2012	✓	✓	✓	Evaluates the real-world effectiveness of the Improved Penalty Avoiding Rational Policy Making algorithm (Improved PARP) in non-Markov multi-agent environments, particularly in the context of a multi-agent soccer environment using a keepaway task.
[27]	2013	✓	✓	×	Introduces a novel approach, backward Q-Learning, that combines SARSA and Q-Learning, addressing the exploration-exploitation dilemma by directly tuning Q-values and demonstrating improved learning speed and performance in simulations such as cliff walk and cart–pole balancing.
[34]	2015	✓	✓	✓	Presents Q-Batch, a novel update-rule for Batch Reinforcement Learning, applied to enhance robotic soccer controllers on physical platforms. Demonstrates superior performance over hand-coded policies in diverse tasks, and in a comparative study, outperforms Q-Learning in terms of policy quality within the same interaction time for a specific task.
[6]	2017	✓	✓	✓	Presents the Temporal-Difference value iteration algorithm with state-value functions, applied to enhance shooting skills for soccer robots in the RoboCup Small Size League, and utilizes a Multi-Layer Perceptron (MLP) as a function approximator, resulting in effective training and successful acquisition of shooting skills under specific experimental conditions.
[25]	2018	✓	✓	✓	The problem of local strategy selection for each class of situations over time is regarded as an RL problem and is solved using a Q-Learning method.
[35]	2018	✓	✓	✓	Presents a decentralized reinforcement learning (DRL) methodology to train individual behaviors in multi-agent systems with multi-dimensional action spaces, proposing three approaches—DRL-Independent, DRL Cooperative-Adaptive (CA), and DRL-Lenient—validated through empirical studies,

					showcasing enhanced performance and quicker learning compared to centralized counterparts, particularly in complex real-world scenarios.
[38]	2019	✓	✓	✓	Presents a delayed rewarding learning algorithm, based on the fact that the performance of the keeper can be evaluated just after the whole procedure of the penalty kick is done.
[7]	2020	✓	✓	✓	Uses the eligibility traces and the replace-trace mechanism to improve the ability of policy selection in real-time confrontation.
[16]	2020	×	×	✓	Contributes by conducting a comprehensive benchmark comparison of widely used path planners from the literature, assessing their performance in the Robocup Small Size League Competition context.
[36]	2020	✓	×	✓	The IEEE Very Small Size Soccer (VSSS-RL) framework is introduced for training robot soccer agents using Reinforcement Learning and sim-to-real methods, achieving success in the 2019 Latin American Robotics Competition and serving as a versatile tool for studying RL in dynamic environments.
[39]	2020	✓	✓	✓	Using the Half Field Offense (HFO) environment proposes a baseline approach for goalkeeper strategy using SARSA with eligibility traces and Tile Coding for the discretization of state variables in RoboCup 2D Soccer Simulation.
[41]	2020	✓	✓	✓	The episodic SARSA(λ) that is an extension of SARSA that uses a decay rate (λ) for the eligibility trace, updating all the recently visited state-action values.
[45]	2020	✓	✓	✓	The X-OMQ(λ) algorithm integrates eXtended Classifier System (XCS) with opponent modeling to enable concurrent reinforcement learners in zero-sum Markov Games, with the goal of learning interpretable action selection rules, optimizing policies through a genetic algorithm, and refining the opponent's model simultaneously.
[47]	2020	✓	✓	✓	Presents the Hierarchical Movement Grouped Deep-Q-Network (HMG-DQN), an improved DQN algorithm specifically developed for addressing the challenges of cooperative competition in RoboCup soccer, demonstrating superior performance in scenarios such as 2v1 and 3v2 break-throughs through its operation at a high hierarchy of movement groups.
[28]	2021	✓	×	✓	Presents CYRUS, the RoboCup 2021 2D Soccer Simulation League champion, highlighting new functionalities like Multi Action Dribble, Pass Prediction, and Marking Decision to improve dribbling, passing, and defensive actions.
[23]	2022	✓	×	✓	Focusing on the study and optimization of the Monte-Carlo tree algorithm and reinforcement learning for defense strategy in the context of soccer.
[42]	2022	✓	✓	✓	Investigates the use of an improved SARSA algorithm to enhance the cooperation mode in RoboCup 2D Soccer games, resulting in improved offensive and defensive capabilities of the team.
[44]	2022	✓	✓	✓	Gives a thorough review of opponent modeling techniques in adversarial domains, covering stochastic, continuous, and concurrent actions, along with sparse, partially observable payoff structures. Introduces a novel framework for method comparison, conducts an analysis of different techniques, and outlines future research directions, highlighting the efficacy of opponent modeling in exploiting strategic weaknesses, particularly in partially observable scenarios.
[4]	2023	✓	×	✓	Develops a scalable multi-agent reinforcement learning solution for a full 11 versus 11 simulated robotic soccer game by enhancing the Proximal Policy Optimization (PPO) algorithm, introducing a faster 2D soccer simulation environment, and emphasizing self-play to achieve stability and higher-level strategy emergence.
[9]	2023	✓	×	✓	Presents an innovative skill set created using custom primitives and an enhanced Proximal Policy Optimization (PPO) algorithm. This skill set plays a crucial role, enhancing sample efficiency, stability, and seamless transitions between behaviors.
[20]	2023	×	×	✓	Presents a Markov process model for soccer, dividing the pitch into nine zones and defining states based on ball location, team possession, and score. Log-linear models capture state transitions, allowing estimation of team strengths concerning scoring, conceding, gaining, or losing possession in specific pitch zones.

[31]	2023	✓	✓	✓	Proposes a novel Consensus Action Policy (CAP) in multi-agent reinforcement learning for improving cooperation in robot confrontation scenarios. CAP evaluates joint actions based on past failures, promoting cooperation by selecting actions with high consensus.
[37]	2023	✓	✓	✓	A specific reward function, known as the shooting angle reward, has been devised to enhance the goal-scoring rate by building upon the fundamental reward function.
[46]	2023	✓	✓	✓	Aims that robot movement parameters such as speed and anmgle are more precisely adjusted by the Q-Learning algorithm.
[3]	2024	×	×	✓	Introduces a Probability-Based Strategy (PBS) for autonomous soccer robots, offering real-time decision-making and adaptability without relying on predefined plays. PBS outperforms traditional architectures like Skills, Tactics, and Plays (STP) in flexibility, implementation time, and strategy customization.
[5]	2024	×	×	✓	The RoboCup Simulation League involves developing AI and team strategy for eleven autonomous software agents playing soccer in a virtual 2D environment, with a central server providing game information and communication, challenging agents to make rapid decisions within 100 ms cycles based on noisy sensor input.
[18]	2024	✓	×	✓	Conducts a systematic review at the intersection of sports analytics and AI research, exploring soccer analysis, evaluation, and decision-making. Emphasizes the observation-orientation-decision-action (OODA) loop in soccer match analytics, covering both real-world and virtual domains, and discusses the potential of bridging the gap between them for improved analysis and decision-making paradigms.

3- Environmental Modeling

In this paper, we present a novel approach to enhance the navigation and obstacle avoidance capabilities of an autonomous soccer robot using eligibility traces. Eligibility traces are a reinforcement learning technique that allows the robot to learn from past experiences and make informed decisions in real-time. The primary objective of our research is to improve the robot's ability to navigate through a soccer field while avoiding obstacles effectively. By integrating eligibility traces into the robot's control system, we aim to enhance its decision-making process, enabling it to adapt to dynamic environments and make intelligent navigation choices. To achieve this, we propose a navigation policy that combines the principles of eligibility traces with obstacle detection and avoidance algorithms. The robot leverages its sensory inputs, such as vision and proximity sensors, to perceive the environment, detect obstacles, and make informed navigation decisions based on the learned eligibility traces. Our experiments and evaluations demonstrate the effectiveness of the proposed approach. The robot exhibits improved navigation performance, successfully avoiding obstacles while efficiently reaching its target positions on the soccer field. The integration of eligibility traces into the robot's control system enables it to adapt and learn from its experiences, leading to more refined navigation strategies over time. The findings of this research contribute to the advancement of autonomous robotics, specifically in the domain of soccer-playing robots. The utilization of eligibility traces in combination with obstacle avoidance algorithms offers a promising approach to enhancing the navigation capabilities of autonomous robots in dynamic and complex environments. The aim is to design a navigation system for soccer robots and

intelligently solve the problem of avoiding obstacles using reinforcing learning. The main advantage of using the proposed system is that by gaining experience and balancing it with exploration, the robot learns how to deal with unexpected situations in a dynamic environment. Considering the possible obstacles for a robot in a soccer game, the aim of performing all the movements is defined as follows:

1. Making robots move towards the ball in the shortest possible time and trace while avoiding the obstacles.

2. Transferring the ball by the robot to the goal line in the shortest possible trace considering the methodology of avoiding colliding with obstacles.

To make the performance simple, only one-half of the field is considered for examination. The rules of the other half, like the first half, can be derived and generalized. This method has been widely employed both in scientific and practical realms and has been commonly utilized across numerous references such as [39], [50], and [51]. [50] introduces Half Field Offense (HFO), a streamlined benchmark in the RoboCup 2D simulation domain, facilitating quick prototyping and assessment of AI, machine learning, and multiagent systems, with standardized interfaces for single and multiagent learning, ad hoc teamwork, and imitation learning experiments, accompanied by reinforcement learning agent benchmark results on diverse HFO tasks. In another instance, [51] presents PLASTIC, a versatile algorithm designed to tackle the ad hoc teamwork challenge, where an agent cooperates with diverse teammates to achieve a common objective. The algorithm, assessed in pursuit scenarios and RoboCup 2D soccer simulations, dynamically adjusts to new teammates by leveraging insights gained from previous interactions and efficiently capitalizing on similarities in teammates' behaviors. The Half Field environment, preserving soccer's inherent complexities, limits the agent's focus on decision-making. It offers consistent interfaces for interacting with the environment and other agents, alongside standardized tools to assess performance. Attention to half of the field, in addition to the mentioned advantage, has the following benefits:

1. Increasing simulation speed: Since a match can be performed only on half of the field, tests can be performed faster and focus on the goal of the study.

2. Ease in recording events and variables: Because a knowledge base can be considered that allows us to directly access information such as gateways, rivals, and other relevant information during the test.

Also, the existence of obstacles such as the rival robot, teammate robot, and other obstacles in the field leads to an increase in the number of different states as well as in calculations. An initiative solution for this purpose is presented in this article, in which first, all obstacles except the gateway are ignored. The field environment tested is shown in Figure 2 taking into account the considerations expressed, where R is the place home of the robot, and the ball reaches the goal by reaching place homes 2 and 3.

Mentioning this point is crucial: Even when facing an empty goal, none of the agents achieve a perfect scoring rate consistently. This phenomenon arises due to the RoboCup 2D simulator's incorporation of noise into the agents' perceptions and actions, resulting in occasional missed shots. Upon introducing a goalkeeper, the scoring efficiency of all offensive agents significantly declines, except for SARSA. Correspondingly, the average attempts required to score a goal increased by 22, showcasing the substantial difficulty disparity between these two scenarios. However, playing as a Keeper isn't any easier. As the results demonstrate, an offensive Helios-agent performs just as effectively when scoring against an empty goal as it does against a randomly acting agent.

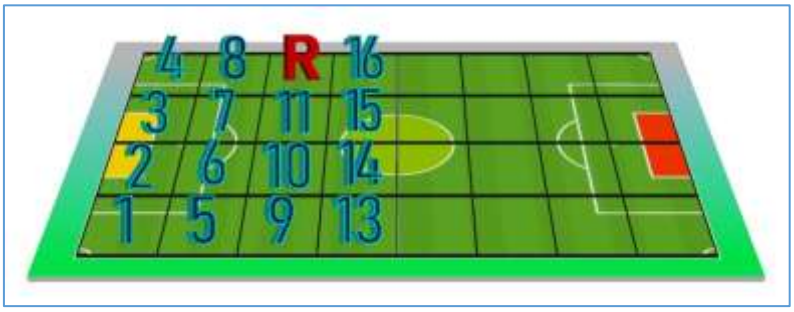

Figure 2. The field environment was tested.

After determining the eligibility traces (to move towards the ball and make goals) based on the current situation, they are investigated for the place homes that have the same cumulative rewards, in terms of their occupation, and the agent moves from right to left in these homes, with the priority of empty homes. This approach results in a significant decrease in calculations as well as simplicity and clarity in the expression of eligibility traces. In the next section, this issue will be addressed in more detail.

4- Determining the Eligibility Traces and Analyzing the Results

Our studies have led us to the fact that the Python programming language is currently the most suitable language and optimal choice for implementing aspects related to RoboCup. For instance, [52] introduces Pyrus, an open-source, Python-based platform specifically crafted for the 2D simulation of RoboCup Soccer (SS2D). It overcomes the limitations of C++ base codes by offering a user-friendly environment, empowering researchers, including beginners, to effectively develop ideas and incorporate machine learning algorithms into their teams for the annual computer-based soccer world cup. Pyrus is publicly accessible on GitHub under the MIT License. Therefore, the experiments are conducted in a Python programming environment in 4 steps. In the first step, regardless of the obstacles, the eligibility traces are determined to get the robot to the ball and finally to score in different situations. During the game, depending on the prevailing conditions, any of the eligibility traces obtained can be selected. In this experiment, the reward value of the target house (Goal) was zero, the reward value for moving away from the ball, -100, the reward value of each step, -1, and the discount rate, 0.1. In the mentioned numerical value, the effect of the discount rate is such that the agent with a coefficient of 0.9 moves

forward according to its quasi-greedy soft policy and experiences new states with a coefficient of 0.1 so as not to be caught in the local optimization. The experiments were performed with experiencing 80,000 replications, the results of which are shown below. It is should be noted that the optimal policy was achieved in the number of repetitions of about 57,800 rounds.

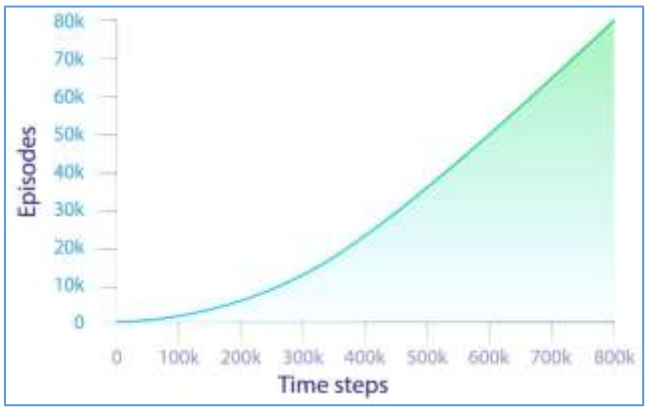

Figure 3. Timeline diagram based on the number of experiences.

Figure 3 shows a timeline diagram based on the number of experiences. In the interpretation of the learning agent's performance in this diagram, it can be said that: becoming closer to the vertical axis indicates that our agent completes the episode with a higher number of states, and the more episodes it completes, the more negative scores it gets and the fewer rewards it gets in total. As can be seen in this diagram, at first the slope of the diagram is low and closer to the horizontal axis, thus, as the learning agent is learning, it has no idea at first and takes many steps, but as it goes forward and learns, the slope of the diagram increases and its speed gets better as well.

In a typical example of this paper, according to the experiments, a numerical value of 4 was obtained for the optimal number of steps to determine the eligibility traces. The results based on the number of different steps are shown in Figure 4 in detail.

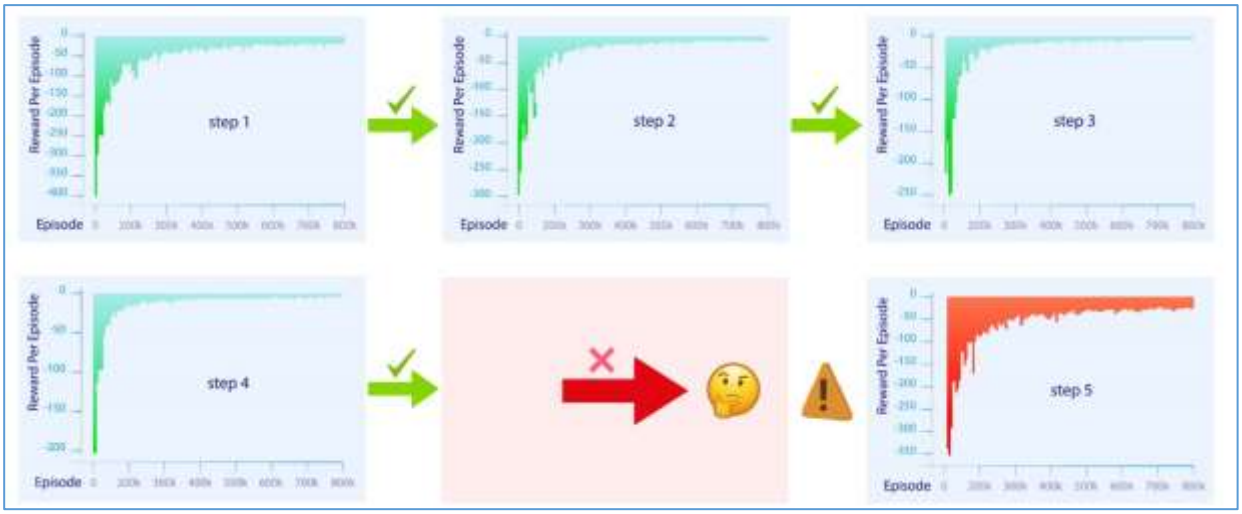

Figure 4. Test result with the number of kind of steps.

As can be seen, as the number of steps increases, the diagram reaches the optimal value more rapidly (the right top of the diagram), and finally, it gets closer to the ideal value, i.e. 100% (the left top). As mentioned earlier, theoretically, if the number of steps is considered too many, some of the states of bad traces get positive rewards as well, which is not desirable. If the number of steps is considered very small, only the states that are close to the goal receive positive effects. Of course, it should be said that the optimal number of steps can vary depending on the problem; in other words, it also depends on the conditions of the problem. As can be seen in Figure 4, by increasing the number of steps suddenly to 5, the learning agent's behavior has not only not been improved in terms of both the convergence speed and the achievement of the ideal value but also has become worse. This emphasizes the importance of fine-tuning the number of steps based on the problem's complexity. The integration of eligibility traces enables the robot to learn from past experiences and dynamically adjust its navigation strategy. Eligibility traces, as depicted in Figures 3 and 4, highlight the learning agent's progression over experiences. Initially tentative and exploratory, the agent refines its behavior, evidenced by an increasing slope in the timeline diagram.

In the second stage of our experimentation, we introduced a layer of complexity by incorporating obstacles into the soccer field environment. The primary objective was to evaluate the robot's decision-making prowess when confronted with obstacles, particularly when faced with the task of choosing between place homes that shared identical cumulative rewards. The placement followed the same priority order established in the initial stage, emphasizing a seamless transition from right to left. The robot's task in this stage was to navigate around the introduced obstacles, showcasing adaptability and strategic decision-making. According to this policy, a number of 761 eligibility traces were obtained. To simplify the expression of eligibility traces, we have considered an innovative method, which we will describe in detail below. In this regard, the <> sign indicates the simultaneous relocations of the ball and the robot, and the underline sign was used under the place homes that have the same cumulative rewards so that placing the robots in these place homes in the first stage has the same priority, but in this stage, it will be different due to the impact of applying the obstacles. Also, the cumulative reward for the place homes in parentheses is a downward value from left to right. Next, similar eligibility traces were placed in a row. In the comprehensive set of 761 eligibility traces obtained in this stage, we implemented an innovative approach to condense and organize similar traces into rows. This process, showcased in Tables 2 to 17, effectively reduced the number of traces to 211. This not only facilitated a clearer interpretation of results but also optimized memory usage, a crucial consideration for real-world implementation. This reduction in the number of eligibility traces saves significant memory as well as increases performance when implemented in the real environment. To better understand, a column is placed in the relevant tables to express the number of summarized eligibility traces along with the summary process. Each row in these tables represents a unique scenario, depicting a distinct decision-making process for the robot. This comprehensive approach not only optimizes memory

utilization but also ensures a systematic and strategic decision-making process for the robot, even in the presence of obstacles. To further explain the issue, the interpretation of the two rows of traces is given below, e.g.

- For row 3: If the conditions of this trace were met since the position of the robot is a place home 1 and the position of the ball is a place home 12, as per the eligibility trace, the robot strategically relocates itself to either place home 6 or 2 (prioritizing the initially empty one) before proceeding to place home 5. This decision is influenced by the identical cumulative rewards observed for place homes 6 and 2 during the experiment.
  - Conditions:
    The position of the robot is in place home 1.
    The position of the ball is in place home 12.
  - Eligibility Trace:
    The robot, based on the eligibility trace obtained, prioritizes placing itself in either place home 6 or 2 (both initially empty) and subsequently in place home 5.
  - Explanation:
    The cumulative reward obtained for place homes 6 and 2 during the experiment was identical.
    The robot, following the strategy, chooses a path that optimizes its movement towards the goal (place home 5) based on the previously learned eligibility trace.
- For row 19: If the conditions of this trace were met since the position of the robot is a place home 2 and the position of the ball is a place home 6, following the eligibility trace, the robot first maneuvers toward the ball to approach the goal. Subsequently, it prioritizes placement in one of the place homes (3, 7, 11, 10, 5, or 9, based on availability) due to identical cumulative rewards for these homes, arranged in a descending order from left to right.
  - Conditions:
    The position of the robot is in place home 2.
    The position of the ball is in place home 6.
  - Eligibility Trace:
    The robot, according to the eligibility trace, first moves towards the ball (Goal) and then, by prioritizing, places itself in one of the place homes 3, 7, 11, 10, 5, or 9 (if empty).
  - Explanation:
    The cumulative reward obtained for these place homes follows a downward value from left to right.
    The robot intelligently selects a path that involves reaching the Goal first and then strategically placing itself in the available place homes based on the learned eligibility trace.

These examples showcase the robot's adaptive decision-making process, considering both its current position and the ball's position, while also incorporating the cumulative rewards associated with

potential movements. The prioritization mechanism ensures that the robot optimally selects its path, demonstrating a scientifically grounded approach to obstacle-rich navigation in dynamic environments.

Table 2. Eligibility traces for robot movement - location of the robot, place home 1.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
1	1	3-4-7-8	2-6-5	1+1+1+1=4
2	1	11	6-2-5	2×1=2
3	1	12	6-2-5	2×1=2
4	1	9-10-13-14	6-5-2	(1+1+1+1)×(2×1)=8
5	1	2	Goal	1
6	1	6	2-3-(7-5-1<>6-10-11-9)	1
7	1	5	2-6-1<>5	1

Table 3. Eligibility traces for robot movement - location of the robot, place home 2.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
8	2	4	3-7-6-5-1	(2×1)+(2×1)=4
9	2	8	7-3-6-5-1	(2×1)+(2×1)=4
10	2	12	7-3-6 -5-1	(2×1)+(2×1)=4
11	2	11	7-6-3-5-1	(2×1)+(2×1)=4
12	2	10	6-5-7-3-1	(2×1)+(2×1)=4
13	2	9	5-6-1-7-3	(2×1)+(3×2×1)=8
14	2	16	7-6-3-5-1	(2×1)+(2×1)=4
15	2	15	7-6-5 -3-1	(3×2×1)+(2×1)=8
16	2	14	6-5-7-3-1	(2×1)+(2×1)=4
17	2	13	5-6-7-1-3	1
18	2	3	Goal	1
19	2	6	2<>6-(3-7-11-10-5-9)	1
20	2	7	2<>7-(3-6-8-11-10-12)	1
21	2	5	2<>5-(6-1-10-9)	1
22	2	1	2<>1-(6-5)	1

Table 4. Eligibility traces for robot movement - location of the robot, place home 3.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
23	3	1	2-6-7-8-4	(2×1)+(2×1)=4
24	3	5	6-2-7-8-4	(2×1)+(2×1)=4
25	3	9	6-2-7-8-4	2×1=2
26	3	10	6-7-2-8-4	(2×1)+(2×1)=4
27	3	11	7-8-6-2-4	(2×1)+(2×1)=4
28	3	12	8-7-4-6-2	(2×1)+(3×2×1)=8
29	3	13	6-7-2-8-4	(2×1)+(2×1)=4
30	3	14	6-7-8-2-4	(3×2×1)+(2×1)=8
31	3	15	7-8-6-2-4	(3×2×1)+(2×1)=8
32	3	16	8-7-6-4-2	3×2×1=6
33	3	7	3<>7-(2-6-8-11-10-12)	1
34	3	2	Goal	1
35	3	4	3<>4-(7-8)	1
36	3	6	3<>6-(2-7-5-1-10-11-9)	1

37	**3**	**8**	**3<>8-(7-4-11-12)**	**1**

Table 5. Eligibility traces for robot movement - location of the robot, place home 4.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
38	**4**	**12-16**	**8-7-3**	**1+1=2**
39	**4**	**11-15**	**8-7-3**	**(1+1)×(2×1)=4**
40	**4**	**10**	**7-8-3**	**2×1=2**
41	**4**	**14**	**7-8-3**	**2×1=2**
42	**4**	**5-6**	**7-3-8**	**(1+1)×(2×1)=4**
43	**4**	**1-2**	**3-7-8**	**(1+1)×(2×1)=4**
44	**4**	**3**	**Goal**	**1**
45	**4**	**7**	**3-2-(6-8-4<>7-10-11-12)**	**1**
46	**4**	**8**	**3-(7-4<>8-11-12)**	**1**

Table 6. Eligibility traces for robot movement - location of the robot, place home 5.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
47	**5**	**13**	**9-10-6-2-1**	**(2×1)+(2×1)=4**
48	**5**	**14**	**10-9-6-2-1**	**(2×1)+(2×1)=4**
49	**5**	**15**	**10-9-6-2-1**	**2×1=2**
50	**5**	**11**	**10-6-9-2-1**	**(2×1)+(2×1)=4**
51	**5**	**7**	**6-2-10-9-1**	**(2×1)+(2×1)=4**
52	**5**	**3**	**2-6-1-10-9**	**(2×1)+(3×2×1)=8**
53	**5**	**16**	**10-6-9-2-1**	**(2×1)+(2×1)=4**
54	**5**	**12**	**10-6-2-9-1**	**(3×2×1)+(2×1)=8**
55	**5**	**8**	**6-2-10-9-1**	**(3×2×1)+(2×1)=8**
56	**5**	**4**	**2-6-10-1-9**	**3×2×1=6**
57	**5**	**2**	**Goal**	**1**
58	**5**	**1**	**2-6-5<>1**	**1**
59	**5**	**6**	**2-3-(7-5<>6-1-11-10-9)**	**1**
60	**5**	**10**	**6-7-5<>10-(11-9-15-14-13)**	**1**
61	**5**	**9**	**6-5<>9-(10-14-13)**	**1**

Table 7. Eligibility traces for robot movement - location of the robot, place home 6.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
62	**6**	**4**	**3-7-2-11-10-1-5-9**	**(2×1)+(2×1)=4**
63	**6**	**8**	**7-3-11-2-10-9-1-5**	**3×2×1=6**
64	**6**	**16**	**11-7-10-3-9-2-5-1**	**(2×1)+(2×1)+(2×1)=6**
65	**6**	**12**	**7-11-10-3-2-9-5-1**	**(2×1)+(3×2×1)+(2×1)=10**
66	**6**	**15**	**11-10-7-9-5-3-2-1**	**(2×1)+(3×2×1)+(2×1)=10**
67	**6**	**14**	**10-11-9-7-5-1-3-2**	**(3×2×1)+(2×1)+(3×2×1)=14**
68	**6**	**13**	**10-9-5-7-11-1-2-3**	**3×2×1=6**
69	**6**	**7**	**3-2-(6<>7-8-4-11-10-12)**	**1**
70	**6**	**2-3**	**Goal**	**1+1=2**
71	**6**	**5**	**2-(6<>5-1-10-9)**	**1**
72	**6**	**1**	**2-(6<>1-5)**	**1**
73	**6**	**9**	**6<>9-(5-10-14-13)**	**1**
74	**6**	**10**	**6<>10-(7-5-11-9)**	**1**

75	**6**	**11**	**6<>11-(7-8-10-12-15-14-16)**	**1**

Table 8. Eligibility traces for robot movement - location of the robot, place home 7.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
76	**7**	**16**	**12-11-8-10-6-4-3-2**	**(2×1)+(2×1)=4**
77	**7**	**15**	**11-12-10-8-6-2-4-3**	**3×2×1=6**
78	**7**	**13**	**10-11-6-12-2-8-3-4**	**(2×1)+(2×1)+(2×1)=6**
79	**7**	**14**	**11-10-6-12-8-2-3-4**	**(2×1)+(3×2×1)+(2×1)=10**
80	**7**	**9**	**10-6-11-2-3-12-8-4**	**(2×1)+(3×2×1)+(2×1)=10**
81	**7**	**5**	**6-10-2-11-3-4-12-8**	**(3×2×1)+(2×1)+(2×1)=10**
82	**7**	**1**	**6-2-3-11-10-4-8-12**	**3×2×1=6**
83	**7**	**2-3**	**Goal**	**1+1=2**
84	**7**	**6**	**2-(5-7<>6-10-11-9)**	**1**
85	**7**	**4**	**3-(7<>4-8)**	**1**
86	**7**	**8**	**3-(7<>8-4-11-12)**	**1**
87	**7**	**11**	**7<>11-(6-8-10-12-15-14-16)**	**1**
88	**7**	**10**	**7<>10-(6-5-11-9-14-15-13)**	**1**
89	**7**	**12**	**7<>12-(8-11-15-16)**	**1**

Table 9. Eligibility traces for robot movement - location of the robot, place home 8.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
90	**8**	**16**	**12-11-7-3-4**	**(2×1)+(2×1)=4**
91	**8**	**15**	**11-12-7-3-4**	**(2×1)+(2×1)=4**
92	**8**	**14**	**11-12-7-3-4**	**(2×1)+(2×1)=4**
93	**8**	**10**	**11-7-12-3-4**	**(2×1)+(2×1)=4**
94	**8**	**6**	**7-3-11-12-4**	**(2×1)+(2×1)=4**
95	**8**	**2**	**3-7-4-11-12**	**(2×1)+(3×2×1)=8**
96	**8**	**13**	**11-7- 12-3-4**	**(2×1)+(2×1)=4**
97	**8**	**9**	**11-7-3 -12-4**	**(3×2×1)+(2×1)=8**
98	**8**	**5**	**7- 3-11-12-4**	**(2×1)+(2×1)=4**
99	**8**	**1**	**3-7-11-4-12**	**3×2×1=6**
100	**8**	**3**	**Goal**	**1**
101	**8**	**4**	**3-(7-8<>4)**	**1**
102	**8**	**7**	**3-(6-8<>7-4-11-10-12)**	**1**
103	**8**	**11**	**7-6-8<>11-(10-12-15-14-16)**	**1**
104	**8**	**12**	**7-12<>8-(11-15-16)**	**1**

Table 10. Eligibility traces for robot movement - location of the robot, place home 9.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
105	**9**	**1**	**5-6-10-14-13**	**(2×1)+(2×1)=4**
106	**9**	**2**	**6-5-10-14-13**	**(2×1)+(2×1)=4**
107	**9**	**3**	**6-5-10 -14-13**	**(2×1)+(2×1)=4**
108	**9**	**7**	**6-10-5-14-13**	**(2×1)+(2×1)=4**
109	**9**	**11**	**10-14-6 -5-13**	**(2×1)+(2×1)=4**
110	**9**	**15**	**14-10-13-6-5**	**(2×1)+(3×2×1)=8**
111	**9**	**4**	**6-10-5-14-13**	**(2×1)+(2×1)=4**
112	**9**	**8**	**6-10-14-5-13**	**(3×2×1)+(2×1)=8**

113	**9**	**12**	**10-14-6-5-13**	**(2×1)+(2×1)=4**
114	**9**	**16**	**14-10-6-13-5**	**3×2×1=6**
115	**9**	**5**	**2-(1-6-10-9<>5)**	**1**
116	**9**	**6**	**2-3-(7-5-1-10-11-6<>9)**	**1**
117	**9**	**10**	**6-5-7-(11-9<>10-14-15-13)**	**1**
118	**9**	**14**	**10-11-9<>14-(15-13)**	**1**
119	**9**	**13**	**10-9<>13-14**	**1**

Table 11. Eligibility traces for robot movement - location of the robot, place home 10.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
120	**10**	**1**	**5-6-9-7-11-13-14-15**	**(2×1)+(2×1)=4**
121	**10**	**2**	**6-5-7-9-11-15-13-14**	**3×2×1=6**
122	**10**	**4**	**7-6-11-5-15-9-14-13**	**(2×1)+(2×1)+(2×1)=6**
123	**10**	**3**	**6-7-11-5-9-15-14-13**	**(2×1)+(3×2×1)+ (2×1)=10**
124	**10**	**8**	**7-11-6-15-14-5-9-13**	**(2×1)+(3×2×1)+ (2×1)=10**
125	**10**	**12**	**11-7-15-6-14-13-5-9**	**(3×2×1)+(2×1)+(3×2×1)=14**
126	**10**	**16**	**11-15-14-6-7-13-9-5**	**3×2×1=6**
127	**10**	**6**	**2-3-(7-5-10<>6-11-9-14-15-13)**	**1**
128	**10**	**7**	**3-2-(6-8-4-10<>7-11-12)**	**1**
129	**10**	**11**	**6-7-8-(12-10<>11-15-14-16)**	**1**
130	**10**	**5**	**2-6-1-(10<>5-9-14-15-13)**	**1**
131	**10**	**9**	**6-5-(10<>9-14-13)**	**1**
132	**10**	**13**	**9-10<>13-14**	**1**
133	**10**	**15**	**11-12-10<>15-(14-16)**	**1**

Table 12. Eligibility traces for robot movement - location of the robot, place home 11.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
134	**11**	**13**	**14-10-15-6-7-16-12-8**	**(2×1)+(2×1)=4**
135	**11**	**9**	**10-14-6-15-7-8-16-12**	**3×2×1=6**
136	**11**	**1**	**6-10-7-14-8-15-12-16**	**(2×1)+(2×1)+(2×1)=6**
137	**11**	**5**	**10-6-7-14-15-8-12-16**	**(2×1)+(2×1)+(2×1)=6**
138	**11**	**2**	**6-7-10-8-12-14-15-16**	**(2×1)+(2×1)+(2×1)=6**
139	**11**	**3**	**7-6-8-10-12-16-14-15**	**(3×2×1)+(2×1)+(3×2×1)=14**
140	**11**	**4**	**7-8-12-10-6-16-15-14**	**3×2×1=6**
141	**11**	**7**	**3-2-(4-6-8-11<>7-12-10)**	**1**
142	**11**	**6**	**2-3-(1-5-7-9-11<>6)**	**1**
143	**11**	**8**	**3-(7-4-12-11<>8)**	**1**
144	**11**	**12**	**7-8-(11<>12-15-16)**	**1**
145	**11**	**10**	**6-7-5-(9-11<>10-14-15-13)**	**1**
146	**11**	**15**	**10-12-11<>15-(14-16)**	**1**
147	**11**	**14**	**10-9-11<>14-(15-13)**	**1**
148	**11**	**16**	**10-12-11<>16-(14-15)**	**1**

Table 13. Eligibility traces for robot movement - location of the robot, place home 12.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
149	12	4	8-7-11-15-16	(2×1)+(2×1)=4
150	12	3	7-8-11-15-16	(2×1)+(2×1)=4
151	12	2	7-8-11-15-16	2×1=2
152	12	6	7-11-8-15-16	(2×1)+(2×1)=4
153	12	10	11-15-7-8-16	(2×1)+(2×1)=4
154	12	14	15-11-16-7-8	(2×1)+(3×2×1)=8
155	12	1	7-11-8-15-16	(2×1)+(2×1)=4
156	12	5	7-11-15-8-16	(3×2×1)+(2×1)=8
157	12	9	11-15-7-8-16	(3×2×1)+(2×1)=8
158	12	13	15-11-7-16-8	3×2×1=6
159	12	8	3-(4-7-11-12<>8)	1
160	12	7	3-2-(4-6-8-11-10-12<>7)	1
161	12	11	6-7-8-(10-12<>11-15-14-16)	1
162	12	15	11-10-12<>15-(14-16)	1
163	12	16	11-16<>12-15	1

Table 14. Eligibility traces for robot movement - location of the robot, place home 13.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
164	13	1-5	9-10-14	1
165	13	2-6	9-10-14	2×1=2
166	13	7	10-9-14	2×1=2
167	13	3	10-9-14	2×1=2
168	13	8-12	10-14-9	2×1=2
169	13	15-16	14-10-9	2×1=2
170	13	9	5-6-(10-14-13<>9)	1
171	13	10	6-7-5-(11-9-14-15-13<>10)	1
172	13	14	10-11-9-(15-13<>14)	1

Table 15. Eligibility traces for robot movement - location of the robot, place home 14.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
173	14	16	15-11-10-9-13	(2×1)+(2×1)=4
174	14	12	11-15-10-9-13	(2×1)+(2×1)=4
175	14	8	11-15-10-9-13	2×1=2
176	14	7	11-10-15-9-13	(2×1)+(2×1)=4
177	14	6	10-9-11-15-13	(2×1)+(2×1)=4
178	14	5	9-10-13-11-15	(2×1)+(3×2×1)=8
179	14	4	11-10-15-9-13	(2×1)+(2×1)=4
180	14	3	11-10-9-15-13	(3×2×1)+(2×1)=8
181	14	2	10-9-11-15-13	(3×2×1)+(2×1)=8
182	14	1	9-10-11-13-15	3×2×1=6
183	14	10	6-7-5-(11-9-15-14<>10-13)	1
184	14	11	7-6-8-(10-12-15-14<>11-16)	1
185	14	15	11-10-12-(14<>15-16)	1
186	14	9	6-5-(10-14<>9-13)	1

187	**14**	**13**	**10-9-(14<>13)**	**1**

Table 16. Eligibility traces for robot movement - location of the robot, place home 15.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
188	**15**	**13**	**14-10-11-12-16**	**(2×1)+(2×1)=4**
189	**15**	**9**	**10-14-11-12-16**	**(2×1)+(2×1)=4**
190	**15**	**5**	**10-14-11 -12-16**	**(2×1)+(2×1)=4**
191	**15**	**6**	**10-11-14-12-16**	**(2×1)+(2×1)=4**
192	**15**	**7**	**11-12-10 -14-16**	**(2×1)+(2×1)=4**
193	**15**	**8**	**12-11-16-10-14**	**(2×1)+(3×2×1)=8**
194	**15**	**1**	**10-11-14-12-16**	**(2×1)+(2×1)=4**
195	**15**	**2**	**10-11-12-14-16**	**(3×2×1)+(2×1)=8**
196	**15**	**3**	**11-12-10-14-16**	**(2×1)+(2×1)=4**
197	**15**	**4**	**12-11-10-16-14**	**3×2×1=6**
198	**15**	**11**	**7-6-8-(10-12-15<>11-14-16)**	**1**
199	**15**	**10**	**6-7-5-(11-9-15<>10-14-13)**	**1**
200	**15**	**12**	**7-8-(11-15<>12-16)**	**1**
201	**15**	**14**	**10-11-9-(13-15<>14)**	**1**
202	**15**	**16**	**11-12-(15<>16)**	**1**

Table 17. Eligibility traces for robot movement - location of the robot, place home 16.

Row Number	Robot's Location	Ball's Location	Eligibility Traces	Number of Eligibility Traces Obtained before Summarizing
203	**16**	**13-14**	**15-11-12**	**1+1=2**
204	**16**	**9-10**	**15-11-12**	**(1+1)×(2×1)=4**
205	**16**	**6**	**11-15-12**	**2×1=2**
206	**16**	**5**	**11-15-12**	**2×1=2**
207	**16**	**4-7**	**11-12-15**	**(1+1)×(2×1)=4**
208	**16**	**5-8**	**12-11-15**	**(1+1)×(2×1)=4**
209	**16**	**12**	**7-8-(11-15-16<>12)**	**1**
210	**16**	**11**	**7-6-8-(12-16<>11-15)**	**1**
211	**16**	**15**	**11-10-12-(14-16<>15)**	**1**

The core objective in second stage was to examine the robot's ability to discern optimal paths among place homes with equivalent cumulative rewards, considering the influence of obstacles. The following points provide a detailed expansion of the process and results of this critical stage:

1- Challenges Posed by Obstacles:

- The incorporation of obstacles mirrors real-world complexities, as robots often operate in dynamic environments where spatial constraints necessitate adaptive navigation strategies.
- Obstacles introduce an additional layer of decision-making complexity, requiring the robot to dynamically adjust its paths based on the changing spatial landscape.

2- Directional Priority Scheme - Right to Left:

- To systematically address scenarios with place homes having equivalent cumulative rewards, a directional priority scheme was implemented, emphasizing navigation decisions from right to left.
- This directional policy plays a crucial role in determining the robot's actions when confronted with multiple place homes with the same cumulative reward.

3- Symbolic Representation for Clarity:

- The symbolic representation, notably the < > signs and underlining, serves as a visual aid to enhance the clarity of decision-making scenarios.
- < > denotes simultaneous movements of the ball and the robot, and underlining highlights place homes with equivalent cumulative rewards.

4- Cumulative Reward Descent:

- Cumulative rewards associated with place homes are structured in parentheses, descending from left to right.
- This downward trend symbolizes a decreasing priority, influencing the robot to consider place homes from left to right during its navigation.

5- Path Summarization for Efficiency:

- A critical aspect of this stage involves path summarization, a technique aimed at optimizing memory usage and computational efficiency.
- Similar eligibility traces are grouped into rows, reducing redundancy and enhancing the overall efficiency of the decision-making process.

6- Interpreting Rows - Example Scenarios:

- Each row in Tables 2 to 17 encapsulates a unique scenario, representing a specific decision-making process of the robot.
- For instance, Row 3 reflects a scenario where the robot, positioned in place home 1 with the ball in place home 12, strategically moves to either place home 6 or 2, prioritizing place home 5.

7- Optimized Memory Usage:

- The summarization technique is not solely for visual clarity but plays a pivotal role in optimizing memory usage.
- The reduction from 761 to 211 traces ensures efficient memory utilization, a crucial consideration for real-world robotic implementations.

8- Real-World Applicability:

- The insights gained from Stage 2 are invaluable for real-world robotic systems, where computational resources are often limited.
- The directional priority scheme and summarization technique contribute to efficient decision-making, enabling robots to navigate effectively in complex, obstacle-laden environments.

In essence, Stage 2 not only simulates the challenges of navigating around obstacles but also demonstrates a sophisticated decision-making process that can be extrapolated to real-world scenarios. The combination of symbolic representation, prioritization policies, and path summarization establishes a robust foundation for intelligent robotic navigation in dynamic and constrained environments.

In the third stage, to increase the efficiency of the proposed system and to prevent additional calculations, a knowledge base is considered to store the desired eligibility traces obtained so far. In addition to the listed benefits, this increases the future decision-making power and helps to understand why a series of actions are performed and follow them. The system begins to learn dynamically. To perform each new move, if the existing conditions correspond to one of the conditions in the knowledge base, the previous eligibility traces will be followed without new calculations, otherwise, a new trace will be determined. If the new trace leads us to the desired goal, it will be added to the other previous traces in the knowledge base along with the environmental conditions. Since the new rules are produced and stored, they can be easily analyzed by experts, so it can be seen what behavior the robot has learned, and in a general perspective, it can be seen what series of actions it has taken in the current situation. This research emphasizes the potential of combining eligibility traces with a robust knowledge base to augment the navigation and decision-making capabilities of robots. It showcases the significant impact on robot performance in steering, obstacle avoidance, and intelligent decision-making in dynamic environments. The findings underscore the value of this approach in advancing robotics and AI systems, especially in addressing challenges posed by dynamic environments and sports competitions. The fusion of these techniques offers promising prospects for enhancing the adaptability and efficiency of robots.

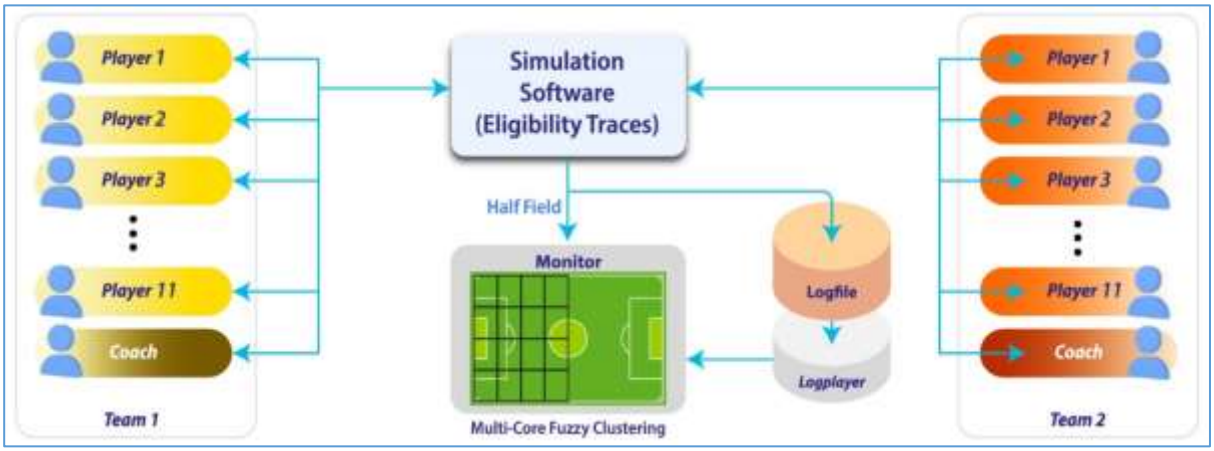

Figure 5. Soccer match simulation.

In the fourth stage, a small simulation of a soccer match is performed according to Figure 5. The multi-core fuzzy clustering method is used to map the coordinates of environmental factors to the real soccer field according to the method proposed by the authors of this paper, which is presented in our previous work in the reference number [53]. This operation is performed in such a way that the implementation environment is classified into 16 clusters so that the representative of each cluster is the respective center. When a new factor is entered, the mapping operation is performed according to the clustering

algorithm. After performing the mapping operation, the system starts based on the locations of the agents and the prevailing conditions. For further clarification, following the intricate mapping process, the system initiates its operations based on the agents' positions, defined as the centers of the clustered segments, alongside the predefined rules residing in the knowledge base. An additional explanation is that, in the dynamic and uncertain environment of the robot, certain objects possess shared attributes across multiple clusters. Addressing this scenario requires considering cluster overlaps by allocating a set of membership degrees to each object. The fuzzy nature of clusters leads to multiple membership assignments, offering a nuanced representation. Fuzzy clustering, by its nature, tends to entail reduced computational burden compared to deterministic methods, simultaneously enabling more straightforward identification and management of ambiguous, noisy, or outlier data points. Our proposed method in [53] pivots on feasibility concepts, leveraging multi-core learning to discern clusters within intricate data structures. Feasibility scores attributed to each datum reflect the extent of properties inherited from respective clusters. This methodology automatically tunes core weights within an optimization framework, circumventing pitfalls associated with inefficacious cores or irrelevant features, and ensuring the system's robustness and adaptability. In the cited reference, our extensive experimentation led us to the conclusion that the proposed method exhibited a more effective performance compared to conventional methods such as KMeans [54], Hierarchical Clustering (HC) [55], and Metric Pairwise Constrained Kmeans (MPCK-Means) [56]. These three prominent techniques offer distinct approaches to partitioning datasets within the specific domain addressed in this paper.

K-Means is an iterative clustering algorithm that partitions a dataset into k distinct, non-overlapping clusters by minimizing the sum of squared distances between data points and their respective cluster centers. MPCK-Means introduces a unique dimension by incorporating distance metric training in each iteration, utilizing both unlabeled data and pairwise constraints. This method can learn specific metrics for individual clusters, enabling diverse cluster shapes. Additionally, Hierarchical Clustering adopts a hierarchical and tree-like structure to organize data into clusters based on distance criteria. Typically using greedy algorithms, it iteratively merges or splits clusters, generating a dendrogram, a tree-like diagram illustrating hierarchical relationships between clusters. Cutting the dendrogram at various levels results in distinct clusters. We replicated our experiments, previously conducted in our prior work, with the latest advancements, leading to the findings presented in Figure 6.

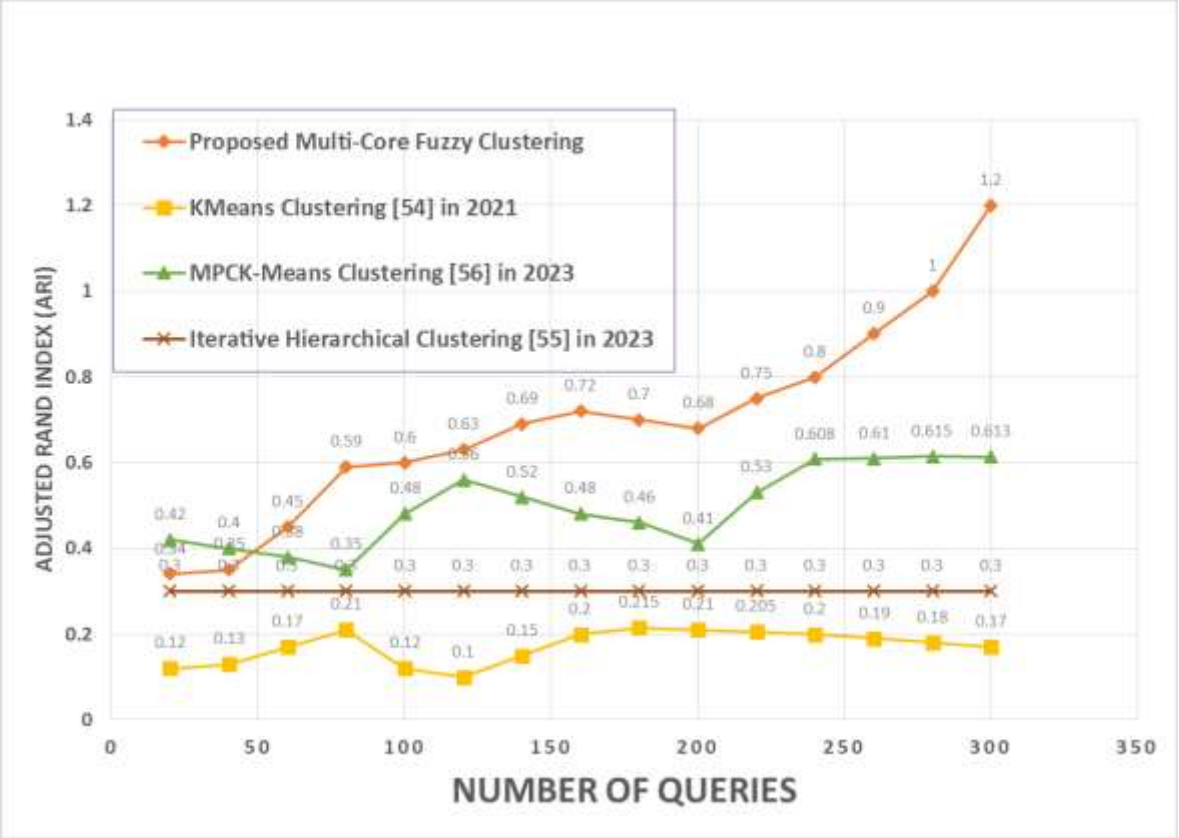

Figure 6. The performance of Multi-Core Fuzzy clustering in comparison to other methods.

The results of this stage of experiments are reported based on the queries and the Adjusted Rand Index (ARI) standard deviation. The ARI standard deviation is a dispersion indicator that shows, on average, how much the data deviate from the mean distance. If the standard deviation of a set of data is close to zero, it indicates that the data points are close to the average and have little dispersion, while a large standard deviation indicates significant data dispersion. Based on the observations from the analysis of other methods in Figure 6, it can be inferred that our proposed multi-core fuzzy clustering method outperforms other clustering methods, namely KMeans, MPC-KMeans, and Hierarchical Clustering. This indicates the robustness of the proposed method.

The superiority of our proposed method over KMeans arises from its non-linear property. This distinctive feature enables our method to model more complex and non-linear relationships within the data, in contrast to KMeans, which assumes linear relationships among the data points. This non-linear characteristic is crucial as many complex data structures cannot be well-described by linear models. Additionally, our proposed method employs a multi-core learning model. This feature allows the algorithm to work concurrently and in parallel on multiple segments of the data. Not only does this enhance the speed of clustering operations, but it also makes the algorithm feasible for large datasets. Conversely, KMeans lacks this parallelization capability, and performing clustering operations on large datasets might pose challenges and demand substantial execution time.

Upon thorough examination, the data reveals a notable disparity in the efficiency levels between our proposed method and Hierarchical Clustering. This discrepancy prompts a closer look at the

fundamental factors contributing to this difference. Hierarchical Clustering, while a widely-used method, exhibits limitations in handling datasets with intricate structures or high dimensions. The inherent nature of this method, which relies on building a hierarchy of clusters based on distance criteria, tends to falter when dealing with data that exhibits non-linear relationships. The failure to accurately model non-linear patterns within the data leads to a decrease in the effectiveness of Hierarchical Clustering. On the other hand, our proposed method outshines Hierarchical Clustering in terms of efficiency. The utilization of non-linear modeling, as well as the incorporation of a multi-core learning model, enables our approach to navigate the complexities of high-dimensional and structurally intricate datasets more adeptly. The concurrent processing capabilities of our method contribute significantly to its efficiency, especially in scenarios where Hierarchical Clustering falls short.

When contrasting our proposed method with MPCK-Means, an algorithm known for its superior performance over KMeans and Hierarchical Clustering, it becomes apparent that our method excels, particularly in datasets characterized by higher dimensions. This marked superiority raises intriguing questions about the underlying mechanisms contributing to this observed trend. MPCK-Means, in its pursuit of learning a suitable metric during clustering, showcases commendable performance, especially when compared to traditional methods like KMeans and Hierarchical Clustering. However, the strength of MPCK-Means appears to not plateau when confronted with an increase in the number of queries beyond a certain threshold. Experimental results, graphically depicted in Figure 6, illuminate this behavior, revealing that the efficiency of MPCK-Means stabilizes and fails to exhibit significant improvement beyond a specific point. In stark contrast, our proposed method not only maintains a competitive edge but showcases a discernible upward trajectory in efficiency as the number of queries increases. This upward trend highlights the adaptability and scalability of our approach, making it particularly well-suited for scenarios where the demands on the clustering algorithm intensify.

In essence, this analysis underscores the nuanced dynamics of our proposed multi-core fuzzy clustering method compared to other methods, emphasizing the advantages of our approach in scenarios where datasets possess higher dimensions and the clustering algorithm faces escalating query demands. The ability of our method to thrive under such conditions positions it as a robust and promising solution in the realm of clustering algorithms. This rigorous evaluation solidified the credibility of our approach in mapping and simulating complex environmental factors on the soccer field, providing enhanced decision-making capabilities for agents within this context. For enhanced clarity, the operational framework is presented below in a segregated manner:

1- Fuzzy Clustering Parameters: The multi-core fuzzy clustering method leverages 16 clusters to categorize the implementation environment, each with a respective center. The number of clusters is selected based on considerations of the soccer field's complexity and the need for nuanced decision-making. Parameters like distance metrics and optimization strategies are fine-tuned to ensure effective clustering.

2- Dynamic Mapping Operation: As new environmental factors are introduced, the clustering algorithm dynamically adjusts its mappings. The system incorporates dynamic adjustments during runtime, ensuring adaptability to evolving conditions. This adaptive mapping operation is pivotal for accurate decision-making in a dynamic soccer field environment.

3- Knowledge Base Utilization: The knowledge base plays a crucial role in decision-making. It contains predefined rules that govern the system's behavior. During the simulation, the system retrieves and employs information from the knowledge base in real-time. This architecture ensures that the robot's decisions align with learned eligibility traces and adapt to the specific conditions encountered.

4- Handling Cluster Overlaps: In dynamic environments, certain objects may share attributes across multiple clusters. To address this, the system allocates membership degrees to objects, considering cluster overlaps. The fuzzy nature of clusters enables nuanced representation, allowing the system to make informed decisions even in scenarios where attributes span multiple clusters.

5- Simulation Visualizations: Visual elements, such as graphs or charts, are integrated into the simulation to provide a clear representation of the system's behavior. Key decision points, environmental changes, and the robot's responses are visually highlighted, offering an intuitive understanding of the simulation.

6- Comparison with Conventional Methods: The proposed multi-core fuzzy clustering method outperforms traditional techniques like KMeans, Iterative Hierarchical Clustering (IHC), and MPCK-Means. Performance metrics and qualitative observations from extensive experiments confirm the method's efficacy in mapping and simulating complex environmental factors on the soccer field.

7- Real-World Relevance: Insights gained from the simulation hold significant real-world applications. The combination of fuzzy clustering and a knowledge base contributes to efficient and adaptive robotic navigation. The system's decision-making process, informed by eligibility traces and dynamic mappings, exhibits promising potential for addressing challenges in dynamic environments and sports competitions.

5- Conclusion

One of the most effective techniques for training autonomous robots is reinforcement learning. Through this approach, robots can acquire the skill to make optimal decisions without relying on intricate programming or strict, pre-defined instructions. Consequently, this method proves advantageous in teaching intricate and multifaceted robotic behaviors. In scenarios like RoboCup competitions, the adoption of this technique can significantly aid in acquiring diverse behavioral patterns. In this paper, an application of the underlying algorithm, a new method for determining eligibility traces, solving navigation problems, and avoiding obstacles by soccer-playing robots was presented. In the proposed method, only one-half of the field was considered for experiments. The resulting traces can be

generalized to the whole field. Also, to optimize and increase the efficiency of the algorithm, obstacles such as rival robots and team robots as well as obstacles on the field, were first ignored (and eligibility traces were determined only by considering the locations of the robot, ball, and gateway) and then is considered. This approach resulted in a significant decrease in computational burden. In total, the experiments were performed in 4 stages. In the first stage, the eligibility traces for the robot to move towards the ball, as well as to score goals regardless of obstacles were obtained. During this stage, a diagram of the results of the experiments was presented and analyzed. In the second stage, after applying obstacles and performing simplification to show the traces, 211 eligibility traces were obtained. In the third stage, the desired eligibility traces leading to the goal were stored in the knowledge base and used in the continuation of the game process. The applications of this knowledge base include: adapting the current conditions to the previous ones and following the previous eligibility traces, understanding why the series of actions are performed, and increasing the expert agent's decision-making power. In the fourth stage, to simulate a real soccer match, the positions of the agents were mapped to the real field based on the clustering method proposed by the authors of this paper, and a small match was arranged. Online experience and high decision-making power in the face of new situations were the key achievements of this paper in the game. Finally, the initiatives of this paper can be summarized as follows:

1. Modeling the environment, removing half of the field symmetrically, and generalizing the eligibility traces obtained to the whole field,
2. Determining the eligibility traces regardless of all the obstacles,
3. Adopting a method for determining eligibility traces considering obstacles with a minimum computational burden,
4. Summarizing and displaying the number of 761 eligibility traces in 211 rows,
5. The possibility of pursuing the reason for performing a series of actions while storing the desired eligibility traces in the knowledge base, and
6. Carrying out simulated math and mapping the agents to the real field.

This approach aims to enhance skill learning in a scalable manner, which will eventually contribute to the development of a foundational model for soccer robotics. The findings of this research can be summarized as follows:

1. Eligibility Traces for Navigation: Successfully determined eligibility traces allowing robots to navigate towards the ball and score goals effectively, considering obstacles and environmental complexities.
2. Efficient Path Planning: Developed a method to reduce computational burden while maintaining efficiency in obstacle avoidance and path planning.
3. Knowledge Base Implementation: Established a knowledge base system for storing and reusing learned paths, improving decision-making in similar scenarios.

4. Real-world Mapping and Simulation: Successfully mapped robot positions and environmental factors to simulate a real soccer match scenario, showcasing practical applicability.
5. Efficiency in Learning and Decision-making: Achieved online experience learning and enhanced decision-making capabilities for robots facing new or changing conditions.

This meticulous approach ensures the soccer robot's ability to make informed decisions in a dynamic environment, considering both cumulative rewards and obstacle navigation. The resulting condensed eligibility traces maintain a strategic decision-making process while optimizing memory usage for real-world implementation.

6- Future Works and Practical Suggestions

The official website for RoboCup competitions can be found at the reference [57]. This platform serves as a comprehensive resource where the rules and regulations governing each league's specific guidelines in the competitions are explicitly outlined. In the context of small-size soccer robots [58], certain design features become crucial for effective gameplay. For instance, these robots commonly incorporate the 'Rotating Dribble' feature, a key element found in robots designed to skillfully handle and manipulate the ball during matches [59]. This dynamic capability significantly influences the strategic aspects of the game, allowing the robots to showcase advanced ball control techniques and enhance their overall performance on the field. The "Rotating Dribble" refers to the rotation of the ball around the robot, enabling it to control and carry the ball. By utilizing this feature, the robot can effectively maintain possession of the ball and create scoring opportunities. The ability to rotate the body and ball around itself while in motion is crucial for ball retention and creating favorable goal-scoring situations [60]. In this study, in the second stage of the quadruple experiments, after the obstacles were involved, we used the right-to-left prioritization of the place homes not currently occupied to investigate the possibility of the robot being placed in the place homes with the same cumulative rewards. To make the implemented method more intelligent and realistic, this prioritization can be applied by considering the proximity of the ball to the dribbling part of the robot as in Figure 7, the details of which will be presented in future works.

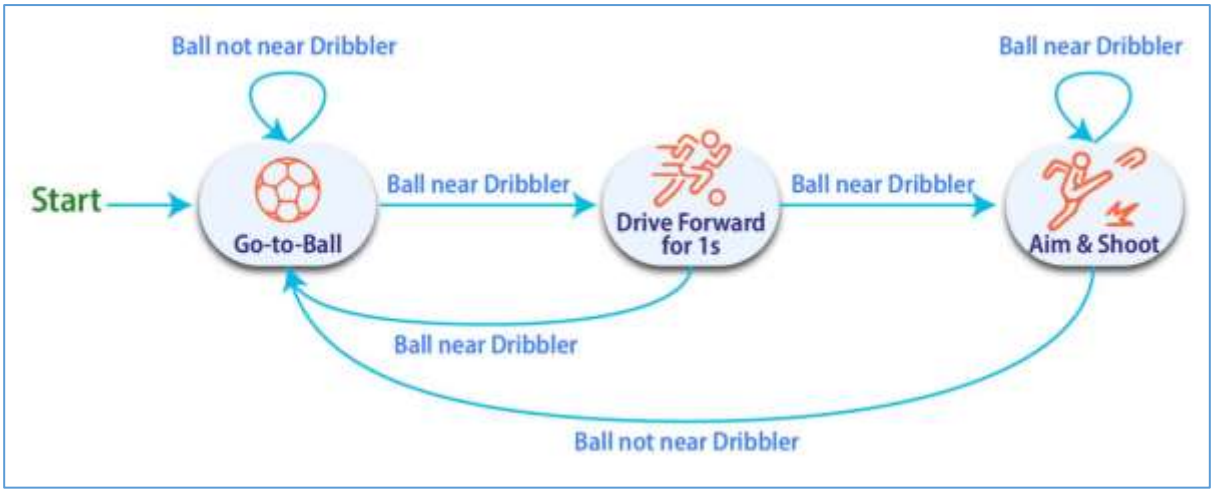

Figure 7. Considering the position of the ball relative to the dribbler part of the robot in the prioritization.

Briefly, future endeavors encompass a comprehensive analysis of the "Rotating Dribble" feature's impact on ball control and retention. we underscored its substantial influence, showcasing how the rotational movement around the robot empowers optimal ball possession and strategic goal-scoring opportunities. Moreover, further enhancements will focus on adapting algorithms specifically tailored to harness the potential of the "Rotating Dribble" feature. This adaptation significantly augments the robot's ball control, strategically enhancing its maneuverability and retention tactics during gameplay. A critical aspect will involve a detailed assessment of the "Rotating Dribble" feature's effectiveness across diverse game conditions. This analysis aims to elucidate its remarkable adaptability and efficacy, showcasing its pivotal role in dynamic gameplay scenarios. Additionally, the future scope involves refining prioritization methods for robot placement in areas with comparable cumulative rewards. By augmenting decision-making paradigms and considering the proximity of the ball relative to the robot's rotational component, this refinement aims to optimize strategic positioning tactics during gameplay.

Revision and enhancement of qualified search models can significantly optimize robots' decision-making in dynamic and obstacle-rich environments. Establishing a knowledge repository, more complete than what has been mentioned in this article, to store effective decision patterns when facing obstacles and diverse conditions can augment efficiency and advance intelligent decision-making for robots. Employing virtual reality simulations and virtual learning frameworks [61] can further enhance the accuracy and efficacy of decision algorithms in real-world settings. Furthermore, for robots to effectively perform specialized tasks alongside humans in a dynamic environment, a learning system that encourages seamless interaction between humans and robots becomes indispensable. This integration facilitates the training of robots by individuals without specialized expertise. Especially, a newly emerging strategy, referred to as "human-in-the-loop," integrates human expertise into the system. This approach has garnered significant attention as a critical element in the evolution of human-centric AI, as underscored in recent studies like [62].

Declarations

Ethical Approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. This article does not contain any studies with animals performed by any of the authors.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' contributions

Seyed Omid Azarkasb and Seyed Hossein Khasteh. These authors contributed equally to this work.

Authors and Affiliations

K.N. Toosi University of Technology, Tehran, Iran.

Seyed Omid Azarkasb

K.N. Toosi University of Technology, Tehran, Iran.

Seyed Hossein Khasteh

Corresponding author

Correspondence to Seyed Omid Azarkasb

Funding

This research was not funded.

Availability of data and materials

A significant amount of data is addressed in this article. The remaining data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions. The authors declare that all the experimental data in this paper are true and valid. Moreover, The authors declare that all experimental data are obtained from detailed experiments.

References

[1] Escobar-Naranjo. J., G. Caiza, P. Ayala, E. Jordan, C.A. Garcia, M.V. Garcia, **"Autonomous Navigation of Robots: Optimization with DQN"**, MDPI, Applied Sciences, Vol. 13, No. 12, 2023.

[2] Ye. J., N. Kang, B. Guan, S. Cai, T. Zhang, Y. Yang, **"Application of Robot Obstacle Avoidance Algorithm and Attack Strategy on ROS"**, Published under licence by IOP Publishing Ltd, Journal of Physics: Conference Series, Vol. 2456, No. 1, The 2nd International Conference on Robotics, Automation and Intelligent Control, Changsha, China, 2023.

[3] Ribeiro, A.F.A, A.C.C. Lopes, T.A. Ribeiro, N.S.S.M. Pereira, G.T. Lopes, A.F.M. Ribeiro, **"Probability-Based Strategy for a Soccer Multi-Agent Autonomous Robot System"**, MDPI, Robotics, Vol. 13, No. 1, 2024.

[4] Smit, A., H.A. Engelbrecht, W. Brink, A. Pretorius, **"Scaling Multi-Agent Reinforcement Learning to Full 11 Versus 11 Simulated Robotic Soccer"**, Spriger Link, Autonomous Agents and Multi-Agent Systems, Vol. 37, Article Number 20, 2023.

[5] RoboCup, **"The RoboCup Soccer Simulator"**, https://rcsoccersim.github.io, Accessed: 2024.

[6] Yoon, M., J. Bekker, S. Kroon, "**New Reinforcement Learning Algorithm for Robot Soccer"**, ORiON, Vol. 33, No. 1, pp. 1-20, 2017.

[7] Hu, C., M. Xu, K-S. Hwang, **"An Adaptive Cooperation with Reinforcement Learning for Robot Soccer Games"**, International Journal of Advanced Robotic Systems, 2020.

[8] Leng, J. B.M. Sathyaraj, L. Jain, **"Temporal Difference Learning and Simulated Annealing for Optimal Control: A Case Study"**, Proceedings of the Second KES International conference on Agent and multi-agent systems: technologies and applications, pp. 495–504, 2008.

[9] Abreu, M., L.P. Reis, N. Lau, **"Designing a Skilled Soccer Team for RoboCup: Exploring Skill-Set-Primitives through Reinforcement Learning"**, License CC BY 4.0, arXiv:2312.14360, 2023.

[10] Kruusmaa, M., **"Global Navigation in Dynamic Environments Using Case-Based Reasoning"**, Springer link, Autonomous Robots, Vol.14, pp. 71-91, 2003.

[11] Zhang, T., H. Mo, **"Reinforcement Learning for Robot Research: A Comprehensive Review and Open Issues"**, International Journal of Advanced Robotic Systems, pp. 1-22, 2021.

[12] Gabel, T., M. Veloso**, "Selecting Heterogeneous Team Players by Case-Based Reasoning: A Case Study in Robotic Soccer Simulation"**, Technical report CMU-CS-01-165, Carnegie Mellon University, 2001.

[13] AdibYaghmaie, F., A. Mobarhani, H. D. Taghirad, **"A Navigation System for Autonomous Robot Operating in Unknown and Dynamic Environment: Escaping Algorithm"**, International Journal of Robotics, Vol. 4, No. 4, 2016.

[14] Pratomo, A.H., A.S. Prabuwono, M.S. Zakaria, K. Omar, **"Position and Obstacle Avoidance Algorithm in Robot Soccer"**, Journal of Computer Science, Vol. 6, No. 2, pp. 173-179, 2010.

[15] Zheng. J., S. Mao, Z. Wu, P. Kong, H. Qiang, "**Improved Path Planning for Indoor Patrol Robot Based on Deep Reinforcement Learning**", MDPI, Symmetry, Vol. 14, No.1, 2022.

[16] Pinheiro, F.C.R., M. Maximo, T. Yoneyama, **"Comparison of Sampling-Based Path Planners for Robocup Small Size League"**, IEEE Latin American Robotics Symposium, Brazilian Symposium on Robotics and Workshop on Robotics in Education, Natal, Brazil, 2020.

[17] Kim. J-H., Y-H. Kim, S-H. Choi, I-W. Park, **"Evolutionary Multi-Objective Optimization in Robot Soccer System for Education"**, IEEE Computational Intelligence Magazine, Vol. 4, No. 1, pp. 31-41, 2009.

[18] Pu, Z., Y. Pan, S. Wang, B. Liu, M. Chen, H. Ma, and Y. Cui, "**Orientation and Decision-Making for Soccer Based on Sports Analytics and AI: A Systematic Review"**, IEEE/CAA Journal of Automatica Sinica, Vol. 11, No. 1, pp. 37–57, 2024.

[19] Sutton, R.S., A.G. Barto, **"Reinforcement Learning: An Introduction"**, A Bradford Book, Second edition, The MIT Press, Cambridge, Massachusetts, London, England, 2012.

[20] Hirotsu, N., K. Inoue, K. Yamamoto, M. Yoshimura, "**Soccer as a Markov Process: Modelling and Estimation of the Zonal Variation of Team Strengths**", IMA Journal of Management Mathematics, Vol. 34, No. 2, pp. 257–284, 2023.

[21] Miyazaki, K., M. Itou, H. Kobayashi, **"Evaluation of the Improved Penalty Avoiding Rational Policy Making Algorithm in Real World Environment"**, Springer Link, Asian Conference on Intelligent Information and Database Systems, pp. 270-280, Part of the Lecture Notes in Computer Science book series (LNAI, Volume 7196), 2012.

[22] Busoniu, L., R. Babuska, B.D. Schutter, D. Ernst, **"Reinforcement Learning and Dynamic Programming Using Function Approximators"**, Automation and Control Engineering, Publisher: CRC Press; first edition, 2010.

[23] Wang, F., X.T. Lin, Y.X. Xiao, **"Alice2022: Team Description Paper"**, In RoboCup Symposium and Competitions, Thailand, 2022.

[24] Stone, P., R.S. Sutton, G. Kuhlmann, **"Reinforcement Learning for RoboCup Soccer Keepaway"**, International Society for Adaptive Behavior, Vol. 13, No. 3, pp. 165-188, 2005.

[25] Shi, H., Z. Lin, K-S. Hwang, S. Yang, J. Chen, **"An Adaptive Strategy Selection Method with Reinforcement Learning for Robotic Soccer Games"**, Institute of Electrical and Electronics Engineers (IEEE), IEEE Access, Vol. 6, pp. 8376-8386, 2018.

[26] Singh, P.S., R.S. Sutton, L.P. Kaelbling, **"Reinforcement Learning with Replacing Eligibility Traces"**, Machine Learning, Vol. 22, pp. 123-158, 1996.

[27] Wang, Y-H., T-H. S.Li, C-J. Lin**, "Backward Q-Learning: The Combination of SARSA Algorithm and Q-Learning"**, ELSEVIER, Engineering Applications of Artificial Intelligence, Vol. 26, No. 9, pp. 2184-2193, 2013.

[28] Zare, N., O. Amini, A. Sayareh, M. Sarmaili, A. Firouzkouhi, S. Matwin, A. Soares, **"Improving Dribbling, Passing, and Marking Actions in Soccer Simulation 2D Games using Machine Learning",** Springer Link, RoboCup 2021: Robot World Cup XXIV, RoboCup International Symposium, Champion team paper, Part of the Lecture Notes in Computer Science book series (LNAI, Volume 13132), pp. 340-351, First Online: 2022, Submitted in arXiv on 2024, 2021.

[29] Sarje, A. A. Chawre, S.B. Nair, **"Reinforcement Learning of Player Agents in RoboCup Soccer Simulation"**, IEEE Fourth International Conference on Hybrid Intelligent Systems, Kitakyushu, Japan, 2004.

[30] Hwang, K-S, S-W. Tan, C-C. Chen, **"Cooperative Strategy Based on Adaptive Q-Learning for Robot Soccer Systems"**, IEEE Transactions on Fuzzy Systems, Vol. 12, No. 4, pp. 569 - 576, 2004.

[31] Xu, M., X. Chen, Y. She, Y. Jin, G. Zhao, J. Wang**, "Strengthening Cooperative Consensus in Multi-Robot Confrontation"**, ACM Transactions on Intelligent Systems and Technology, 2023.

[32] Celiberto Jr, L.A., J. Matsuura, R.A.C. Bianchi, **"Heuristic Q-Learning Soccer Players: A New Reinforcement Learning Approach to RoboCup Simulation"**, Springer Link, 13th Portuguese

Conference on Aritficial Intelligence, Progress in Artificial Intelligence, Part of the Lecture Notes in Computer Science book series (LNAI, Volume 4874), pp. 520-529, 2007.

[33] Xiong, L., G. Jing, Z. Zhenkun, H. Zekai, **"A New Passing Strategy Based on Q-Learning Algorithm in RoboCup"**, IEEE International Conference on Computer Science and Software Engineering, pp. 524-527, 2008.

[34] Cunha, J., R. Serra, N. Lau, L.S. lopes, A.j.R. Neves, **"Batch Reinforcement Learning for Robotic Soccer Using the Q-Batch Update-Rule"**, Spriger Link, Journal of Intelligent & Robotic Systems, Vol. 80, pp. 385-399, No. 3-4, 2015.

[35] Leottau, D.L., J. Ruiz-del-Solar, R. Babuska, **"Decentralized Reinforcement Learning of Robot Behaviors"**, ELSEVIER, Artificial Intelligence, Vol 256, pp. 130-159, 2018.

[36] Bassani, H.F., and et al., **"A Framework for Studying Reinforcement Learning and Sim-to-Real in Robot Soccer"**, Transfer Learning for Human & AI, License CC BY-NC-SA 4.0, arXiv:2008.12624, 2020.

[37] Yu, L., K. Li, S. Huo, K. Zhou, **"Cooperative Offensive Decision-Making for Soccer Robots Based on Bi-Channel Q-Value Evaluation MADDPG"**, ELSEVIER, Engineering Applications of Artificial Intelligence, Vol. 121, 2023.

[38] Zolanvari, A., M.M. Shirazi, M.B. Menhaj, **"A Q-Learning Approach for Controlling a Robotic Goalkeeper during Penalty Procedure"**, Second International Congress on Science and Engineering, Hamburg, Germany, pp. 1-12, 2019.

[39] Barbosa, V.G.F, R.F.O. Neto, R.V.L.G. Rodrigues, **"A Baseline Approach for Goalkeeper Strategy using SARSA with Tile Coding on the Half Field Offense Environment",** 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), pp. 195-202, 2020.

[40] Leng, J., C.P. Lim, **"Reinforcement Learning of Competitive and Cooperative Skills in Soccer Agents"**, Applied Soft Computing, Vol. 11, No. 1, pp.1353-1362**,** 2011.

[41] Homem, T.P.D., P.E. Santos, A.H.R. Costac, R.A.da.C. Bianchib, R.L. de Mantarasd, **"Qualitative Case-Based Reasoning and Learning"**, ELSEVIER, Artificial Intelligence, Vol. 283, 2020.

[42] Zhan, W., S. Qu, **"Cooperation Mode of Soccer Robot Game Based on Improved SARSA Algorithm"**, Hindawi, Wireless Communications and Mobile Computing, License: CC BY 4.0, Vol. 2022, Article ID 9190687, 11 pages, 2022.

[43] De Luna Amat, M, "**An Explanation of How AI Is Changing the World Through Football**", Telefonica Tech, 2024.

[44] Nashed, S.B., S. Ziberstein, **"A Survey on Opponent Modeling in Adversarial Domains"**, Journal of Artificial Intelligence Research,Vol. 73, pp. 277-327, 2022.

[45] Chen, H., C. Wang, J. Huang, J. Kong, H. Deng, **"XCS with Opponent Modelling for Concurrent Reinforcement Learners"**, ELSEVIER, Neurocomputing, Vol. 399, pp. 449-466, 2020.

[46] Li, Y., Y. Song, A. Rezaeipanah, **"Generation a Shooting on the Walking for Soccer Simulation 3D League using Q-Learning Algorithm"**, Springer Link, Journal of Ambient Intelligence and Humanized Computing, Vol. 14, pp. 6947-6957, 2023.

[47] Wang, Z., Y. Zeng, Y. Yuan, Y. Guo, **"Refining Co-operative Competition of Robocup Soccer with Reinforcement Learning"**, IEEE Fifth International Conference on Data Science in Cyberspace (DSC), pp. 279-283, Hong Kong, China, 2020.

[48] Jaradat, M.A.K, M. Al-Rousan, L. Quadan, **"Reinforcement Based Mobile Robot Navigation in Dynamic Environment"**, ELSEVIER, Robotics and Computer-Integrated Manufacturing, Vol. 27, No. 1, pp. 135-149, 2011.

[49] Nakahara, H., K. Tsutsui, K. Takeda, K. Fujii, **"Action Valuation of on- and off-ball Soccer Players Based on Multi-agent Deep Reinforcement Learning"**, License CC BY-SA 4.0, 2023.

[50] Haushnecht, M., P. Mupparaju, S. Subramanian, S. Kalyanakrishnan, P. stone, **"Half Field Offense: An Environment for Multiagent Learning and Ad Hoc Teamwork"**, In AAMAS Adaptive Learning Agents (ALA) Workshop, Singapore, 2016.

[51] Barrett, S, A. Rosenfeld, S. Kraus, P. stone, **"Making Friends on the Fly: Cooperating with New Teammates"**, ELSEVIER, Artificial Intelligence, Vol. 242, pp. 132-171, 2017.

[52] Fadelli, I., "**An Open-Source and Python-Based Platform for the 2D Simulation of Robocup Soccer**", Tech Xplore, https://techxplore.com/news/2023-08-open-source-python-based-platform-2d-simulation.html, 2023, Accessed: 2024.

[53] Azarkasb, S.O., S.H. Khasteh, **"A New Approach for Mapping of Soccer Robot Agents Position to Real Filed Based on Multi-Core Fuzzy Clustering"**, 26th IEEE International Computer Conference, 2021.

[54] Bei, H., Y. Mao, W. Wang, X. Zhang, **"Fuzzy Clustering Method Based on Improved Weighted Distance"**, Mathematical Problem in Engineering, Vol. 5, Hindawi, 2021.

[55] Romanazzi, A., D. Scocciolini, M. Savoia, N. Buratti, **"Iterative Hierarchical Clustering Algorithm for Automated Operational Modal Analysis"**, ELSEVIER, Automation in Construction, Vol. 156, 2023.

[56] Randel, R., D. Aloise, A. Hertz, "**A Lagrangian-Based Approach to Learn Distance Metrics for Clustering with Minimal Data Transformation**", In book: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 127-135, , 2023.

[57] RoboCup, https://www.robocup.org, Accessed: 2024.

[58] Belleville, S., C. Christensen, A. Espeland, L. Rinaldi, N. Rogers, B. Schwantes, E. Vadeboncoeur, Y. Zhao, **"Small Size Soccer Robots"**, A Major Qualifying Project Report submitted to the faculty of WORCESTER POLYTECHNIC INSTITUTE, Digital WPI, 125 Pages, 2023.

[59] Goncalves, A., and et al., **"ITAndroids Small Size League Team Description Paper for RoboCup 2023"**, RoboCup federation , 2023.

[60] Martins, F.B., M.G. Machado, H.F. Bassani, P.H.M. Braga, E.S. Barros, **"rSoccer: A Framework for Studying Reinforcement Learning in Small and Very Small Size Robot Soccer"**, License CC BY-NC-SA 4.0, Part of the Lecture Notes in Computer Science book series (LNAI, Volume 13132), 2022.

[61] De Medeiros, T.F., M. Máximo, T. Yoneyama**, "Deep Reinforcement Learning Applied to IEEE Very Small Size Soccer Strategy"**, Latin American Robotics Symposium, Brazilian Symposium on Robotics, Workshop on Robotics in Education, Natal, Brazil, 2020.

[62] Jeon, H., D-W. Kim, B-Y. Kong, **"Deep Reinforcement Learning for Cooperative Robots Based on Adaptive Sentiment Feedback"**, ELSEVIER, Expert Systems with Applications, Vol. 243, 2024.

Seyed Omid Azarkasb received his B.Sc. degree in computer software engineering from Kashan Branch Azad University in 1996 and 2001. He studied artificial intelligent systems at Qazvin University of Technology and got his M.Sc. in 2008. Now, he is a visiting professor and Ph.D. student at K. N. Toosi University of Technology, Tehran, Iran.
His research interests include Robotics, Intrusion detection systems, Machine learning methods, Internet of Everything (IoE), Fog and Cloud Computing.

Seyed Hossein Khasteh received the B.Sc. degree in electrical engineering, the M.Sc. degree in Artificial Intelligence, and the Ph.D. degree in Artificial Intelligence all from the Sharif University of Technology, Tehran, Iran.
He is currently an Assistant Professor with the Computer Engineering Department, K. N. Toosi, University of Technology, Tehran, Iran.
His current research interests include social network analysis, machine learning, and big data analysis.