R2D2 Soccer Simulation 2D Team Description Paper Robocup World Cup 2024

Seyed Hassan Majid Zonouzi, Mohammad Hesam Nasiri, Sanaz Moosapour, Sina Taheri Behrooz, Sana Mousavi, Seyed Mostafa Atyabi, Fatemeh Houra Haghighatkhah

Shahid Beheshti University, Tehran, Tehran, The Islamic Republic of Iran

s.majidzonouzi@mail.sbu.ac.ir, mohammadh.nasiri@mail.sbu.ac.ir, s.moosapour@mail.sbu.ac.ir, sin.taheri@mail.sbu.ac.ir, san.mousavi@mail.sbu.ac.ir, se.atyabi@mail.sbu.ac.ir, fa.haghighatkhah@mail.sbu.ac.ir

Abstract. The paper details the R2D2 team's (R3CESBU) algorithm and development clarifications for RoboCup's 2D soccer simulation league. It introduces goalie, shooting, and marking advancements, emphasizing the ranking algorithm for chain action states and dynamic positioning. R2D2 employs artificial intelligence techniques, combining behavioral cloning and game log parsing to enhance kickable agents and goalies. The research contributes to the evolution of competitive play in the RoboCup 2D Simulation League. The offence tactics involve improvements in the field evaluator, through-pass, unmarking, and shooting algorithms, resulting in significant performance enhancements.

Keywords: RoboCup, World Cup, Soccer Simulation

1 Introduction

We use agent-2d base code to develop our ideas and strategies for the 2D soccer simulation league [1]. After last year's success and solid defense, we decided to emphasize the offensive side for this year. Significant changes were made to the Field Evaluator, and the Through-Pass generator was tweaked. On the other hand, the shooting was majorly improved, and the unmark function was changed concerning the new field evaluator.

2 Behavioral cloning to imitate Helios team

We propose a new approach to creating team models by combining imitation learning [2] with elements of deep learning [3] in a data-driven framework. We use a simulated environment to make the process more efficient and scalable. Our machine-learning techniques enable us to construct a seamless data pipeline that can be applied to multiple teams, eliminating the need for handcrafting scenarios. We use behavioral cloning, an imitation learning technique that enables models to mimic the behavior of expert players. The state of the agent (player positions, velocities, and orientations on the field) serves as input to the model, and the model outputs the appropriate action for the agent. Our approach distinguishes itself from alternative solutions by tailoring the implementation and identifying critical features unique to each model, enabling the creation of highly specialized datasets. We have also developed new tools to generate an effective dataset encompassing all games, which will be a valuable resource for future research endeavors. We demonstrate significant improvements in action accuracy, on-target shots, and goalkeeping saves through behavioral cloning [4] and deep learning, contributing to the evolution of competitive play in the RoboCup 2D Simulation League.

2.1 Related Works

The ITAndroid team's approach, which involved mining data from matches involving the Japanese team Helios, bears similarities to our research, but notable differences exist. While both studies leverage neural networks for imitation learning, our research took a more specialized approach. We generated two distinct neural networks to imitate the actions of the Helios team, specifically focusing on kickable and goalie agents. In contrast, the ITAndroid team adopted a unified strategy, employing a single neural network to imitate actions across all player agent types. Our research concentrated on these two specific player categories to achieve the highest accuracy and optimize game results. This strategic divergence underscores the unique strengths. [5]

In the five-section strategy, the goalie adjusts their position based on the ball's location, a strategy implemented by R3CESBU last year. In the "BA_Safe" section, they remain still to conserve stamina. In "BA_DribbleBlock," they position between the goal and the ball. In "BA_DefMidField," they maintain a position equidistant from the goal and nearest post. In "BA_CrossBlock," they move to the nearest post. In "BA_Danger," they track the ball while a defender provides support. [6]

2.2 Data Generation

We utilized the Yushan team's data mining tool [7] tailored for 2D football simulation to enhance insights from our dataset. This widely used tool transforms game logs into CSV files, iterating through cycles to record cycle-specific data. Inspired by this tool, we developed a customized version capable of processing log files and storing data. Our dataset comprised 100,000 kicks by the Helios team, with 20% for validation and 80% for neural network training. Real-game scenarios served as performance benchmarks, eliminating the need for a separate test dataset. This section details our data mining tools, dataset partitioning, and real-game performance evaluation.

2.3 Deployment

We utilize the CppDnn library [8] to deploy our model in the game environment. This C++ tool integrates Keras DNN functionality into our applications, enabling us to efficiently convert and utilize our saved model within our C++ codebase for real-time gameplay.

2.4 Results

In soccer 2D simulation research, achieving the best results and highest win rate is paramount. Our comprehensive analysis, spanning over 1,000 games, has evidenced significant improvements in our team's passing system. Specifically, we have achieved a 5 percent increase in the passing success rate, a finding depicted in Fig. 1. This enhancement has been facilitated by the application of behavioral cloning techniques, reflecting our unwavering commitment to refining our gameplay. Moreover, we have recorded notable advancements in shoot accuracy, underscoring the effectiveness of our methodological approach. These improvements in passing and shooting, alongside goalkeeper save accuracy (as demonstrated in Figs. 1 and 2 for passing and shooting accuracy, respectively, and Fig. 3 for goalkeeper save accuracy), underscore our dedication to enhancing every facet of our performance for more tremendous success on the field.

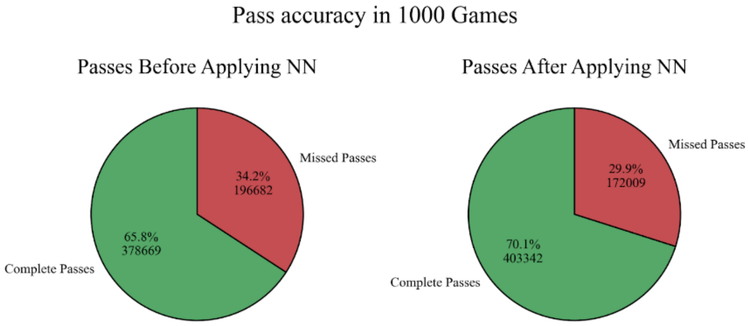

Fig. 1. Pass accuracy before and after using Behavioral Cloning

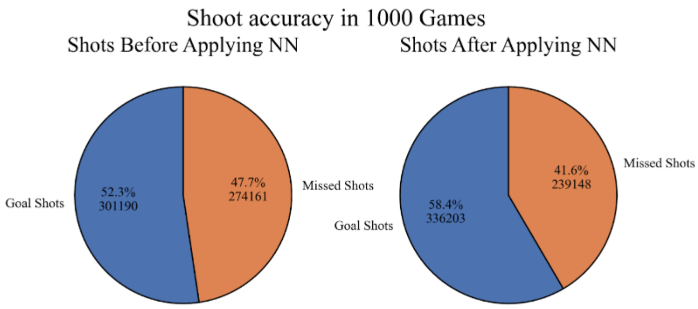

Fig. 2. Shoot accuracy before and after using Behavioral Cloning

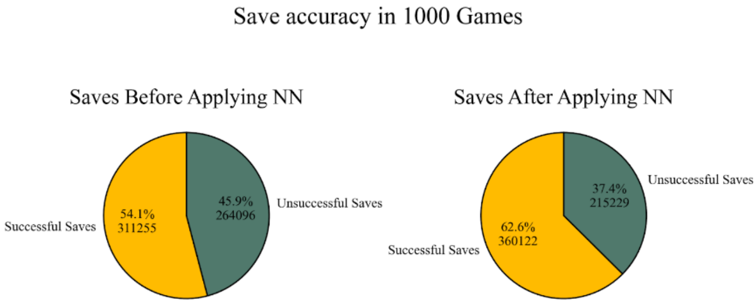

Fig. 3. Save accuracy before and after using Behavioral Cloning

3 Offense Tactics and Strategies

Major changes were made to our team's offensive play style this year. Here, we will discuss some of the more important ones.

3.1 Field Evaluator

One of our team's main problems in the previous year was the lack of a reliable evaluation function for code-generated actions. So, some parameters were added to address this problem. Alongside the static evaluation method that uses a constant array, we now check some of the other conditions of each state. The most important of these conditions is the number of players present around the ball in the state. The score is reduced if more players are around the ball in any given state. This encourages the players to move the ball to areas with less opponent density. Some of the other notable conditions are the wideness of the team in new states and the number of opponents that can follow the ball in a set number of steps after moving to any new state. These changes have drastically changed how our team plays, and the players choose new and more desirable states.

Another major change to the evaluator was the way passes were scored. In the previous version, our team did not utilize passing enough; as a result, there was limited space for the players to create promising goal-scoring opportunities. So, in the newer version, depending on the state of the game and the opponent we are facing, the score of all pass actions is dynamically increased in the chain action graph. This mechanism allows us to exploit small positioning errors in opponents and keep possession of the ball to reduce the risk of conceding goals. Although some mechanisms should be used to ensure a repetitive pass is not generated between 2 players. Like adding a penalty to the score of passes with the same source player and destination player as the previous pass.

3.2 Through-Pass

The team's through-pass algorithm was fine-tuned and tweaked. A main problem of the through passes was the field evaluator, so after fixing that, a part of the issue was solved, although the agent was still ignoring some obvious passes. So overall, the risk of through passes increased drastically, and the field checked for valid passes expanded. As a result, players might attempt through passes to players who may not be able to receive the ball. In many instances, this results in successful and precise through passes, although in some instances, the ball possession is lost. We have tried to balance and minimize the risk of possession loss in these instances.

Dynamic Positioning System. We introduced a dynamic positioning system that analyzes real-time game data to optimize our forward’s positions. This system considers factors like opponent positioning, ball trajectory, and player stamina, enabling our attackers to exploit gaps in the opposition's defense more effectively. By constantly adjusting our positioning, we've created a fluid and unpredictable offense that adapts to the game's flow, making it harder for opponents to anticipate our moves.

Adaptive Passing Mechanism. Our adaptive passing mechanism has been overhauled to prioritize passes that break through defensive lines. This system uses advanced algorithms to evaluate potential passing channels, dynamically adjusting for risk and reward based on the game's context. This approach has significantly increased the quality of our scoring opportunities, leading to a higher conversion rate of chances into goals.

3.3 Unmarking

After the changes to the Field Evaluator, some changes were needed to improve the unmark function. The previous version of the unmark function moved the players to positions where no direct opponent could threaten the pass from the player currently possessing the ball. This method has a major flaw: the players might use their stamina and move to new positions with a small chance of being chosen as a target in the evaluator function. To address this issue, The evaluator was used in the unmark function. The positions generated by the unmark function are now given to the evaluator, and the position with the highest score is chosen as the new position for unmarking. The enhancements to the unmark function extend beyond the inclusion of the evaluator. Like the through-pass generator, the parameters of the unmark function were tweaked and optimized to promote the efficient stamina usage of the players. In the new version of unmark, players are encouraged to stay closer to their designated position retrieved from the formation. This way, players don’t move too far away from their intended positions, creating potential defensive exploits.

Coordinated Attack Patterns. We've developed a set of coordinated attack patterns executed based on specific triggers observed during the match. These patterns involve synchronized movements and passes designed to disorient the opponent's defense. By practicing these patterns, our team can execute complex maneuvers that capitalize on our opponents' split-second decisions and mistakes, leading to clear scoring opportunities.

3.4 Shoot

We implemented three different algorithms for choosing the best target point in shoot action.

The first one is that we divide the goal area into 2 parts [9], and if the middle of the goal area is an ideal shooting target (according to the distance between the point and the goalie, the distance between the nearest opponent and the point, the distance between the kicker and the point ...), we execute the kick. Otherwise, we pick up the better half and repeat the algorithm, using a recursive function, until we find a good point. This approach has a major flaw. There is a chance that the half that was not chosen had the better shooting target. This results in some of the promising targets being completely overlooked.

The second algorithm is that we choose 32 points in the goal area [10] and assign a score to each of them, and we choose the point with the maximum score. This algorithm solves the first algorithm's problems, but another problem exists. In this algorithm, the point generation is static. This means that if the striker is closer to one of the sides, there will be no difference in the points generated by the shooting function. Intuitively, we want more points checked on the left side if our striker is on the left side of the goal.

The third algorithm that solves the second algorithm's problem is to draw a line between the kicker and the left goal and a line between the kicker and the right goal. We draw lines in this area with a 5-degree distance as it displayed in Fig. 4, so with this algorithm, the number of points in each position is not equal to another, and the considered points are more. Hence, the chosen target point is more accurate; again, we assign a score to each point and pick the point with the maximum score.

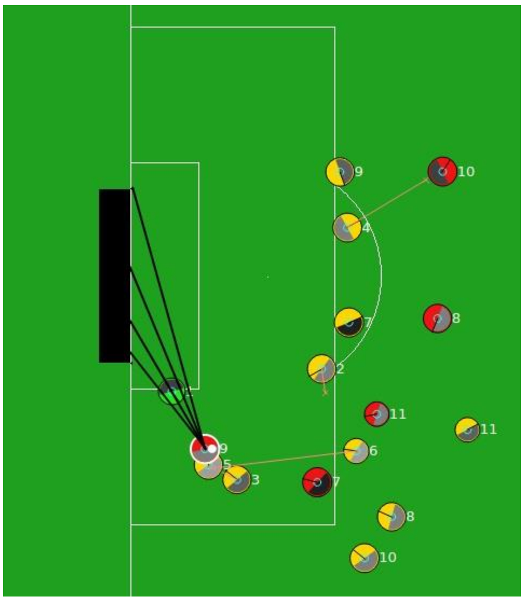

Fig. 4. Finding the best shoot angle

4 References

1. Akiyama, H., & Nakashima, T. (2013). Helios base: An open-source package for the RoboCup soccer 2D simulation. In Proceedings of the Robot Soccer World Cup (pp. 528-535). Berlin, Heidelberg: Springer.
2. Codevilla, F., Müller, M., López, A., Koltun, V., & Dosovitskiy, A. (2018). End-to-End Driving Via Conditional Imitation Learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4693-4700). Brisbane, QLD, Australia.
3. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
4. Zhan, E., Zheng, S., Yue, Y., & Sha, L. Generative Multi-Agent Behavioral Cloning. Caltech.edu.
5. Fidalgo, D. T. M., Coimbra, F. V., Gottschild, G. F., dos S. S. Filho, J., & Marco. (2020). ITAndroids Team Description Papers. In RoboCup 2020 World Cup, Canada.
6. Nasiri, M. H., et al. (2023, July). R3CESBU: Innovative Approaches in Goalkeeping and Team Coordination for RoboCup Soccer Simulation 2D. In RoboCup 2023, Bordeaux, France.
7. Cheng, Z., et al. (2018). YuShan2018 Team Description Paper for RoboCup2021. In RoboCup 2018 Symposium and Competitions, Worldwide, Brazil.
8. Khayami, R., et al. (2014). CYRUS 2D simulation team description paper 2014. In RoboCup 2014 Symposium and Competitions: Team description papers.
9. Noohpisheh, M., et al. (2018). Razi Soccer 2D Simulation Team Description Paper 2018.
10. Aghayari, R., Shahbazy, S., Iranmanesh, M., & Nikfetrat, A. (2022). IraNad Soccer 2D Simulation Team Description Paper 2022.