Easter2026: Team Description Paper

Yinuo Xie[1][0009−0007−6822−9265], Yuan Li[2], Xin Liu[3], Zixi Jia[3], Xingyang Li[3], and Shiguang Wen[3]

[1] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China 20237563@stu.neu.edu.cn
[2] School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China 20246766@stu.neu.edu.cn
[3] Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China 20246803@stu.neu.edu.cn jiazixi@mail.neu.edu.cn lixingyang@mail.neu.edu.cn wenshiguang@mail.neu.edu.cn

Abstract. The Easter team is making its first attempt to participate in international competitions, with the long-term vision of advancing machine learning and game theory at all levels of the game. To this end, we plan to deploy a concise state modeling and passing network modeling algorithm to simplify state-space search and achieve optimal decision-making. The team's base code is built on the Helios base.

Keywords: RoboCup · 2D Soccer Simulation · Costmap · Machine Learning.

1 Introduction

Since receiving awards in RoboCup China Open 2022, the Easter team has been dedicated to learning and applying artificial intelligence techniques in the complex RoboCup 2D simulation environment. Facing experienced opponents, this year we have shifted our focus from domestic competitions to international ones. We are well aware of the challenges of starting from scratch; therefore, our core philosophy is: based on understanding classic frameworks, we aim to achieve significant performance improvements through low-cost algorithm optimizations. We use the widely adopted Helios base[1] as our foundation code, and the Cross Language Soccer Framework[8], draw on public research ideas from strong teams such as Cyrus and ITAndroids, and make improvements targeting our own weaknesses (e.g., the goalkeeper's tendency to be out of position, insufficient exploitation of shooting opportunities). This paper will focus on our work in three areas: goalkeeper modeling, shooting decision-making, and basic formation switching.

2 Related Works

During development, we referred to a large body of public literature from the RoboCup community, which provided important inspiration for our research:

- The formation model based on triangulation and dynamic positioning adjustments proposed by the Helios team[2] made us realize the importance of positional information in team collaboration.
- The research on goalkeepers by the ITAndroids team[7] impressed us deeply, especially their modifications to the Dash model in response to server version updates and the method of optimizing goalkeeper positioning using the angle bisector, which directly inspired the work in Section 3 of this paper.
- The exploration of shooting algorithms by the R2D2 team[9] is very comprehensive, particularly the method of dividing the shooting cone to find the best shooting point, providing a clear mathematical foundation for improving our shooting logic.
- The Simurgh team[6], as another new team also using the Helios base, demonstrated that a team can make progress even without complex machine learning models, through their approach of optimizing passing and shooting decisions using simple rules (if-else).

3 Our Works

3.1 Improvements in Goalkeeper Positioning and Ball Interception Logic

In early tests, we found that the goalkeeper often stood too close to the goal line when the opponent shot, and hesitated to rush out when facing a one-on-one situation. Inspired by the work of ITAndroids[7], we implemented the following improvements:

Bisector Intersection Positioning We modified the goalkeeper's default positioning logic. When the ball is in the opponent's half, the goalkeeper moves up appropriately; when the ball enters our defensive third, we no longer simply keep the goalkeeper at the same Y-coordinate as the ball. Instead, we calculate the angle formed by the ball and the two goal posts, and position the goalkeeper on the angle bisector, while adding a random error ε to handle unexpected situations. This maximizes the blocking of shooting angles. The final positioning is shown in Fig. 1.

Active Catch Trigger We refined the trigger conditions for the goalkeeper's 'Catch' action. When the opponent delivers a through pass and the ball speed is sufficiently high, the goalkeeper no longer waits but actively rushes to the ball's landing point for interception. This algorithm is rule-driven and implemented through if-else condition judgments. First, the player's current position and the opponent's defensive situation are evaluated. Then, the distances between this player and all visible teammates are calculated. Based on the above information, the optimal interception option is determined according to the following rules:

1. Distance priority: If the distance between the goalkeeper and the ball's landing point is less than that of any defender, and the ball speed exceeds a threshold, the goalkeeper immediately rushes out. This rule ensures the goalkeeper can handle threatening balls at the first moment, preventing defenders from being unable to recover in time.
2. Opponent threat assessment: If the goalkeeper is far away but an opponent forward is approaching the landing point and unmarked, the system calculates the opponent's shooting angle and speed. If the opponent might form a one-on-one with the goalkeeper after receiving the ball, the goalkeeper will anticipate in advance and expand the defensive range, even rushing out of the penalty area to compress the opponent's shooting space. At this point, the algorithm references the goalkeeper's 'rush-out tendency' attribute; if the value is high, active interception is triggered.
3. Out-of-box handling: When the landing point is outside the penalty area and the goalkeeper rushes out, the algorithm enables 'ClearBall' mode, allowing the goalkeeper to use clearance actions, but the probability of error is reduced based on their ball control ability. If the goalkeeper's technical attributes are insufficient, the system triggers cautiously to avoid leaving an open goal.

This algorithm aims to formulate reasonable goalkeeper strategies by combining player roles, real-time match situations, defensive intensity, and available passing routes. It prioritizes passing to teammates who are well-positioned and in advantageous receiving positions, while comprehensively considering factors such as defensive coverage and passing distance.

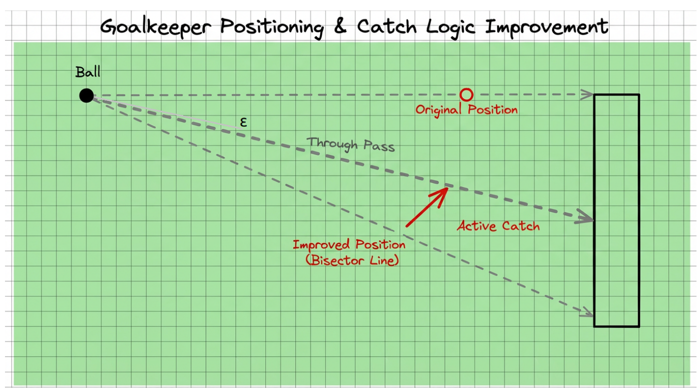

Fig. 1. Goalkeeper Positioning and Defence

3.2 Costmap-based Kicking Planning

When analyzing match logs, we found that forwards, when having an opportunity inside the penalty area, often chose to directly blast the ball towards the center of the goal, resulting in easy saves by the goalkeeper. To increase the variety and threat of shots, we referred to the third shooting algorithm of R2D2[9]. Additionally, inspired by the Costmap mechanism used in ROS[5], combined with a quadratic function of the target point, we implemented a new shooting point planning method:

Discretization encoding. From the position of the ball-holding player, we draw lines to the left and right goal posts, forming a shooting cone area. Within this cone, we draw rays at 5-degree intervals, and the intersection of each ray with the goal line serves as a candidate shooting point. The line connecting the player to each candidate point is called the kicking direction.

Generating Costmap and path weight f. Inspired by the ROS costmap concept, we consider the positions of each player and generate an expansion radius. Combining the $1-\text{sigmoid}(x)$ function, we construct a function field where the path weights increase within the field. For each kicking direction l, let r be the distance from a certain position to the goalkeeper, and d_i be the distance to the i-th other player. According to the formula:

$$f = l + \int_l [1 - \text{sigmoid}(r)]dl + \int_l \sum_i [1 - \text{sigmoid}(d_i)]dl$$

Generating heuristic function g. Let k be an adjustable coefficient. Directly use the kicking angle θ to construct the distribution of the heuristic function:

$$g = k\theta^2$$

Select direction and execute attack. Choose the point with the lowest $f + g$ as the shooting target. This dynamic division method ensures that regardless of whether the attacking player is on the left or right, the generated candidate points cover more threatening areas.

Costmap for Whole-map Passing We are recently planning to extend the costmap to the entire field. For the game, we make an assumption: we must ensure that the ball is either at a player's feet or on its way to a player during a pass, without considering kicking the ball to other areas for players to retrieve. Under this tactic, the search space of states can be greatly reduced. We encode the feasible positions of the ball as a network; ideally, the ball can only be passed through these pathways. Not every pass will succeed at all times; when an opponent occupies a certain edge in the network, the pass may fail. Therefore, we introduce a Costmap for the presence of opponents, which increases the weight of some passing edges, and these edges are not considered during passing. Ultimately, when choosing a pass, players use A* search to automatically plan the optimal path to score through sequential passes, or the optimal path to pass to the least defended area. In this way, passing can globally consider the entire passing network to plan the attack path, rather than focusing on the immediate few players, achieving the greatest degree of team collaboration.

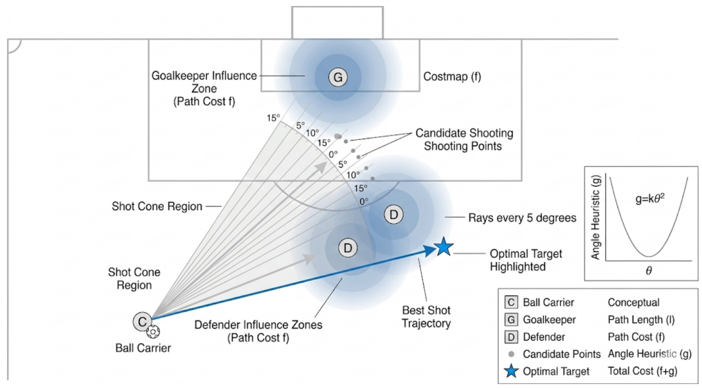

Fig. 2. Shooting costmap planning

3.3 End-to-End Learning of Formations for Targeted Defense

During a match, accurately identifying the opponent's formation allows us to adaptively adjust our own division of labor to achieve targeted defense[4]. HELIOS uses Gaussian mixture models to classify position formations, which is based on the premise that solutions exist for various types of formations. However, we have not accumulated such complex tactical rules, and we urgently need an end-to-end learning method that automatically generates our appropriate positions based on the opponent's formation.

Model Construction We adopt a multi-layer perceptron, with the input layer being the opponent's positions and the output layer being the suggested positions. We treat the 11 players of each team as 11 nodes on each side of the network, each node corresponding to a player's position on the field at a certain moment (x, y coordinates). The edges between nodes represent spatial interactions between players.

Model Training The model's dataset comes from past match results, where the losing team's positions are used as samples and the winning team's positions as labels. The loss function is the mean squared error between the suggested positions and the winning team's positions. After learning, the network provides recommended positions for players. Therefore, during a match, players in idle states will actively move to the recommended positions, ensuring quicker and more active responses when situations change. However, when specific tasks exist or the situation is special, the recommended positions only serve as a scoring mechanism and do not restrict the player's actions, allowing players to handle more dangerous situations, such as clearing the ball in the midfield or defensive areas.

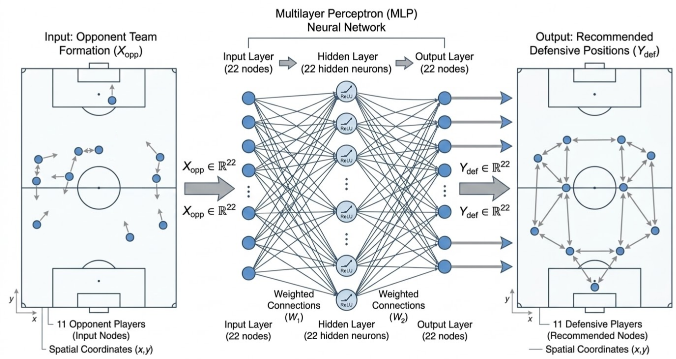

Fig. 3. Formation Learning

4 Future Discussion

This paper introduces the preliminary work of the Easter team in RoboCup. By drawing on mature community experience and optimizing for our own weaknesses, we have made substantial progress in goalkeeper, shooting, and tactical formations. As a new team, we still have many shortcomings. In future work, we plan to:

- Introduce data analysis: Learn from the YuShan team[3] by using data mining tools to analyze match logs, identify areas of abnormal player energy consumption, and optimize running strategies.
- Explore machine learning: Attempt to apply supervised learning or simple reinforcement learning in some simple scenarios (e.g., 1v1 defense), mimicking the defensive behavior training of ITAndroids[7].
- Improve the passing network: Enhance current passing decisions by not only basing them on distance but also dynamically considering the space around the receiving player and opponent pressure, as well as predicting future deformations of the passing network.

Finally, through reasonable state encoding, we hope to simplify the robot soccer problem into a computer board game problem, which can then be efficiently tackled using mature algorithms (e.g., reinforcement learning, MCTS).

References

1. Akiyama, H., Nakashima, T.: Helios base: An open source package for the robocup soccer 2d simulation. In: Behnke, S., Veloso, M., Visser, A., Xiong, R. (eds.) RoboCup 2013: Robot World Cup XVII. pp. 528–535. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
2. Akiyama, H., Nakashima, T., Fujikawa, T., Hishiki, A., Yano, S., Okayama, S.: Helios2024: Robocup soccer simulation 2d competition champion. In: Barros, E.,

Hanna, J.P., Okada, H., Torta, E. (eds.) RoboCup 2024: Robot World Cup XXVII. pp. 495–501. Springer Nature Switzerland, Cham (2025)
3. Cheng, Z., Zhang, L., Dan, D., Xiong, H., Qin, H.: YuShan2024 Team Description Paper for RoboCup2024. In: RoboCup 2024: Symposium and Competitions. Eindhoven, Netherlands (2024)
4. Futai, L., Yan, Z., Chenhao, Z., Zihao, S., Xiaorui, Z.: An aircraft formation recognition method based on deep learning. In: 2022 7th International Conference on Communication, Image and Signal Processing (CCISP). pp. 457–460 (2022). https://doi.org/10.1109/CCISP55629.2022.9974477
5. Lu, D.V., Hershberger, D., Smart, W.D.: Layered costmaps for context-sensitive navigation. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 709–715 (2014). https://doi.org/10.1109/IROS.2014.6942636
6. Maalekpour, A., et al.: Simurgh Soccer 2D Simulation Team Description Paper Robocup IranOpen 2024. In: RoboCup IranOpen 2024: Symposium and Competitions. Tehran, Iran (2024)
7. Vasconcelos, D.M., Ramos, L.F.B., Maximo, M.R.O.A., Segura, N., Almeida, V.F.: ITAndroids 2D Soccer Simulation Team Description Paper 2024. In: RoboCup 2024: Symposium and Competitions. Eindhoven, Netherlands (2024)
8. Zare, N., Sayareh, A., Sadraii, A., Firouzkouhi, A., Soares, A.: Cross language soccer framework: An open source framework fornbsp;thenbsp;robocup 2d soccer simulation. In: RoboCup 2024: Robot World Cup XXVII. p. 152–163. Springer-Verlag, Berlin, Heidelberg (2025). https://doi.org/10.1007/978-3-031-85859-8_13, https://doi.org/10.1007/978-3-031-85859-$8_1$3
9. Zonouzi, S.H.M., Nasiri, M.H., Moosapour, S., Behrooz, S.T., Mousavi, S., Atyabi, S.M., Haghighakhah, F.H.: R2D2 Soccer Simulation 2D Team Description Paper Robocup World Cup 2024. In: RoboCup 2024: World Cup Symposium and Competitions. Eindhoven, Netherlands (2024)