KETE Soccer Simulation 2D Team Description Paper 2026

Enes ÇELEĞEN, Enes Yavuz ALPAY, Şuayp Talha KOCABAY, Kutay DEMİRBAŞ

TUBITAK Science High School, Kocaeli, Türkiye

{haberenes, makalidap, kocabaysuayptalha08,
kutaydemirbas2009}@gmail.com

Abstract. This paper presents the tactical planning and implementation framework of KETE, a newly established team participating in the 2026 RoboCup Soccer Simulation 2D league. The study focuses on processing potential opponent tactics to develop a robust, defense-oriented strategy characterized by opportunistic offensive transitions. By leveraging Machine Learning (ML) algorithms, the system aims to enhance on-field coordination and player agility. The proposed architecture ensures that agents can make synchronized, high-speed decisions, providing a competitive edge in dynamic match environments.

Keywords: RoboCup 2D Soccer Simulation, Tactical Planning, Machine Learning, Multi-Agent Systems (MAS), Defensive Strategy, Autonomous Agents

1 Introduction

In the dynamic world of 2D football strategy, KETE's mission is to build a team defined by defensive strength, discipline, and collective intelligence. Our philosophy centers on creating a resilient structure where every player contributes to a unified defensive system. By combining advanced learning approaches, we aim to develop a team that anticipates threats, adapts to challenges, and maintains balance under pressure. Rather than reacting to the opponent, KETE strives to control the game through organization, awareness, and cohesion. Our ultimate goal is to transform defensive stability into a foundation for consistent success and decisive transitions.

1.1 The Game

The simulation environment is meticulously engineered as a standard 11v11 association football match, delivering a high-fidelity 2D competitive experience that mirrors the strategic complexity of real-world soccer. Within this framework, the primary objective transcends the traditional goal of outscoring the opponent; it focuses on the sophisticated orchestration of eleven autonomous agents designed to execute the most effective tactical actions under dynamic constraints (as shown in Figure 1). Our developmental focus lies in the synthesis of robust algorithmic coding and advanced model training, ensuring that each agent possesses the situational awareness required for optimal decision-making. By leveraging high-dimensional data from the 2D simulator, we aim to train these models to recognize spatial patterns, manage stamina efficiently, and maintain a disciplined defensive posture. Ultimately, the goal is to bridge the gap between raw computational logic and fluid, collective intelligence, transforming a group of individual scripts into a cohesive, goal-oriented tactical unit capable of dominating the pitch through superior positioning and synchronized execution.

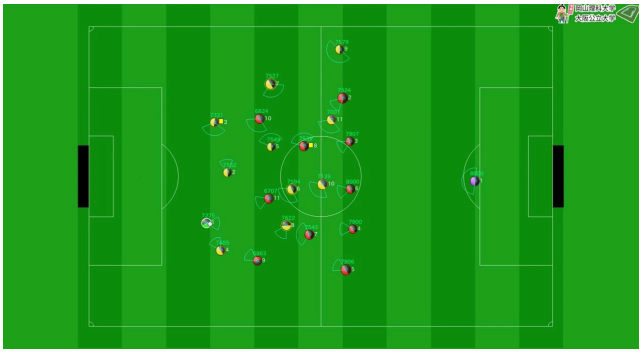

Figure 1. A screenshot from the game.

Drawing a significant conceptual parallel, the simulation's core mechanics are profoundly inspired by the popular game HaxBall, offering a streamlined yet highly competitive physics-based environment (as illustrated in Figure 2). This top-down, 2D perspective prioritizes precision in collision physics, momentum management, and spatial positioning. By adopting this intuitive framework, the simulation strips away visual complexity to focus on the raw strategic essence of the sport. For our team, this means the training models must account for rapid velocity changes and circular hit-box interactions, demanding a higher level of micro-positioning and predictive movement. Integrating this HaxBall-like fluidity with our defensive-heavy tactical layer allows the agents to maintain a "tight-knit" formation, where every bounce and interception is calculated with mathematical rigor, ensuring a seamless blend of arcade-style agility and professional-grade strategic depth.

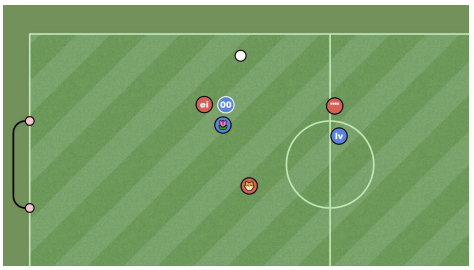

Figure 2. An image of HaxBall, the 2D football game.

1.2 Reinforcement Learning

Expanding our tactical scope, Reinforcement Learning (RL) serves as the sophisticated guiding mentor that governs the behavioral evolution of our agents. In the high-stakes environment of a 2D football simulation, RL empowers players to transcend static programming by adapting and refining their decision-making processes through continuous, real-time feedback loops. Within our defensive-heavy framework, the RL agent operates within a high-dimensional state space where every successful interception, disciplined containment, and strategic clearance is quantified as a positive reinforcement (as illustrated in Figure 3).

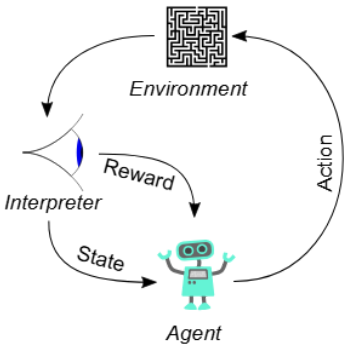

Figure 3. An illustration of reinforcement learning.

Unlike rigid heuristic systems, our RL-driven model treats the defensive landscape as a dynamic puzzle. Each encounter with an opponent becomes a computational lesson; the agent learns to evaluate the trade-off between aggressive pressing and maintaining structural integrity. By optimizing Reward Functions that prioritize "low-risk, high-gain" maneuvers, the system shapes players who are proficient in navigating complex offensive surges without breaking the collective formation. Furthermore, this learning methodology ensures an adaptive and responsive transition strategy. When a turnover occurs, the RL-trained agents instantaneously pivot from a deep-block defensive posture to an explosive counter-attacking phase, identifying the most efficient vectors for ball progression.

1.2.1 Q-Learning

Within our computational framework, Q-Learning operates as the master strategist, serving as the primary architect for the high-level decision-making processes of our agents. In the context of a 2D football simulation, where the environment is characterized by rapid transitions and stochastic movements, Q-Learning provides a rigorous mathematical structure for evaluating the long-term utility of specific actions. Rather than pursuing a purely aggressive offensive, our implementation of the Q-Table is optimized to prioritize defensive stability and the identification of optimal "transition windows" (as illustrated in Figure 4).

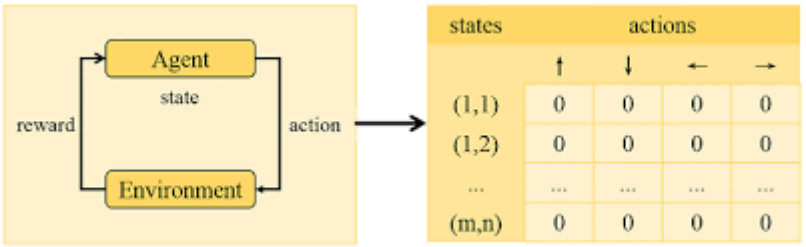

Figure 4. An illustration of Q-Learning.

The system functions by systematically exploring an expansive state-action space, where each agent learns to map its current environmental state—such as its position relative to the ball, teammates, and the opponent's defensive line—to a specific optimal action. Through an iterative process of Temporal Difference (TD) learning, the model evaluates the success or failure of various tactical maneuvers. In our defensive-oriented paradigm, "success" is redefined; it is not merely scored goals, but the successful minimization of the opponent's expected goal probability and the maintenance of a compact formation.

As the system evolves, it fine-tunes the Q-values, allowing players to develop an innate sense of "spatial intelligence." This enables them to act as agile, effective guardians of the pitch who can anticipate offensive surges and intercept passing lanes with surgical precision. By balancing exploration and exploitation, Q-Learning ensures that the team does not just react to the game, but actively shapes it, crafting a resilient structure that waits for the perfect moment to exploit an opponent's overextension. This strategic foresight transforms the collective unit into a disciplined force, where every individual decision is calibrated to contribute to the holistic defensive integrity of the squad.

1.3 Technical Strategy

While Reinforcement Learning (RL) broadly conceptualizes learning through environmental interaction, our architecture specifically implements a dual-layered MDP framework to separate continuous, low-level actuator control from discrete, high-level tactical reasoning. The environment is modeled as a Markov Decision Process defined by the tuple $(S,\ A,\ R,\ P, \gamma)$, where S is the state space, A is the action space, $R: S \times A \rightarrow R$ is the reward function, P represents the transition dynamics, and $\gamma \in [0, 1)$ is the discount factor.

To achieve our dual-perspective logic, we decouple this global MDP into two distinct formulations: micro-positioning optimized via Proximal Policy Optimization (PPO) and macro-strategy optimized via Deep Q-Networks (DQN).

1.3.1 Micro-Positioning via Proximal Policy Optimization (PPO)

Micro-positioning addresses the continuous control problem of an individual agent navigating the 2D physics environment (e.g., precise momentum management, collision avoidance, and trajectory interception). Because the action space for movement in the RoboCup 2D simulator involves continuous variables (dash power, turn angles), traditional value-based methods are inefficient. Therefore, we utilize Proximal Policy Optimization (PPO), an actor-critic policy gradient method that ensures stable, incremental policy updates.

We define the micro-positioning MDP as follows:

- **State Space (S_{micro}):** A continuous vector containing the agent's absolute coordinates, current velocity, stamina level, and relative polar coordinates (distance and angle) to the ball and the nearest opponent.
- **Action Space (A_{micro}):** A continuous vector defining actuator commands, specifically $a = [p_{dash}, \theta_{turn}]$, where $p_{dash} \in [-100, 100]$ is the dash power and $\theta_{turn} \in [-180^{\circ}, 180^{\circ}]$ is the turn angle.
- **Reward Function (R_{micro}):** The reward is designed to reinforce optimal defensive tracking and stamina preservation. It is calculated as:

$$R_{micro}(s_t, a_t) = w_1 \Delta d_{ball} + w_2 c_{interception} - w_3 \Delta stamina$$

  where Δd_{ball} is the reduction in distance to the ball (or target marking zone), $c_{interception}$ is a sparse binary reward for successfully blocking a pass, and $\Delta stamina$ penalizes exhaustive, inefficient dashing.

PPO optimizes the policy $\pi_\theta(a|s)$ by maximizing the clipped surrogate objective function, ensuring the agent learns agile, HaxBall-inspired collision mechanics without catastrophic policy divergence during training.

1.3.2 Macro-Strategy via Deep Q-Networks (DQN)

Macro-strategy dictates the collective, high-level decision-making of the team, such as shifting the defensive block, triggering the "Onion Defense" press, or initiating explosive transitions. Because these tactical shifts are distinct, categorical choices rather than continuous movements, we model this as a discrete action problem solved via a Deep Q-Network (DQN).

We define the macro-strategy MDP as follows:

- **State Space (S_{macro}):** A discretized representation of the pitch. The state vector includes the spatial distribution of both teams (centers of mass for defense, midfield, and attack lines), ball possession status, and the current game phase.
- **Action Space (A_{macro}):** A discrete set of tactical directives broadcasted to the agents: A_{macro} = {Retreat, Hold Line, Squeeze/Press, Explosive Transition}.
- **Reward Function (R_{macro}):** The objective is to minimize the opponent's expected goal probability and exploit turnovers. The reward is formulated as:

$$R_{macro}(s_t, a_t) = \alpha I_{turnover} + \beta(D_{ideal} - D_{actual}) - \eta I_{conceded}$$

  where $I_{turnover}$ is a positive reward for regaining possession in the opponent's half, $(D_{ideal} - D_{actual})$ penalizes deviation from the ideal inter-layer distance

of our "Onion Defense" formation, and $I_{conceded}$ is a heavy penalty for allowing a shot on goal.

The DQN estimates the optimal action-value function $Q^*(s,a)$, governed by the Bellman equation:

$$Q^*(s,a) = E_{s'\sim P}[R_{macro}(s,a) + \gamma \, max_{a'} Q^*(s',a')]$$

1.3.3 Algorithmic Integration

The true efficacy of KETE's system emerges from the hierarchical integration of these methods. The DQN (Macro) evaluates the global state space S_{macro} at a lower frequency (e.g., every 10 simulation cycles) and selects a discrete tactical action $a_{macro} \in A_{macro}$. This tactical directive modifies the reward weights (w_1, w_2, w_3) or the target tracking zones in the PPO framework (Micro). Consequently, the continuous, high-frequency actions of the agents (A_{micro}) are dynamically constrained and guided by the global Q-values, ensuring that individual agility mathematically serves the collective defensive structure.

2 Game Strategy

Our tactical philosophy is built upon the principle of Defensive Luring and Explosive Transitions. The core objective is to concede spatial dominance in non-critical areas to entice the opponent into our defensive third, effectively "trapping" their key players in high-density zones before launching a lethal counter-attack.

2.1 Defensive Phase

When the opponent is in possession, the team adopts a disciplined, low-block formation. The primary goal is not immediate ball recovery but the systematic denial of passing lanes into the box. We employ a "Squeezing" mechanic where, as the opponent enters our half, the defensive and midfield lines contract. This creates a high-pressure pocket designed to bait the opponent into overcommitting. Once the opponent's transition is overextended, a coordinated press is triggered to force a turnover in a congested area (as shown in Figure 5).

This multi-layered defensive structure, often referred to as the "Onion Defense," continuously optimizes the distance between layers to minimize exploitable gaps while maintaining flexibility for lateral coverage. The inter-layer density is adjusted dynamically to prevent interior passes and cross-channel progression, ensuring that opponents face a compact, difficult-to-penetrate formation (as illustrated in Figure 6).

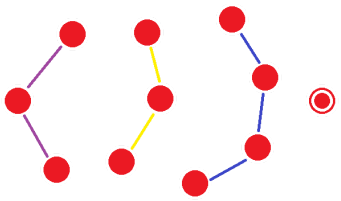

Figure 5. Multi-Layered "Onion Defense" Architecture: Dynamic Inter-Layer Distance Optimization for Penetration Denial.

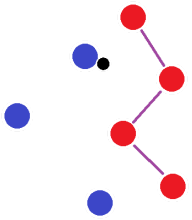

Figure 6. Dynamic "Onion Defense" Model: Optimizing Inter-Layer Density to Neutralize Interior Passes and Cross-Channel Progression.

2.2 Midfield Dynamics

The midfield acts as the tactical pivot of the team. In the defensive phase, midfielders maintain a "Screening" posture to protect the defensive line. Upon reclaiming possession, the objective shifts instantaneously to Rapid Verticality. Instead of lateral ball circulation, the midfield is programmed to identify the most direct vertical lane to exploit the space vacated by the lured opponent. This "Bait-and-Switch" maneuver is the cornerstone of our offensive strategy.

2.3 Offensive Phase

Our offensive mechanics are optimized for speed and efficiency rather than sustained possession. When the ball is won, the forwards utilize "Diagonal Stretching" runs to pull apart the opponent's backline, which is typically caught in a high-line position. The aim is to finish the attack within a few high-velocity touches, minimizing the opponent's time to regroup.

2.4 Goalkeeper Mechanics

The goalkeeper is integrated into the system not just as a shot-stopper but as the initial playmaker. Beyond high-reflex saving models, the goalkeeper is trained in "Immediate Distribution" (ID) logic. After a save or a claim, the keeper prioritizes long, accurate throws or kicks to the wings, bypassing the opponent's counter-press and initiating the counter-attack before the defense can stabilize.

3 Who are We?

The driving force behind this project is Team KETE, a synergistic collective of four software researchers and enthusiasts: Enes Çeleğen, Enes Yavuz Alpay, Şuayp Talha Kocabay, and Kutay Demirbaş. As students of the TÜBİTAK Science High School, we represent a community dedicated to scientific excellence and high-level engineering.

Although 2026 marks our debut year in the RoboCup Soccer Simulation 2D league, we approach this challenge with a foundation built on rigorous technical training and a shared passion for innovation. Each member of KETE brings a unique set of competencies to the table, backed by various achievements in software development.

What defines KETE is not just our individual proficiency, but our cohesive team structure. We believe that the most elegant solutions emerge from the intersection of deep analytical thinking and fluid collaboration. Our internal synergy allows us to tackle multifaceted problems—such as the "Onion Defense" logic or multi-agent reinforcement learning—with a unified vision. As a debutant team, we are committed to pushing the boundaries of the simulation environment, transforming our academic background and enthusiasm for code into a competitive, intelligent, and resilient strategic entity.

References

1. Akiyama, H., & Nakashima, T. (2013). Helios base: An open source package for the RoboCup soccer 2D simulation. In Robot Soccer World Cup XVII (pp. 528-535). Springer, Berlin, Heidelberg.
2. Akiyama, H., Nakashima, T., & Hatakeyama, K. (2022). HELIOS2022: Team Description Paper. In RoboCup 2022 Symposium and Competitions. Bangkok, Thailand.
3. Fathi, E., & Mazloum, S. (2023). EMPEROR Soccer Simulation 2D Team Description Paper 2023. In RoboCup 2023 Symposium. Bordeaux, France.
4. Gabel, T., Eren, B., Eren, Y., Sommer, F., & Godehardt, E. (2023). FRA-UNIted — Team Description 2023. In RoboCup 2023 Symposium.
5. Krishnan, S., Lam, M., Chitlangia, S., Wan, Z., Barth-Maron, G., Faust, A., & Reddi, V. J. (2022). QuaRL: Quantization for Fast and Environmentally Sustainable Reinforcement Learning. arXiv preprint arXiv:1910.01055.
6. Marian, S., Luca, D., Sacuiu, R., Sarac, B., & Cotalea, O. (2023). OXSY 2023 Team Description Paper. In RoboCup 2023 Symposium.
7. Noda, I., & Matsubara, H. (1996). Soccer server and researches on multiagent systems. In Proceedings of the IROS-96 Workshop on RoboCup (pp. 1-7).
8. Prokopenko, M., & Wang, P. (2018). Gliders2d: Source Code Base for RoboCup 2D Soccer Simulation League. arXiv preprint arXiv:1812.10202.
9. Sayareh, A., Zare, N., Amini, O., Firouzkouhi, A., Sarvmaili, M., & Matwin, S. (2023). Observation Denoising in CYRUS Soccer Simulation 2D Team For RoboCup 2023. arXiv preprint arXiv:2305.19283.
10. Tomoharu, N., Akiyama, H., Suzuki, Y., Ohori, A., & Fukushima, T. (2018). HELIOS2018: Team Description Paper. In RoboCup 2018 Symposium.
11. Yamagata, T., McConville, R., & Santos-Rodriguez, R. (2021). Reinforcement Learning with Feedback from Multiple Humans with Diverse Skills. arXiv preprint arXiv:2111.08596.
12. Zare, N., Keshavarzi, A., Beheshtian, S. E., Mowla, H., Akbarpour, A., Jafari, H., et al. (2016). Cyrus 2D Simulation Team Description Paper 2016. In RoboCup 2016. Leipzig, Germany.
13. Zare, N., Amini, O., Sayareh, A., Sarvmaili, M., Firouzkouhi, A., Matwin, S., & Soares, A. (2021). Improving Dribbling, Passing, and Marking Actions in Soccer Simulation 2D Games Using Machine Learning. In RoboCup 2021: Robot World Cup XXIV. Springer.
14. HaxBall: A Physics-Based 2D Soccer Game. Retrieved from https://www.haxball.com (Accessed: March 2026).