Tactical Digital Twin: Reinforcement Learning in Football

Michele Curci
Politecnico di Torino
Matr. 337117
michele.curci@studenti.polito.it

Lorenzo Ferrandi
Politecnico di Torino
Matr. 349309
lorenzo.ferrandi@studenti.polito.it

Xueyufei Zhang
Politecnico di Torino
Matr. 336472
xueyufei.zhang@studenti.polito.it

Abstract

This paper presents an integrated framework for football tactical analysis using Multi-Agent Reinforcement Learning (MARL). We extend a standard simulated environment with three novel modules: (1) a What-If Analysis engine initialized by real-world StatsBomb event data, (2) an Adversarial Learning component for robust policy stress-testing, and (3) a Customizable Player Extension to account for heterogenous agent attributes. The code can be found here.

1 Introduction

The integration of advanced data analytics has fundamentally transformed professional football, moving the sport from subjective observation toward a quantitative, evidence-based discipline. While traditional metrics such as Expected Goals (xG) and spatial tracking provide a descriptive overview of match performance, they often fail to capture the underlying causal dynamics of tactical decisions. To address this limitation, the concept of a "Tactical Digital Twin" has emerged—a generative simulation environment where historical match data can be replayed, modified, and optimized through autonomous agents.

The core challenge in creating such a twin lies in the inherent complexity of football as a dynamic, multi-agent system. Player interactions are governed by a continuous state-action space, partial observability (the "fog of war" on the pitch), and the need for simultaneous cooperation and competition. Traditional rule-based AI often lacks the fluidity to model these emergent behaviors. Consequently, Reinforcement Learning (RL) has become a primary candidate for modeling these interactions, as it allows agents to discover optimal strategies through trial and error within a physics-constrained environment.

This project introduces a comprehensive RL-based framework tailored for offensive tactical modeling. By building upon the standard 120×80m pitch geometry and leveraging the StatsBomb Open Dataset, we propose an environment that is not only a training ground for agents but also a laboratory for "What-If" analysis. Our framework extends the state-of-the-art by incorporating adversarial training to ensure defensive realism and heterogeneous agent attributes to model the physical diversity found in professional squads.

1.1 Research Questions

To guide our investigation into the efficacy of MARL for tactical modeling, we define the following core research questions:

- **RQ1: Counterfactual Reproducibility.** To what extent can agents trained via Proximal Policy Optimization (PPO) successfully resolve high-leverage tactical scenarios extracted from real-world historical event data?
- **RQ2: Robustness through Adversarial Play.** How does the introduction of an asynchronous adversarial evolution mechanism impact the stability and emergent coordination of offensive passing policies compared to static defensive baselines?
- **RQ3: Impact of Agent Heterogeneity.** How do variations in physical and technical player attributes—specifically locomotion speed and shooting precision—alter the feasibility and selection of optimal tactical trajectories within the simulation?

2 Related work

Recent literature on policy learning and decision evaluation in soccer—both from real-world data and in simulated environments—highlights several consolidated methodological directions. First, it emphasizes the importance of defining state representations that capture tactical structure and spatial constraints, moving beyond purely event-based descriptions and motivating modeling choices that are consistent with performance analysis in data-rich settings [4]. Second, in multi-agent reinforcement learning settings, the limitations of sparse reward signals are widely acknowledged; accordingly, reward-shaping techniques have been proposed to inject contextual information and tactical priors, providing denser and more informative feedback during training [1]. In parallel, offline reinforcement learning approaches trained on historical datasets (tracking and event data) have been used to estimate value functions and enable counterfactual analyses of decisions, yielding quantitative measures of action quality relative to plausible alternatives [3]. Moreover, action-valuation methods show how incorporating context (e.g., space control) into value estimation makes decision evaluation more sensitive to the positioning of teammates and opponents [2]. Finally, the need for standardized training and evaluation protocols (benchmarks, self-play and opponent populations, comparable metrics) is increasingly stressed to ensure reproducibility and meaningful comparisons, especially as scenario complexity grows [5]. Accordingly, our project adopts spatially informed state and reward design and evaluates policies with decision-centric metrics in targeted scenarios, rather than relying solely on terminal outcomes.

3 Method

In this section, we detail the theoretical and algorithmic framework underpinning the tactical simulation. Our approach integrates state-of-the-art Reinforcement Learning (RL) techniques to navigate the high-dimensional, continuous state space of a football pitch. The methodology is built upon two pillars: the Proximal Policy Optimization (PPO) algorithm, which ensures stable policy updates in a complex environment, and Multi-Agent Reinforcement Learning

(MARL), which governs the collaborative and competitive interactions between autonomous entities. By combining these, we create a robust learning pipeline capable of evolving from random exploration to structured tactical coordination.

3.1 Proximal Policy Optimization (PPO)

Proximal Policy Optimization is employed as the primary learning algorithm due to its balance between ease of implementation, sample efficiency, and robustness to hyperparameter tuning. PPO is an **Actor-Critic** method that addresses the instability of standard Policy Gradient methods, where a single large step in the policy space can collapse the learning process. The core of PPO is the **clipped surrogate objective function**. Traditional policy gradients rely on the estimator:

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[\log \pi_\theta(a_t|s_t)\hat{A}_t] \tag{1}$$

which often leads to destructively large policy updates. PPO constrains these updates by considering the probability ratio $r_t(\theta)$ between the new policy and the old policy:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{2}$$

The objective function $L^{CLIP}(\theta)$ is defined as:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{3}$$

Where $\hat{A}_t$ is the estimated advantage at time t, and ϵ is a hyperparameter (typically 0.1 or 0.2). This clipping mechanism ensures that the policy does not move too far away from the previous iteration, effectively creating a “trust region” that maintains training stability even in the volatile dynamics of a football match.

Simultaneously, the **Value Function (Critic)** is trained to minimize the Mean Squared Error (MSE) between the predicted value and the actual discounted returns, assisting the Actor by reducing the variance of the advantage estimates through Generalized Advantage Estimation (GAE).

3.2 Multi-Agent Reinforcement Learning (MARL)

Football is inherently a multi-agent problem characterized by a shared environment where the transition dynamics depend on the joint actions of all agents: $P(s'|s, a_1, \ldots, a_n)$. We adopt a **Decentralized Execution** framework, where each agent makes decisions based on its local observations, ensuring the model remains scalable and realistic.

3.2.1 *Centralized Training, Decentralized Execution (CTDE).*

To mitigate the non-stationarity problem—where an agent’s environment changes as other agents learn—we utilize the CTDE paradigm. During training, the algorithm has access to global state information and the actions of all agents to stabilize the critic’s value estimates. However, during the “What-If” evaluation or real-time simulation, the **Actor** (the policy) only receives local observations $\mathbf{o}_i$, such as the relative distance to the ball, teammates, and goalposts.

3.2.2 *Parameter Sharing.*

To accelerate convergence in our 2v1 and 2v0 scenarios, we implement **homogeneous parameter sharing** among agents of the same team. By sharing the weights of the neural networks across attackers, the agents benefit from a collective pool of experience. This approach is particularly effective in football, as the fundamental skills required—such as finding space or tracking the ball—are universal across offensive roles.

The coordination is further refined by the **Reward Shaping** mechanism, where agents receive a global team reward for goals scored or successful passes, encouraging the emergence of cooperative behaviors like “wall passes” and decoy runs that would be impossible to learn in a strictly individualistic RL setup.

4 Experiment

The experimental framework is designed to evaluate the adaptability of MARL policies across diverse tactical scenarios. Our experiments utilize a custom-built simulation environment based on the **Ray RLlib** library for distributed training and the **StatsBomb** open dataset for scenario initialization. The following subsections detail the data acquisition process, the architectural constraints of the environment, and the three core extensions: *What-If Analysis*, *Adversarial Learning*, and *Realistic Player Customization.*

4.1 Dataset

To ground the simulation in professional football dynamics, we leverage the **StatsBomb Open Dataset**, one of the most comprehensive publicly available sources of high-fidelity football event data. The dataset provides a granular log of every technical action occurring during a match, offering a rich repository of spatial and temporal information.

4.1.1 *Data Characteristics and Granularity.*

The StatsBomb data follows a specific event-stream format, where each entry represents a discrete action such as a pass, shot, tackle, or dribble. Each event is accompanied by:

- **Spatiotemporal Metadata:** Precise (x, y) coordinates of the event origin and, where applicable, the destination (e.g., the end-point of a pass).
- **Contextual Attributes:** Detailed modifiers including body part used, pressure indicators, and technical outcomes (e.g., completion status).
- **High-Frequency Tracking-Lite:** While not full 25fps tracking, the inclusion of "freeze frames" for shots provides the locations of all players in the frame at the moment of the event.

4.1.2 *Pre-processing and Spatial Normalization.*

The raw dataset utilizes a proprietary coordinate system where the pitch is mapped to a 120×80 unit grid. To ensure compatibility with the Reinforcement Learning environment’s neural network inputs, we implemented a pre-processing pipeline to normalize these values into a continuous $[0, 1]$ range. This transformation preserves the relative spatial relationships and pitch geometry while facilitating faster gradient convergence during the training phase.

4.2 Environment

To train agents capable of complex tactical reasoning, a custom simulation environment was developed, which bridges the gap between abstract reinforcement learning benchmarks and the dynamic nature of real-world football. The environment is built upon a continuous state-action space, determined by a hybrid physics engine and a hierarchical reward structure.

4.2.1 Physical Infrastructure and Objects. The simulation takes place on a standard football pitch ($120m \times 80m$). To facilitate neural network convergence, all spatial coordinates are normalized to the range $[0, 1]$.

A core innovation of the environment is the **Dual Dynamics System** for the ball, designed to balance computational efficiency with physical realism. The ball toggles between two distinct states:

- **Free Physics State:** When the ball is not controlled by any agent, it obeys Newtonian mechanics. The realistic friction with a decay coefficient ($\mu \approx 0.15\%$) was modeled, simulating the natural deceleration of a rolling football on grass.
- **Owned State:** When a player moves within the control radius ($0.11m$) of the ball, the physics engine switches to an attachment mode. This allows the agent to dribble and change direction without the instability of continuous rigid-body collision calculations.

4.2.2 Agent Architecture: Perception and Action. The agents are designed to mimic human cognitive limitations and capabilities, applicable across both single-agent and multi-agent scenarios.

1. Constrained Perception (Field of View): Unlike traditional simulations that provide global state access, our agents operate under partial observability. We implemented a **Field of View (FOV)** mechanism where agents can only accurately perceive objects within a specific fan-shaped area. Regions behind the player (the "Blind Spot") are not visible. This constraint forces the policy to learn active information-gathering behaviors, such as turning to scan for teammates or opponents before executing a pass.

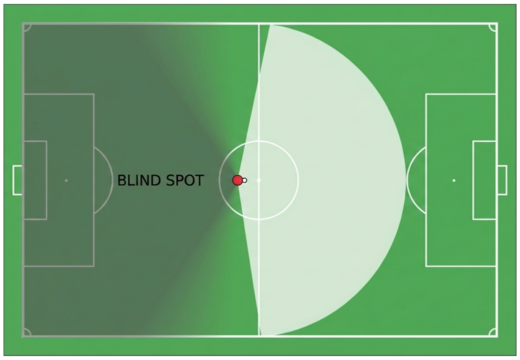

Figure 1: Fan-Shaped Field of View

2. Continuous Action Space: The action space is continuous and high-dimensional. At each timestep, an agent outputs:

- A movement vector $\vec{v} \in \mathbb{R}^2$ controlling direction and speed.
- Discrete triggers for high-level skills: *Shoot*, *Pass*, and *Tackle*.

Crucially, skill execution is **probabilistic**. Parameters such as shot power and accuracy are subject to Gaussian noise, modeling the inherent imperfection of human technique.

4.2.3 Curriculum of Scenarios. The environment supports a curriculum learning approach, progressively increasing task complexity:

- **Single-Agent Drills:** Specific sub-tasks were defined such as *Move* (dribbling to a target), *Shot* (scoring against a static keeper), and *View* (object tracking). These scenarios serve to set the agent's basic motor skills.
- **Multi-Agent Tactical Scenarios:** Agents trained in isolation are currently evaluated in N vs 1 scenarios, with N vs N settings planned for future work. Here, role-specific behaviors (*Attacker*, *Defender*, *Goalkeeper*) emerge, and the challenge shifts from ball control to team strategy.

4.2.4 Reward Strategy. Defining a reward function for football is challenging due to the extreme sparsity of goals. To address this, we implemented a composite reward structure:

$$R_{total} = (1 - \alpha)R_{dense} + \alpha R_{sparse} \tag{4}$$

Dense Rewards (Tactical Heatmaps): To encourage effective spatial positioning, we utilize *Elliptical Reward Grids*. These are potential fields that assign continuous value to regions of the pitch based on tactical relevance (e.g., the "Golden Zone" in front of the goal). Agents receive positive feedback for navigating into these high-value zones while maintaining possession.

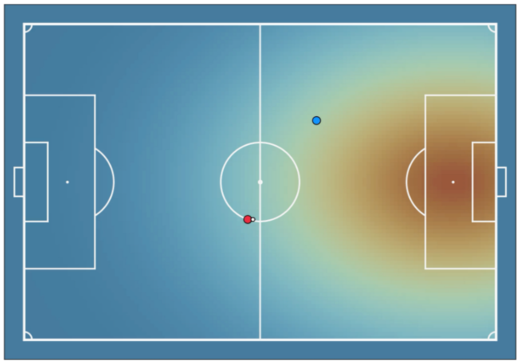

Figure 2: Elliptical Reward Grids

Sparse Rewards (Events): High-value rewards are assigned for specific game events to reinforce task objectives, including +1.0 for scoring a goal, positive rewards for successful passes and tackles, and penalties for losing possession or sending the ball out of bounds.

4.3 Adversarial learning

In the application scenarios of the Tactical Digital Twin, it's unable to provide adaptive pressure if the oppnents are static, and agents would find it difficult to learn how to handle complex situations.

To address these shortcoming, we developed and used an Adversarial Multi-Agent Reinforcement Learning (MARL) framework. In this section, we will disscuss the following three core components in detail: the **asynchronous evolution** mechanism, the **opponent pool** strategy for robustness, and the **entropy regularization** techniques used to stabilize training.

4.3.1 Asynchronous Evolution. Traditional self-play algorithms often suffer from training instability when agents update their policies simultaneously. To address this, we adopt an *Asynchronous Evolution Mechanism.*

Let π_θ denote the policy of the attacking team (Attacker) parameterized by θ, and π_ϕ denote the policy of the defending team (Defender) parameterized by ϕ. Instead of updating θ and ϕ at the same time, we alternate the training phases in cycles.

The training process revolves around a switch frequency f_{switch}. A complete cycle consists of two sequential phases: Attacker optimization followed by Defender optimization. Consequently, the cycle length is $T_{cycle} = 2 \times f_{switch}$. Specifically, we defined a small switch frequency $f_{switch} = 500$ episodes in our expriments, due to the limited computational resources and the resulting small number of total training episodes.

- **Phase A (Attacker Optimization):** For f_{switch} episodes, maximize $J(\theta)$ while ϕ is fixed.
- **Phase B (Defender Optimization):** For the subsequent f_{switch} episodes, maximize $J(\phi)$ while θ is fixed.

This alternating freezing ensures that the learning agent faces a static environment relative to the opponent's behavior, allowing the PPO algorithm to converge to a counter-strategy before the opponent adapts.

Independent Policy Mapping – Unlike parameter-sharing approaches often used in homogeneous MARL, we implement **Independent Policy Mapping**. Each agent (e.g., `att_1`, `def_1`) maintains its own distinct policy network.

This architectural choice is crucial for the "Realistic Players" objective (discussed in subsequent sections), as it allows us to put heterogeneous attributes (speed, shot power) into specific agents without affecting other agents within a uniform network.

4.3.2 Opponent Pool. A common failure mode in adversarial RL is *catastrophic forgetting* or cyclic non-transitivity (e.g., Rock-Paper-Scissors dynamics), where an agent learns to beat the current opponent but loses the ability to defeat previous versions.

To ensure robustness, we implemented a historical **Opponent Pool**. At the end of each training phase (every f_{switch} episodes), the current policy snapshot is saved to a pool $\mathcal{P}$. During training, the opponent policy π_{opp} is sampled as follows:

$$\pi_{opp} = \begin{cases} \pi_{latest} & \text{with probability } p \\ \pi \sim \text{Uniform}(\mathcal{P}) & \text{with probability } 1-p \end{cases} \tag{5}$$

In our experiments, we set $p = 0.5$. This ensures that agents spend 50% of their time mproving their strategy against the state-of-the-art opponent, while the remaining 50% is spent validating their strategy against diverse historical behaviors, preventing overfitting to a single playstyle.

4.3.3 Entropy Regularization. To balance exploration and exploitation, we implemented an exponential entropy decay schedule. High entropy is favored in the early cycles to encourage agents making diverse tactical discovery, while low entropy is enforced in later cycles to stabilize performance.

The entropy coefficient β at episode e is updated according to:

$$\beta(e) = \beta_{start} \cdot \left(\frac{\beta_{end}}{\beta_{start}}\right)^{e/E_{total}} \tag{6}$$

where E_{total} is the total number of training episodes.

For attackers, we decayed β from 0.05 to 0.002. For defenders, who have a reactive task, we used a lower range from 0.01 to 0.0005. This differences reflects that attackers must generate creative solutions to score, whereas defenders prioritize consistent positioning in football game.

4.4 Realistic players

We introduced the **Realistic Players** extension to reduce a common simplification in simulated football environments, namely the assumption that all agents share identical *physical* and *technical* capabilities. Such homogeneity can lead to policies that exploit unrealistic dynamics and can obscure how individual constraints affect feasibility, timing, and action quality.

To better approximate real match conditions, this extension aims to emulate a more *heterogeneous* playing field, where the outcome of the same high-level decision depends on the specific attributes of the acting player. The goal is not to increase complexity for its own sake, but to enable a more faithful analysis of learned behaviours under controlled variations of player capabilities, in line with the intended use of the environment as a tactical experimentation tool.

From an implementation perspective, we introduced additional *single-agent configuration* options and extended the corresponding *single-agent scenario* definitions. Importantly, we designed the extension as a controlled intervention: across all experiments we kept the same *environment interface*, the same *observation/action structure*, and the same *training pipeline*), modifying only a small set of parameters depending on the task setting. This ensures that any behavioural differences can be attributed to the injected player heterogeneity rather than to changes in the learning setup.

4.4.1 Movement setting: speed heterogeneity. In the **Movement** setting, we introduce heterogeneity by modifying the attacker's locomotion capability. Specifically, we implemented two movement profiles (**Slow Player** vs **Fast Player**) by applying a *multiplicative speed modifier* to the player's movement parameter, resulting in an otherwise identical agent with different effective pace. This allows the same policy-learning setup to be run under different physical constraints, isolating the role of mobility in the feasibility of movement trajectories and positioning decisions.

To evaluate performance in the *Movement* setting under speed heterogeneity, we track the **episodic return** $R_e = \sum_{t=0}^{T-1} r_t$ and report a smoothed learning curve via the **moving average** over a window of $w = 100$ episodes,

$$\bar{R}_e = \frac{1}{w} \sum_{i=e-w+1}^{e} R_i. \tag{7}$$

This reduces the variance typical of on-policy training and allows a clearer comparison between profiles trained under the same environment interface and training pipeline. As shown in Figure 3, the **fast** profile increases more rapidly and converges to a substantially higher reward regime than the **slow** profile, indicating both improved sample efficiency and a higher achievable performance plateau when movement constraints are less restrictive. Consistently with the curve separation, the final 10% of training yields a mean return of 9.327 for **fast** versus 3.453 for **slow**.

Overall, this gap indicates that stricter kinematic constraints make it harder for the agent to consistently reach advantageous spatial configurations within the same training budget; moreover, qualitative inspection of the renderings suggests that the agent tends to engage in longer dribbling phases around the opponent, resulting in episodes that last longer on average and delaying access to high-reward situations.

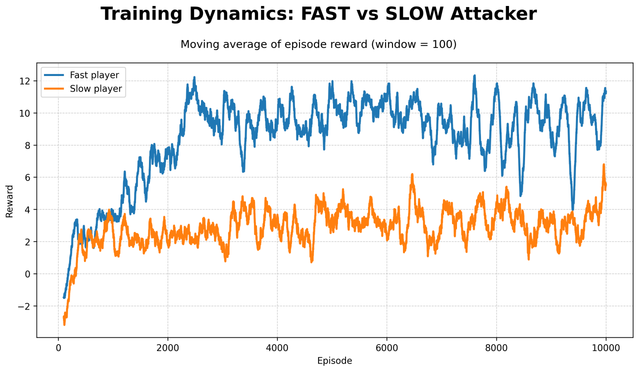

Figure 3: Moving average of episodic return in Move setting (Fast vs Slow)

4.4.2 Shooting setting: shot execution heterogeneity and centralized goalkeeper constraint. In the **Shooting** setting, we introduce heterogeneity at the level of action execution by defining two distinct shooting profiles, **Strong Shot** and **Weak Shot**. These profiles differ not only in *shot power* but also in *shot precision*, which we model as increased or reduced *angular noise* in the shot direction (i.e., variability in the effective shooting angle). In other words, **Strong Shot** corresponds to higher power and tighter *angular dispersion*, while **Weak Shot** corresponds to lower power and larger dispersion around the intended target angle.

To provide a simple and interpretable defensive constraint, we reposition the opposing defender as a *goalkeeper surrogate*: we place it *centrally on the goal line*, such that it can effectively intercept only centrally aimed shots. This configuration is intended to encourage the attacker to learn shot placement without relying on trivial, straight-to-center trajectories, while keeping the rest of the environment and learning pipeline unchanged.

In the **Shooting** setting, we observed an unintended behavioral pattern rather than an improvement in shot preparation.

To describe where the policy decides to shoot from, we log the *shot distance* (distance between the ball position at shot time and the goal center) and analyze its distribution across test cases. As shown in Figure 4, shots are concentrated at relatively **long range**, with the **Strong** profile even more skewed towards larger distances than **Weak**. This means that, instead of moving closer to the goal to create a better shooting situation, the agent often chooses to shoot as soon as possible even when far from goal. A likely reason is that the Strong profile makes these early shots “good enough” to be selected by the policy (because they are more powerful and less noisy), so the agent has less incentive to spend extra steps approaching the goal; as a result, the policy settles on a simple behavior: **shoot early from far**.

4.5 What-if

The *What-If* extension serves as a counterfactual evaluation framework designed to explore alternative outcomes of historical match events. By utilizing the normalized StatsBomb data, we transition from a purely descriptive analysis of football tactics to a generative one, where trained agents are tasked with resolving real-world scenarios under the constraints of the learned policy.

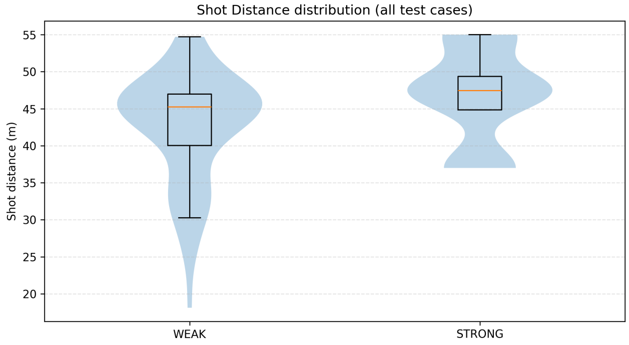

Figure 4: Shot distance distribution in Shot setting (Strong vs Weak)

4.5.1 Historical State Injection. The primary technical challenge in the *What-If* module is the accurate mapping of historical events into the environment's initial state vector $\mathbf{s}_0$. We developed a deterministic initialization wrapper that overrides the default random positioning of the `FootballMultiEnv`. Specifically, for a chosen event (e.g., a pass-shot assist), the starting coordinates of the passer (A_1) and the intended recipient (A_2) are extracted and injected into the environment.

This allows us to evaluate the *emergent* behavior of the agents: given the exact same starting positions as professional players, will the RL agents choose the same passing lane, or will the PPO policy discover a more optimal trajectory that was not exploited in the real match?

4.5.2 Counterfactual Evaluation and Rollouts. To ensure statistical significance, each historical scenario is subjected to N evaluation rollouts (typically $N = 20$). During these rollouts, stochasticity is disabled (`explore=False`) to assess the deterministic preference of the learned policy.

We track several key performance indicators (KPIs) during these rollouts:

- **Agent-Specific Reward Distribution:** We calculate the mean and standard deviation of the rewards accumulated by each agent (Att_1, Att_2) across all runs. This metric serves as a proxy for policy stability and individual contribution to the team objective, allowing us to identify if specific agents are incurring high penalties for possession loss or sub-optimal positioning.
- **Aggregated Action Statistics:** To measure the tactical output of the "What-If" scenario, we aggregate the frequency of discrete successful and unsuccessful actions. These include:
  - **Passing Efficiency:** A comparison between total passes attempted and passes successfully completed.
  - **Offensive Threat:** The total volume of shots taken and the resulting conversion rate into goals scored.

- **Temporal Possession Metrics:** We track the "Possession Time" for each agent, measured in simulation steps. This distribution, often visualized via boxplots, reveals the ball-retention capabilities of the agents under the specific spatial constraints of the historical scenario.

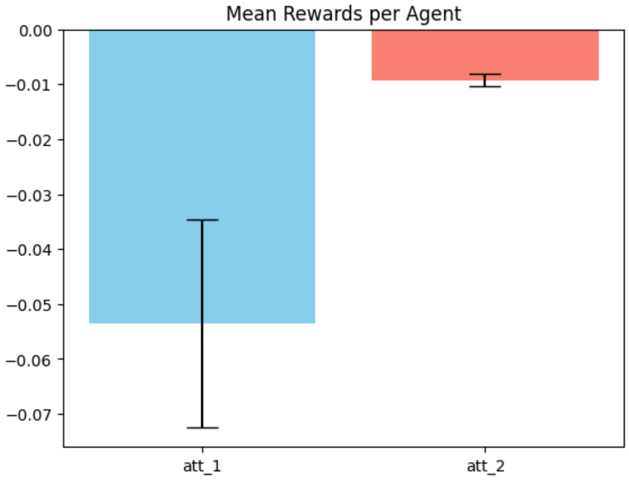

Figure 5: Average rewards per agent in what-if evaluation

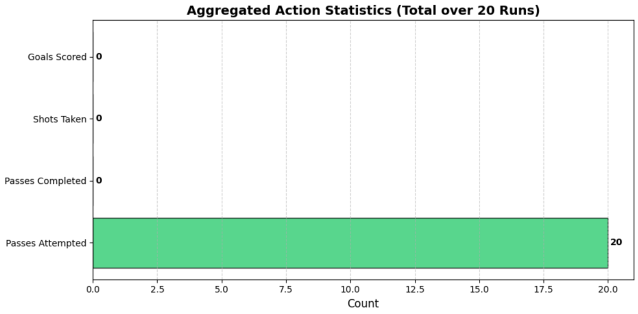

Figure 6: Aggregated statistics in what-if evaluation

By comparing these metrics against the historical "ground truth," we can quantify the effectiveness of the PPO policy in solving specific tactical problems and identify scenarios where the simulated agents outperform or underperform relative to professional standards.

5 Conclusion

This research established a modular "Tactical Digital Twin" framework by integrating real-world event data, adversarial training, and agent heterogeneity into a Multi-Agent Reinforcement Learning environment. While each of the three proposed extensions—What-If Analysis, Adversarial Learning, and Realistic Player Customization—was successfully implemented and integrated into the simulation pipeline, the empirical results reveal significant challenges in policy convergence.

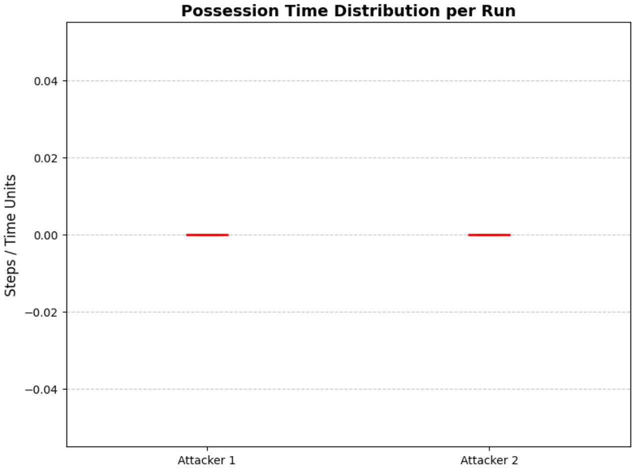

Figure 7: Possession time in what-if evaluation

In response to our research questions, the study yielded the following insights:

- **RQ1:** The model demonstrated difficulty in resolving historical *What-If* scenarios. As evidenced by the 0% pass completion rate in evaluation rollouts, the agents failed to translate historical coordinates into successful tactical execution, often failing to coordinate the timing of the pass with the receiver's run.
- **RQ2:** Adversarial training successfully introduced defensive pressure, but rather than fostering robust offensive play, it often exacerbated the "sparse reward" problem, leading attackers to converge toward sub-optimal, overly-conservative behaviors.
- **RQ3:** Player heterogeneity (speed and shooting precision) did alter the physical feasibility of certain actions, but the underlying PPO policy was not sufficiently advanced to exploit these unique attributes for tactical advantage.

Ultimately, the quantitative results—characterized by high negative rewards and a lack of goal-scoring emergence—indicate that the current model has not achieved a level of play suitable for professional tactical forecasting. The agents struggle with the high-dimensional coordination required for football, resulting in behaviors that, in their current state, are of limited practical utility for match analysis.

However, the "failure" of the agents to learn does not diminish the value of the architectural framework. The infrastructure developed here provides a rigorous, modular foundation for future research. By decoupling the environment logic from the agent attributes and historical data injection, we have built a sandbox that is ready for more sophisticated algorithms. Future work involving **Curriculum Learning**, **Imitation Learning** (to pre-train agents on StatsBomb trajectories), and **Transformer-based architectures** could leverage this existing pipeline to achieve the tactical intelligence that the current PPO implementation lacked. While the current "players" have yet to learn the game, the "stadium" is now fully built and ready for a more capable generation of digital athletes.

References

[1] Chaoyi Gu, Varuna De Silva, Corentin Artaud, and Rafael Pina. 2023. Embedding Contextual Information through Reward Shaping in Multi-Agent Learning: A Case Study from Google Football. In *2023 IEEE 13th International Conference on Pattern Recognition Systems (ICPRS)*. IEEE, Guayaquil, Ecuador, 1–8. doi:10.1109/ICPRS58416.2023.10179030

[2] Michael Pulis and Josef Bajada. 2022. Reinforcement learning for football player decision making analysis. In *StatsBomb Conference*. Hudl StatsBomb, London, United Kingdom, 1–23. https://www.um.edu.mt/library/oar/handle/123456789/131785

[3] Pegah Rahimian, Balazs Mark Mihalyi, and Laszlo Toka. 2024. In-game soccer outcome prediction with offline reinforcement learning. *Machine Learning* 113, 10 (2024), 7393–7419. doi:10.1007/s10994-024-06611-1

[4] Robert Rein and Daniel Memmert. 2016. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus* 5, 1 (2016), 1410. doi:10.1186/s40064-016-3108-2

[5] Yan Song, He Jiang, Haifeng Zhang, Zheng Tian, Weinan Zhang, and Jun Wang. 2024. Boosting Studies of Multi-Agent Reinforcement Learning on Google Research Football Environment: the Past, Present, and Future. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), Auckland, New Zealand, 1772–1781. https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p1772.pdf