

Optimising Spatial Teamwork Under Uncertainty

Gregory Everett¹, Ryan J. Beal², Tim Matthews², Timothy J. Norman¹, Sarvapali D. Ramchurn¹

¹School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom

²Sentient Sports, United Kingdom

gae1g17@soton.ac.uk, ryan.beal@sentientsports.com, tim.matthews@sentientsports.com,
T.J.Norman@soton.ac.uk, sdr1@soton.ac.uk

Abstract

We introduce a novel method for assessing agent teamwork based on their spatial coordination. Our approach models the influence of spatial proximity on team formation and sustained spatial dominance over adversaries using a Multi-agent Markov Decision Process. We develop an algorithm to derive efficient teamwork strategies by combining Monte Carlo Tree Search and linear programming. When applied to team defence in football (soccer) using real-world data, our approach reduces opponent threat by 21%, outperforming optimised individual behaviour by 6%. Additionally, our model enhances the predictive accuracy of future attack locations and provides deeper insights compared to existing teamwork models that do not explicitly consider the spatial dynamics of teamwork.

1 Introduction

Team Formation (TF) problems in multi-agent systems involve heterogeneous agents collaborating to achieve common goals such as task completion or risk minimisation. In TF problems, the environment is often dynamic and agents face uncertainties about the positions and intentions of other agents. In some settings, agents compete against adversaries, adding complexity to decision-making. Understanding these uncertainties and choosing the characteristics of optimal teams is crucial for developing efficient TF systems.

Recent research explores various criteria for team formation, such as task proximity (Capezzuto, Tarapore, and Ramchurn 2021), agent waiting times (Amador, Okamoto, and Zivan 2014), and the alignment of agent roles to tasks (Aswale and Pincioli 2023). For example, in social ridesharing, agents form teams to minimise travel times (Bistaffa et al. 2017), while in fire response, teams are formed within spatial and temporal constraints to minimise job completion time (Chen et al. 2021). These methods assess team value using predefined metrics such as task completion time or team suitability for nearby tasks (Parker et al. 2016), often overlooking specific agent-to-agent interactions. Beal et al. (2020) studies directed agent interactions in past teams to predict future team performance. However, the approach does not consider spatial constraints commonly found in TF problems (O’Leary and Cummings 2007).

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In many real-world teams, agents form spatial arrangements to optimise team dominance across an area. For example, Orcas encircle prey to minimise the chance of escape and officers spread to cover criminal hot spots in security settings. While previous TF models have explored agent proximity and spatial constraints, such as information limitations (Bulka, Gaston, and Desjardins 2007) or the impact of spatial dispersion on agent communication (O’Leary and Cummings 2007), they do not focus on optimising agent communication. Studies in UAV teams consider agent coordination to maximise task coverage in an area (Baker et al. 2016), however, no existing model studies optimising agent interactions to maximise spatial dominance against an adversary. We define this concept as spatial teamwork.

Against this background, we develop a novel spatial teamwork model focused on nearby agents, addressing agent uncertainties regarding the intentions of teammates and adversaries. We use a stochastic Multi-agent Markov Decision Process (MMDP) to model these dynamics. To navigate these uncertainties, Monte Carlo Tree Search (MCTS) is used to explore future scenarios and determine effective spatial teamwork in dynamic conditions. This approach allows agents to act individually or as subteams, where they coordinate actions with a shared objective function.

We validate our approach using Association football data, focusing on defensive strategies to minimise goal concession probability. Players are modelled as agents with their own actions, ability to coordinate with teammates and objectives. They operate with incomplete information regarding future ball and player positions. Football transitions, including player movements, are learned from real-world data.

This paper advances the state of the art as follows:

- We propose a novel model of spatial teamwork where a dynamic defence domain with incomplete information is modelled as an MMDP.
- An algorithm based on MCTS and linear programming is proposed to optimise individual and teamwork-based decision-making.
- We learn agent behaviour from real-world data and use it to simulate future scenarios in a dynamic environment.
- We compare our algorithm to real-world outcomes and show that our model reduces opponent threat by ~21%, outperforming optimised individual behavior by ~6%.

By so doing, we set a benchmark for optimising spatial teamwork in dynamic real-world domains with adversaries.

The rest of the paper is structured as follows. In Section 2, we discuss related work. Section 3 introduces the teamwork model and Section 4 discusses MCTS to optimise agent contributions. In Section 5, we apply our model to football and evaluate our approach in Section 6. We critically review our findings in Section 7 and then conclude.

2 Related Work

Before we present our spatial teamwork model we review existing research related to TF and team optimisation.

Team Formation in Spatiotemporal Domains Several studies address agent coordination with spatiotemporal constraints and considerations, such as agent proximity, travel time and task deadlines (Amador, Okamoto, and Zivan 2014). Ramchurn et al. (2010) optimise task completion in disaster response by considering agent travel distances and deadlines using mixed integer programming. Capezzuto, Tarapore, and Ramchurn (2021) extend this approach with a distributed algorithm for larger settings, tested on a London fire brigade dataset. Similarly, Scerri et al. (2005) develop a task allocation algorithm for extreme teams with interdependent tasks and time windows, such as UAV teams or hospitals. These approaches focus on task allocation but do not address spatial interactions between agents.

Bistaffa et al. (2017) model a social network for coalition formation to minimise agent travel times and fuel costs in social ridesharing. However, agent teams are limited to graph connections. Amador, Okamoto, and Zivan (2014) balance officer workload and incident importance in law enforcement. These studies assign whole teams to tasks and do not consider coordination between individual agents. Conversely, Mutzari, Gan, and Kraus (2021) apply cooperative game theory in multi-defender Stackelberg security games where agents disperse to independent targets. In contrast, our paper emphasizes spatial coordination to maximise area dominance instead of independent tasks. Additionally, unlike Mutzari, Gan, and Kraus (2021), where the attacker has complete knowledge, our problem has behavioural uncertainties for both agents and adversaries. Our model focuses on these uncertainties instead of environmental uncertainties addressed in previous literature (Silver and Veness 2010).

Previous TF research has addressed spatiotemporal constraints using metrics such as agent arrival time to assess team utility (Ramchurn et al. 2010). Other studies have explored how spatial proximity affects teammate communication (O’Leary and Cummings 2007). However, there is limited research on spatial interactions between teammates and their impact on team outcomes. While studies on UAV teams have coordinated agents to maximise task coverage (Baker et al. 2016), no study, to our knowledge, has optimised agent coordination for maximising spatial dominance against adversaries. Our study targets optimising spatial arrangements of agents to defend entire spatial planes, differing from previous studies that focus on independent targets or tasks.

Team-Based Optimisation Beal et al. (2020) present the first approach to explore agent teamwork through directed

interactions, using real-world football passes to identify successful player pairs. Our approach extends this by optimising spatial interactions and decision-making among multiple agents. While Beal et al. (2020) successfully derive value from directed agent interactions, they do not consider spatial proximity and constraints, which are important in many real-world teams (O’Leary and Cummings 2007).

MCTS is valued in team formation research for efficiently sampling large state spaces. Wu and Ramchurn (2020) use MCTS to generate search trees to optimise coalitions based on team utility but do not consider spatiotemporal constraints. Baker et al. (2016) use MCTS to optimise agent actions, leveraging information sharing between trees with the max-sum algorithm. In contrast, our study separately optimises individual and team actions with MCTS, using linear programming to determine effective teams.

Previous work in team sports has focused on optimising team formation, such as Matthews, Ramchurn, and Chalkiadakis (2012), who optimise fantasy football teams, and Beal et al. (2020), who select optimised football lineups. In contrast, our spatial teamwork model optimises agent interactions to improve team collaboration. While some studies have optimised player behaviour (Rahimian, Oroojlooy, and Toka 2021; Wang et al. 2024), they do not address spatial interactions and communication between players. Previous work has also explored deep multi-agent reinforcement learning for team information sharing and task decomposition in simulated football environments (Li et al. 2021; Yang et al. 2022). In contrast, our approach focuses on spatial dominance in real-world adversarial settings.

3 Spatial Teamwork Model

In this section, we model an attack-defense scenario inspired by real-world systems, including team sports and security settings, such as port security (Shieh et al. 2012) and nature security (Xu et al. 2017). We define the environment, agents, objectives, and agent subteams.

3.1 Basic Definitions

We define a defence scenario G as a sequence of timesteps $G = \{t_1, \dots, t_L\}$, where L is the number of timesteps and may vary for different scenarios. There are variable time intervals between timesteps. These scenarios may represent waves of attacks in security settings or possession phases in sports. Each scenario results in an outcome Ω , such as the loss of possession in sports or a failed attack in security.

Agents operate in a two-dimensional Euclidean space Λ , consisting of locations $\lambda = (x, y)$, representing coordinates on the plane. In contrast to typical defence scenarios with independent targets, this scenario requires defence of the entire plane, where each location λ has a value depending on strategic and temporal factors. The value of a location λ at timestep t is denoted as $V_{\lambda, t} \in \mathbb{R}$.

Each scenario has a set of agents $k \in K$. The agents are partitioned into two adversarial teams, a defending team K^D , and an attacking team K^O . These designations are consistent throughout the scenario timesteps. At each timestep t , each agent k has a set of possible actions $a_t^k \in \mathcal{A}_t^k$. These

actions refer to the movement of an agent from their current location λ_t^k to a new location λ_{t+1}^k . We also define agent-specific characteristics, such as role and physical condition, that may impact agent behaviour as ζ_t^k . The collective locations of a set of agents, defined as their spatial structure, are formed by both teams at each timestep t based on the actions chosen by each agent at the previous timestep. These spatial structures can be denoted as C_t^D and C_t^O where C_t^D contains all defending agent locations $\lambda_t^d \forall d \in D$. We build on these definitions to explain team objectives in the next subsection.

3.2 Team Objective

Attackers and defenders aim to maximise control over the space Λ . At timestep t , the control that defenders have over a location λ is represented as a probability, $\Pr(\Upsilon_{\lambda,t}|C_t^D, C_t^O)$, where $\Upsilon_{\lambda,t}$ indicates that the defenders K^D control λ at time t , which depends on the spatial structures C_t^D and C_t^O . Defensive utility is defined as a penalty corresponding to the attacking team's dominance over Λ , weighted by the value of the locations $\lambda \in \Lambda$. Defending agents aim to minimise this penalty. We represent team K^D 's utility at time t as:

$$U_t(K^D) = - \int_{\lambda} V_{\lambda,t} \cdot (1 - \Pr(\Upsilon_{\lambda,t}|C_t^D, C_t^O)) d\lambda \quad (1)$$

The optimal spatial structure for defenders K^D at timestep t is defined as $C_t^{D*} = \arg \max_{C_t^D} U_t(K^D)$. The defenders aim to minimise attacking control (i.e., maximise their utility) throughout an entire scenario, defined as:

$$U_G(K^D) = \sum_t^T U_t(K^D) \quad (2)$$

Defenders face multiple challenges when optimising $U_G(K^D)$. Firstly, there are spatiotemporal movement constraints requiring agents to move feasible distances between locations within timestep intervals (see Section 5). Additionally, agents do not know the actions that will be chosen by their adversaries and teammates. Agents must therefore plan their movement strategically to maximise future utility.

While attackers share similar goals and complexities, we focus on defenders for this paper.

3.3 Agent Contribution and Subteam Formation

At each timestep t , agents can form subteams with nearby teammates to optimise spatial coverage. A subteam is defined as $\Psi \subseteq K^D$, with Ψ as the set of subteams. Each agent can only belong to one subteam at t , determined by proximity-based K-means clustering, with the number of clusters chosen by maximising the Silhouette coefficient (Rousseeuw 1987). Subteam members share an objective function and action at t , providing awareness of their subteam members' actions, similar to human or intelligent systems communication. We define the objective function as:

Individual Agents Agents choose actions to improve their contribution to team utility. Using cooperative game theory, we value an agent's impact by their marginal contribution. The marginal contribution of agent k^d to the team K^D at

timestep t is defined as $\Theta_t^d = U_t(K^D) - U_t(K^{D \setminus \{d\}})$ where $U_t(K^{D \setminus \{d\}})$ is the utility achieved by the team if the agent k^d was not in the team K^D at timestep t .

Agent Subteams Instead of maximising individual contributions, agents may spatially coordinate with teammates in a subteam Ψ to maximise their joint marginal contribution. The marginal contribution of a subteam Ψ at timestep t is $\Theta_t^\Psi = U_t(K^D) - U_t(K^{D \setminus \Psi})$. Agents must use context to decide whether to act individually or coordinate in a subteam. Subteams offer certainty about each member's actions and can improve defensive efficiency, such as by surrounding adversaries. However, individual actions may be more effective for defending nearby vulnerable locations.

3.4 Spatial Teamwork Multi-Agent MDP

Markov Decision Process models can define how a stochastic environment changes as a decision-maker interacts with it. An MMDP extends this to multiple decision-making agents, each with their own action sets. We formulate a scenario G as an MMDP defined as $\mathcal{M} = \langle S, \{\mathcal{A}^k\}_{k \in K}, \Gamma, \{R_k\}_{k \in K}, \gamma \rangle$ with a set of states S , a set of actions \mathcal{A}^k for each agent k , Γ is the transition function, R_k is a reward function for each agent k and γ is a discount factor. Each MMDP state $s \in S$ is a tuple $s = \langle C_t^K, \zeta_t^K \rangle$ where C_t^K and ζ_t^K are the agent locations and characteristics defined previously in Section 3.1.

Each agent's action space includes all possible spatial movements within a realistic distance between timesteps (see Section 5, this may differ depending on the application domain). The transition function $\Gamma = \Pr(s'|s, \{a^k\}_{k \in K})$ represents the probability distribution over possible next states s' given the current state s , which contains information on agent locations and characteristics, and each agent's selected action. The reward of action a for agent k chosen at state s depends on the actions of other agents and the resulting state s' , denoted as $R_k(s, a^k, s')$. This reward reflects the agent's immediate contribution to team utility at state s' , $\Theta_{t'}^k$. The discount factor determines the preference for immediate versus future rewards.

The MMDP reaches a terminal state when the scenario G ends, such as when possession is lost in sports or an attack fails in security. The probability of a scenario concluding is captured in the transition function Γ . In the next section, we discuss techniques for optimising agent and team rewards.

4 Optimising Team Decision-Making

In this section, we explain the process of maximising team reward by optimising agent actions and spatial teamwork.

4.1 Agent Objective

Agents aim to maximise their long-term contribution beyond just the current timestep. This is defined below.

Agents An agent k aims to maximise long-term reward over a scenario from state s using the Bellman equation:

$$\mathcal{V}(s) = \max_{a^k} \left(\sum_{s'} \Pr(s'|s, a^k) \cdot (R_k(s, a^k, s') + \gamma \mathcal{V}(s')) \right) \quad (3)$$

Where $\mathcal{V}(s)$ is the Q-value of the optimal action at state s . The agent must balance immediate rewards with potential future gains, despite uncertainty about future actions of teammates and opponents, requiring them to predict behaviour and future state transition probabilities.

Subteams Agents in a subteam choose a joint action to maximise their collective long-term contribution, $\mathcal{V}(s)$.

$$\mathcal{V}(s) = \max_{a^\Psi} \left(\sum_{s'} \Pr(s'|s, a^\Psi) \cdot (R_\Psi(s, a^\Psi, s') + \gamma \mathcal{V}(s')) \right) \quad (4)$$

Due to the many possible actions for each agent, there are $|\mathcal{A}|^{|K|}$ possible next states from a current MMDP state s . Despite the use of action discretisation in Section 5, there are still approximately 1×10^{26} possible new states from the current state. Given this complexity, computing exact solutions for Equations 3 and 4 is infeasible. To address this, we approximate agent behaviour (in Sections 5.2 and 5.3), and use MCTS to estimate Q-values of actions by simulating many possible scenario outcomes resulting from actions.

4.2 Monte-Carlo Tree Search

MCTS is an anytime search algorithm used to optimise actions. We apply MCTS at a timestep t for all agents $k \in K^D$ and all potential subteams $\Psi \in \Psi$ by iteratively selecting actions and simulating the rest of the scenario G . The MCTS algorithm runs as follows:

MCTS Algorithm

- 1) **Selection** - Select the most promising action from the root using UCB1 (Auer, Cesa-Bianchi, and Fischer 2002) until an unexplored or terminal node is reached. Each node, N_s , with state s , leads to a child node with state s' , determined by action a and transition function Γ .
- 2) **Expansion** - Expand a node N_s by randomly selecting an unexplored action.
- 3) **Simulation** - At leaf node N_s , approximate $\mathcal{V}(s)$ using cumulative reward in an MMDP simulation until a terminal state is reached.
- 4) **Backpropagation** - Backpropagate the value of the new child node N_s up the tree to the root.

Our transition function, for which we use a deep learning-based model to compute (in Section 5), can be inefficient for single simulations. To improve MCTS convergence speed, we parallelise MCTS expansion and simulation. Unlike traditional leaf parallelization (Cazenave and Jouandeau 2007), our approach uses a single thread and instead uses parallel tree nodes to batch process state transitions using Γ .

At node N_s , we expand a random action and perform M transitions (100 in this paper) to reach M leaf nodes $N_{s'} = \{N_{s^0}, \dots, N_{s^M}\}$. We batch process these nodes using the transition model Γ to produce new states s' . Each new leaf node is then simulated until reaching a terminal state, generating estimated values $\mathcal{V}(s') = \{\mathcal{V}(s^0), \dots, \mathcal{V}(s^M)\}$. In backpropagation, the Q-value of the parent node N_s is updated with the average cumulative reward across all expanded nodes $\mathcal{V}(s')$, adjusting for the node count M .

4.3 Optimising Decision Selection

Given that we have optimised actions for each defending agent $k \in K^D$ and subteam $\Psi \in \Psi$ at timestep t , the goal is to maximise the long-term utility of the entire team $U_G(K^D)$. As agents and subteams independently select actions without knowledge of the rest of the team's actions, it is a crucial step to evaluate whether each agent should act individually or within their subteam. To find the combination of individual and subteam actions that maximises $U_G(K^D)$, we treat this as a linear programming problem:

$$\begin{aligned} & \text{maximise} && U_G(K^D) \\ & \text{subject to} && \sum_{k \in K^D} k_z + \sum_{\Psi \in \Psi} |\Psi| = |K^D| \\ & && \sum_{\Psi \in \Psi} |\Psi| > 1 \end{aligned} \quad (5)$$

Where k_z is a decision variable, set to 1 if agent k acts individually and 0 if in a subteam. To solve this linear program, we extract all combinations of individuals and subteams and approximate $U_G(K^D)$ for each using one thousand MMDP scenario simulations, similar to MCTS simulation. The action set that maximises $U_G(K^D)$ is chosen as the team's joint action, $a_*^{K^D}$. We term this optimisation strategy *spatial teamwork*. The entire process, including MCTS for agents and subteams, to identify $a_*^{K^D}$ is shown in Figure 1 and called the teamwork optimiser. The teamwork optimiser allows for inter-agent interaction to maximise joint subteam contributions, extending beyond a simple individual strategy where each agent selects their own optimised action.

5 Applying Spatial Teamwork to Football

In this section, we apply our model to attack-defence scenarios in football, illustrating its relevance and providing context for agent actions and state transitions in the sport.

5.1 Multi-Agent Environment in Football

We define a period of possession in football as a defence scenario G , consisting of timesteps t representing events (e.g., a pass or shot), and ending with an outcome, Ω , which is either a goal or a possession loss. The football pitch, Λ , is discretised into a grid of $(P \times Q)$ zones denoted as Φ , where $\phi_{p,q}$ represents a specific zone. In this paper, the grid comprises (25×16) zones. The value of pitch zones, $V_{\phi,t}$, are computed using the popular Expected Threat model¹, indicating the probability of a goal in the next 5 events from zone ϕ . Additionally, we update team utility to accommodate zones:

$$U_t(K^D) = - \sum_{p \in P} \sum_{q \in Q} V_{\phi_{p,q},t} \cdot (1 - \Pr(\Upsilon_{\phi_{p,q},t} | C_t^D, C_t^O)) \quad (6)$$

Where $\Pr(\Upsilon_{\phi_{p,q},t} | C_t^D, C_t^O)$ is the probability of defenders K^D winning possession if the ball arrives at zone $\phi_{p,q}$. This is calculated using zone centroids and a physics-based

¹Expected Threat: <https://karun.in/blog/expected-threat.html>. Last accessed July 12, 2024.

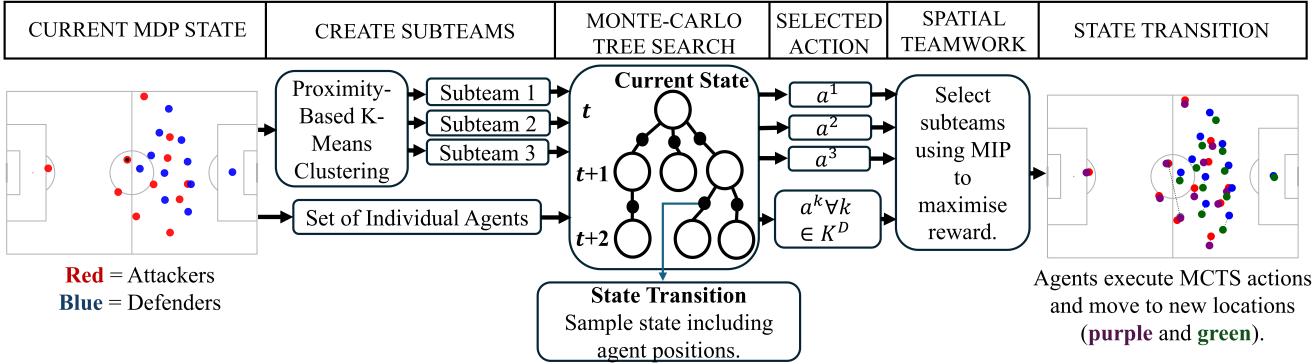


Figure 1: The agent optimisation process. Starting from the current MMDP state, K-means clustering determines subteams. MCTS approximates optimal actions for each subteam and agent, and MIP determines which subteams should form, resulting in an action for each agent. Executing these actions causes state transitions, as shown here with a football example.

ball control model (Spearman 2018). In football, K^D and K^O are the defending and attacking teams, each with 11 players. For each scenario G , the defending team K^D is determined by ball possession. The locations of players on team K^D at time t represent the team’s spatial structure C_t^D .

To accommodate the football domain, we define an MMDP state s as a tuple $s = \langle C_t^K, \zeta_t^k, \phi_t^B \rangle$ where ϕ_t^B is the current zone that the ball B occupies. Additionally, for this domain, the player characteristics, ζ_t^k , are a 3-tuple, $(\kappa_x^k, \kappa_y^k, J^k)$, containing the x and y velocity of the player and their role in the team (e.g., Center Defender) respectively. Player velocity at a new state s' is calculated as the average velocity required to move from their location in state s to their location in state s' within the time between timesteps. The immediate reward R_k represents the contribution of a player k towards control of valuable pitch space at time t . In this paper, we set the discount factor to 1.

5.2 Transition Function for Football

The transition function in an MMDP predicts state changes probabilistically. In our football framework, it models ball and player movements from state to state. We learn transition probabilities from real-world data. Our transition function, $\Gamma = \Pr(s'|s, \{a^k\}_{k \in K})$ is simplified using domain knowledge by making player positions at the next state conditional on ball movement. We define this as follows:

$$\Gamma(s', s) = \Pr(\Phi_{t+1}^B | s) \cdot \prod_{k \in K} \Pr(\lambda_{t+1}^k | s, \{a^k\}_{k \in K}, \Phi_{t+1}^B) \quad (7)$$

Where $\Pr(\Phi_{t+1}^B | s)$ is the probability of the ball moving to each zone, or possession being lost, given the current state s and $\Pr(\lambda_{t+1}^k | s, \{a^k\}_{k \in K}, \Phi_{t+1}^B)$ is the probability of player k ’s next location given the current state, actions and ball movement. We use a ball transition model (Spearman 2018) to compute $\Pr(\Phi_{t+1}^B | s)$ and sample from these probabilities during state transitions. For successful ball transitions, we extend a player location prediction model (Everett et al. 2023) and train it using data from 34 real football games to predict player locations in future states. Unlike Everett et al. (2023), which uses only on-ball data, our

model uses state information including player locations and velocities, ball locations, player roles, and time elapsed between states (calculated using a ball travel model (Spearman 2018)). This deep-learning model, combining convolutional and graph neural networks, achieves the lowest mean Euclidean error (2.31m) compared to the same baselines in (Everett et al. 2023) such as XGBoost (2.47m), graph neural network (2.56m), and a simple spline (5.03m). As player movement depends on ball movement, from a state s and actions $\{a^k\}_{k \in K}$, the number of possible new non-terminal states equals the total number of pitch zones. Details on player actions are given in the next subsection.

5.3 Player and Subteam Actions

We model player actions as directional biases that increase the likelihood of players moving towards certain areas. These biases adjust player velocity data in our transition model, increasing movement probability towards the desired direction while accounting for ball movement, teammates, and opponents. Adversaries receive no velocity alteration, imitating the average real-world behavior. We discretise actions to reduce the action space size, speeding up MCTS convergence. The action space is described as follows:

Player Actions Each individual player selects an action from one of eight possible directions, corresponding to points on a compass-like axis: Up, Top Right, Right, Bottom Right, Down, Bottom Left, Left, and Top Left. The selected action alters the player’s velocity by 2 m/s in the specified direction. We set a maximum speed of 5 m/s, which is extracted from Spearman (2018).

Subteam Actions Subteam actions ensure all players execute the same action, knowing that their sub-teammates will do the same, modelling player communication. For example, if the action is ‘Up’, all subteam members increase velocity in that direction. To enhance flexibility, we use domain knowledge to expand the action space beyond directional bias: Avoid ball, Towards opponent, Avoid opponent, Spread out and Close in. To focus subteam actions on coordination rather than flexibility, we align each agent’s velocity with the individual action closest to the chosen subteam action.

6 Evaluation

In this section, we evaluate our model against real-world actions and numerous baselines, listed as follows:

- **Random** - Agents choose their movements randomly.
- **Real-world** - Threat accumulated in the real-world data.
- **Simulated Real-world** - Agents follow their real-world actions within our MMDP simulations, bound to the closest action in the action set.
- **Individual optimiser** - All agents optimise their contribution individually without any subteams.
- **Pair-Based optimiser** - Agents choose actions in pairs based on the Team-Centred model in (Beal et al. 2020).
- **(Spatial) Teamwork optimiser** - Agents have the option to coordinate in subteams.

Inspired by (Beal et al. 2020), our Pair-Based strategy segments teams into agent pairs to maximise pass success frequency within pairs from past games. Unlike Beal et al. (2020), who focus on lineup selection and in-possession attacking link-ups, we modify this method to in-game defensive decision-making by splitting the 10 outfield defenders into 5 pairs (excluding the goalkeeper). This pair formation does not explicitly consider spatial context and proximity. Our MCTS model selects effective actions for these pairs.

We compute results for all events (e.g., passes and shots) in a 34-game real-world football dataset in the K League 1 supplied to us by BePro Group Ltd.

6.1 Experiment 1 - Model Performance in Dynamic Defence Simulations

We assess our selected actions against real-world decisions and numerous baselines for each event in our dataset by simulating one thousand instances of the next 5 events using our football MMDP model. Table 1 shows the accumulated threat (goal concession probability) for each strategy.

The teamwork optimiser reduces threat by 20.8%, which is 5.7% more than the individual optimiser, highlighting the benefits of spatial collaboration. Similar threats in real-world and simulated real-world scenarios support the MMDP model’s accuracy of the real-world. Optimising agent behaviour is critical in high-threat situations; therefore, we focus on these scenarios in Table 2. The results in these high-threat scenarios draw similar conclusions.

Strategy	Average Reward	Reduction
Random	-0.058 ± 0.001	-9.4%
Real-World	-0.053 ± 0.001	0.0%
Simulated Real-World	-0.051 ± 0.001	3.8%
<i>Individual Optimiser</i>	-0.045 ± 0.001	15.1%
<i>Pair-Based Optimiser</i>	-0.047 ± 0.001	11.3%
Teamwork Optimiser	-0.042 ± 0.001	20.8%

Table 1: Average reward comparison for simulated play sequences using various decision-making strategies.

Strategy	Average Reward	Reduction
Random	-0.084 ± 0.001	-18.3%
Real-World	-0.071 ± 0.001	0.0%
Simulated Real-World	-0.074 ± 0.001	-4.2%
<i>Individual Optimiser</i>	-0.065 ± 0.001	8.5%
<i>Pair-Based Optimiser</i>	-0.068 ± 0.001	4.2%
Teamwork Optimiser	-0.061 ± 0.001	14.1%

Table 2: Average reward comparison for simulated play sequences in high threat situations ($\geq 1\%$ goal probability).

We analysed how the difference between the average rewards (teamwork optimiser vs. real-world reward) correlates with the number of goals each team conceded in our dataset. Despite noise from low scoring rates across 34 games, a significant Pearson correlation coefficient of 0.71 ($p < 0.05$) confirms the expected relationship between these variables.

Additionally, for every event, we compared threat levels at the next real-world event with those from an augmented next event where defenders chose the optimised actions. This myopic approach assesses performance beyond the MMDP simulation. Results showed that teams reduce threat at just the real next event by $1.5\% \pm 0.6$ and $2.3\% \pm 1.2$ by using the individual and teamwork optimisers respectively.

6.2 Experiment 2 - Attack Location Prediction Using Subteam Collaboration

We have shown that our optimiser reduces overall team threat. Next, we evaluate the subteam performance metric (Θ_t^Ψ) against individual metrics (Θ_t^d) for predicting the zone of the next attack (i.e., passes or shots). The zone that each subteam centroid resides in, and the performance metric of that subteam, are used to form a grid of features matching the shape of Λ , and a simple convolutional neural network model utilises these spatial features to predict the next attack zone. Table 3 compares the predictive ability of subteam metrics to individual metrics over five-fold cross-validation.

Metric	Individual	Subteam
Passes	0.156 ± 0.003	0.147 ± 0.001
Shots	0.0100 ± 0.0004	0.0093 ± 0.0002

Table 3: Log-loss scores of contribution metrics as features for predicting future attack zones.

We find that subteam contributions better predict opponent attack locations than individual contributions. This emphasizes the value of analyzing agent coordination, as it uncovers the impact of spatial teamwork on the predictability of future attacks in dynamic settings such as team sports.

6.3 Experiment 3: Impact of Subteam Configurations on Threat Reduction

In this experiment, we assess how varying numbers of subteams impact threat reduction in real-world scenarios, aim-

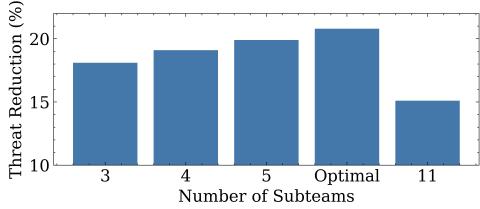


Figure 2: Effects of number of subteams on threat reduction.

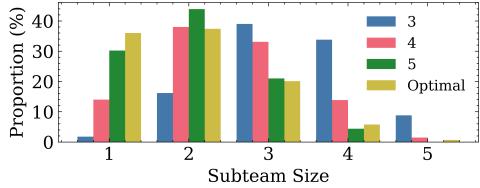


Figure 3: Distribution of subteam sizes for varying cluster numbers. Bar colours represent the number of subteams.

ing to identify the optimal number of players for effective spatial teamwork. Figure 2 compares threat reduction for various numbers of subteams using the teamwork optimiser. We drive deeper insight into the varying subteam numbers by looking at the distribution of subteam sizes in Figure 3.

These results show that clustering into five subteams or using the optimal k-means silhouette score is most effective for threat reduction. Figure 3 shows that these clusters often form pairs, suggesting that spatial teamwork is most effective in pairs, aligning with Beal (2022), which found that teamwork in football passing is best represented by pairs.

In many domains, agents have specific roles, such as player positions in football. Our model most commonly suggests subteam pairs as: (Center Defender, Center Defender), (Center Forward, Center Forward), (Right Defender, Right Midfield) and (Left Defender, Left Midfield). This aligns with human expert views on important role-based pairs.

6.4 Experiment 4: Case Study Analysis of Defensive Optimisations

In this experiment, we compare real-world defensive scenarios with model-optimised decisions to show their impact on game outcomes. Figure 4 highlights a specific defensive situation, demonstrating a 0.4% decrease in goal probability ($U_t(K^D)$) due to the model’s recommendations. This case study shows how suggested adjustments in positioning and teamwork can significantly reduce goal likelihood and improve team preparation in a visually compelling way.

7 Discussion

We validate our spatial teamwork model with real-world football datasets, showing its effectiveness in improving spatial dominance in dynamic settings. Football is a suitable domain due to its rich spatiotemporal data and the clear objective of preventing goals and winning matches. Our model also has potential in other spatiotemporal defence domains, including patrol, security, and other team sports. The model

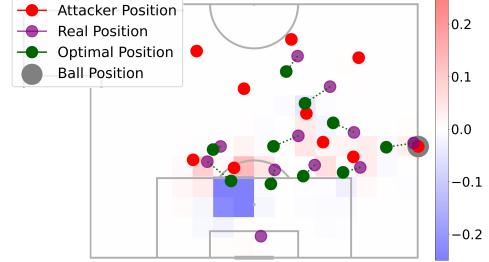


Figure 4: Optimised positioning to reduce threat. There is a 3.0% goal probability compared to 3.4% with real-world positions. The heatmap shows the change in goal probability (%) for each zone when shifting to optimised positioning.

may also be applicable beyond defence, such as for attack or emergency response. Future research will extend the model to other domains as we obtain more high-quality datasets.

The Pair-Based Optimiser improves on real-world performance but is less effective at reducing threats compared to other optimisers. This suggests that while optimising actions in pairs is valuable (as shown in Experiment 3), accounting for spatial context and proximity is crucial when optimising for specific scenarios to avoid suboptimal communication where individual actions may be preferable. Our findings in Experiment 3 may further explain the effectiveness of the teamwork model in (Beal et al. 2020) due to the implicit link between spatial proximity and passing in football, and we show that integrating spatial context into pair formation is effective for in-game player decision-making.

Our approach optimises agent behaviour by simulating outcomes and minimising average threat. Future work could explore strategies such as a minimax approach to minimise the threat of the worst-case scenario. We plan to evaluate various approaches and the impact of game state, such as maintaining a winning position, on optimised strategies.

The model’s primary use is post-scenario analysis to identify poor teamwork patterns and suggest improvements. It can build databases of similar scenarios to assess typical team responses. In football, this may inform team training.

8 Conclusion

In this paper, we propose a novel approach to agent co-ordination in dynamic environments, setting a benchmark for optimising spatial teamwork in adversarial settings. We model defence against adversaries as an MMDP and use MCTS to compute effective decisions. We apply this model to football defence, learning play sequences from real-world spatiotemporal data. Our MCTS approach reduces opponent threat by up to 21% compared to real-world outcomes, with an additional 6% reduction achieved using our spatial teamwork model over individual actions. We also show that our model better predicts future attack locations compared to an individual-based benchmark. Finally, we explain how our model results may offer deeper insight into previous teamwork models that do not explicitly consider the relationship between spatial proximity and teamwork.

Acknowledgments

We thank Bepro Group Ltd for supporting and providing the data resources for this work. The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work. Gregory Everett was supported by Sentient Sports and Sarvapali Ramchurn was supported by the UKRI Trustworthy Autonomous Systems Hub (EP/V00784X/1) and Responsible AI UK (EP/Y009800/1).

References

- Amador, S.; Okamoto, S.; and Zivan, R. 2014. Dynamic multi-agent task allocation with spatial and temporal constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.
- Aswale, A.; and Pincioli, C. 2023. Heterogeneous Coalition Formation and Scheduling with Multi-Skilled Robots. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5402–5409. IEEE.
- Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47: 235–256.
- Baker, C. A.; Ramchurn, S.; Teacy, W.; and Jennings, N. R. 2016. Planning search and rescue missions for UAV teams. In *ECAI 2016*, 1777–1782. IOS Press.
- Beal, R.; Changder, N.; Norman, T.; and Ramchurn, S. 2020. Learning the value of teamwork to form efficient teams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7063–7070.
- Beal, R. J. 2022. *Artificial intelligence in team sports*. Ph.D. thesis, University of Southampton.
- Bistaffa, F.; Farinelli, A.; Chalkiadakis, G.; and Ramchurn, S. D. 2017. A cooperative game-theoretic approach to the social ridesharing problem. *Artificial Intelligence*, 246: 86–117.
- Bulka, B.; Gaston, M.; and Desjardins, M. 2007. Local strategy learning in networked multi-agent team formation. *Autonomous Agents and Multi-Agent Systems*, 15: 29–45.
- Capezzuto, L.; Tarapore, D.; and Ramchurn, S. D. 2021. Large-scale, dynamic and distributed coalition formation with spatial and temporal constraints. In *Multi-Agent Systems: 18th European Conference, EUMAS 2021, Virtual Event, June 28–29, 2021, Revised Selected Papers 18*, 108–125. Springer.
- Cazenave, T.; and Jouandeau, N. 2007. On the parallelization of UCT. In *Computer games workshop*.
- Chen, J.; Guo, Y.; Qiu, Z.; Xin, B.; Jia, Q.-S.; and Gui, W. 2021. Multiagent dynamic task assignment based on forest fire point model. *IEEE Transactions on Automation Science and Engineering*, 19(2): 833–849.
- Everett, G.; Beal, R. J.; Matthews, T.; Early, J.; Norman, T. J.; and Ramchurn, S. D. 2023. Inferring Player Location in Sports Matches: Multi-Agent Spatial Imputation from Limited Observations. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 1643–1651.
- Li, C.; Wang, T.; Wu, C.; Zhao, Q.; Yang, J.; and Zhang, C. 2021. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3991–4002.
- Matthews, T.; Ramchurn, S.; and Chalkiadakis, G. 2012. Competing with humans at fantasy football: Team formation in large partially-observable domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, 1394–1400.
- Mutzari, D.; Gan, J.; and Kraus, S. 2021. Coalition formation in multi-defender security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 5603–5610.
- O’Leary, M. B.; and Cummings, J. N. 2007. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *MIS quarterly*, 433–452.
- Parker, J.; Nunes, E.; Godoy, J.; and Gini, M. 2016. Exploiting spatial locality and heterogeneity of agents for search and rescue teamwork. *Journal of Field Robotics*, 33(7): 877–900.
- Rahimian, P.; Oroojlooy, A.; and Toka, L. 2021. Towards optimized actions in critical situations of soccer games with deep reinforcement learning. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–12.
- Ramchurn, S. D.; Polukarov, M.; Farinelli, A.; Truong, C.; and Jennings, N. R. 2010. Coalition formation with spatial and temporal constraints. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 3-Volume 3*, 1181–1188.
- Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20: 53–65.
- Scerri, P.; Farinelli, A.; Okamoto, S.; and Tambe, M. 2005. Allocating tasks in extreme teams. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 727–734.
- Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. Protect: A deployed game theoretic system to protect the ports of the united states. In *Proceedings of the 11th international conference on autonomous agents and multiagent systems-volume 1*, 13–20.
- Silver, D.; and Veness, J. 2010. Monte-Carlo planning in large POMDPs. *Advances in neural information processing systems*, 23.
- Spearman, W. 2018. Beyond expected goals. In *Proceedings of the 12th MIT Sloan sports analytics conference*, 1–17.
- Wang, Z.; Veličković, P.; Hennes, D.; Tomašev, N.; Prince, L.; Kaisers, M.; Bachrach, Y.; Elie, R.; Wenliang, L. K.; Piccinini, F.; et al. 2024. TacticAI: an AI assistant for football tactics. *Nature communications*, 15(1): 1906.
- Wu, F.; and Ramchurn, S. D. 2020. Monte-Carlo tree search for scalable coalition formation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 407–413.

Xu, H.; Ford, B.; Fang, F.; Dilkinia, B.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Nsubaga, M.; et al. 2017. Optimal patrol planning for green security games with black-box attackers. In *Decision and Game Theory for Security: 8th International Conference, GameSec 2017, Vienna, Austria, October 23-25, 2017, Proceedings*, 458–477. Springer.

Yang, M.; Zhao, J.; Hu, X.; Zhou, W.; Zhu, J.; and Li, H. 2022. Ldsa: Learning dynamic subtask assignment in co-operative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 1698–1710.