Deep Reinforcement Learning for Football Player Decision Analysis

Michael Pulis

Supervisor: Dr Josef Bajada

October, 2023

Submitted in partial fulfilment of the requirements for the degree of Masters in Artificial Intelligence (Hons).

L-Università ta' Malta
Faculty of Information &
Communication Technology

To my parents,

Without whom, work on this thesis would not have been possible.

Abstract

Analysis of a football player's decision-making process often relies heavily on easily interpretable statistics such as the goals scored, and the assists provided by the player. While these statistics are useful, relying solely on them leads to more nuanced high-level performances being overlooked. This is because results-based analysis does not account for the ever-present role of luck that distorts the outcome. A team can consistently generate higher-quality goal scoring opportunities than their opponents throughout a match, but still end up losing due to unfortunate finishing, or an outstanding goalkeeping display by the opponents. Recent advances in statistical analysis of football events have yielded more objective metrics such as Expected Goals (xG) and Expected Threat (xT) to address this problem. These metrics have been used to develop Possession Value Models (PVMs) that can be used to evaluate the decision making within players. However, these models do not take into account the context within which actions were made, since they rely solely on event data about the actions itself.

To evaluate player decisions objectively in context, we propose a novel model which we call Decision Value (DV), generated through offline Deep Reinforcement Learning. This model was trained on a dataset of past matches, consisting of the actions performed by elite-level football players. The dataset consists of both event data and also tracking data, which provides the coordinates of the teammates and opposition players. This data was preprocessed and augmented further into a new dataset, which incorporates the details of the actions and the coordinates of the teammates and opposition players, together with the reward obtained as a result of the action. Having such a richer dataset allowed the model to learn to evaluate decisions within the context that they were made. The IQL algorithm was used to perform offline reinforcement learning.

Acknowledgements

This work was partially funded by the 2021 MDIA scholarship program.

The dataset was provided as a part of the StatsBomb 2022 Research Challenge.

Contents

List of Figures

List of Tables

1 Introduction

Football is a low scoring game, where influential moments are often few and far between when considering the entire 90 minute duration. A player's performance is often judged through cursory glances of easily interpretable statistics, such as goal contributions made by the attacking players, or tackles and interceptions made by the defensive players. Another factor that makes football decision analysis difficult is the role of luck. Good decision making is not determined by the actual outcome of the action, as the tight margins present during an actual game can result in a good pass by a midfielder being squandered by poor finishing from the attacker. This issue has been identified and at least partially addressed in other sports, such as ice-hockey (Macdonald, 2012), lacrosse (Myers et al., 2021), and baseball (Baumer et al., 2015). In football, the issue has started to be tackled by objective measure such as Expected Goals (xG), and the introduction of Possession Value Models (PVMs) such as Expected Threat (xT) (Singh, 2018) and Valuing Actions by Estimating Probabilities (VAEP) (Decroos et al., 2020). The xG metric allows for each shot to be analysed objectively, such that it is quantified by the probability that an average elite level football player would have scored from an identical scenario. This type of analysis allows for the effects of randomness to be minimised, and allows players to be analysed objectively, as we can identify players that are over-performing their xG. This would indicate that a player is either experiencing a lucky streak, or they are extremely efficient with their chances. Conversely, if a striker is consistently under-performing their xG would indicate that a striker is enduring some back luck, indicating that they are actually performing better than traditional metrics would suggest, or they have worse than average finishing abilities.

PVMs can then be trained to value all actions on the pitch in an objective manner. This is done by identifying the likelihood increase/decrease that a particular action has on the team's chance of scoring or conceding a goal after the current action is made. The primary issue with current metrics is their inability to consider the location of the surrounding players when valuing player actions and decisions.

1.1 Motivation

The use of evaluation of player decision making in football has been a largely subjective affair. The emergence of recent metrics such as xG, xT and VAEP have allowed for more objective evaluation of player decisions to take place. Pairing this fact with the recent advances in artificial intelligence and reinforcement learning, and the volume of data that is now being obtained from each football game, opens an opportunity to lay the groundwork for a framework that can value football player decisions. By utilising data that can simply be obtained from broadcast footage instead of on-player tracking devices or systems that involve multiple tracking cameras to be installed within the stadium, this research aims to offer a viable option to value player decisions within the context that they are made, whilst also lowering the barrier to entry for clubs that require these advanced statistics where the aforementioned sophisticated and expensive technology is not available. This could allow for teams with lower spending budgets to take advantage of the system especially when using the developed model to prepare a shortlist of players that a team's limited scouting department should focus on. Equally, this approach will allow established teams to search for the best decision makers in relatively obscure leagues within which they do not typically search for players, due to the logistical limitation of sending scouts to each possible league.

1.2 Aims and Objectives

The main goal of this project is to research and develop a system that can take the context within which a decision is made within a football game into account to obtain an objective model that can be used for analysis and improvement of football players' decision making process. To address the aim of this research project, the following objectives will be set:

Objective 1: Generate an augmented dataset suited for Deep Reinforcement Learning using existing football tracking and event analysis.

One of the challenges of applying deep reinforcement learning (DRL) to football tracking and event analysis is the scarcity of suitable datasets that capture the complex dynamics of the game. Existing datasets are either too small or do not contain adequate data to be used for DRL algorithms. Therefore, this objective aims to utilise existing datasets to generate an augmented dataset that is optimised for performing DRL within the context of football tracking and event datasets.

Objective 2: Research and implement a Deep Reinforcement Learning model to learn to value football player decisions.

By using the aforementioned dataset created in Objective 1, a Deep Reinforcement Learning model will be chosen to learn to value player decisions within the context that they are made. This will be carried out after analysis of the state of the art algorithms that are available, whilst also considering how suited each algorithm is to the particular scenario required within this work. This will also involve hyper-parameter tuning which will help to determine which algorithm will be used to train the final model.

Objective 3: Utilise this RL model to develop a football player decision making evaluation metric that can also be used to evaluate team performance.

The Deep Reinforcement Learning model trained within Objective 2 will then be used on a section of the augmented dataset developed within Objective 1 to evaluate the top decision making players. This will be carried out through various different qualitative approaches that will measure decision making quality within various different scenarios. The evaluation will also be coupled with quantitative evaluation to compare the model's output with traditional indicators of success. The output will also be compared with existing possession valuation frameworks (such as xT).

1.3 Proposed Solution

To achieve the aims and objectives outlined within the previous section, the following solution is proposed.

- Identify datasets that contain paired event and tracking data, and load and preprocess the dataset. This will allow the filtering of unnecessary actions and reduce the effect of noise within the dataset.
- Convert the event data and tracking coordinates into a flexible format that will allow for representation of a non-fixed number of players per scenario, such as an image representation of the scenario.
- Use the `d3rlpy` library[1] containing implementations of the state-of-the-art algorithms to find the ideal algorithm through hyper-parameter searching with `Optuna` by ensuring that the critic and actor networks converge successfully, as well as the td-error.

[1] `https://d3rlpy.readthedocs.io/en/latest/`

- After training the model on data obtained from one season of football, predictions will be made on decisions made within a different season.
- The model will be used to obtain the expected return associated with taking different actions within the same scenario. The top performing players within each different position will be obtained, as well as analysing the developed model's alignment with club transfer strategies.
- The model's ability to order teams will be compared with the actual league table ordered by points. The strength of the correlation will be compared with correlations obtained using different metrics such as xG and other PVMs.

The system will require strong GPUs with a large amount of VRAM. Thus, the training will need to take place on cloud-based machines that will be rented throughout the duration of this research.

1.4 Contributions

Through this research project, we will be making the following main contributions to literature regarding the use of Deep Reinforcement Learning within football player decision making analysis.

- Introducing a novel methodology to prepare event and tracking data for use within a DRL context.
- Provide a reward function that can integrate the context within which an action is made, whilst also accounting for the expected effect that the action will have on the value of possession.
- Show that the proposed methodology can be used to effectively value player decisions within their context.

The work carried out within this research was published within the StatsBomb Conference in September of 2022, which took place at Wembley Stadium, the national stadium of the English football team, and was presented in front of several team scouts and heads of data science from the elite teams within the top 5 football leagues in Europe.

1.5 Chapter Overview

This section contains a brief summary of the contents of each of the chapters within this project.

1. Background and Literature Review In this chapter, the history of objective football analysis is discussed. The chapter also includes the history of both traditional and deep reinforcement learning, and finally the latest research published at the intersection of the two topics.

2. Methodology In this chapter, the exact details of the implementation chosen for this project are shown. This includes the augmentations carried out on the StatsBomb research dataset, provided after this project was accepted into the 2022 StatsBomb conference, as well as how the DRL algorithms discussed in the previous chapter are used to address the aims of this research. The plan for evaluation of the system is also outlined within this chapter.

3. Results and Evaluation In this chapter, the results obtained from the methodology outlined in the previous chapter are shown. This includes discussion of the necessity of the augmentations made to obtain a dataset compatible with the scope of this research. It also includes the results of the training process, as well as the possible ways in which the developed model can be used to evaluate individual players, as well as football teams.

4. Conclusion In this chapter, the entirety of the research, methodology and results are re-visited and summarized. The key takeaways from the research are extracted and compared with the aims and objectives set out at the beginning of the research to confirm if these have been met sufficiently. The chapter also contains a thorough discussion of the limitations of the work, as well as suggestions for future work.

2 Background

This chapter introduces the key terms and concepts associated with football analysis and reinforcement learning. First, we will introduce the rules of football together with the data that is used for analysis and statistical modelling. This is followed by tools and techniques that help to determine ball possession and position strength. We then introduce the field of RL, focusing on offline RL.

2.1 What is Football?

In this work we address the sport of Association Football, more commonly referred to as *football* within most of the world, or *soccer* in Northern America. Football is a game that is played between two teams. Each team has 10 outfield players and 1 goalkeeper, and games are played over two halves that are each 45 minutes long. Games are won by the team that scores the most goals throughout the entire game. In most cases, the game is said to be a tie or a draw if neither team manages to outscore the other after both halves have been played, with the exception of knock-out format tournaments, which are not being considered within this work. Teams are assigned 3 points for winning a game, 1 point for achieving a draw and 0 points for losing a game. The player that shoots the ball into the opposition goal is said to be the goal-scorer, and their teammate that provides the pass immediately preceding the action leading to the goal is said to have provided the assist. Teams are typically made up of three main types of outfield players, being defenders, midfielders and attackers, illustrated in Figure 2.1.

2.2 Types of Data used in Football Analysis

Data from football games normally takes one of two forms, namely event and tracking data. Event data contains play-by-play tracking of events. Each data point would include the action taken (pass, dribble, clearance, etc.) as well as the players involved in the action

Position Number	Position Abbreviation	Position Name
1	GK	Goalkeeper
2	RB	Right Back
3	RCB	Right Center Back
4	CB	Center Back
5	LCB	Left Center Back
6	LB	Left Back
7	RWB	Right Wing Back
8	LWB	Left Wing Back
9	RDM	Right Defensive Midfield
10	CDM	Center Defensive Midfield
11	LDM	Left Defensive Midfield
12	RM	Right Midfield
13	RCM	Right Center Midfield
14	CM	Center Midfield
15	LCM	Left Center Midfield
16	LM	Left Midfield
17	RW	Right Wing
18	RAM	Right Attacking Midfield
19	CAM	Center Attacking Midfield
20	LAM	Left Attacking Midfield
21	LW	Left Wing
22	RCF	Right Center Forward
23	ST	Striker
24	LCF	Left Center Forward
25	SS	Secondary Striker

Figure 2.1: Player Positions, Adapted from (StatsBomb, 2020)

itself. Other details are also included, such as the body part used to take the action, the originating and destination positions in applicable cases, if any defensive pressure was being applied on the attacking team, and other information.

Tracking data simply includes the coordinates of the players on the pitch. There are two main sources for football tracking data. The first, and most reliable source, is data taken directly from trackers worn by the players themselves during the match. This ensures that there are always 22 correct player coordinates. The other source for tracking data is camera footage. The positions of the players can be extracted using tracking algorithms that use a combination of computer vision and deep learning (Linke et al., 2020; Naik et al., 2022). The extracted coordinates are mostly reliable, however they do not guarantee that all players are present at the same time. This is due to the fact that most of the time, some players are out of the view of the broadcast camera. The two data types are illustrated in Figures 2.2 and 2.3.

2.3 Traditional Visuals used for Football Analysis

The aforementioned data types associated with football analysis are often used to provide helpful visuals to coaches and performance analysts. A few key examples are shown within the following subsections.

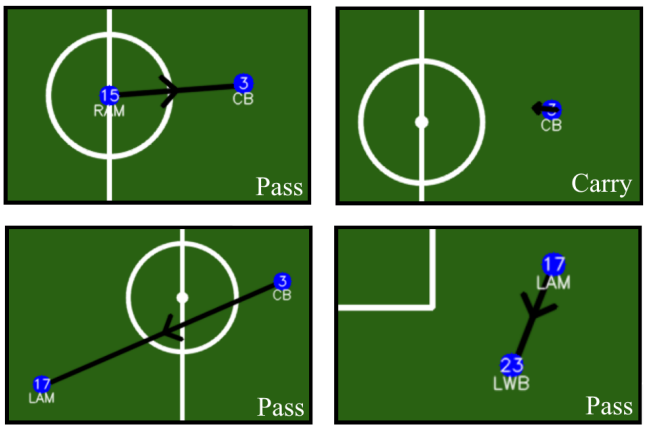


Figure 2.2: Event Data

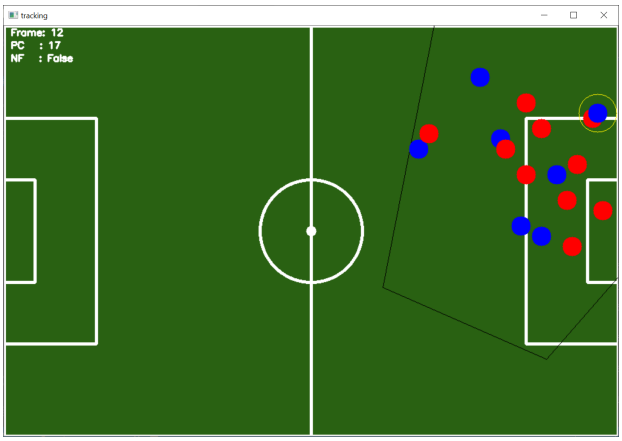


Figure 2.3: Tracking Data

2.3.1 Pass Networks

The *Pass Network* diagram is generated using event data (Caicedo-Parada et al., 2020). The average coordinates of each player when they touch the ball are calculated, which are then used to plot their location on the diagram. The intensity of the connection between the players is found by counting the number of times a pass is carried out between the two players. An example of this type of diagram is shown in Figure 2.4.

This type of diagram can be used to analyse the team's structure throughout the match, as well as to identify which patterns of play are emerging. For example, the pass network shown in Figure 2.4 shows that the goalkeeper tends to pass the ball to the left sided center-back, however play tends to shift towards the right, as Antonio Valencia and Juan Mata tend to receive the ball more often during the match.

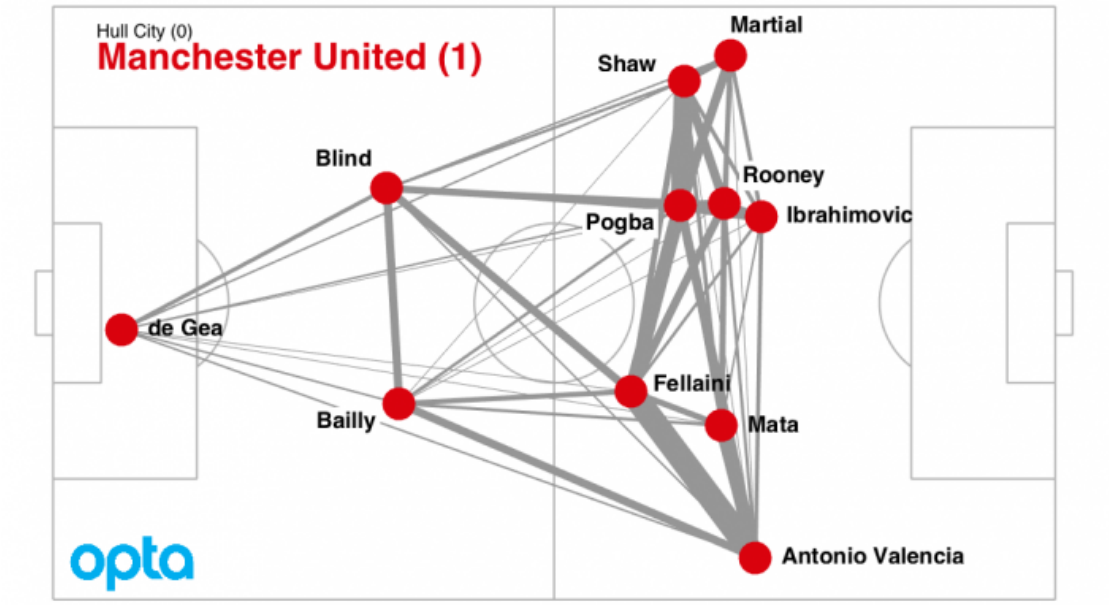


Figure 2.4: Pass Networks, Adapted from (Sumpter, 2017)

2.3.2 Voronoi Diagrams

This diagram is an example of a visualisation that is created using tracking data (Efthimiou, 2021; Kim, 2004). The purpose of Voronoi diagrams is to split the surface into a number of polygons, such that each polygon only contains one point. Each polygon is drawn such that it represents the area of the diagram that is closer to its point than to any other point. This is adapted to football whereby the points are the coordinates of the players, and a class is assigned to each player based on the team that the player belongs to. The polygons are then coloured by their class, with the final result representing a diagram that shows which areas of the pitch belong to which team. An example of this is shown in Figure 2.5[1].

2.4 Expected Goals (xG)

Using *shots taken* or *shots on target* as a metric of good offensive decision making can be misleading, as you have no guarantee as to the quality of said shots (Eggels et al., 2016). This problem is addressed by xG. The xG model outputs the probability of a shot being scored when considering the factors surrounding the attempt, such as the position, body part used, positioning of the goalkeeper, angle & distance to goal and more. Thus, the

[1] https://donsetpg.github.io/blog/2020/12/24/Narya/

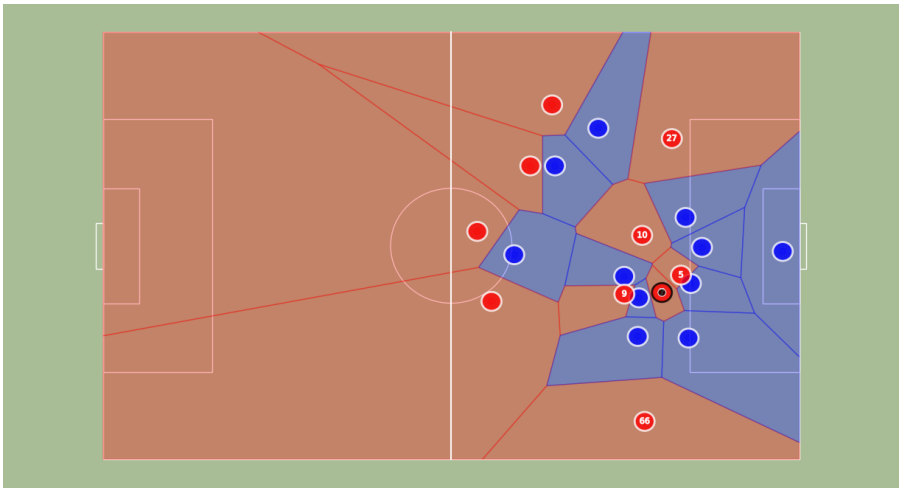


Figure 2.5: Voronoi Diagrams

xG model can be used as an objective metric for analysing the quality of the goal scoring chances created, irrelevant of if the opportunity was scored or not, which eliminates the element of luck from the analysis. Different implementations of the models use different features to train the models, depending on the information that is available to the model. [2,3]. The initial xG implementation carried out by Eggels et al. (2016) used the context of the shot (free-kick, open-play, corner, etc), the body part used, the distance and angle to goal, the number of opposition players between the shooter and the goal, and a similar feature containing the number for the number of team-mates. The models themselves are usually trained using Tree-based algorithms such as Decision Trees or XGBoost (Eggels et al., 2016; Robberechts and Davis, 2020). An example of the output of the StatsBomb xG model is shown in Figure 2.6, as shots taken with opponents obstructing the line to goal, or taken from tight angles have a lower xG value than the other attempts.

The xG model can then be used to analyse teams by summing the total xG generated by a particular team resulting in the xGF (xG For the team), summing the total xG conceded by the team resulting in xGA (xG Against the team). Further models have also been created since the original xG model, namely the Post Shot xG (PSxG) model. The PSxG model differs from traditional xG in that it also takes the shot's trajectory into account. This allows the model to consider the value of shooting into particular zones of the goal. PSxG is a valuable model, especially at evaluating goalkeeper performance, however for this work, we chose to use traditional xG as the scope is not to reward a striker for being a clinical finisher, rather to reward them for deciding to shoot in the first place.

[2] https://understat.com

[3] https://www.driblab.com/analysis-team/how-good-is-driblabs-expected-goals-xg-model/

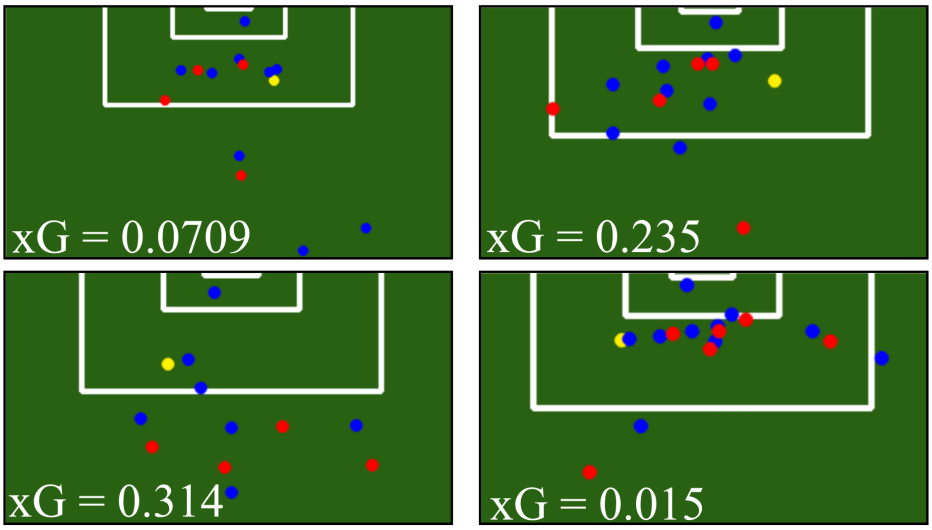


Figure 2.6: Expected Goals Visualisation. Yellow = Player taking the shot, Red = Teammates, Blue = Opponents.

2.5 Possession Value Models (PVMs)

In similar fashion to how xG models attempt to describe the value of a particular shot attempt, Possession Value Models (PVMs) attempt to model how valuable a particular action is, with each different model making different assumptions about what constitutes a valuable action.

2.5.1 Expected Threat (xT)

The xT model (Singh, 2018) aims to provide an objective model into how 'threatening' an area of the pitch is. The underlying assumption made when defining this model is that players tend to move the ball with the intention of increasing their team's chance of scoring. This notion is defined as follows, where xT is denoted by Θ, where :

$$\Theta_{x,y} = s_{x,y}g_{x,y} + (m_{x,y}\sum_{z=1}^{16}\sum_{w=1}^{12} T_{(x,y)\rightarrow(z,w)}\Theta_{z,w}) \tag{2.1}$$

- $s_{x,y}$ is the estimated likelihood of deciding to shoot from zone (x, y)
- $g_{x,y}$ is the chance of scoring from zone (x, y)
- $m_{x,y}$ is the likelihood of deciding to move the ball from zone (x, y)
- $T_{(x,y)\rightarrow(z,w)}$ corresponds to the likelihood of moving from zone (x, y) to (z, w)

The likelihoods are calculated by using historic data that contains the likelihood of the ball to transition from each grid cell to each other cell. To calculate how *threatening* a particular zone (x, y) is, two sub-values are summed. The first value is the likelihood that a shot is taken, multiplied by the likelihood that the shot is scored ($s_{x,y}g_{x,y}$). The second sub-value is determined by how likely the ball is to be moved, multiplied by the value of it being moved there from (x, y). To find the value of moving the ball from (x, y) to a given zone (z, w), $T_{(x,y)\rightarrow(z,w)}$ is used. This is then multiplied by $\Theta_{z,w}$ to obtain the value of moving to (z, w). This is done for all possible zones ($\sum_{z=1}^{16}\sum_{w=1}^{12}$) since it is not known where the ball will be moved to. The metric is defined recursively. Thus value returned by Θ is initially set to 0 for all zones, and it is then found by running the algorithm on each zone iteratively, until the values converge. Thus, the xT model aims to encapsulate the how valuable a zone is by summing the value of the possible actions taken from it. Consider a visualisation of moving the ball from zone A to zone B in Figure 2.7:

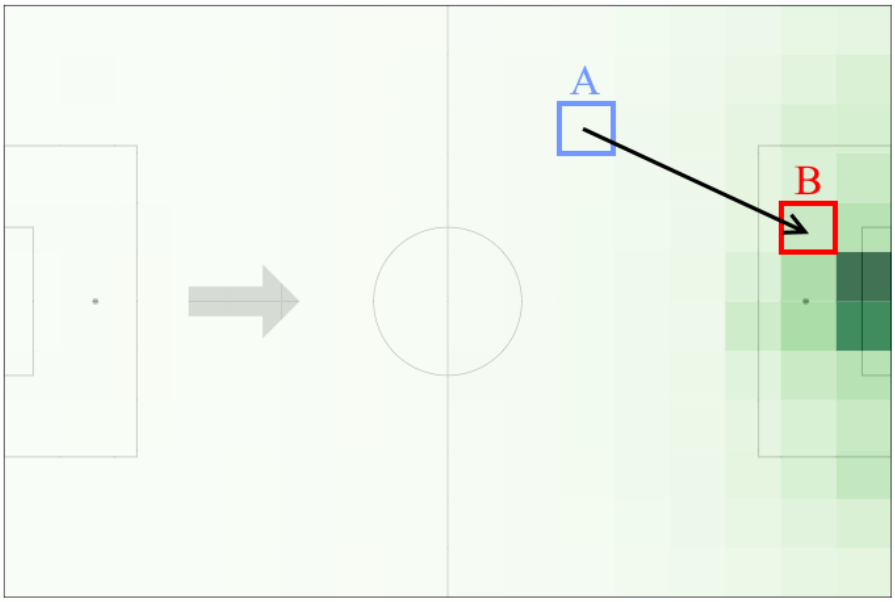


Figure 2.7: Map of xT value per zone (Darker = higher xT). Adapted from (Singh, 2018)

Zones A and B have xT values of 0.022 and 0.126 respectively. This means that progressing the ball from A to B increases the xT by 0.104, or by 472%. This indicates that performing this action increases the value of possession significantly. The inverted scenario can also be considered, where the ball is moved from zone B to A. Here the resultant xT difference would be negative, showing that the decision was one that reduced the value of the possession.

2.5.2 VAEP

The Valuing Actions by Estimating Probabilities (VAEP) model (Decroos et al., 2020) is an alternative PVM that aims to estimate the value of each action taken by a player, in similar fashion to the xT model. However, the key assumption used in VAEP is fundamentally different to that of the xT model, which works on the assumption that the sole aim of each action is to increase your team's chance of scoring. On the other hand, The VAEP model is designed with the assumption that each action is carried out with two intentions in mind, to increase your team's chance of scoring, and to decrease your opponent's chance of scoring. Thus, VAEP makes use of P_{score} and $P_{concede}$ that are represented as follows:

$$\Delta P_{score}(a_i) = P^k_{score}(a_i, t) - P^k_{score}(a_{i-1}, t) \tag{2.2}$$

$$\Delta P_{concede}(a_i) = P^k_{concede}(a_i, t) - P^k_{concede}(a_{i-1}, t) \tag{2.3}$$

$\Delta P_{score}(a_i)$ captures the offensive value of the action a_i for team t by calculating how much the team's chance of scoring has increased by performing action a_i. The function P^k_{score} calculates how likely team t is to score within the next k actions. Thus by finding the difference between value of $P^k_{score}(a_i)$ and $P^k_{score}(a_{i-1})$ (where a_{i-1} refers to the action preceding a_i), we can quantify the offensive value of performing a_i. Here, a positive difference indicates an increase in offensive value and thus an increased chance of scoring within the next k actions, and similarly a negative difference indicates a lower offensive value and a lower chance of scoring within the next k actions. Similarly, $\Delta P_{score}(a_i)$ captures the defensive value of the action a_i by making use of the function $P^k_{concede}(a_i, t)$ to calculate the increase or decrease in the likelihood that performing a_i decreases or increases team t's chance of conceding within the next k actions.

The $P^k_{score}(a_i, t)$ and $P^k_{concede}(a_i, t)$ functions are based on two separate gradient boosted tree models used to predict the likelihood of a goal being scored or conceded within the next k actions. The features used are grouped into three categories, the first of which being features taken directly from the event. These are the action type, result, (x, y) coordinates for the start and end location and the time elapsed since the start of the game. The second category contains features that are calculated, and that encode information about previous events. These are the distance and angle to the opposition goal, the time elapsed and distance covered between the previous and current action, and whether or not possession has changed teams. The final category of features contain the context of the game. These are the match score after the current action concludes and the goal difference due to the current action (Decroos et al., 2020). The formula for $VAEP(a_i)$ is

defined as follows:

$$VAEP(a_i) = \Delta P_{score}(a_i, t) - \Delta P_{concede}(a_i, t) \tag{2.4}$$

Thus, the value is derived from rewarding a_i based on how much it increases team t's chance of scoring whilst also penalising a_i for how much it increases team t's chance of conceding.

2.5.3 On-Ball Value (OBV)

OBV is a proprietary PVM developed by StatsBomb[4](StatsBomb, 2021), that has a similar structure to VAEP. The exact details of the implementation are not made public, however the key differences from similar models were highlighted when the model was introduced (StatsBomb, 2021). The main advantage is that the OBV model is trained using StatsBomb's highly sophisticated xG model, allowing it to value shot actions more accurately than other PVMs. The data science team also made use of two separate tree-based models to predict the likelihood of scoring, and the likelihood of conceding. The features used comprise of 'pitch details', such as the coordinates of the start and the end of the event and the distance and angle to goal. Details of the event itself are also included as features to train the model, such as the body part that was used, or if the event was carried out whilst the player in possession was under pressure from the opposition. Information that encodes possession history was not included within the OBV model's features used for training, such as including which index in the possession chain the current action is. This was done to avoid biasing the model in favour of stronger teams that are more likely to have longer possession chains.

2.6 Pitch Control Models (PCMs)

In the context of football analysis, PCMs refer to models that quantify which areas of the pitch *belong* to which team. A PCM allows us to quantify the probability that a team t would be able to retain possession of the ball if it were to be spontaneously dropped at a location (x, y) on a football pitch, with values ranging from 0 to 1. If the value returned by the PCM is close to 1, it would indicate that team t would be expected to keep possession of the ball. If the value were closer to 0 however, team t would be expected to lose the ball. The values obtained from the PCM are overlayed onto the actual pitch which also contains the locations of the players. The colour of the value obtained from the PCM reflects the

[4] https://statsbomb.com/

degree with which the ball is controlled by each team. Several different features can be used to design a PCM, such as the location of the team mates and opponents, the ball, and the velocity of each player. The most common technique used to create these models is to make use of Voronoi Diagrams (Kim, 2004; Perl and Memmert, 2016). The Voronoi Diagrams are typically modified to include other considerations such as the player velocity, or by combining a PVM model such as Expected Threat or VAEP with a PCM to reward higher value areas that are also more likely to result in possession being retained (Higgins et al., 2023; Spearman et al., 2017). An example of a PCM is shown in Figure 2.8.

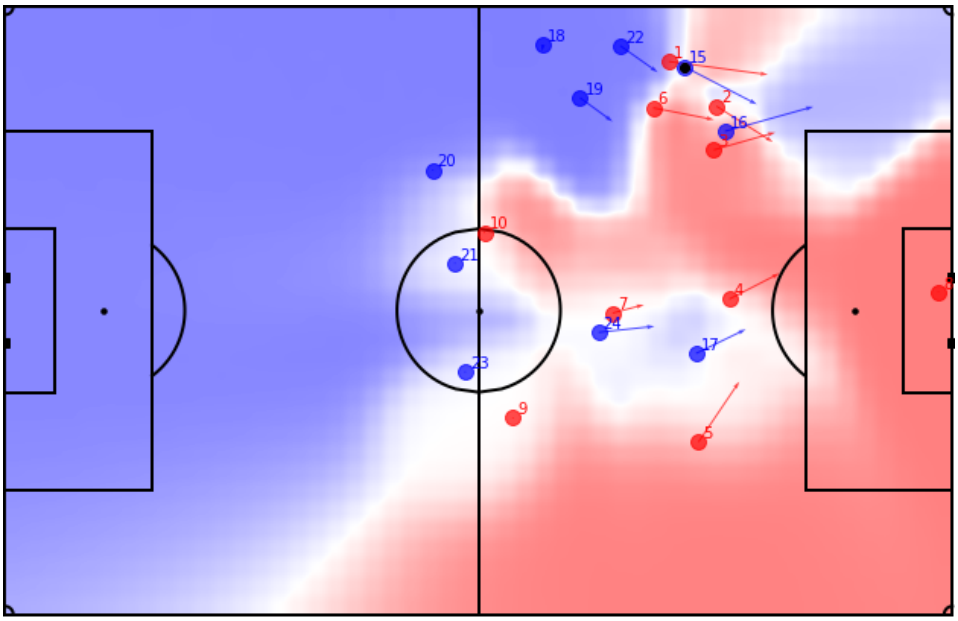


Figure 2.8: Pitch Control Model Example (Spearman et al., 2017)

2.7 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning methodology that solves sequential decision making problems by exploring the long-term effect of taking different actions on the environment. Upon taking an action, the agent is given a reward based on the outcome of the action (Sutton and Barto, 2018). By aiming to maximise the cumulative rewards achieved over time, the agent learns to perform optimally within its environment. The goal of RL is to find a policy π that successfully chooses the right action A_t that maximises the return (Sutton and Barto, 2018), and the process can be visualised in Figure 2.9.

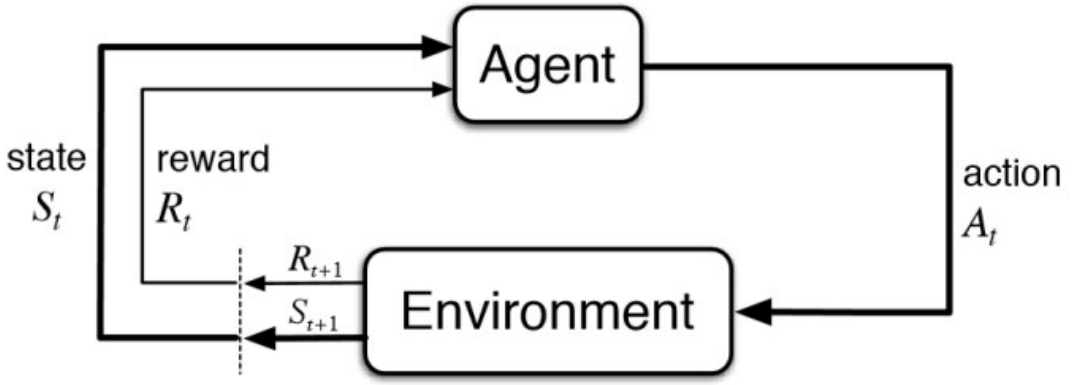


Figure 2.9: Reinforcement Learning Interaction Loop (Sutton and Barto, 2018)

Return. The agent's goal is to obtain the maximum rewards in the long run. This concept is represented by the symbol G_t, which refers to the return at timestep t. This can be formulated in Equation 2.5 (Sutton and Barto, 2018).

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots R_T \tag{2.5}$$

Where T refers to the final timestep within the episode. This formulation seems to align with the agent's goal of representing the long-term cumulative reward. However, the definition is problematic, as for longer episodes, the reward will tend towards infinity, and the influence of the action at timestep t on timestep T will be detached semantically. To address this, the formulation in Equation 2.6 is used (Sutton and Barto, 2018).

$$G_t = r_{t+1} + \gamma^1 r_{t+2} + \gamma^2 r_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.6}$$

Thus it can be seen how γ is used to diminish the value of rewards the further away they are from the current timestep t.

Policy Function. The policy function refers to the agent's strategy. For stochastic environments, the policy function is defined as a probability distribution over all possible actions, as a function of the state it is provided with, using the following notation $\pi(s)$. In deterministic environments, the policy function will simply output the selected action (Sutton and Barto, 2018).

State-Value Function. This function is a function that is applied on states to obtain the expected return from that particular state. It is commonly a learning target within RL

algorithms, as is defined in terms of the policy function π, as can be seen in Equation 2.7 (Sutton and Barto, 2018).

$$V_\pi(s) = E_\pi[G_t|S_t = s] \tag{2.7}$$

This is due to the fact that the rewards are given based on the actions taken according to the policy function.

Optimality. In RL, optimality refers to the fact that for a finite MDP, there will always exist a behavioural policy that obtains the highest possible long term reward.

Different approaches exist for performing RL, such as *on-policy*, or *off-policy* learning. In on-policy systems, the same policy that is being evaluated and improved upon during training is used to take decisions in the environment. In off-policy the target policy is the one being improved, while the behaviour policy is used to explore and discover more information about the environment, thus informing the target policy how it can be improved. This can lead to the agent arriving at the optimal policy earlier since it is more likely to explore the environment. The on-policy paradigm is used within the SARSA (State, Action, Reward, State', Action') algorithm. To update the Q-function estimate TD Learning is typically used, which refers to the idea that the update step for the Q-function is calculated by finding the difference between the existing estimate of expected return the current state-action pair and the actual reward received from the current state-action pair. The formula for this can be seen within Equation 2.8 (Sutton and Barto, 2018).

$$E_t = r_{t+1} + \gamma V_{t+1} - V_t \tag{2.8}$$

The current estimated return for the current state-action pair is represented by V_t. The actual reward gained from the current action r_{t+1} is added to the expected return of the next state-action pair V_{t+1}. Thus by updating the difference between the two values, the Q-function's estimates are shifted towards the new information provided by the latest reward. The update is scaled back by the learning rate γ to reduce noise and ensure that it does not update too far in any direction within a single time-step. The SARSA update step can be seen within Equation 2.9 (Sutton and Barto, 2018).

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \tag{2.9}$$

Thus, it can be seen how TD-Learning is used. The on-policy nature of the SARSA algorithm emerges from the fact that action A_{t+1} is chosen by utilising an ϵ-greedy policy derived from the learned Q-function. The ϵ-greedy policy uses the ϵ parameter to decide whether it will use the learned Q-function, or to choose a random action when obtaining

A_{t+1}. This is done to introduce an element of randomness and exploration that avoids the policy from becoming stuck in local minima. The Q-Learning approach is similar, however it does not make use of the current policy when obtaining A_{t+1}. The update step for Q-Learning can be seen in Equation 2.10 (Sutton and Barto, 2018).

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{2.10}$$

It can be seen how the learned policy is not used. Instead, the Q-function is used directly to obtain the action that has the highest estimated expected return, thus making Q-Learning off-policy as opposed to the on-policy approach of SARSA. The Q-Learning approach can also make use of ϵ-greedy exploration, however it is not based on the current policy as is done in SARSA.

2.7.1 Policy Gradients (PG)

Policy Gradients refers to a class of algorithms used to perform reinforcement learning, introduced in Williams (1992). They aim to tackle RL in a different way to SARSA and Q-Learning. The method was developed as a solution to the issue with using a regular Q-Table within large and continuous action spaces. Instead of using a typical Q-Table and Q-function, this approach only requires a single approximator that is trained to work as the policy function by directly outputting the action probabilities for a particular state. During training of the approximator (which be can any type of model, typically a neural network or a tree-based model) the gradient to be used during learning is calculated such that the learned policy maximises the expected return obtained from the reward function. The policy parameter update is usually performed once per episode, after the rewards are obtained for each action within the episode, however, it can also be updated per-time step. Different techniques can be used to calculate the policy gradient, such as TD based techniques or Monte Carlo based techniques (Silver et al., 2014; Yoo et al., 2021), Policy gradients have been successfully used in many different areas such as contextual recommendations (Pan et al., 2019) and large-scale robot control (Khan et al., 2020).

2.7.2 Actor-Critic

The actor-critic technique is an RL technique that utilises the advantages of PG and Q-Learning. In actor-critic, two separate models are being used simultaneously, called the actor and the critic. The purpose of the actor is to select actions based on the current state. This is typically achieved by making use of the aforementioned policy gradient technique to obtain an optimal policy. The critic is used to evaluate the decisions made by the

actor. This is typically done through the use of Q-Learning, where the critic learns a separate state-action value function which allows it to compare the quality of the decisions made by the actor. During training, the critic makes use of the output of the actor, usually to scale the direction in which it will update. Similarly, the critic utilises the actor during its update, usually by using the actor's policy to obtain the next state and thereby calculate the expected return. A visualisation of the two networks using each other's feedback can be seen in Figure 2.10. The advantages of actor-critic as opposed to actor or critic only networks are the numerical stability as well as addressing the slow convergence.

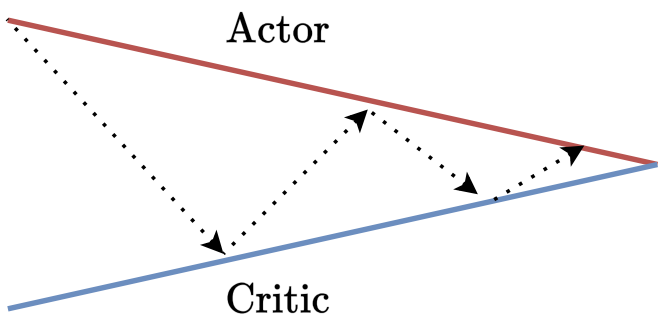


Figure 2.10: Actor-Critic visualisation

An emerging machine learning paradigm that has been used is called Inverse Reinforcement Learning (IRL). The goal is to try to obtain the ideal reward function by learning from 'expert' agents, and it has been used in several different fields such as autonomous driving (You et al., 2019) and modelling of brain activity (Jara-Ettinger, 2019). Recently it has also been used for sports analysis in cricket (Vohra and Gordon, 2021) and American Football (Takayanagi et al., 2022). It has also been used recently for football analysis (Muelling et al., 2013; Rahimian and Toka, 2022). Part of the utility of IRL is the explanatory power it offers. However within this work the main focus is not to understand why players made a decision, rather it is focused on trying to obtain an optimal policy for valuing player decisions.

2.8 Deep Reinforcement Learning

Across several areas in literature, deep learning has emerged as a powerful and often better alternative to classical machine learning techniques (LeCun et al., 2015). Within the RL paradigm, deep learning techniques offer the ability to estimate large action spaces efficiently. One of the first works that made effective use of Deep Learning within RL was

carried out by Mnih et al. (2015). In this work, the authors attempt to develop a single algorithm that can play several different Atari 2600 games, where the only observations are the game score, and an image from the game's screen. They achieved this by using a Q-Learning based approach, and made innovative use of a CNN as the Q-Function approximator. The model was then used as the decision policy function in $\pi_(s)$ that outputs the ideal action to take in a particular state s. The model structure can be seen within Figure 2.11.

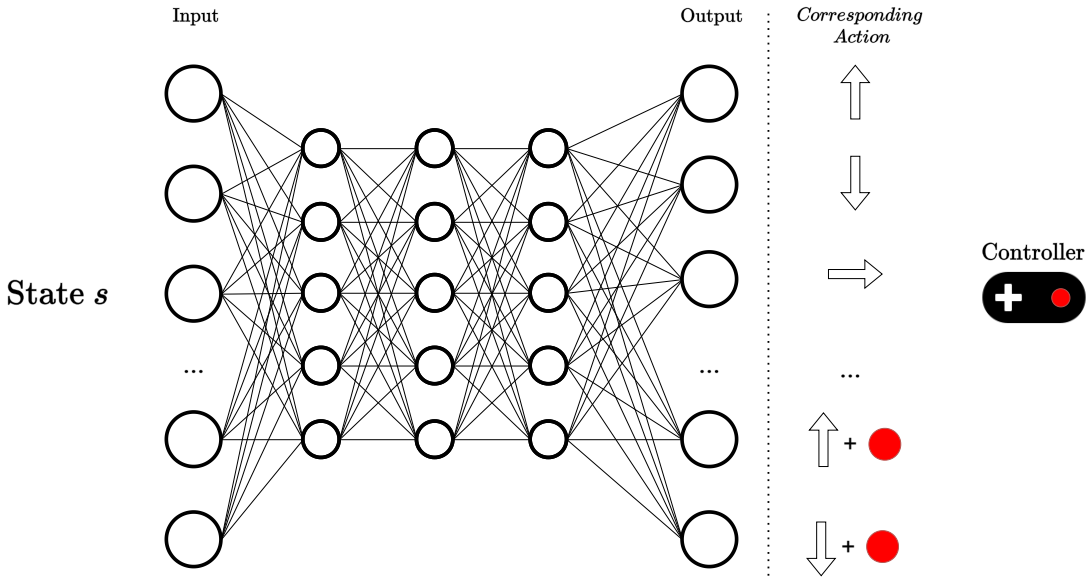


Figure 2.11: DQN $Q(s)$ Network, Adapted from (Mnih et al., 2015)

The model was given the state as the input, as it does not need the particular action that was chosen to be part of the input since the action space is discrete. The output layer already contains a mapping of all possible actions. To train the model, the Bellman loss function is used, defined in Equation 2.11.

$$L(w) = \mathbb{E}\left[\left(r + \gamma V(s') - V(s)\right)^2\right] \tag{2.11}$$

Here, w refers to the weights of the neural network, r refers to the observed reward and γ is the discount factor. $V(s)$ and $V(s')$ refer to the estimated values of the current and next state, and $\mathbb{E}$ refers to the expectation operator. By using this loss function, an approximation of the optimal state-action value function $Q^*(s,a)$ is achieved. To obtain a policy function that selects the ideal action for each state, the maximal value from the last layer of the $Q(s,a)$ model is carried out, as this represents the optimal action. This is represented mathematically as $\pi(s) = \underset{a \in A}{\text{argmax}}(Q(s,a))$.

The main drawback of the DQN approach is its lack of support for large or continuous action spaces. DQN also tends to suffer from overestimation bias. To address the shortcomings, the Deep Deterministic Policy Gradients (DDPG) approach was developed by Lillicrap et al. (2015). To allow for continuous action spaces, an actor-critic approach is used. The actor makes use of the Deterministic Policy Gradient (DPG) approach developed by Silver et al. (2014). The DPG algorithm utilises a neural network to approximate the policy function. In doing so, it allows the policy to output a vector containing continuous outputs without having to discretise the action space. The critic makes use of a variant of Q-Learning, making it an off-policy approach with a neural network is used as an approximator. The main variation from traditional Q-learning is that a copy of the Q-network is made and used to calculate the model's loss, called the target network. The weights of the target network and the actual learned networks are slowly merged over a number of timesteps to reduce noise during training and minimise overfitting.This actor-critic structure can be seen in Figure 2.12.

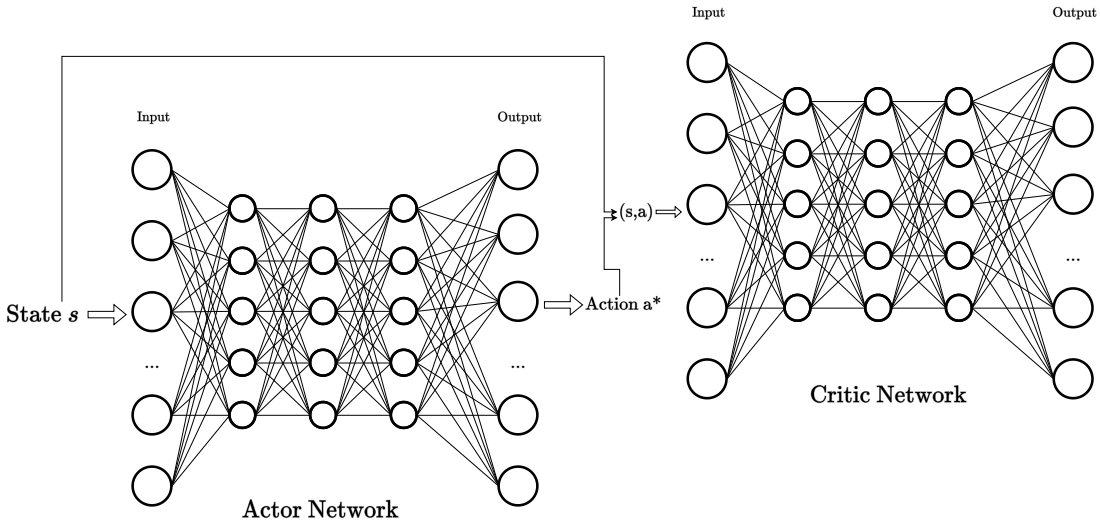


Figure 2.12: DDPG Structure, Adapted from (Liessner et al., 2018)

The main drawback of DQN and DDPG is overestimation bias. This occurs when the Q-function approximation models value certain state-action pairs too highly, resulting in imperfect policies. There is also the issue that actor-critic algorithms tend to become stuck in local minima, thus once the agent discovers a viable path, they might become blind to alternate actions that might yield higher rewards. To address this issue several different techniques were proposed, one of which is the Soft Actor-Critic (SAC) (Haarnoja et al., 2018). This algorithm rewards the agent for taking actions within their environment whilst also maximising the entropy of its decisions. Here, entropy refers to the measure

of randomness present within the policy function's probability distribution. The definition can be seen within Equation 2.12.

$$H(\pi) = -\sum_{a} \pi(a|s) \log \pi(a|s) \tag{2.12}$$

The formula in 2.12 describes the entropy being referred to within SAC, also commonly referred to as the Shannon entropy, which is used to describe the randomness or entropy within a probability distribution. By using the entropy as a factor within a weighted average within the SAC's policy network's objective function, the model is encouraged to maximise the entropy. Since the objective function also takes the traditional Q-function into account, it will not resort into a random policy. This helps SAC to mitigate the issue of getting stuck within local minima. Other approaches to tackle the issues of the DDPG algorithm include the Twin Delayed Deep Deterministic Policy Gradients (TD3) approach (Fujimoto et al., 2018). In this work, the authors make three key contributions. The first of which is to train two Q-function networks *(or critics)* and using the smallest of the two values within the Bellman loss function. In doing so, the overestimation bias is minimised. The second contribution is the delay of the policy function update with respect to the Q-function updates, and the final contribution is the inclusion of randomness to the target function in similar fashion to SAC. Further improvement upon the TD3 approach was made in the TD3 + Behaviour Cloning (TD3+BC) method (Fujimoto and Gu, 2021). This method extends upon the original work by extending the policy function by including a behaviour cloning term. The effect of the added behavioural cloning is to increase regularisation, and the authors found that this addition lead to competitive performance with respect to the state-of-the-art offline RL algorithms.

The aforementioned deep approaches have allowed RL based algorithms to tackle problems where the environment is represented by an image. This image based DRL approach has been used in various different applications, such as stock price forecasting from price history graph images (Lee et al., 2019), autonomous driving (Kendall et al., 2019) and control of physical systems (Nair et al., 2018). The DRL algorithms discussed so far were designed with online RL in mind, thus they cannot be used to train agents within an offline setting without modification.

2.9 Offline RL

Another aspect where RL algorithms vary is in whether the training is done *online*, or *offline*. In online training, the agent learns by taking actions within an interactive environment to update its policy. In contrast, offline learning is used to learn from datasets of

prior actions (Levine et al., 2020). A visualisation of this paradigm can be seen in Figure 2.13.

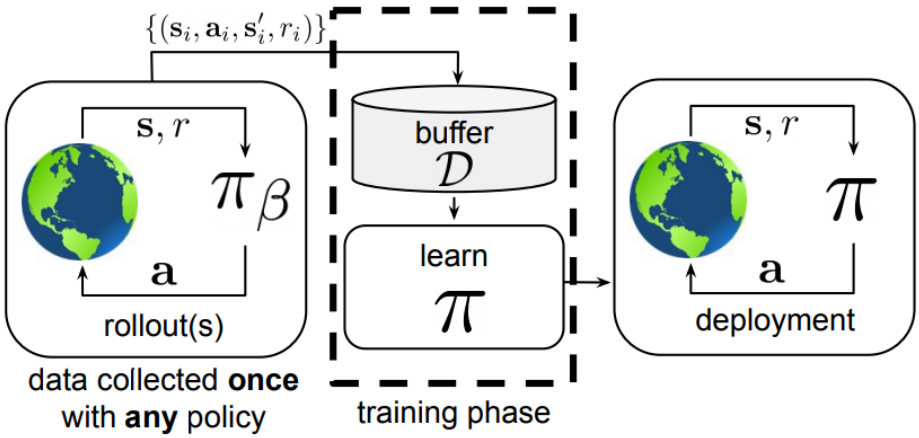


Figure 2.13: Offline RL, Adapted from (Levine et al., 2020)

2.10 Types of Data used in Football Analysis

One of the early works in this field was carried out by (Ernst et al., 2005). In this work, the authors proposed the idea of approximating the Q-function by training on a dataset that contains 4 data points for each timestep t, $< s_t, a_t, r_t, s_{t+1} >$, where s_t is the state at $t)$ and a_t is the action taken at t. Similarly, r_t is the reward for t. The state obtained after performing a_t in s_t results in the next state denoted by s_{t+1}. The dataset could represent a single episode, or could also be the concatenation of several episodes, where terminal states indicate when one episode ends and the other one starts.

The authors defined the notion of a *fitted Q Iteration algorithm*. This was necessary, as the traditional tabular Q-function is only practical in the case of a discrete action space that is relatively small. Whilst the approximator was not necessary for the offline aspect of their work, the authors found that for larger discrete action spaces, or continuous action spaces, the tabular Q-function representation is not practical. This is similar to the approach taken within Section 2.8. The results showed that even when compared with traditional online RL techniques, the offline RL approach paired with the fitted Q-function was able to reduce the complexity of the state, which allowed it to perform well when compared with the traditional Q-Function. The fundamental work outlined by (Ernst et al., 2005) is a basis upon which several other more modern Offline RL techniques have been written.

One of the first offline RL algorithms that made use of deep learning as an approximator successfully was the actor-critic algorithm titled Conservative Q-Learning (CQL) (Kumar et al., 2020). The authors of the Conservative Q-Learning (CQL) algorithm demonstrated how the other algorithms such as SAC, that are not designed primarily for offline RL, tend to perform sub-optimally when applied to offline RL scenarios. One of the reasons proposed for this sub-optimal policy is the effect of dataset-drift. This refers to the scenario where the distribution present within the data does not represent the real world distribution. This is applicable since these algorithms are dealing with offline data, thus they must attempt to learn an optimal behaviour policy from actions performed by past agents, which may not have been optimal. One of the first steps taken to address this was by utilising importance sampling. This was incorporated by re-weighting the Q-values for state-action pairs by the ratio between the probability obtained from the learned policy and the probability of the state-action pair observed from the dataset. Thus, actions that are more likely to be carried out in particular states are weighted higher, to encourage the actor model to learn to overcome the distributions present within the dataset.

Further of RL algorithms has emerged, where the primary goal of the algorithms is to perform offline pre-training, such as the Advantage-Weighted Actor-Critic (AWAC) algorithm Nair et al. (2020). The AWAC takes a different approach to the CQL algorithm, as one of the main contributions is that the actor is incentivised to conform with the data present in the dataset. The issue of dataset-drift is mitigated due to the other incentives given to the actor apart from the conformity to the dataset. The conformity incentive is used during offline training, and also when online training is performed. The authors found that the constraint improves performance of the algorithm when compared with other algorithms such as SAC, AWR and BEAR. Further innovation within the state-of-the-art Critic Regularized Regression (CRR) Wang et al. (2020). CRR is a simple offline RL algorithm that utilises the concept of filtering to perform updates selectively, which helps improve training performance. Due to its simplicity and performance, CRR is commonly used in literature (Lambert et al., 2022), (Konyushkova et al., 2020).

One of the latest works that tackles offline RL was carried out by Kostrikov et al. (2021), called Implicit Q-Learning (IQL). This work aims to address one of the key issues of offline RL algorithms, which is the critic having to evaluate state-action pairs that it had not encountered in the offline data. This was tackled by creating a third network called the Value network, which is trained using the expected regression loss. The value network predicts the estimated return using only the state as input to the network. In IQL, The state-value function is trained using expectile regression, defined in Equation 2.13:

$$L_2^\tau(u) = |\tau - \mathbb{1}(u < 0)|u^2 \tag{2.13}$$

This is then used in the loss function defined in 2.14:

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim D}[L_2^\tau(Q_\theta(s,a) - V_\psi(s))] \tag{2.14}$$

L_2^τ is an asymmetric expectile loss function, where u refers to the difference between the predicted and the target values, and ψ and θ represent the weights of the value and critic networks respectively. The τ parameter is a value between 0 and 1 that refers to the percentile of the error distribution that the L_2^τ tries to minimise. This allows the function to focus on penalising values of u that are within the τth expectile. The effect of varying τ can be seen in Figure 2.14. Lower values of τ will cause the function to minimise the bottom percentile of the error distribution, thereby making the model more conservative, and similarly, higher values of τ will make the function focus on minimising the top percentile of the error distribution, making the model more likely to take more risky decisions. The $L_V(\psi)$ function in Equation 2.14 returns a scalar value that represents the average expectile loss (calculated using L_2^τ in Equation 2.13) between the target $Q_\theta(s,a)$ and the state value function V_ϕ over a sample of D state-action pairs. Here, ϕ represents the weights of the approximator network. Thus, the equations in 2.14 and 2.13 are the loss functions used to learn the state-value function using expectile regression.

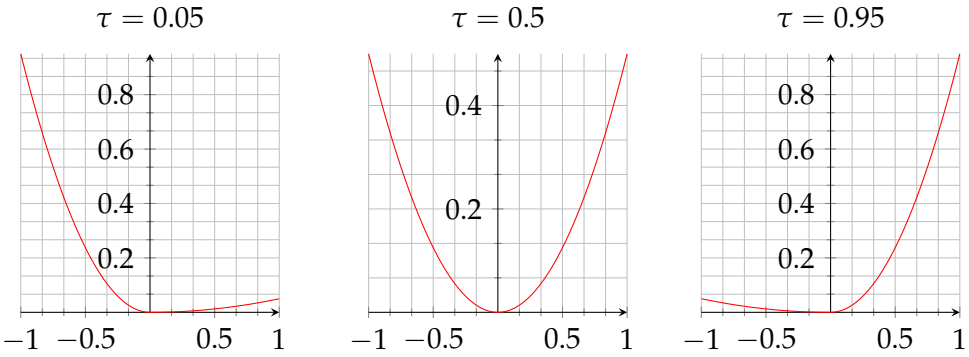


Figure 2.14: Effects of varying the value of τ on L_2

To train the Q-Function, the following loss function is used:

$$L_Q(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}[(r + \gamma V_\psi(s') - Q_\theta(s,a))^2] \tag{2.15}$$

The advantage of using expectile regression to estimate a state-valuation function can be seen when comparing Equation 2.15 with Equation 2.9. By utilising the value network ($V_\phi(s')$) to estimate the expected value of the next state s', the loss function does not

need to query the Q function each time, and the model also will not have to estimate the value of out-of-distribution actions. This optimisation step improves the computational performance of the model. Otherwise, Equation 2.15 makes use of the traditional structure associated with the equation for Q-Function loss. The reward obtained is added to the discounted expected return of the next state, after which the value of $Q_\theta(s, a)$ is subtracted. The scalar value that is obtained is then squared. The final loss function that must be outlined is the one used for the policy function, which can be seen in Equation 2.16.

$$L_\pi(\phi) = \mathbb{E}_{(s,a)\sim D}[e^{(\beta(Q_\theta(s,a)-V_\psi(s)))} \log \pi_\phi(a|s)] \tag{2.16}$$

In Equation 2.16, there are two components. The second segment, $\pi_\phi(a|s)$ is used to obtain the probability that the proposed transition is carried out. Here ϕ represents the weights of the actor network. The log function returns increasingly large negative values for smaller input values. Thus, within $log(\pi_\phi(a|s))$, the smaller the likelihood that a is chosen, given state s, the larger the negative loss. Within the first part of the equation the difference between $Q_\theta(s, a)$ and $V_\psi(s)$ is calculated. In doing so, the loss function can quantify how much the model is over valuing the state-action pair when contrasted with the state valuation function. This difference is then multiplied by a hyper-parameter β. This parameter can be used to increase or decrease the influence of the first half of the equation. For greater values, the policy loss function will aim to obtain the maximum Q-function, whilst for smaller values, the policy loss function will only seek to replicate behaviour from the observed actions, for which values can be between 0 and 1. The output obtained from multiplying the difference by β is exponentiated using the exponential function, which simply returns the value of e^x. In doing so, the value is converted into a positive value that will be increase drastically for higher values, thereby increasing the policy loss significantly if the disparity between $Q_\theta(s, a)$ and $V_\psi(s))$ is large. Thus it can be seen how multiplying the values obtained from the first and second halves of Equation 2.16, the policy loss function can be used to weigh the difference between the state-action value and the state-value estimation by the current estimation of the action's likelihood.

The simplicity, efficiency and state-of-the art performance (Kumar and Kuzovkin, 2022) has led to its adoption in a variety of different offline RL tasks (Hussing et al., 2022), (Shah et al., 2022), (Prudencio et al., 2022). Offline DRL has since been used in several different use cases, including robotics (Sinha et al., 2022), health-care (Fatemi et al., 2022) and advertising (Wang et al., 2022)

3 Literature Review

In this chapter, a thorough review of literature will be presented that offers a comprehensive summary of both the fundamental works, as well as the most recent contributions to the various topics, such as the meaning of decision making, the application of RL to football analysis, as well as the recent works that attempt to obtain meaningful insights from the combination of both event and tracking data.

3.1 Decision Making Analysis in Sport

The human decision making process is an active area of research that takes into account the many factors associated with making a decision. Decision making is said to be 'intentional, consequential and optimizing' (Chia, 1994; March, 1989), and whilst the original context for this quoted section is related to organisational leadership, the sentiment remains true for the domain of sport. Work by Gréhaigne et al. (2012) also highlighted the various crucial aspects associated with decision making within sports, both those on an individual level player location, tactical knowledge, ability & experience, as well as the factors that apply on a team based level *(team tactics & cohesion)*. In a review of literature carried out by (Silva et al., 2020), several other key works that look into the process of decision making within various age-categories of different sports, such as football, basketball and handball were analysed. Decision making in youth football was also found to relate to the direct area of the pitch that players subconsciously focus on at different moments during play (Vaeyens et al., 2007). They showed that higher level players are more likely to observe their environment more quickly and effectively, indicating that proper observation of the environment is a key factor within the elite level sports. Woods et al. (2015) looked into the difference in judgement between the decision making quality of Australian under-18 elite-level football players, and non-football playing participants under the age of 18. To measure the decision making quality, professional coaches were asked to identify the correct next action. The results showed that the elite level players could be identified reliably solely by their decision making ability, indicating that it

is an important trait for elite level footballers. The ability for player decision making to be improved through coaching sessions was also analysed by Pizarro et al. (2019). The results showed that improvements in futsal player decision making was recorded, measured using an objective performance metric, was found after the players were subjected to several coaching sessions.

3.2 Football Decision Analysis

One of the earliest and most influential works that utilised match data for football decision analysis was carried out by Rudd (2011). The scope of this work was to quantify the contribution that players make towards creating high quality goal scoring opportunities. This was done by splitting the attacking zones of the pitch into 6 separate zones. Event data from a season of the English Premier League was then used to create a transition matrix between the aforementioned states, as well as considering other states such as throw ins, corners and penalties. Using this matrix, the likelihood that the current state, s, transitions to a goal state us defined as $P(s)$. Thus, by considering a sequence of states, and finding $P(s)$ for each state, the authors could identify which actions lead to an increased chance of a goal in the next action. This information could then be used to identify which players contribute the most to creating high quality opportunities, whilst also identifying the players that decrease the goal-scoring opportunities.

3.3 Machine Learning in Football Analysis

Early use of Machine Learning (ML) methods in football analysis were mainly focused around handling big-data, and creating pipelines that allow for squad tactical analysis (Rein and Memmert, 2016). ML has also been used in football analysis to represent players as points on a 2D scatter plot, after which clustering was applied to identify which players have similar playing styles (García-Aliaga et al., 2021). Clustering was also used to automatically identify player positions from a large dataset of football event data. A holistic performance metric (called PlayeRank) was then obtained by correlating each possible action performed within each position, against the outcome of the game, to automatically find the coefficient with which each action contributes towards a 'good performance'. The results showed that the highest scoring players from each cluster aligned with the highest scoring players predicted by football scouts. This shows that ML techniques applied on football event data can produce valuable player evaluation. ML has also been applied to football event data to predict which team will take the next shot within a possession

sequence, reporting an accuracy of 75.2% (Kusmakar et al., 2020). Within the same work, the authors devised metrics to identify player-to-player interaction networks, showcasing the versatility of ML techniques on football datasets.

3.4 Reinforcement Learning in Sports Analysis

RL has been used successfully in performing analysis of player performance and decision metrics in hockey (Liu and Schulte, 2018) and basketball (Yanai et al., 2022). To develop a model that can assess the performance on NHL players, the authors made use of DRL based on a dataset that contained 3 million data points (Liu and Schulte, 2018), which closely resemble the event based dataset type outlined in Section 2.10. The dataset contains information pertaining to the location of the player in possession of the puck and additional features that encode the context of the game such as the timestamp and the score differential. In this work, the terminal action in each episode was considered to be when a goal was scored. Thus the first episode begins at the start of the game, while subsequent episodes begin after the restart from a goal action. The authors made use of offline SARSA learning, where an approximator is used to estimate the Q-Function.

Thus, they are using an on policy algorithm. Since they wish to use a continuous and large action space, the authors chose to use an approximator for the Q-Function. To achieve this they used a dynamic LSTM that made use of a dynamic trace length. The trace length refers to the number of inputs that the LSTM can process, thus allowing the LSTM to process a variable amount of events. Three separate Q-functions are trained, being those for the home, away or neither team. The feature list used is roughly identical to the one used within VAEP, however within the context of hockey. Thus it includes information such as the (x, y) coordinates of the puck, its velocity, angle between puck and goal, and other similar features. The authors highlighted that the scope of the DRL model was to develop a model to be used as a behavioural analytics tool for real world players, as opposed to developing a model with the intent of controlling artificial agents (Liu and Schulte, 2018). By utilising the trained Q-function using the on-policy SARSA approach, the authors developed the Goal Impact Metric (GIM), which aims to find which players' actions caused the highest increase the teams expected return for their respective teams within their dataset. This was done by first defining a players impact for a particular team, as can be seen in Equation 3.1.

$$\text{impact}^{\text{team}}(s_t, a_t) = Q^{\text{team}(s_t, a_t)} - Q^{\text{team}(s_t-1, a_t-1)} \tag{3.1}$$

Thus, it can be seen how the *impact* of a particular action resembles the xT and VAEP

implementations, as the difference between the team's likelihood of scoring between consecutive actions, making this work especially influential when considering that it precedes the work carried out in VAEP. The impact of each player is then calculated by iterating over all state-action pairs that can be seen within the dataset for a particular action and multiplying the impact of each state-action pair by the number of times that the player performs the action within the particular state (which is typically 1 due to the continuous nature of the state representation). This can be seen within Equation 3.2

$$GIM^i(D) = \sum_{s,a} n_D^i(s,a) \times \text{impact}^{\text{team}}(s,a) \tag{3.2}$$

The results showed that the GIM metric could successfully identify undervalued players. This was demonstrated as two undervalued players were identified amongst the 20 highest GIM players before the players were offered high salary increases. The result was also found to correlate with the number of goals scored by the player. The results illustrates the suitability of offline deep reinforcement within sports contexts at valuing player actions and decisions. In a similar vein to the work carried out by (Liu and Schulte, 2018), (Yanai et al., 2022) set out to develop a system that would be able to evaluate basketball players' decision making, called Q-Ball. To achieve this, the authors trained a DDPG model in offline mode, thus making it an actor-critic based model that was trained on a dataset of precomputed actions as opposed to a simulated environment. It was trained on the combined dataset from two different sources. The two data sources are similar in nature to the event and tracking data highlighted in Section 2.10. This allows the model to value actions within the context that they were performed in, as the model is informed of the action that was carried out, as well as the location of the surrounding players at that moment in time. The reward function was created such that a reward of +2 or +3 if a successful shot is made, corresponding to the increase in score. In the case of the player surrendering possession to the opposite team, a negative reward of -0.5 is assigned. Otherwise, a reward of 0 is given. This reward function teaches the model to learn to value good shot actions, and to punish possession loss. Assigning a reward of 0 if the possession is retained prevents the model from simply rewarding longer possession chains. Qualitative analysis of the model's output showed how Q-Ball was able to identify players that are performing well and that have high potential, as well as to determine which teams have the best performing players with respect to the average Q-Ball value obtained by their players. This work illustrates how a combination of event and tracking data can be used to train an offline actor-critic based DRL algorithm to develop a metric for evaluating player decisions in a team sport. Even though it is a different sport, the fundamental techniques and results are relevant to a footballing scenario.

3.5 Reinforcement Learning in Football

The earliest use of Reinforcement Learning (RL) within football based environments was in the area of physical robots (Duan et al., 2007; Riedmiller et al., 2009). These were trained to play and compete against each other in physical environments. Virtual environments, such as the one developed by Google in (Kurach et al., 2019) proved to be a popular way to train RL models to learn to take the ideal action within a football game. To allow the agents to learn effectively, various state representation types were proposed. The first consisted of a rendered image of the entire pitch from the view of a virtual broadcast camera, containing the players as well as a 2D mini-map. The minima. The second type of image observation consisted of a list of matrices, with each matrix being a 72×96 grid containing 0s. The first matrix would then have values of 1 corresponding with the location of the players, and similarly the other matrices contain 1s at the location of the opposition and the ball. The action space considered within the Google RL environment was a discrete action space. The actions were split into dribble, pass, tackle, shoot and sprint. The action space also contained the possible directions within which the action could be performed. This environment was used to identify viable deep reinforcement learning approaches for football decision making. Albeit in simulated environments, multi-agent systems were trained successfully using TD3 and CQL (Huang et al., 2021; Lee et al., 2021).

RL techniques have also been used on historic football event data for player decision analysis. One example that makes use of traditional machine learning techniques to model player behaviour as an explicit Markov Decision Process (MDP) was developed in Van Roy et al. (2021). The states are considered to be the different locations on the pitch, identified by a grid split shaped 12×16, similarly to the split used in xT (Singh, 2018). The authors use a simple reward function where the discount factor is set to 1, and a reward of 1 is assigned for actions that result in a goal, with a reward of 0 being assigned otherwise. The actions were modelled as either shot or movement actions, and a the transition function is created by estimating likelihood that the an action is successful. In the cases of movement success is determined by the likelihood that a pass or carry is successful in moving the possession without surrendering it to the opposition, whilst in the case of a shot it is defined as the likelihood that a goal is scored from that shot.

The policy function is also learned from the dataset, as it provides the probability distribution for the possible actions within all possible states. The final trained MDP was used to evaluate players to identify the highest risk taking players, and the most conservative players. It was also used to identify areas of the pitch where teams are underperforming, such as situations in which teams are opting not to shoot when shooting

would have been a better alternative, amounting to a number of goals lost over the duration of a season. The results show how RL applied to event data within football can provide meaningful insights into player and team decision making.

3.5.1 Deep Reinforcement Learning in Football Analysis

One of the first applications of DRL within the context of football was carried out within Liu et al. (2020). In this work, the authors made use of TD learning to apply on-policy SARSA learning. They used an LSTM based approximator to estimate the Q-function. The authors stated that the focus was not on obtaining an optimal policy. Instead, they tackle the prediction problem and are focused with obtaining a model to be used for player decision analysis, in a similar fashion to the aforementioned work in Yanai et al. (2022). The model was trained on event based data, thus the environment is only partially observed. The features computed for each event are the time remaining in the game, the x and y coordinates, the current goal-difference, the action taken and its outcome, the ball velocity, the event duration in seconds, the angle between the ball and the goal, and the team identifier (home/away) in possession of the ball. The reward given was formulated as a vector of length 3 of the form $[g_{tHome}, g_{tAway}, g_{tNeither}]$.

The reward at time-step t for the reward function was set at 0 for all elements of the vector, until the time-step where either the home or away team scores a goal. At this point, the respective element would then be set to 1. In the event that neither team scores until the end of the game, the element corresponding with *neither* would then be set to 1 at that final time-step. The episodes themselves were split into goal scoring episodes. Thus, the Q-function corresponds to the probability that either the home or the away team scores at the end of the episode. From this, the authors were able to obtain the GIM metric in similar fashion to their previous work that was discussed within Section 3.4 (Liu and Schulte, 2018). The metric was compared with other player decision valuation models such as VAEP, as well as traditional indicators of success such as Goals Scored, and Assists Provided. The results showed that the metrics developed within this work had significant explanatory power with respects to player evaluation.

One of the first examples of DRL being used in offline mode to football player decision making was developed by (Rahimian et al., 2021). In this work, the authors sought to develop a model that could propose alternative decisions that could have been made in the critical moments of the game. They were defined as the moment leading up to loss of possession, or when a shot is taken. The action types considered were pass, shoot, foul or clearance. Dribble and take-on actions were not considered, as in these critical scenario, these would not be viable options, according to the authors. The data used

within the work was composed of a combination of event and tracking data from 104 European football games. The dataset was confidential and never made public. Thus, the model is made aware of both the action itself being performed, as well as the context within which the action was performed. The dataset was converted into episodes where each episode terminates when the team in possession of the ball surrenders possession to the opposition.

A CNN-LSTM based model was first trained to predict the next action without the use of RL. This was used as the behaviour policy, based on the actions taken by players in the past. The policy gradient technique was then used to perform offline DRL by retraining the behavioural policy and tuning the weights towards the values returned by the reward function, thereby obtaining an optimised policy directly. The reward function takes the different types of actions into consideration when calculating the reward. If the action is a shot, then the xG of the shot is assigned as the reward. If the event is not a shot but possession is retained, then the reward given is the difference between the value of possession at the start of the action and the end, thereby rewarding the agent for taking actions that increase the value of possession. In the case that possession is lost, then the reward assigned is a negative constant, to discourage the agent from losing possession. The results showed that the optimised policy was able to obtain a higher average return than the behavioural policy. Qualitative evaluation also confirmed that the model's predictions make intuitive sense within a footballing context. These include cases where the optimal policy recommended shooting in cases where the player opted not to shoot from areas that would have obtained a high xG as well as suggesting that a clearance is better suited than a foul within certain defensive situations. This showed that offline DRL applied on a combination of event and tracking data to evaluate football player decision making can obtain valuable insights.

3.6 Combining Event and Tracking Data

One of the earliest and most influential works that combines tracking and event data was carried out by Fernández et al. (2019). In this work, the authors made use of a formulaic approach to modelling the expected value obtained for taking a particular action. The three main action types that were considered within this work were *pass*, *shot* and *ball-drive*. A simplified representation of the method used to value a particular tracking frame, which contains the coordinates of all 22 players can be seen in Equation 3.3.

$$EPV(t) = V(Pass) \times P(Pass) + V(Shot) \times P(Shot) + V(Ball\_drive) \times P(Ball\_drive) \tag{3.3}$$

Thus, the expected value that would be obtained from each action is multiplied by the respective probability that the action type itself is chosen by the player, and the final result is the sum for all considered action types. Action probability was determined through the use of convolutional neural networks, and the value of the action itself was determined through a combination of features in the case of pass and ball-drive. Expected Goals were used in the case of the shot action.

The results from the work were mostly evaluated through qualitative and empirical observations of real world-game scenarios. These found that the EPV returned results in line with domain expert knowledge in several different scenarios. The authors argued that the model's ability to consider the value of different actions independently provided additional ways which decision making could be analysed by action type. Thus, the work carried out by Fernández et al. (2019) showcases how tracking data can be used to obtain meaningful insights into granular player decision making moments, especially when combined with existing tools for football performance analysis such as heatmaps and pass networks.

More recently, work has been carried out that combines both the event and tracking data, such as that provided by the StatsBomb 360 dataset[1]. This dataset solves the issue of manually aligning event and tracking datasets by providing both the event and tracking data aligned within the same dataset. In their work, StatsBomb proposed a metric called Line-Breaking-Passes[2]. This metric aims to identify which passes break opposition lines. An opposition line refers to the virtual line that is created by the opposing players. Typically, 'lines' are created by the midfield and defensive players, where they assume the shape of multiple lines parallel to the half-way line across their own half to try to reduce the space that is available to the attacking team, and to protect their own goal. These lines created by the opposition are typically quite close to each other, thus penetrating the lines is quite a difficult task.

In this work carried out by StatsBomb, the tracking and event data were both utilised to identify which passes are able to break the aforementioned defensive lines. The tracking data allowed the authors to identify the location of the lines setup by the opposition players, and the location of the team mates. The event data enabled the identification of passes which managed to 'break' the opposition lines, thereby representing difficult

[1] https://github.com/statsbomb/open-data

[2] https://statsbomb.com/articles/soccer/statsbomb-360-exploring-line-breaking-passes/

passes that move the ball into key areas of the pitch. An example of this can be seen in Figure 3.1.

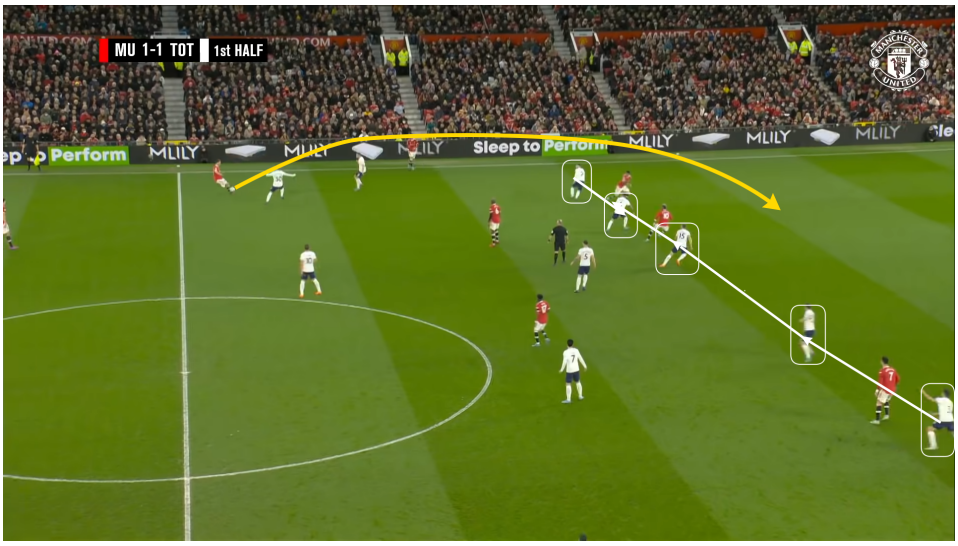


Figure 3.1: Line Breaking Pass example

Here, the pass marked in yellow intercepts the defensive line formed and shown in white, thereby making it a 'line breaking pass'. In defining the metric, the authors were able to compute a list of passes that are able to identify the players that break lines the most with their passing, and also performed clustering on the passes themselves to find out which types of line breaking passes were made by the players. This work demonstrates the valuable insights that can be obtained when combining both types of data.

Further research that made use of both tracking and event data to evaluate decision making for pass actions by players was carried out by Burriel and Buldú (2021). In this work, the authors attempted to create a set of functions that could numerically represent the risk and reward characteristics of passes that occur throughout a match. To achieve this, the authors devised a function that could predict the likelihood of a pass being intercepted through factors such as the location of the surrounding teammates and opposition players by utilising the tracking data. This was done by considering the typical ball speed, the coordinates of the start and the end of the pass action, and other factors. This allowed the authors to represent the risk associated with a particular pass to any other coordinate on the pitch (within the broadcast camera's range).

To represent the reward associated with the pass, the authors made use of the aforementioned EPV model (Fernández et al., 2019). By considering the difference observed from the EPV model from the start to the end of the pass action, the authors were able to increase or decrease in the value of possession through the pass. The results showed that

by utilising the tracking and event data, the PVM and the formulas defined in their work, the authors were able to quantify which players make high risk & reward passes. Clusters emerged from the graphs, such as defenders that make a high volume of low risk passes, and attacking players that often take higher risk/reward passes. The work focused on the actions carried out by F.C Barcelona players, and the results also highlighted the excellent quality of pass decision making exhibited by Lionel Messi, which aligns with what most analysts and pundits would expect. While having different research objectives, these works reinforce the hypothesis that combining event and tracking data together can lead to a more informed situation analysis than that drawn by using either in isolation, as well as the value that can be obtained from utilising PVMs to observe the effect of an action by comparing beginning and the end coordinates of a particular game state.

Tracking and event data was also used to perform analysis of the decision making of football players in the final third[3] with regards to shot decision making in the work also carried out (Larrousse, 2019), where they made use of Voronoi Diagrams and data provided by StatsBomb to analyse shot actions. For each player that made a shot action, their neighbouring teammates were identified. By using data from past actions to predict the pass transition probabilities from each of the zones of the pitch, the likelihood that a pass is performed to any other zone of the pitch is calculated. The authors made use of this pass probability and the Voronoi Diagrams to compute the probability of a pass being made to any of the teammates in a polygon that is directly adjacent the one that the shot taker is in. An xG model was then used to compute the likelihood of a goal being scored for each neighbouring teammate. If a neighbouring teammate was found to be in a likely passing position and the theoretical xG of the teammate's shot is found to be higher, then the original shot taker is said to have had a better option. The results showed that the best performing teams had players that chose the best options at shot action moments.

Further work that utilises offline DRL was carried out in (Rahimian et al., 2022). In this work, tracking and event data were used to obtain an optimal decision policy. The input given to the model is an 11 channel matrix, where each layer is a feature engineered matrix designed to maximise the model's learning potential, such as a matrix containing 1s corresponding with the coordinates of the teammates, and a similar one containing the coordinates of the opposition, and other layers representing the angle and location from goal. A deep policy network is then trained to predict the two different probability surfaces, being the surface indicating success and the surface indicating where the ball is likely to be played. One of the novel concepts introduced with this work was the formu-

[3] For the sake of analysis, the football pitch is often split into thirds, with the dividers being drawn parallel with the half-way line. The team's attacking third refers to the third of the pitch closest to the opposition's penalty area.

lation of the reward function. The reward assigned was different depending on the phase of play that was being carried out at that moment in time, reflecting the different requirements of that particular phase. The results showed that the model was able to offer powerful insights regarding the optimal playing style in different zones of the pitch within the Belgian League, further showcasing the utility of offline DRL applied on combined football and tracking data.

3.6.1 Conclusion

In this chapter, the seminal works associated with RL were compared and contrasted. Following this, offline RL was then discussed, by thoroughly discussing the influential contributions and how each improved upon the existing state-of-the-art. Furthermore, the RL contributions were discussed in the context of sport and football in particular, leading up to the latest papers that make use of RL to evaluate football players and team performances.

4 Methodology

Building upon the work carried out in literature, we propose an RL model that will make use of historic player data obtained from elite-level football games to address the problem of objective player decision analysis.

4.1 O1: Optimised Dataset

The starting point for creating a dataset that is optimsed for performing DRL upon is to identify existing datasets that contain enough information to allow for the required data and context to be extracted and restructured into a format that lends itself better to the desired task.

4.1.1 Candidate Dataset

As was outlined in Section 2.10, the two most common data types used to achieve this goal are the event and tracking datasets. This data is usually the product of companies whose clients are footballing organisations, thus it is difficult to find free samples of such a dataset. However, there are still a few public datasets that can be found.

4.1.1.1 Publicly Available Datasets

The Wyscout dataset[1] contains event data from 1,826 games from the highest level football leagues from England, Spain, France, Germany and Italy. The best attribute of the dataset is the sheer volume of data, as useful insights can be obtained and tested correctly using the data provided. The main drawback however is that the dataset does not contain tracking data of any sort, thus there is an element of context that is missing that makes it unsuitable for this project. Another public dataset is the Metrica Sport[2]. This dataset contains high-quality paired event and tracking data. The main issue with this

[1] https://github.com/koenvo/wyscout-soccer-match-event-dataset
[2] https://github.com/metrica-sports/sample-data

dataset, however, is that the data is only provided for 3 matches. Thus this is only suitable for demonstrations or experiments with the data. A similar dataset was also released by SkillCorner, that contained 9 full matches[3].

Thus, the most suitable that was found was the StatsBomb Open Data dataset[4]. This dataset contains paired event and tracking data from 82 full football games, taken from sets of free data packs that the company has released over the years, including data from the 2020 Men's Euros, the 2020/21 men's La Liga, and the Womens' 2022 Euros amongst other competitions. This dataset was selected as the most suitable dataset, as thousands of paired event and tracking actions could be used to formulate the optimised dataset. Thus it was selected as the dataset of choice for this project. The main limitation of the dataset was the number of games that were made available, as even though the number is unprecedented, training and evaluation would not be straightforward given the number of games and types of competitions provided. This information is summarised in Table 4.1.

Table 4.1: Publicly Available Datasets

Dataset	**Type**	**Number of Matches**
WyScout	Event	1,826
Metrica	Event & Tracking	3
SkillCorner	Event & Tracking	9
StatsBomb Open Data	Event & Tracking	81

4.1.1.2 StatsBomb 2022 Conference Dataset

During the course of development, the StatsBomb company announced the launch of their yearly conference at the end of May of 2022. Researchers were invited to present their ideas to the company, which would then be presented in the conference in September of the same year. The researchers that were accepted would be provided with a dataset that has the same structure as the aforementioned StatsBomb Open Dataset, however the data would be sourced from two entire elite level seasons of the researchers' choice. The work carried out within this research was accepted, published (Pulis and Bajada, 2022) and presented at Wembley Stadium in London[5]. Thus, StatsBomb provided us with the paired event and tracking data from the 2020/21 and 2021/22 seasons of

[3]https://github.com/SkillCorner/opendata
[4]https://github.com/statsbomb/open-data
[5]https://www.youtube.com/watch?v=gBMXw-PHuCY

the English Premier League, containing 580 games in total, and this was the dataset used throughout this work.

4.1.1.3 Dataset Statistics

In this section, information and statistics about the StatsBomb research dataset will be provided. This will provide insight into the dataset that will provide the context within which design decisions were made. In order to process the data, the Soccer Player Action Description Language (SPADL) library (Decroos et al., 2018) was used, and the references to the action names within this section are defined within the SPADL documentation[6]. The goal of the SPADL library is to obtain a universal protocol for describing football player actions. This means that datasets can be fed from different data providers such as StatsBomb, WyScout or Opta, which all have different specifications. The SPADL library processes the different datasets into a single regular format, thus the implementation is not dependant on a particular data provider. The first analysis that was carried out pertained to the number of actions per action type, which can be seen in Figure 4.1.

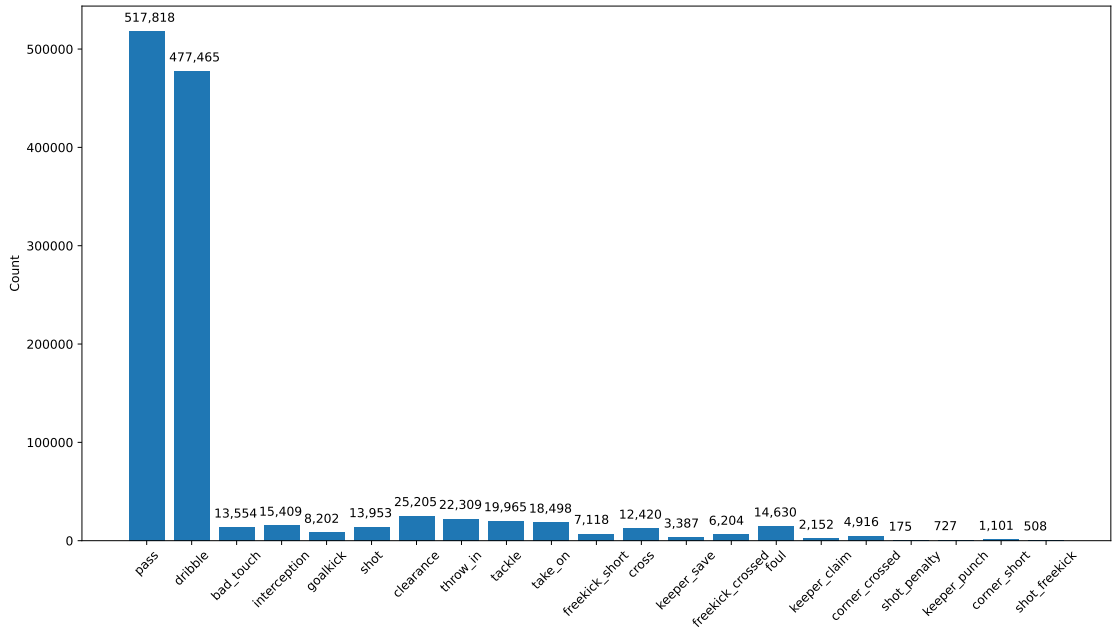


Figure 4.1: Counts of action type within the dataset

The vast majority of actions are either passes or carries, accounting for 82% of all actions taken during a game. This is to be expected as the two events are the only way that teams can move the ball across the pitch and eventually towards the opposition goal,

[6]https://socceraction.readthedocs.io/en/latest/documentation/SPADL_definitions.html

in order to attempt to score. Carries are not considered to have possibility of losing the ball. Whilst the granularity of the dataset is impressive and allows for in-depth analysis of the performance of players, for the scope of this work the actions considered were reduced to the most salient ones:

- **Pass:** Kicking the ball towards a teammate to transfer possession.
- **Carry:** Moving the ball across the pitch in an uncontested manner.
- **Take-On:** Attempting to directly move the ball past an opposition player in a One-on-One scenario.
- **Clearance:** To kick the ball away from wherever it currently is, usually as far away as possible from your own goal, and is normally carried out as a last resort as it surrenders possession.
- **Shot:** To attempt to score a goal by directly kicking the ball towards the opposition's goal.

4.1.2 Augmented Dataset

The objective of creating an augmented dataset is to obtain a dataset that captures events within the context that they were performed. In order to perform offline DRL, the dataset must contain synchronised values for observation, the action, the reward, and the terminal flag *(referring to if the episode terminates after the action)*. The observation is obtained from the tracking data, that represents the positions of the players on the pitch at the moment the action was taken. The action is the action that the player takes on the football pitch. The reward is the value of the action that will be outlined later on, and the terminal flag depends on if the possession terminates after the action is taken.

The event based data from StatsBomb is loaded into the SPADL library to reduce the reliance on a single data provider. This results in a dataset of actions described by the SPADL schema instead. The main difference is that the StatsBomb dataset splits passing the ball and receiving it into two different events. In SPADL, these are combined into a single event. A different schema called Atomic-SPADL exists that also splits passes into the pass and the receipt, however for this work the SPADL dataset was chosen. The pitch is considered to be 105 units wide and 68 units long. Events within the SPADL schema are represented as a list of dictionaries that each contain the relevant details for the particular action. An example of the SPADL dataset is shown in Table 4.2.

It is important to note that even though action 4 follows action 3 directly, the x and y coordinates do not follow. This is due to the fact that actions are modified such that the

Table 4.2: SPADL Dataset Example

index	team_id	action_name	x	y	end_x	end_y	result_id
0	29	pass	10	20	15	20	1
1	29	carry	15	20	22	20	1
2	29	take_on	22	20	23	21	0
3	101	tackle	82	47	82	47	1
4	101	carry	82	47	85	24	1

team in possession of the ball is attacking towards the right side of the pitch. This is done to ensure that any analysis that takes place does not need to take which team is attacking towards which side into account. Since offline DRL requires the data to be structured in an episodic nature, the augmented dataset must be restructured to be a chain of actions that terminate when possession is lost.

4.1.2.1 Actions

The format shown in Table 4.2 is not directly compatible with the inputs required to perform DRL, they must be converted into purely numeric format. This is done by representing them using a vector of length 7. The first 5 elements of the array are a one-hot-encoded vector that represents the action taken at that particular time step. The last two elements of the vector represent the x and y locations of the destination of the action. The x and y coordinates are scaled such that the center of the pitch is mapped to $(0, 0)$, $(105, 68)$ is mapped to $(1, 1)$, and $(0, 0)$ is mapped to $(-1, -1)$. An example of an action being encoded using this technique is shown in Figure 4.2.

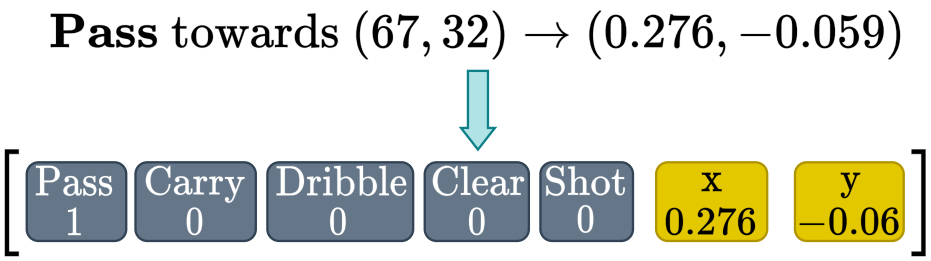


Figure 4.2: Action Vector Representation

By choosing this representation, the dataset will contain a structured numeric representation for each action type, that considers both the action that was taken, as well as

the location it is applied to.

4.1.2.2 Terminal Actions

In order to determine whether an action terminates the chain of possession or not, the outcome of the SPADL action can be observed. The SPADL schema contains definitions for the if the action was carried out successfully or not. This allows us to infer if the action was terminal or not. These definitions can be read from Table 4.3.

Table 4.3: Success flag per action in SPADL

Action	**Successful If**
Pass	Reaches teammate
Carry	Always successful
Take On	Keeps Possession
Clearance	Always unsuccessful
Shot	Goal is scored

A pass will cause possession to be terminated if it does not reach the teammate. Similarly, a take on will be terminal if it the player loses possession after attempting it, and a clearance inherently resigns possession to the opposition. Thus, in the case of a pass, take on, and clearance, the SPADL **result_id** can be used to directly indicate if possession was lost. In the case of shot, it was determined that the possession chain will be terminated after each shot. The carry action is always determined to be successful within the SPADL schema, thus to accurately determine if a carry causes possession to be lost, result of the carry within the SPADL schema is not used, and instead the following action and its outcome must be observed. In the case where a carry is followed by an action that interrupts the possession chain or hands possession to the opposition, the carry will be determined to have been terminal. This is usually due to tackles or fouls by the opposition, or a bad touch by the player in possession.

The final consideration that must be taken when considering if possession is said to be terminal is due to noise in the dataset. For a small percentage of the events in the dataset, a frame containing the location of the surrounding players is not available. In these cases where the sequence is interrupted, it is listed as terminal as well, however it is noted, and is not given a negative reward. The process by which this is carried out is visualised in Figure 4.3.

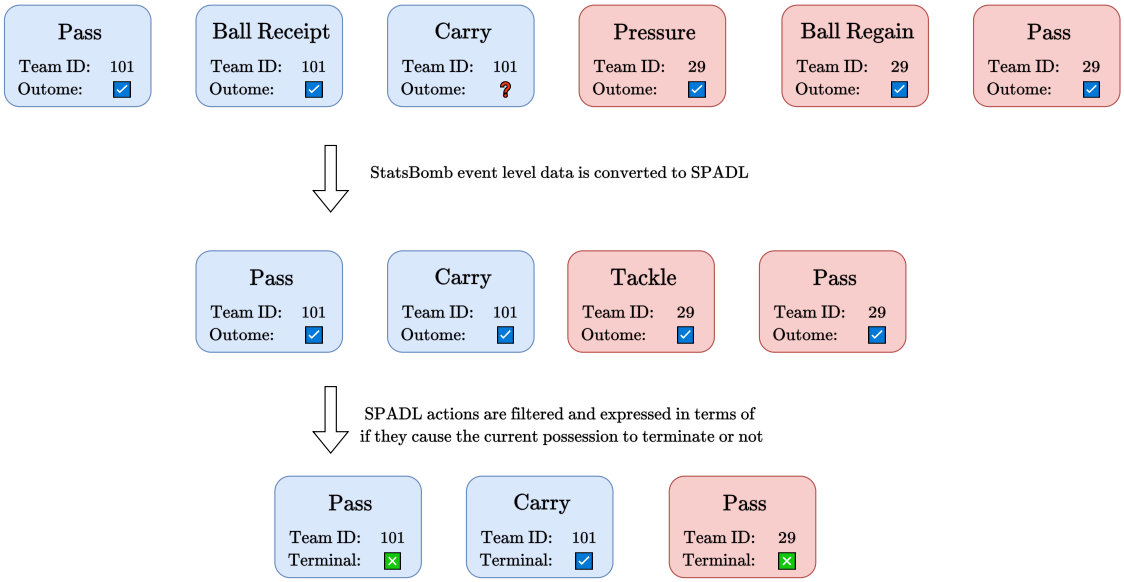


Figure 4.3: Converting the StatsBomb dataset to episodes of possession chains

4.1.2.3 Observations

The purpose of providing observations to the model is to provide the model the context that the action was performed in. For our purpose, we utilised the 360 Data that was provided with the research competition dataset. This contains tracking data that is automatically paired with the event data, obtained from the broadcast camera. An example of the data that is provided with this dataset is shown in Tables 4.4 and 4.5.

Table 4.4: Visible area

x	y
10	0
20	120
40	120
50	0
10	0

Table 4.5: Player locations

actor	teammate	goalkeeper	x	y
True	True	False	10	40
False	True	False	10	60
False	True	True	5	50
False	False	False	20	30
False	False	False	20	60

The visible area contains the polygon that describes the section of the pitch that is visible from the broadcast camera. The freeze frame describes the players that are visible within the freeze frame. The actor flag indicates if the player is the protagonist within the event with a matching id from the events dataset. The teammate flag indicates if the player is a teammate of the actor. The goalkeeper flag indicates whether or not the player is a goalkeeper, and the x and y flags indicate the location of the player on the pitch.

The data within this table can be visualised within Figure 4.4. The visible area polygon is represented by the blue polygon that indicates which area of the pitch is visible to the broadcast camera. The red players are the teammates of the player, whilst the blue players are the opposition players. The yellow player is the actor in possession of the ball.

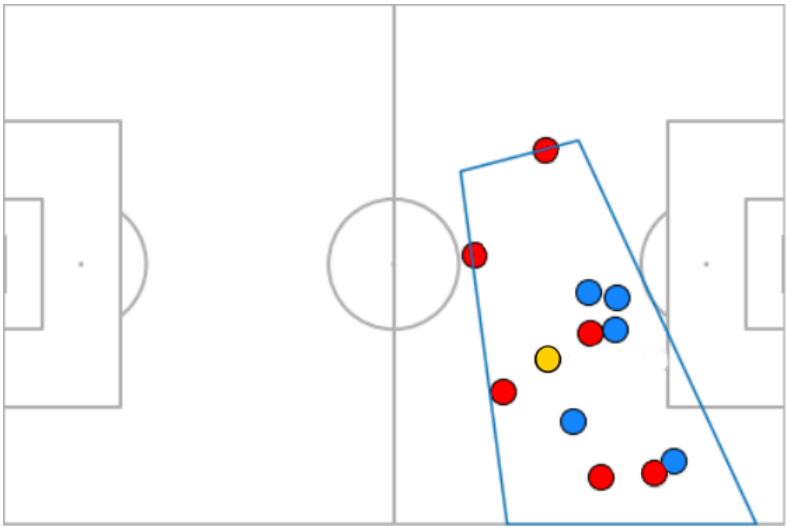


Figure 4.4: Tracking Data Example

To utilise this data to create an observation. Several representations were considered. The implementation that was used makes use of an image that has four channels. The first three channels contain images that correspond with the location of the actor, teammates and opponents respectively. The final channel of the image contains the output of a PCM. This is used as part of the observation to allow the model to also understand which areas of the pitch belong to which team. To generate this, the `scipy.spatial.Voronoi`[7] class was used to generate a set of polygons for each visible player, such that each polygon represents the area of the pitch closest to the player it encompasses. After obtaining the polygons, the `open-cv` [8] library was used to obtain these points and convert them into a PCM that can be used within a footballing context. The lines between neighbouring polygons were blurred to allow for a gradual transition of pitch control between areas that belong to the teammates and the opposition, instead of a hard boundary between them. The entire process is shown in Figure 4.5.

The `Voronoi` class outputs a set of polygons and lines, whereby the dotted lines are said to extend towards infinity in the same direction. This can be seen overlayed over the player coordinates to show how the Voronoi diagram successfully computes the

[7] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Voronoi.html`

[8] `https://github.com/opencv/opencv-python`

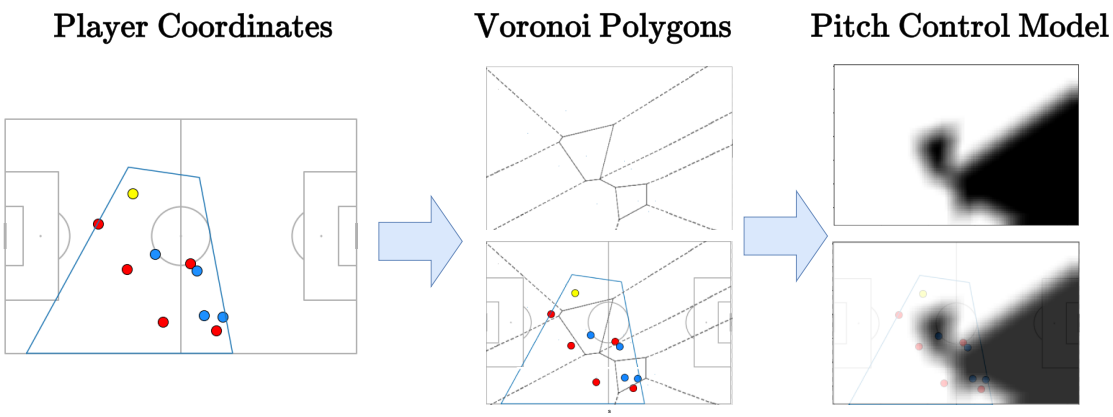


Figure 4.5: Obtaining the PCM

ideal polygons for each scenario. The Pitch Control Model in Figure 4.5 is the output of the `opencv` process, overlayed onto the player coordinates, with black areas of the PCM representing areas controlled by the opposition, and white areas of the PCM representing areas controlled by the teammates. The value itself for a particular action is extracted by accessing the pixel value (which will be a decimal between 0 and 1) of the PCM at the coordinates of the action's destination.

4.1.2.4 Reward Function

In our proposed reward function we combine two aspects; the value of the position on the pitch, and the likelihood of scoring from that position if the action is a shot.

Possession Value: To quantify the effect of player decisions, a PVM will be used to obtain a value for performing a certain action within the given context. Three options were considered.

- Expected Threat
- VAEP
- OBV

For this work, the OBV model was chosen, as the precalculated values are provided within the StatsBomb dataset, which make use of their xG and PSxG models which are trained using their state-of-the-art data.

Shot Value: To objectively value shot actions, the StatsBomb xG model will be used, which is provided as a pre-computed value for each shot within the research dataset. The PSxG is also provided as a precomputed value for each shot, however it was not used since the purpose is to reward shots that are taken from good positions rather than rewarding the player for their shot trajectories.

The definitions for the notation used in the following equations can be seen in Table 4.6:

Table 4.6: Definitions

Symbol	Definition	Range
a	Action	$\mathbb{R}^7$
s	Current state	$4 \times 105 \times 68$
s'	State following s after taking action a	$4 \times 105 \times 68$
$\mathsf{c}(s, s')$	Returns 1 if possession is retained between s and s', otherwise returns 0	$\{0,1\}$
$\mathsf{type}(a)$	Returns the type of the action	{pass, carry, take-on, clearance, shot}
$xG(s, a)$	xG value of shot a taken in state s	[0,1]
$PV(s, a)$	The value obtained from performing action a according to a PVM	[-1, 1]
$\mathsf{PC}(s, a)$	The value of possession at the end of a within state s according to a PCM	[0,1]

The reward function is defined as follows:

$$R(s, a, s') = \begin{cases} xG(s, a) & \text{if type}(a) = \text{shot} \\ R_p(s, a) & \text{if (type}(a) \notin \{\text{shot, clearance}\}) \wedge \mathsf{c}(s, s') = 1 \\ -n & \text{otherwise} \end{cases} \tag{4.1}$$

As is defined in Equation 4.1, if a is a shot then the reward value assigned for that action is equal to the xG obtained from the shot. This means that players will be rewarded for taking shots in proportion to the probability of scoring. In the case that action a is not a shot, the reward value depends on whether performing action a has caused possession to be retained or lost, represented by the function $c(s, s')$. If the possession of the ball is retained or the action is a clearance, the R_p function is used instead, where $R_p(s, a)$ is defined as follows:

$$R_p(s, a) = \begin{cases} \Delta PV(s, a) \times PC(s, a) & \text{if } \Delta PV(s, a) \text{ is positive} \\ \Delta PV(s, a) \times \big(1 - PC(s, a)\big) & \text{if } \Delta PV(s, a) \text{ is negative} \end{cases} \tag{4.2}$$

The aim of the function is to reward the model for increasing the value of possession. This is accounted for by the $\Delta PV(s,a)$ factor. If $\Delta PV(s,a)$ is positive, then the value of possession is increased by carrying out the action. However, we must also account for the context that the action is carried out in. This is done by multiplying the $\Delta PV(s,a)$ by $PC(s,a)$. The output of $PC(s,a)$ ranges from 0 to 1, where 0 corresponds with an area of the pitch that is controlled entirely by the opposition, and 1 corresponds to an area of the pitch that is controlled entirely by the teammates, according to our PCM. Thus, by scaling the $\Delta PV(s,a)$ by $PC(s,a)$, we are essentially scaling it back by how likely we are to retain possession, thereby rewarding an action less if it is more likely to cause possession to be lost.

If $\Delta PV(s,a)$ is negative however, the reward is calculated by multiplying $\Delta PV(s,a)$ by $1 - PC(s,a)$. Since $PC(s,a)$ will output a value close to 0 if the action is likely to lose possession of the ball, it will bring the value of $\Delta PV(s,a)$ closer to 0. However, in the case that $\Delta PV(s,a)$ is negative, then doing so would increase the reward for performing a negative action into a risky area of the pitch. By multiplying it by $1 - PC(s,a)$ instead, we are ensuring that if a negative action will only be reduced towards 0 if it is performed into an area of the pitch that is controlled by the team-mates. Thus, performing a conservative action that loses possession will not be scaled back. However, if an action with a negative $\Delta PV(s,a)$ is carried out into an area that is controlled by the opposition, it will be given the entire negative reward, since $1 - PC(s,a)$ will return a value close to 1. An example of this calculation being carried out is shown within Figure 4.6.

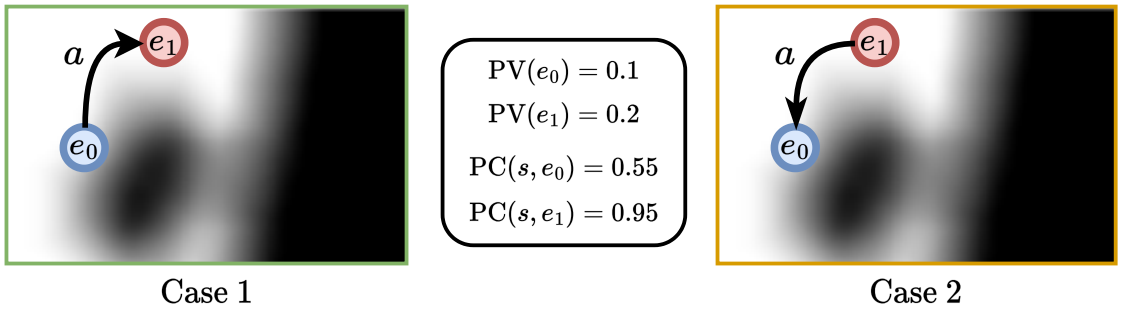


Figure 4.6: Obtaining the PCM

Considering Case 1 from Figure 4.6, the ball is progressing from e_0 to e_1. In this case $\Delta PV(s,a)$ is a positive increase of 0.1. Since the ball is being passed into an area that is mostly controlled by the teammates, illustrated in Figure 4.6 as e_1 is mostly surrounded by white, then the reward that is assigned is only scaled back slightly, which reflects that the gain in possession value is being made in a secure area of the pitch. When considering Case 2 from Figure 4.6, $\Delta PV(s,a)$ is a instead a -0.1. However, the ball is being played

$e_0 \rightarrow e_1$	$e_1 \rightarrow e_0$
$\therefore R(s,a) = \Delta PV(s,a) \geq 0$	$\therefore R(s,a) = \Delta PV(s,a) < 0$
$= \Delta PV(s,a) \times PC(s,a)$	$= \Delta PV(s,a) \times \big(1 - PC(s,a)\big)$
$= (0.2 - 0.1) \times 0.95$	$= (0.2 - 0.1) \times (1 - 0.55)$
$= 0.1 \times 0.95$	$= -0.1 \times 0.45$
$= 0.095$	$= -0.045$

Table 4.7: Reward computation for the two cases shown in Fig. 4.6

into an area that is mostly controlled by the teammates, according to the output of the PCM, which is also illustrated in Figure 4.6. Thus, the penalty is scaled back accordingly to reflect that the negative effect on possession value still presents a high chance of retaining possession.

If the possession of the ball is lost and the shot was not an action, then the agent will simply be rewarded with a negative constant, to deter behaviour that causes possession to be lost. Clearance actions are not penalised for causing possession loss however, since they are always the terminating action. Instead the R_p function is always used for clearances. Thus, by making use of both the PC and PV functions to calculate the reward, the model will be able to learn which actions lead to possession being moved to the most valuable areas of the pitch whilst also restricting itself to move the ball the to areas of the pitch which should not result in possession being surrendered to the opposition. The events that are labelled as excused within Section 4.1.2.2 are not considered to have caused possession to be lost, even if the action is terminal.

4.1.2.5 Episodes

Each state within an episode is represented as a 4 channel image that contains the location of the actor, teammates, opposition and a PCM that corresponds with the players position. The actions are represented as a vector of length 7 that encodes both the action type and its destination. The rewards are defined by considering both the output of both the PVM and the PCM, as defined in Equation 4.1, and the terminal flags are defined by whether or not the action terminated the possession chain. This will allow the augmented dataset to be partitioned into episodes, where each episode consists of actions from a single team. This is illustrated in Figure 4.7.

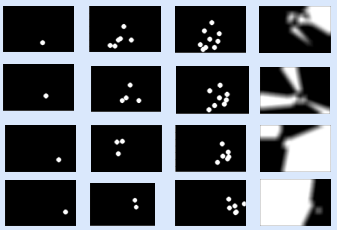
	Observations	Action Vectors	Rewards	Terminal Flag
Episode 1		$[0, 1, 0, 0, 0, -0.63 - 0.71]$	-0.01	0
		$[1, 0, 0, 0, 0, -0.40 - 0.82]$	-1	1
Episode 2		$[0, 1, 0, 0, 0, 0.26, 0.54]$	0.05	0
		$[1, 0, 0, 0, 0, 0.49, 0.69]$	0.03	0
		$[0, 1, 0, 0, 0, 0.74, 0.47]$	0.03	0
		$[0, 0, 1, 0, 0, 0.74, 0.47]$	-1	1

Figure 4.7: Sample from the augmented dataset

4.1.2.6 Evaluation

To evaluate the created dataset, the distribution of the computed DV for all actions will be analysed using heat-map visualisations, as is commonly carried out in the evaluation of football data and PVMs (García-Aliaga et al., 2021; Van Roy et al., 2020). The calculated dataset values will also be evaluated qualitatively against real world samples (Sotudeh).

4.2 O2: DRL model for Player Decision Analysis

Considering the nature of the observations and the actions, the model makes use of a continuous state representation, and continuous control for the action vector. The model must also make use of offline learning, as it will be trained on the dataset extracted from games that was obtained from the augmented dataset designed in the previous section.

4.2.1 Data Preparation

The data is loaded into `d3rlpy` by making use of the `MDPDataset` class that takes 4 arrays containing synchronised values corresponding to the observations, actions, rewards and terminals. The dataset contains data from the 2020/21 season, and the 2021/22 season, which will be referred to as Season 1 and Season 2 respectively. The training set and test sets were obtained by splitting the data from Season 1, where an 80/20 split was used accordingly using the `train_test_split` function from `sklearn`[9]. The data from Season 2 is used to evaluate the results of the model in the Evaluation section of the research, ensuring that it is not carried out on data that the model is trained on. Once the episodes are extracted from the dataset, they are also shuffled to reduce model bias.

4.2.2 Model Structure

To train the actor, critic and value functions using the aforementioned loss functions, `d3rlpy` encodes the image inputs by using a CNN. Each one uses the same CNN architecture, which is based on DQN (Mnih et al., 2015). The layers of the model can be seen in Figure 4.8.

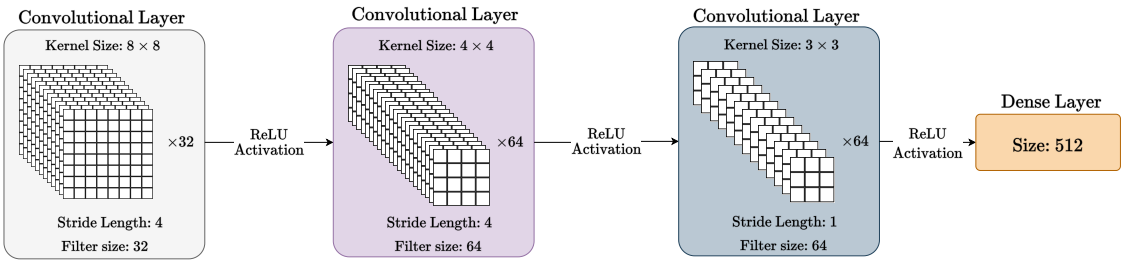


Figure 4.8: CNN Layers

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`

The training was carried out on a machine with an A100 GPU and 90GB of GPU RAM, on Paperspace's Gradient[10] service.

4.2.3 Choice of Algorithm

To select the algorithm that will be used within this work, the algorithms implemented in the `d3rlpy` library that are compatible with offline learning, as well as continuous state and action spaces considered as candidates. These can be seen within Table 4.8.

Algorithm
Conservative Q-Learning
Implicit Q-Learning
TD3+BC
Advantage Weighted Actor-Critic
Critic Regularized Regression

Table 4.8: Offline DRL Continuous Control Algorithms

Hyper-paramter tuning was performed on each algorithm using the Optuna library Akiba et al. (2019). This library was chosen to perform the hyper-parameter optimisation as opposed to GridSearchCV [11] or RandomSearchCV [12] due to its performance, and ability to prune the search space efficiently and converge onto the best possible values without needing to check all possible permutations as both of the other alternatives rely on a more manual process that requires a list of candidate values to be supplied, and both take longer to complete. A reduced version of the dataset containing games from Season 1 was chosen for training, which is an approach used when performing tuning on the entire dataset is too costly (Poloczek et al., 2017). The dataset was split into train and test episodes. Identifying the target of the optimization is a particularly difficult challenge in offline RL, as there is no exploration that can be performed with the learned action policy (Paine et al., 2020; Wang et al., 2021). Thus given the available data, three target objectives were set. This being the minimising of each algorithms respective actor and critic loss functions, as well as the minimisation of their TD-Error. The details of the configuration used to train the models can be seen in Table 4.9.

[10] https://www.paperspace.com/gradient

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

[12] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html`

Constant Hyper-Parameters		
Detail	**Value**	
Epochs	70	
Trials	25	
Episodes	16,000/4,000	
Tunable Hyper-Parameters		
Actor Learning Rate	1e-10	1e-7
Critic Learning Rate	1e-8	1e-5

Table 4.9: Optuna Hyper-Parameter Tuning Details

The hyper-parameters use stem from experimentation that was performed when starting off at the baseline values (Seno et al., 2021). The range for the actor learning rate was set to be lower than that of the critic learning rate (Gershman and Lai, 2020). The Optuna library works by accepting lower and upper bounds for values for certain hyper-parameters. The library will then converge onto the ideal values for these hyper-parameters. Initially the library will set the hyper-parameters randomly, however it will quickly identify which direction it should tune the weight in, thus avoiding having to attempt all the possible permutations. The 'trials' parameter indicates how many trials the library will attempt per algorithm. The range of values for the actor and critic learning rates was set after manual experimentation with the values. The best trial for each algorithm is decided by the set of outputs with smallest distance from the origin on the Pareto front. An example of such a point can be seen in Figure 4.9, drawn for the hyper-parameter candidates of the IQL algorithm. The colour of the points in the Pareto front diagram shown in Figure 4.9 shows the distance of each set of values of the target objectives if the were plotted in 3D Euclidean space from the origin. Each point in the Pareto front represents a tuple of values for the target objectives, such that none of the values in the tuple can be increased without affecting the other target objectives negatively.

4.3 O3: Player Decision Making Evaluation

To measure the decisions that are being made by football players using the model, we used both the qualitative and quantitative metrics defined below.

Obtaining a Decision Score: To evaluate a particular decision, the chosen and trained DRL algorithm was used. The game state was encoded as a vector and passed to the

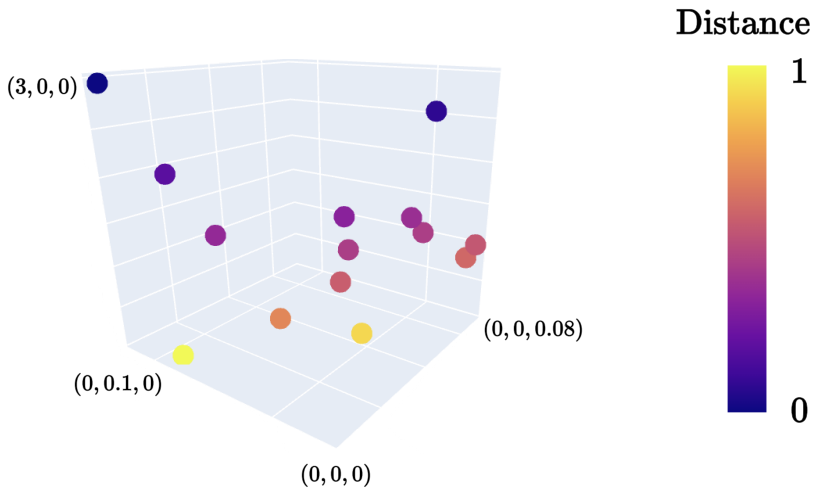


Figure 4.9: Example Pareto front.

`predict_value` function of `d3rlpy`. The function takes two inputs, the state s and the action a. The function will then return the expected return for the state-action pair. In doing so, we can compare the different possible actions that could have been taken within a particular state, and their effect on the expected return which we will refer to as the Decision Value (DV) obtained within that particular action.

4.3.1 Model Analysis

By providing different inputs to the model, especially hypothetical scenarios extracted from real world events, we evaluated the DV model's real world applicability. This can be carried out by obtaining a particular state s and defining several hypothetical actions that could be carried out, with different action types. This can be done by preparing several action vectors that represent the desired hypothetical action, and identifying their expected return within the selected state s. The resulting values are then used to observe whether the DV model provides outputs that entail logically with the desired goal of preventing possession loss and maximising the teams chance of scoring. This method for testing hypothetical actions is a common approach to evaluating model output (Rahimian et al., 2021, 2022).

4.3.2 Player Analysis

In this section, we will outline the different ways that the DV model can be used to evaluate individual player decision making. This will be done by describing the methodology

that will be used to achieve this, considering the model developed and the available data.

4.3.2.1 Analysis by Position

The first analysis makes use of the DV model to order football clubs by their players' decision making over the 2021/22 season from within the StatsBomb research dataset. This data was not provided to the DV model during training. The only inputs given to the model are those stated within Section 4.1 and thus, the actions within the dataset are effectively anonymised, both with respect to the player making them, as well as the team they play for. This will allow the analysis to be unbiased and objective.

The average DV obtained by each team can then be compared with the other analytical models that are typically used to analyse team performance, such as xG, xGF, xGA and the points obtained, as defined in Section 2.4. It can also be compared with other PVMs such as xT. It will not be compared with VAEP as the VAEP model requires us to know if the action terminates in a successful or unsuccessful state, whilst the OBV model is not made public. This will enable us to compare the explanatory power of the different metrics with regards to the league table.

4.3.2.2 Analysis by Action

The actions that players take can also be grouped by their type (Decroos et al., 2020; Van Roy et al., 2020). This will allow us to identify which players are obtaining the most DV through passes that move the possession into more threatening areas, or which players have the highest disparity in DV obtained between take-ons and passes. This will also allow us to analyse defending players and attacking players in more detail, as the DV obtained from clearances and shots can be analysed to identify which players are making the best decisions in these key moments.

4.3.3 Team Analysis

We will outline the different ways that the DV model can be used to evaluate an entire team by observing the DV obtained by their individual players. This section will describe the approach that will be used, taking into account the model created and the data at hand.

4.3.3.1 DV League Table

The DV model will be used to order the teams in the 2021/22 Premier League season by the mean DV they received for actions that their players carried out during each game.

The ordering obtained will be compared with the actual order obtained when sorting by points achieved, as well as the order obtained by using other metrics such as xG and OBV. The mean DV is used instead of the total, as otherwise teams would obtain higher values for simply performing more actions and simply retaining possession for the longest possible period. In league football, retaining possession of the ball for long periods is not correlated with higher win probabilities (Collet, 2013; Gronow et al., 2014).

4.3.3.2 DV By Zone Per Team

By analysing the average DV obtained by each team per zone, it will be possible to identify for which zones, each team is performing below the league average. In doing so, it will be possible to identify the zones of the pitch that teams are excelling in, or other zones that teams need to improve. This approach is also commonly carried out in football analytics (Singh, 2018; Van Roy et al., 2021). This can also be done to predict which zones of the pitch require investment. Within football analysis, the pitch is not split into standard zones, apart from those indicated by the line markings. Thus dividing the zones is typically done based on the desired task at hand, as different tasks require the pitch to be partitioned in different ways. One common way in which the pitch is split is into 18 different zones (Kim et al., 2019). For this work, we divided the pitch into 9 different zones, that can be seen within Figure 4.10.

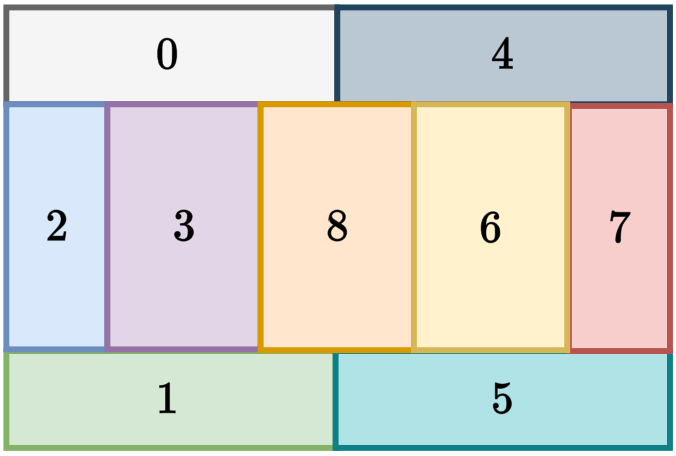


Figure 4.10: Pitch Zones

5 Results and Evaluation

In this chapter, the results obtained from the experiments that were outlined in Chapter 4 will be shown and discussed. This includes discussing the augmented dataset created to be compatible with offline RL. It will also discuss the results used to choose the ideal algorithm, and the results of the training process. Finally, it will contain the results of the obtained DRL model that will be used to evaluate decision making at both an individual and team level to illustrate the model's applicability as a performance evaluation tool.

5.1 O1: Augmented Dataset

Throughout this section, the results obtained using the methodology outlined in Section 4.1 to create an augmented dataset will evaluated. This included the use of a 4 channel image as the observation, the use of a vector of length 7 to represent the actions, the reward function and the flag that determines when an episode is terminated. This will be done by first analysing the resultant dataset with respect to the statistical trends present within the augmented dataset, which will be followed by comparison with real match footage that will highlight the explanatory power offered by the format chosen to create the dataset.

5.1.1 Observation and Action Analysis

To illustrate the distribution of the actions within the augmented dataset, the actions performed by the players within Season 2 were plotted on a heatmap for each action. The heatmap contains the destination for each action itself. The heatmaps for the *Pass*, *Carry* and *Take On* actions can be seen within Figure 5.1.

In the first heat-map pertaining to passes, it can be seen how the most common destination for passes to be made is the left or right side of the pitch, towards the opposition's half. The same is true of carry actions. This is probably due to the fact that defending teams are happy for their opposition to circulate possession in these areas. However, as

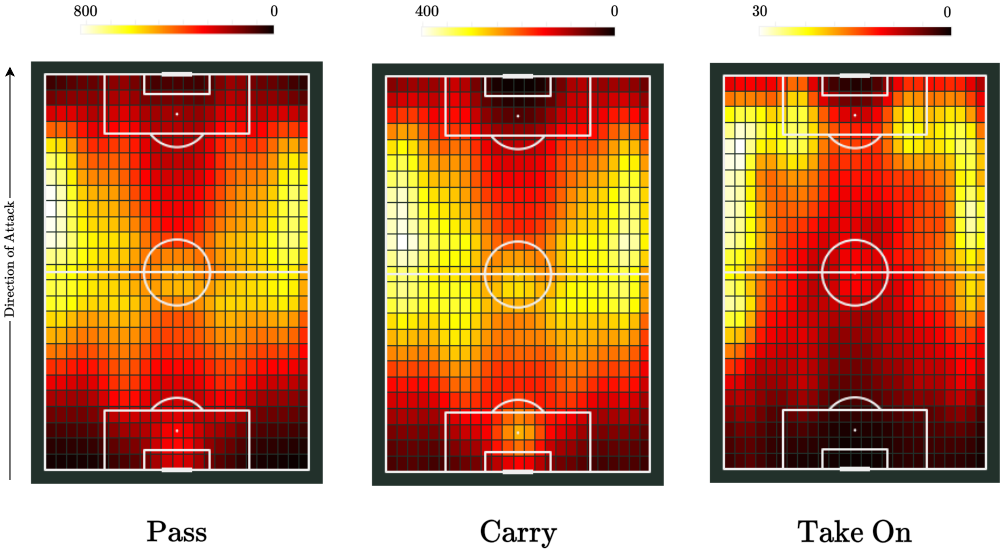


Figure 5.1: Heatmap of action destination for Pass, Carry and Take-On actions

they approach more valuable areas of the pitch closer to the centre of their oppositions half or towards the penalty area, defending teams start to actively restrict passes or carries from being made into these areas. This area of the pitch is commonly referred to as the Position Of Maximum Opportunity (POMO), a term that originates from English Football Association member Charles Hughes (Robson and Hayward, 2006). The term refers to the fact that actions that move possession into or around the oppositions penalty area will increase the team's likelihood of scoring. The defending teams' aversion to allowing their opponents to enter the POMO can be observed in Figure 5.1.

Figure 5.2 contains the destination of clearance and shot actions. It can be seen how the majority of clearances leave the pitch at the extremities of the opposition's defensive half towards either corner flag. Clearances also commonly land just in front of the opposition penalty area. It is also interesting to note the number of clearances that land close to the teams' own penalty box, which is typically considered to be a dangerous area to surrender possession, indicating that a number of clearances are made in critical situations, where the destination is not as important as stopping the threat posed by the opposition.

To illustrate the granularity allowed by the format chosen for the augmented dataset, a sequence of images were obtained from a match between Manchester United and Tottenham Hotspur. The images are shown alongside their respective augmented dataset entries, as well as the corresponding action vectors, which can be seen in Figure 5.3.

The areas of the pitch that contain team mates and opposition players can be seen in

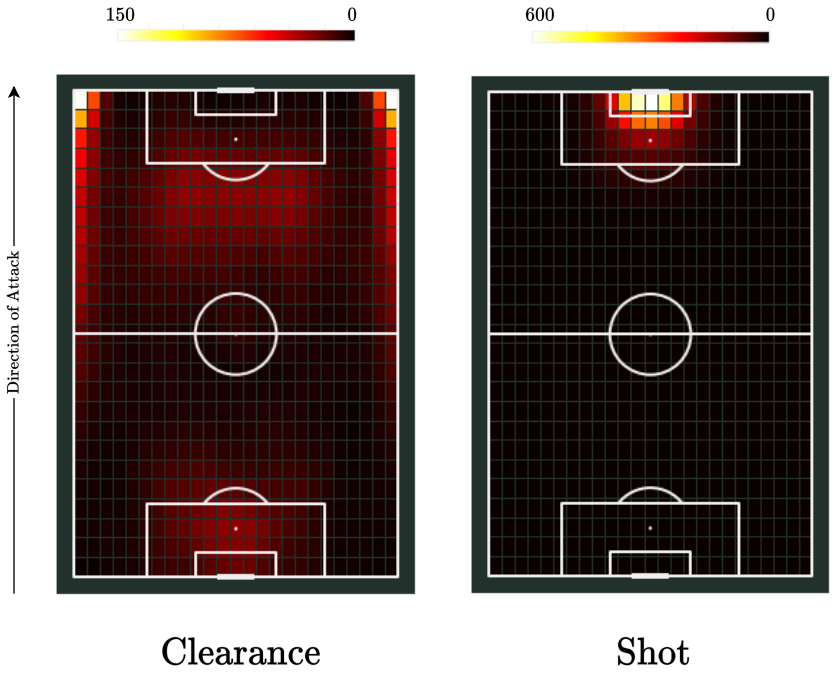


Figure 5.2: Heatmap of counts of actions destined to each bin for Take On and Shot actions

white circles. The PCM corresponding with this first action can also be seen below the image. In the first action, the PCM contains a white cone on the right side of the pitch, indicating that should a ball be played into the area, the team mate would be expected to retain possession of the ball. This observation is also made by the player in possession of the ball, who plays a pass into the space. This action is represented in the action vector by a value of 1 at the index corresponding with a pass, and the values of 0.65 and -0.54 at x and y respectively. The values indicate that the pass is played at 65% of the way into the opposition's half, and played 46% of the way into the pitch when starting from the top horizontal side line and aiming towards the bottom horizontal side line.

The next action is an immediate pass. The PCM indicates that the area where a team-mate can safely place the ball is quite small, thus requiring high technical skill. The x and y values show that the pass is placed 86% of the way into the opposition half, and the value of 0 for the y indicates that the pass is placed precisely half-way between the top and bottom horizontal lines of the pitch. The final action within the sequence is a shot. This is represented by a value of 1 at the corresponding index of the action vector. The x value of 1 indicates that the shot is made at the right side of the pitch. The y value of -0.02 indicates that the ball is shot 2% percent of the way from the middle of the goal

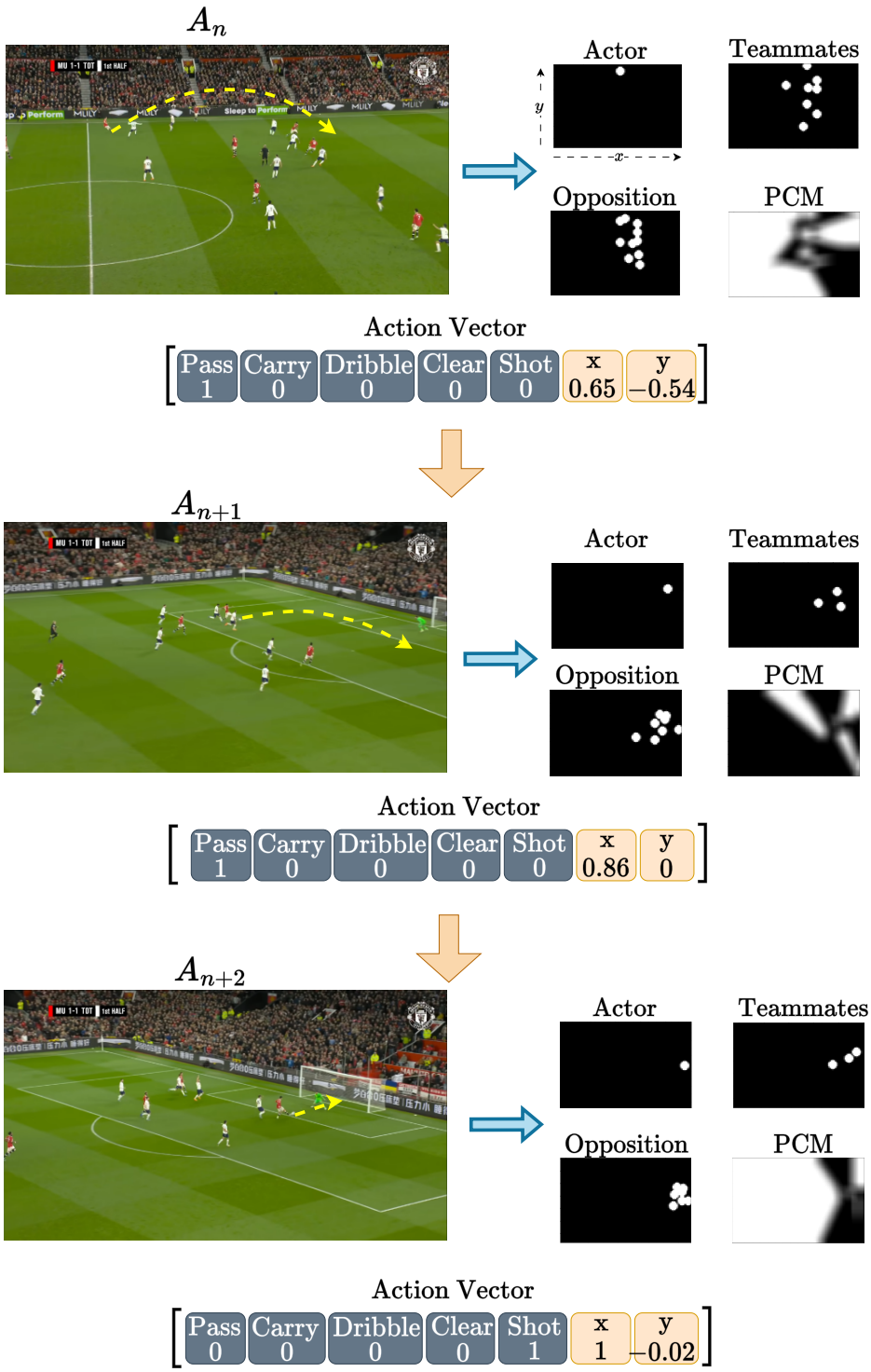


Figure 5.3: Augmented Dataset compared with real match photos

towards the left side of the goal, thus indicating that it's almost exactly at the centre of the pitch.

The examples show how the augmented dataset is able to represent a sequence of games with great explanatory power. When contrasted with alternatives such as using a discrete action space to represent an action's category, or using a grid or tile-based, layout would lose the granularity of the actions being performed. Using this representation, the exact location of the visible players and teammates can be used.

5.1.2 Reward Analysis

This section will analyse the outputs of the reward function R defined in Equation 4.1. This will be done to illustrate the output of the reward function when compared with real world actions, to illustrate the validity of the precomputed rewards provided to the model. The distributions of the outputs of R for each action type was calculated, and can be seen within Figure 5.4. The outliers and extreme values were excluded from the graph to allow for the granular distributions to be analysed effectively.

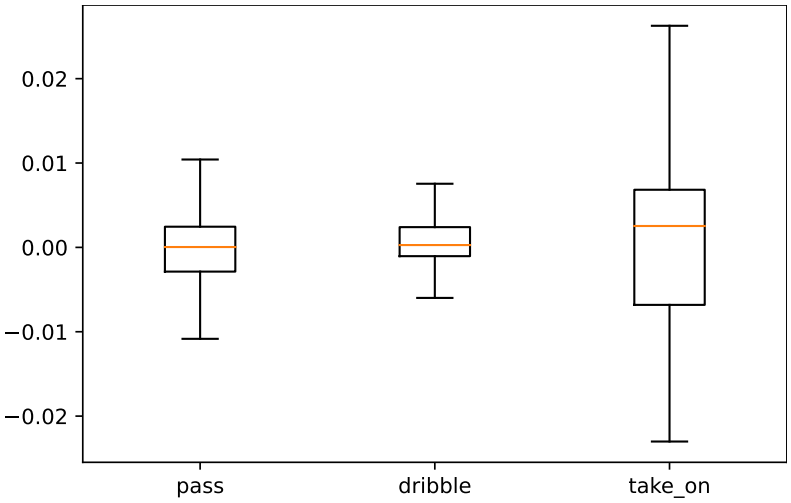


Figure 5.4: Boxplots of R for Pass, Carry and Take-on

The pass and carry actions received similar distributions and median values of R, This could be due to the fact that a large portion of these actions do not greatly affect the value of possession. Passes and carries are mostly performed without pressure from the opposition players, thus signifying that there is a lower risk associated with each action. The take on boxplot offers interesting insight as to the range of the R given to take on actions. The median R is considerably higher than that of passes and carries, yet the range of R values is far greater. This can be explained mostly by the nature of the action, as it is a

high risk action that can either result in possession being moved towards a more valuable part of the pitch, or to possession being surrendered immediately. Take on actions are also commonly performed when the player with the ball at their feet is under pressure, thus making the action more high risk. Figure 5.5 shows the R for clearance and shot actions. These are both actions performed within critical moments during a game, and the results show that the range of R given for shot actions varies significantly more than that of clearance actions. The data shows that most shot actions received a negative R. To obtain further understanding into the R given, a few decisions made by players from Season 2 have been selected and will be discussed in further detail.

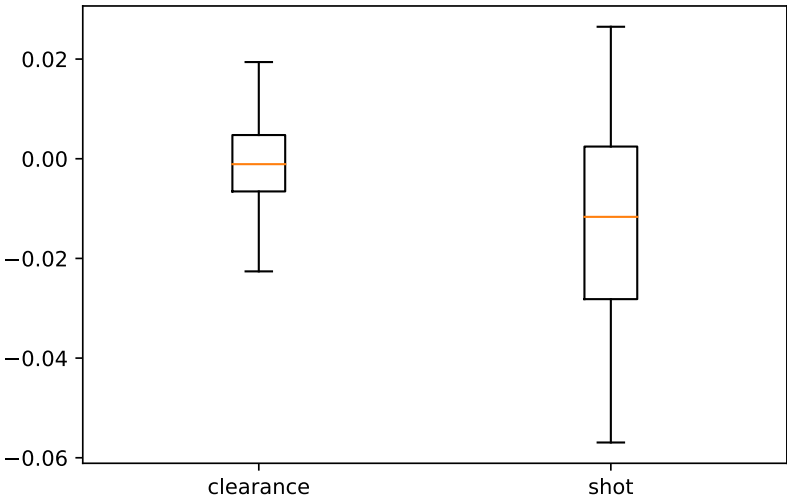


Figure 5.5: Box-plots for Clearance and Shot actions

Action: Pass
Reward: 0.003
Terminal: 1

Action: Pass
Reward: -0.0002
Terminal: 0

Action: Carry
Reward: 0.041
Terminal: 0

Figure 5.6: Actions with corresponding R values

The three actions that can be seen within Figure 5.6 are examples of actions that took place within Season 2, that consist of two pass actions and a carry action, and in each example, the player in possession of the ball is attacking the goal that is on the left side of the pitch. R is the product of two factors, being the PVM value and the PCM value. For these figures, the reward shown is the one calculated using R_p in Equation 4.2 before the -1 is assigned in the case of non-shot actions. The first pass action made by Bruno Fernandes moves the ball towards the opposition penalty box, thus the pass action is valued highly by the PVM component. Initially, the R given for this action is only of 0.003. This is due to the fact that the PCM supplies the context, where the positive value is scaled back towards 0 since it is being played into an area that has significant chance of surrendering possession. The corresponding terminal flag shows that the action did in fact cause possession loss, thus the final R given would be the negative constant -1.

The second action, which is a backwards pass made to Fred, receives a R value of -0.0002. The pass is rated as a negative action by the PVM, since it is moving the possession to a less valuable zone by moving the ball further away from the opponent's goal. The PVM does not take the location of the surrounding players into account however. The context provided by the PCM ensures that the R is scaled back from a highly negative value and back towards 0. The third action shown within Figure 5.6 shows a carry action that is made directly towards the opposition penalty area, towards an area of the pitch that is not crowded by opposition players. This is reflected in the R score that is given, as it is the highest value assigned from the chosen sample actions.

These actions show how the R values provide a valuable starting point for the DRL model to train on, as the combine both the value of moving possession of the ball to the particular action on the pitch for different action types, as well as the context that needs to be taken in to account when considering the value of the actions. To visualise the R in the case of shot actions, three examples of different shots taken during Season 2 were captured and shown in Figure 5.7.

In the case of shot actions, the R is equal to the xG of the shot being taken, as was

Figure 5.7: Shot actions and corresponding xG (=R)

defined in Equation 4.1. In the first example, the Leeds United right-back Luke Ayling receives the ball outside of the opposition's penalty box. The player decides to shoot and receives an xG value of 0.02, according to the model provided by StatsBomb. This means that a shot taken in those circumstances is only expected to be scored 2% of the time. Thus, in this case deciding to shoot would not be advisable. In the second shot, Kai Havertz has received the ball inside the opposition penalty box, and decides to shoot. The shot only receives an xG value of 0.09 in this case. This might be considered counter-intuitive when considering the scenario at face value. However, when the position of the defending players is taken into account, especially the goalkeeper who has left the goal-line and approached the shooting player to narrow down the percentage of the goal that is visible to the striker, the xG value accurately represents the difficulty associated with this chance.

In the last example, a shot is made by Cristiano Ronaldo for Manchester United, after possession is surrendered erroneously in a vulnerable position by Norwich's defender, after being pressured by another Manchester United player. This leads to an easy chance that was scored by Ronaldo, which was given an xG of 0.75. Thus it can be seen through these three examples how the xG value, which is analogous to the R value in this case, is an appropriate measure that will allow the model to value shot decisions within their respective states accordingly.

5.1.3 Terminal Action Identification

The terminal flag indicates an action resulting in the loss of possession, and thus the termination of the episode, as described in Section 4.1.2.2. Table 5.1 shows the percentage of each action that resulted in a terminal state.

By definition, shot actions have a 100% chance of termination. The actions that risk possession loss the most apart from shots are clearance actions. Within Season 2, all actions that had the clearance type were observed to have caused possession loss, thus being a terminal action. The most safe action categories and pass and carry actions, which

Action Type	Terminal
Pass	45%
Carry	13%
Take On	66%
Clearance	100%
Shot	100%

Table 5.1: Percentage of actions which led to a terminal state

is to be expected. Take on actions are the most likely to cause termination, apart from shot and clearance actions.

Within Section 4.1.2.3, the underlying assumption being made is that the values obtained from the PCM are valuable features that can be used to effectively predict how likely an action is to be terminal. This would be done by obtaining the PCM value at the coordinates of the destination of the current action, where values closer to 0 would indicate that the area is more likely to be controlled by the opposition, and areas close to 1 are more likely to be controlled by the teammates. Thus, the assumption is that actions that receive PCM values closer to 1 are more likely to retain possession, and actions that receive values closer to 0 are more likely to cause possession loss. To challenge the assumption, data from 10,000 randomly selected actions was obtained. Figure 5.8 shows the PCM values for terminal and non-terminal actions.

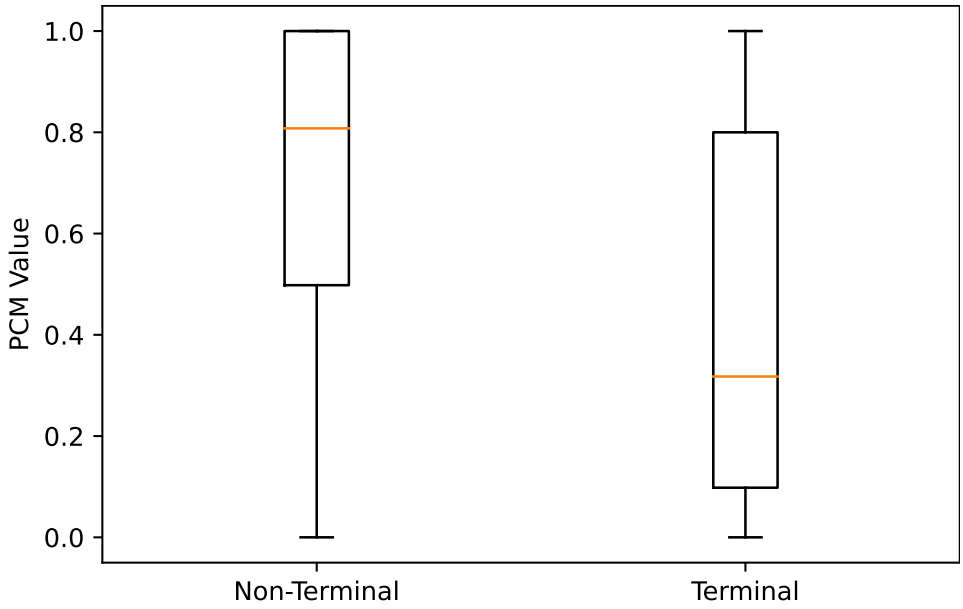
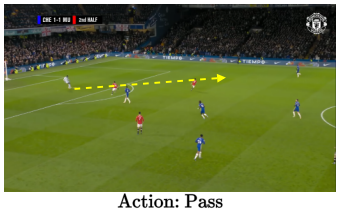


Figure 5.8: PCM Values grouped by Terminal Flag

The data shows that the median value obtained from the PCM at the destination coordinates for actions that were non-terminal was roughly 0.8, with the inter-quartile range being between 0.5 and 1. For terminal actions however, the median PCM value was found to be significantly lower at 0.38, and the inter-quartile range varied between 0.18 and 0.8. This shows that the PCM developed using the methodology outlined in Section 4.1.2.3 can be used as a valuable feature that can model whether the possession can be lost if the ball is played into areas that receive a low PCM value, and vice-versa. For further context, three examples chosen from Season 2, each containing actions that were determined to be terminal can be seen within Figure 5.9.

Action: Clearance
Reward: -0.07
Terminal: 1



Action: Pass
Reward: -0.02
Terminal: 1

Action: Take On
Reward: -0.03
Terminal: 1

Figure 5.9: Terminal action examples

The first image contains an example of a terminal action that is caused by a clearance. The clearance initially receives a relatively large negative R value of -0.07. This is due to the fact that the ball is not cleared far away enough from goal, as the clearance roughly lands at the same location that it was made from. This leads to possession being lost within the penalty box, which is an undesirable outcome, as well as a terminal one. The second example shows the Chelsea FC goalkeeper Edouard Mendy attempting to pass the ball to a teammate. The decision is a poor one, however, as the pass is played directly at an opposition player, surrendering possession of the ball and thereby making the action a terminal one. In the final example the defending player decides to perform a take on action within his own team's defending penalty box. The action is unsuccessful, as the player loses possession. This is reflected in the fact that the output of R is initially set to -0.03 and it is correctly identified as terminal, thereby making the final reward given to be -1.

5.2 O2: Model Training

This section outlines the results obtained when testing the performance and computational cost of each DRL algorithm available within the d3rlpy library, how this lead to a main algorithm being chosen, and the results obtained when training the algorithm at length. The tuples closest to the origin obtained from the hyper-parameter tuning for each algorithm can be seen in Figure 5.2.

Algorithm	Best TD-Error	Critic Loss	Actor Loss	Avg. Compute Time
CRR	0.07	0.07	0.03	17,233s
TD3+BC	0.11	0.23	0.107	7,635s
AWAC	0.082	0.157	254,997	7,255s
IQL	**0.06**	0.135	5.71	10,851s

Table 5.2: Optuna Hyper-Parameter Tuning Results

Most of the algorithms that were attempted achieved adequate results in at least two of the target objectives. However, some were not viable due to the time consideration, and others were not considered due to their non-convergence on the actor loss. Thus, the three algorithms that were considered at the end of this process were the IQL, AWAC, and TD3+BC algorithms. They were set to train with the hyper-parameters whose target objective values had the smallest euclidean distance from the origin. A chart containing the actor and critic losses for the TD3+BC algorithm can be seen within Figure 5.10.

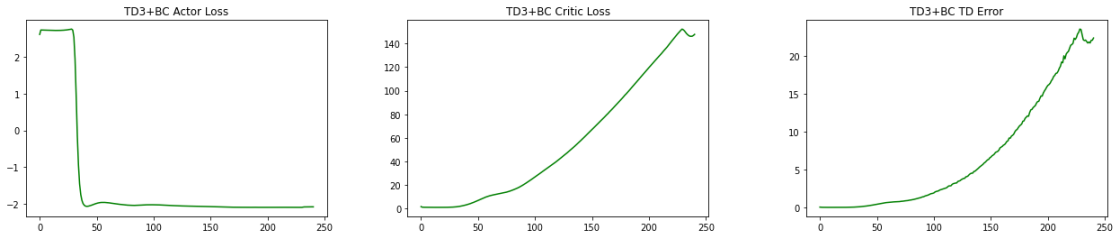


Figure 5.10: Train Losses and Test TD Error for TD3+BC Algorithm

The results in this figure show that the TD3+BC algorithm starts to over-fit heavily. Thus, the IQL and AWAC algorithms were chosen as the candidate algorithms for within this work. Whilst further experimentation into the ideal parameters to use for all the algorithms could be performed, the methodology outlined above to select the AWAC and IQL algorithms was deemed to be satisfactory considering the results obtained in this section and the following, especially when taking into account the financial resources

required to continue to perform such tests. To identify the final algorithm, the results obtained in this section must be framed within a footballing context.

5.2.1 Choice of Algorithm

The IQL model was trained for 781 epochs, with the critic and actor learning rates set to 3.4×10^{-7} and 9.8×10^{-8} respectively. The AWAC model was trained for 714 epochs with the critic and actor learning rates set to 1.5×10^{-5} and 9.7×10^{-8} respectively. All other parameters were left to their default values. Both were trained on all of the training dataset split obtained from Season 1, which was a dataset of 81,670 episodes. The test dataset split had 20,418 episodes. The model training was stopped when the losses stopped decreasing. The final hyper-parameters used for the model are obtained directly from the process carried out in the previous section. The charts of the actor and critic losses on the dataset containing games from Season 1 for both algorithms can be seen in Figures 5.11 and 5.12.

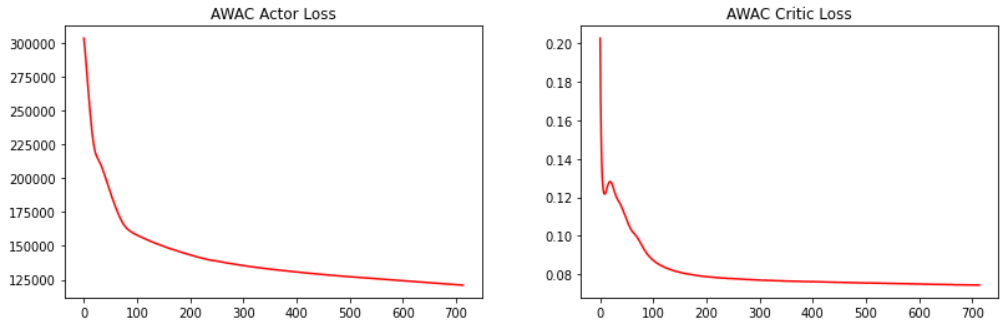


Figure 5.11: Training Losses for AWAC Algorithm

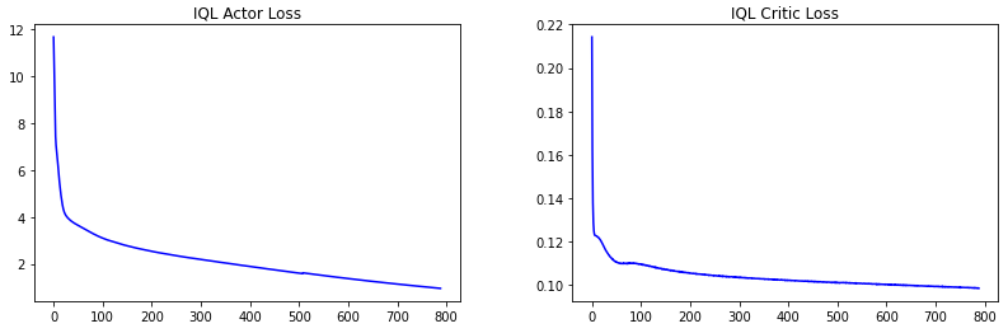


Figure 5.12: Training Losses for IQL Algorithm

The results show that the actor and critic networks for both algorithms are able to learn, and that they do not over-fit. The charts in Figure 5.13 contain diagrams showing

the TD-Error of both of the algorithms. The results show that both policies are converging.

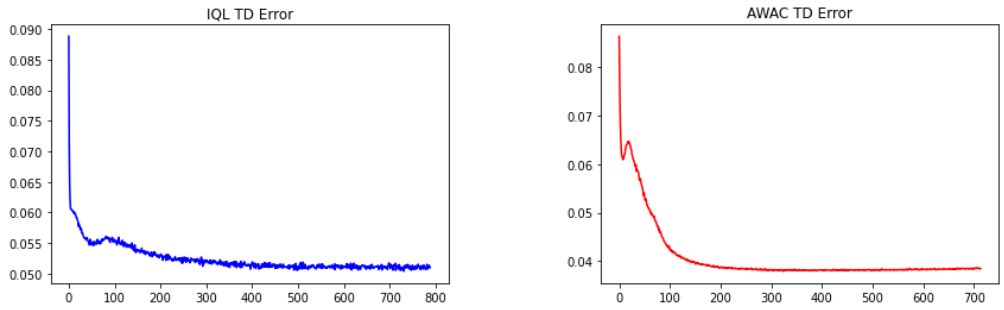


Figure 5.13: Test Dataset TD-Error for IQL & AWAC Algorithms

In order to identify the ideal model for use within this section, a scatter plot containing the points achieved and mean DV obtained for the player decisions can be seen in Figure 5.14. The mean DV is used instead of the total since using the total would result in teams obtaining higher values for simply retaining possession, which while useful, is not a direct indicator of decision making quality.

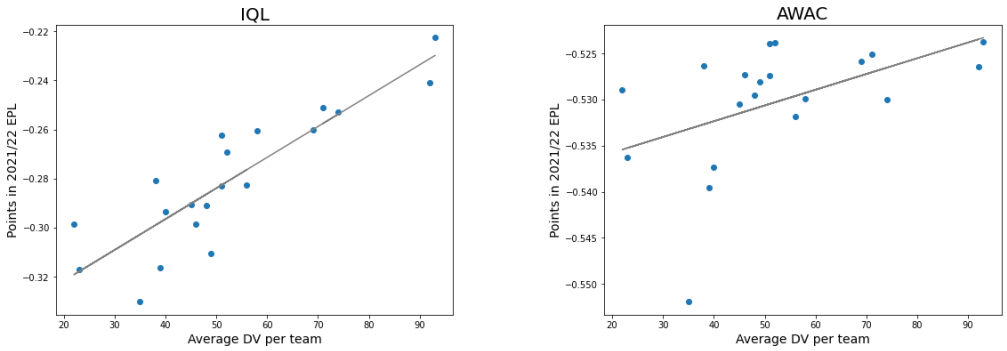


Figure 5.14: Algorithm Explanatory Power

The two charts indicate that using the mean DV obtained for each team from IQL model are highly correlated with the actual total points achieved by the teams during the second season, which contains unseen data. This shows that the DV obtained using the IQL model has better explanatory power at describing the relationship between decision making and final league position than the AWAC model. Thus, the IQL model will be used as the model of choice within the remainder of this chapter.

5.2.2 Conclusion

The results shown in this chapter confirm that the methodology defined for creating a dataset in Section 4.1, which was then trained using the methodology defined in Section

4.2.3 following the hyper-parameter tuning described in this section allowed for both the actor and critic networks to converge to networks that are aligned with the reward function defined within the augmented dataset. Thus, Objective O2 defined within Section 1.2 has been met in a satisfactory manner.

5.3 O3: DRL Model For Player Analysis

Within this section, the IQL model trained in Section 5.2.1 will be used to evaluate the decision making of football players within Season 2 of the dataset provided to tackle O2 within Section 1.2. This data was not used for training, thus the results from the models are unbiased. To obtain further understanding of the output of the DV model, the relationship between the chance that playing the ball into a particular area of the pitch resulting in possession loss was compared with the mean DV associated with that part of the pitch. This was done to observe if the model's valuations were simply mirroring possession loss, or if more nuanced detail was learned. To achieve this, the pitch was split into granular bins (25$\times$ 25), resulting heatmaps are shown within Figure 5.15.

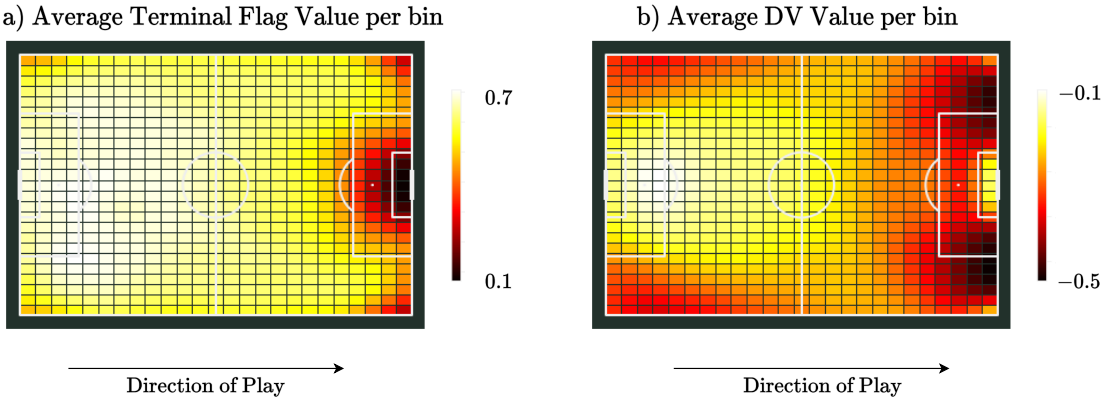


Figure 5.15: Mean DV by action destination

Figure 5.15a shows the areas of the pitch associated with a high likelihood of the action in that bin being terminal shown in darker colours, whilst lighter colours show the areas of the pitch that have low average values for the terminal flag, indicating the chance that possession is lost when the ball is played into the area in darker colours. Figure 5.15b shows the mean DV obtained for actions where the marked zone is the destination of the action is shown. Thus, the second heatmap shows the expected return for a ball that is played into each bin. Intuitively, the heatmaps show that keeping the ball within the team's own defensive half corresponds with lower risk in general. Whilst the two heatmaps are similar, they are also distinctly different in key areas. Playing the ball closer to either touch-line can be seen to have a relatively low mean DV. This is due to the fact that balls played into these areas are not likely to cause possession loss themselves, however they place the player with possession of the ball in an awkward position that causes possession to be lost within the next few actions. Similarly, higher mean DV values are given for actions made within the opposition half when compared with the relatively

low likelihood of immediate possession loss indicated by the average terminal flag value. Within the opposition's penalty area, the DV model learns to value actions highly the closer they get to the opposition goal as they increase the value of possession, as opposed to the terminal actions heatmap, which shows darker values for moving closer to the opposition goal.

This analysis shows the difference between simply valuing actions for how likely they are to terminate possession versus the actual output from the DV model, confirming that the DV model has learned to go beyond simply estimating the likelihood of possession loss.

5.3.1 Team Performance Prediction with DV

Table 5.3 shows the total number of points achieved, the PSxG conceded minus the goals allowed (PSxG-GA), and the mean DV obtained by each teams for decisions made within the 2021/22 season. By ordering the teams by the mean DV obtained throughout the season, we can determine which teams are making the best and worst decisions, on average. The PSxG values conceded by each team were also obtained for each team[1]. The PSxG-GA metric will allow us to analyse the goalkeeping performance of each team, and the quality of shots that each team has faced, as the difference indicates how many fewer goals the tam should have conceded (positive difference) versus how many less they should have conceded (negative difference).

The mean DV correctly predicts the best performing team in the league that season, Manchester City, who went on to win the competition. When comparing the mean DV obtained throughout the season in comparison to the actual league table finish, 11 out of 20 teams were guessed within an error of 1 place with respect to the actual ranking according to the points they obtained throughout the season, with 6 of the 20 teams being guessed correctly. Further analysis was carried out to identify the DV model's explanatory power with the data that it is trained on. The Spearman correlation between each metric and the final points obtained by each team can be seen in Table 5.4.

When considering that the DV model is trained on a reward function that operates on xG and OBV, of which OBV is only trained using event data, the DV metric's ability to predict points more accurately than the metrics used to train it suggests that the added context is crucial for evaluating player decisions and how they affect their teams' performance.

When analysing the difference between points gained and mean DV with respect to segments of the league table, two trends emerge. The DV model is able to order the top

[1] https://fbref.com/en/comps/9/2021-2022/keepersadv/2021-2022-Premier-League-Stats

Team	Points	PSxG-GA	Avg DV	Avg DV Order	Actual Order	Delta
Man City	93	-0.2	-0.22	1	1	0
Liverpool	92	1.6	-0.24	2	2	0
Spurs	71	-1.4	-0.25	3	4	-1
Chelsea	74	-0.4	-0.25	4	3	+1
Arsenal	69	0.6	-0.26	5	5	0
Man Utd	58	1.1	-0.26	6	6	0
Brighton	51	-1.5	-0.26	7	9	-2
Leicester	52	-1.1	-0.27	8	8	0
Leeds	38	-15.6	-0.28	9	17	-8
West Ham	56	-1	-0.28	10	7	+3
Wolves	51	7.3	-0.28	11	10	+1
Aston Villa	45	-5.5	-0.29	12	14	-2
Crystal Palace	48	-5.5	-0.29	13	12	+1
Southampton	40	-7.8	-0.29	14	15	-1
Norwich	22	-14.4	-0.3	16	20	-4
Brentford	46	-4.6	-0.3	15	13	+2
Newcastle	49	-7.7	-0.31	17	11	+6
Everton	39	-5.1	-0.32	18	16	+2
Watford	23	-13.7	-0.32	19	19	0
Burnley	35	-3.5	-0.33	20	18	+2

Table 5.3: Table comparing mean DV and actual total points

Total xG	**Total xGD**	**Mean DV**	**Mean OBV**
0.84	0.86	**0.87**	0.78

Table 5.4: Spearman Correlation between points gained and analysis models

teams accurately, however the lower half of the table shows lower levels of predictability. This could show that the distinction in the quality of decision making between teams that finish in the top rankings in the Premier League is higher than the difference in decision making within the lower places in the league, which means that these results are more likely to be affected by factors and events that do not take place when the team is in possession of the ball, such as the defensive structure, the teams pressing abilities or the team fitness levels.

Further insight into the DV model's team ranking can be drawn by analysing the cases with the largest difference between predicted and actual ranking. The first such case would be of Leeds United. They finished 17th in the league, narrowly escaping relegation. However, their mean DV suggests that they should have finished in the top half of the table. The Post Shot xG (PSxG) model can be used to evaluate the goalkeeper shot

stopping ability. Traditional xG models work by evaluating the scenario within which the shot is taken, to allow obtain a probability that a shot taken within said scenario is converted into a goal. In comparison, the PSxG model takes the location within the net that the ball arrives at after the shot is taken.

Thus, shots that end up within the top right corner of the net will receive a higher PSxG than shots that end up in the middle of the goal, which is easier to reach for goalkeepers. By evaluating the difference between the PSxG and the actual goals conceded by a football team, we can measure the shot stopping ability of goalkeepers. In the case of Leeds United, they obtained a PSxG minus Goals of -15.6. This means that over the course of an entire season, considering data obtained after the shots were taken, they conceded almost 16 goals more than they should have. This might suggest that their goalkeepers should have conceded far fewer goals based on the quality of the finishing that they faced, which could explain why they finished in a lower position than predicted by the xG model. Similar effects can be observed for Norwich, Southampton, Brighton, Spurs, and Aston Villa. Conversely, Wolves obtained a higher ranking than their DV suggested, and they achieved a large positive PSxG difference. Whilst the difference cannot account for all cases, it provides context for cases where the difference is considerable. The difference between the mean DV obtained by the teams is visualised in Figure 5.16. In this figure, the x-axis is one dimensional, however the team logos have been moved along the x-axis for clarity.

In Figure 5.16, the teams appear to become naturally clustered by the mean DV obtained. the visualisation shows how the DV model clearly indicates that Manchester City are making the best decisions on the ball, and how the distance between then and the next best team was considerably big during the 2021/22 season. A cluster of team emerges at 4th position, and similarly for teams that finished between 9th and 16th, and in the bottom few places within the league. This visualision provides further context into the results obtained by the mean DV of each team, showing how the model is able to evaluate team performance in a way that aligns with expectations.

5.3.2 Player Analysis by Position

This section will evaluate the DV model's ability to identify the best performing players in different position categories. This is done by calculating which players obtained the highest DV within their positions, considering that they performed at least as many actions in their position as the league average, to avoid including noise.

The data in Table 5.5 places Alisson and Ederson at the two highest places on the list. The two goalkeepers are regularly cited as being the best goalkeepers at possessing

Figure 5.16: Team disparity between mean DV Visualised

the ball[2,3,4]. Alisson was also the goalkeeper chosen within the PFA Team of the Season for the 2021/22 campaign. Ederson and Alisson have both managed to achieve multiple assists in the English Premier League, and they are the two goalkeepers the two with the highest transfer market value. They are also amongst the best goalkeepers with regards to pass completion rates, and both were selected to represent Brazil for the 2022 World

[2] https://www.footballtransfers.com/en/statistics/players/best-football-players/ball-playing-keeper

[3] https://www.football365.com/news/ranking-every-premier-league-no-1-goalkeeper-by-ball-p

[4] https://www.espn.com/soccer/english-premier-league/story/4700756/goalkeeper-awards-2021-22-best-shot-stoppermost-improvedbreakout-star-and-more

Table 5.5: Goalkeeper Results

Player Name	Mean DV	Average xT
Alisson	-0.25	0.05
Ederson	-0.27	0.33
Mendy	-0.29	0.028
Meslier	-0.31	0.026
Sanchez	-0.32	0.043
Lloris	-0.34	0.029
de Gea	-0.34	0.02
Guaita	-0.34	0.029
Schmiechel	-0.36	0.038
Ramsdale	-0.37	0.046
Martinez	-0.37	0.039
McCarthy	-0.37	0.034

Cup. Similarly, the inclusion of Aaron Rammsdale and David Sanchez can be justified by their high pass completion rate. Both goalkeepers were selected for inclusion within their respective national teams at the 2022 World Cup, as was Hugo Lloris for France.

Whilst these statistics do not pertain to their ability at traditional goalkeeping, such as positioning and reflexes, the results show that the mean DV obtained by goalkeepers correlates heavily with the other metrics typically associated with goalkeepers performance, such as their value, individual statistics as well as national team selection at elite level national tournaments. Comparison of the DV and xT metrics shows a difference in ranking between average xT and average DV shows some similarities, such as Alisson receiving a high average xT value. However, the two metrics differ for other goalkeepers, such as Aaron Ramsdale. Whilst Ramsdale is often praised for his ability on the ball, particularly his accurate long passes which might be the cause of his high xT, the DV model provides the expected return for the actions, which in the case of Ramsdale and the difficult passes he typically attempts, results in a relatively lower value being provided.

In Figure 5.17, the DV obtained by center backs for all non-clearance actions has been plotted on the y-axis, with the DV obtained solely from clearance actions being plotted on the x-axis. The chart shows which defenders are obtaining high DV at tasks traditionally associated with defending in clearances, being plotted against their non-clearance DV in general. The DV model highlights Ruben Dias, and Aymeric Laporte as some of the best defenders in the league. This is corroborated by their winning of the Premier League, as

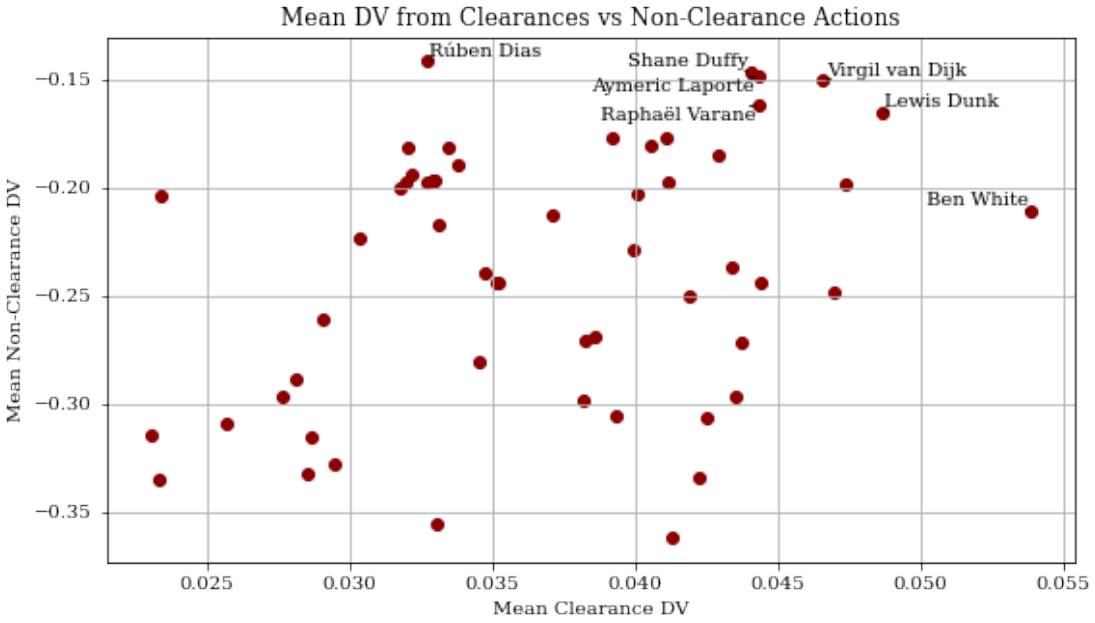


Figure 5.17: Mean DV Obtained by Defenders

well as their inclusion in their respective national team's World Cup squad. Ruben Dias is also listed as the defender with the highest reported transfer market value, and the highest mean DV value during the 2021/22 season. Virgil van Dijk is also listed as one of the better defenders in the chart. The defender was included in the PFA Team of the Year. Similarly, 2018 World Cup winner Raphael Varane achieved a high DV value.

During the 2021/22 season, Ruben Dias managed a pass accuracy of 93.3% (which places him in the 98th percentile for defenders) and 5.2 take-ons that moved possession forward (88th percentile). He also contributed 0.1 xG per game (94th percentile). The same is true for Shane Duffy, Brighton FC's defender whose contribution often goes unnoticed. He achieved 7.33 long passes which puts him in the 93rd percentile for defenders. He was also never dispossessed whilst on the ball, indicating that he has good decision making on the ball. The results in this section highlight the DV metric's ability to identify defensive players that are making the right decisions when they are in possession of the ball.

In Figure 5.18, the mean DV obtained by players that played mostly on either the left or right attacking side during the 2021/22 season were selected. These players were then sorted by the mean DV they obtained during the season. The results show that the top performing decision makers include the likes of Paul Pogba, Phil Foden, Gabriel Jesus, Riyad Mahrez and Raheem Sterling. These are in line with the results achieved within the league itself. Pogba achieved 0.6 assists per 90 minutes, placing him in the 99th

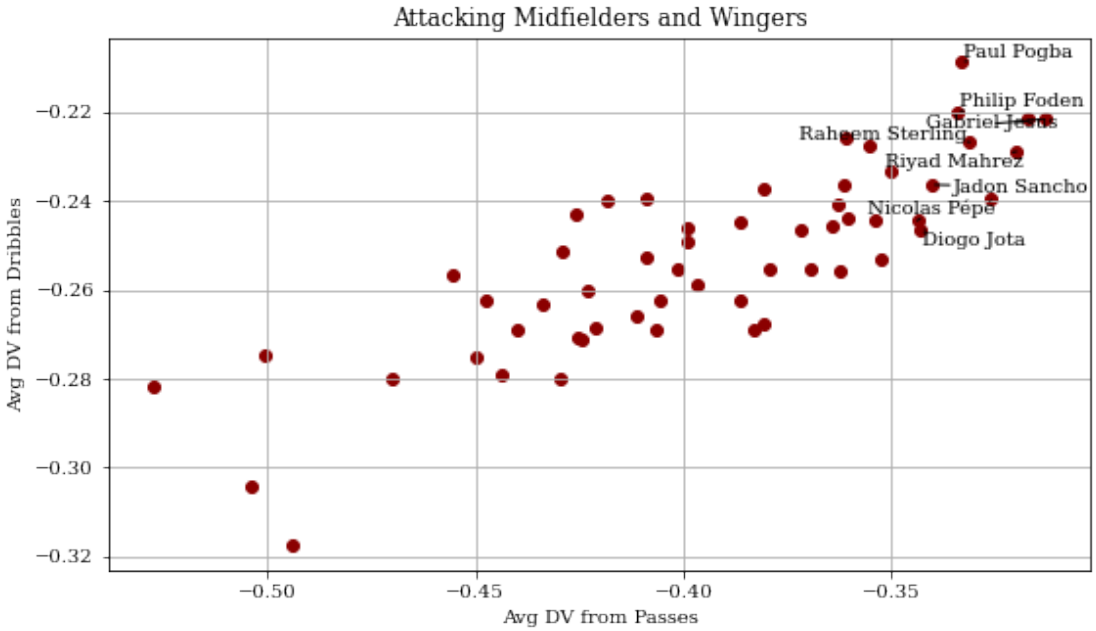


Figure 5.18: Mean DV For Attackers

percentile when compared with Attacking Midfielders and Wingers. He also attempted 59.58 passes per game (96th percentile), of which 7.81 were progressive passes (98th percentile). He also achieved a pass accuracy of 82.4% (90th percentile). He did not win any major trophies nor did he participate in the 2022 World Cup, however this is attributable to his poor luck with injuries throughout the campaign that caused him to miss 16 games during the season. Similarly, Foden, Sterling, Jesus and Mahrez all achieved high percentile values in these areas, and all three of them won the league at the end of the season. The data in Figure 5.18 also shows how the mean DV obtained by players for passes and carries is highly correlated, indicating that a player's ability to make good decisions, thereby obtaining a high DV, seems to be an inherent property of the player themselves.

This final set of results, shown in Figure 5.19 shows which attacking players achieved the highest mean DV per shot taken during the 2021/22 season on the y-axis, versus the players that took the most shots per game on average. The players towards the top right corner indicates the high-volume shot takers that are making the best decisions regarding the scenarios within which they should be taking shots. This data can be compared directly with the 2021/22 Premier League top scorer charts, which can be seen within Table 5.6. Other players, such as Jamie Vardy obtained high mean DVs, but did not take enough shots during matches to rival the league's top scorers. This is supported by the

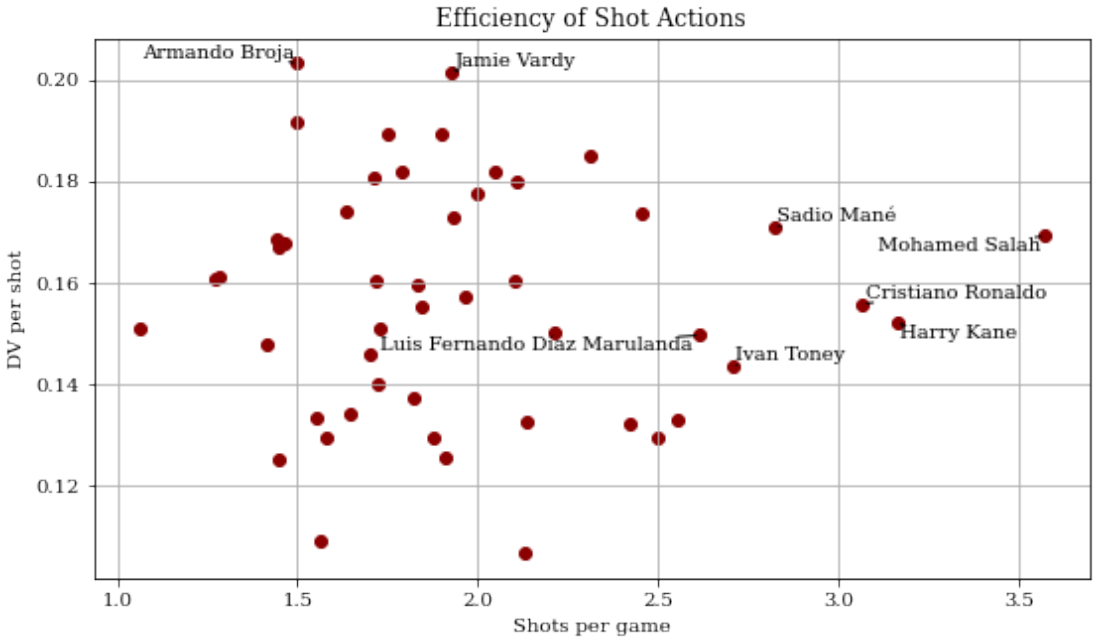


Figure 5.19: Mean DV for shots

fact that Vardy achieved an 0.54 xG from non-penalty shots per game during the season, which puts him within the 94th percentile of strikers.

Player Name	Goals
Mohammed Salah	23
Son Heung-Min	23
Cristiano Ronaldo	18
Harry Kane	17
Saudio Mane	16

Table 5.6: Top Goalscorers in the 2021/22 EPL Season

The data shows that the players that consistently achieved high DV for their shot actions and consistently took a high number of shots during games ended up as some of the league's top scorers. Players such as Cristiano Ronaldo, Mohammed Salah, Saudio Mane, and Harry Kane all emerge as players that both appear closest to the top right corner of Figure 5.19.

5.3.3 Qualitative Action Analysis using DV

Throughout the remainder of this section, the DV model's ability to compare the value of possible actions within their respective scenarios will be carried out. This will allow for analysis of our claim that the DV model can account for the location of surrounding players and opposition accordingly when obtaining a DV for a particular action. The scenarios within this section are taken from games within Season 2, and hypothetical actions that could have been taken within the scenario are plotted. The first such example can be seen within Figure 5.20.

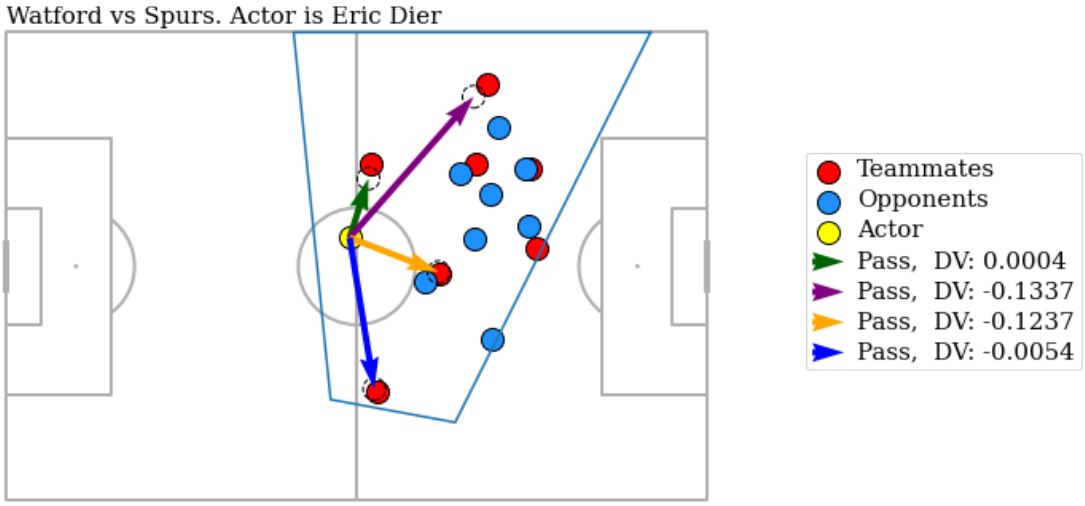


Figure 5.20: Scenario 1

Figure 5.20 shows four hypothetical decisions have plotted, consisting of four passes to different locations on the pitch. The pass marked in purple is the decision with lowest expected return, thereby it is the action with the most risk associated. This is due to the fact that the pass would be made to an area that is significantly far away from the actor which presents several opportunities for possession to be lost during the action. The ball would also be played into a more dangerous area through this action, since there is little support from teammates in this zone of the pitch. The action in orange also receives a similarly low DV score. This could be due to the fact that the ball is being played into an area that is dominated by opposition players.

The pass marked in green obtained a relatively low DV. This indicates that similar passes are relatively safe to play. The pass marked in blue obtains a relatively high DV, indicating a high expected return. This is in line with expectations given the context of the action. The results are particularly interesting when considering the corresponding

xT gains for the passes, where for purple, orange and blue would have respective values of 0.029, 0.025, and 0.013. This further shows how the xT model fails to account for the presence of opposition players.

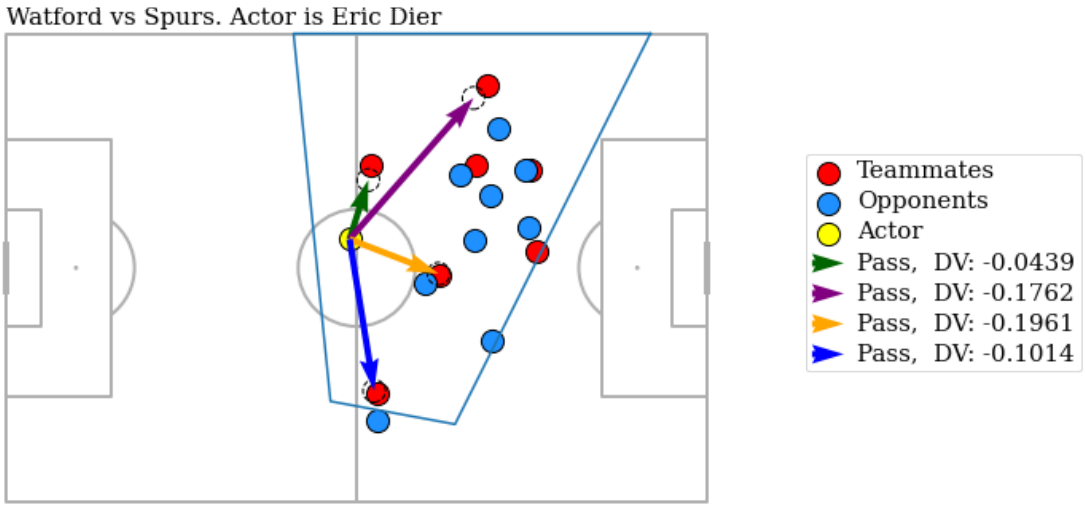


Figure 5.21: Altered Scenario 1

Figure 5.21 contains a fictitious scenario where an opposition player has artificially been added to the scenario to observe the impact it would have on the DV assigned to making the same decision in different scenarios. The result in Figure 5.21 shows how the DV for the pass marked in blue becomes twice as 'risky' when compared with the same decision within 5.20.

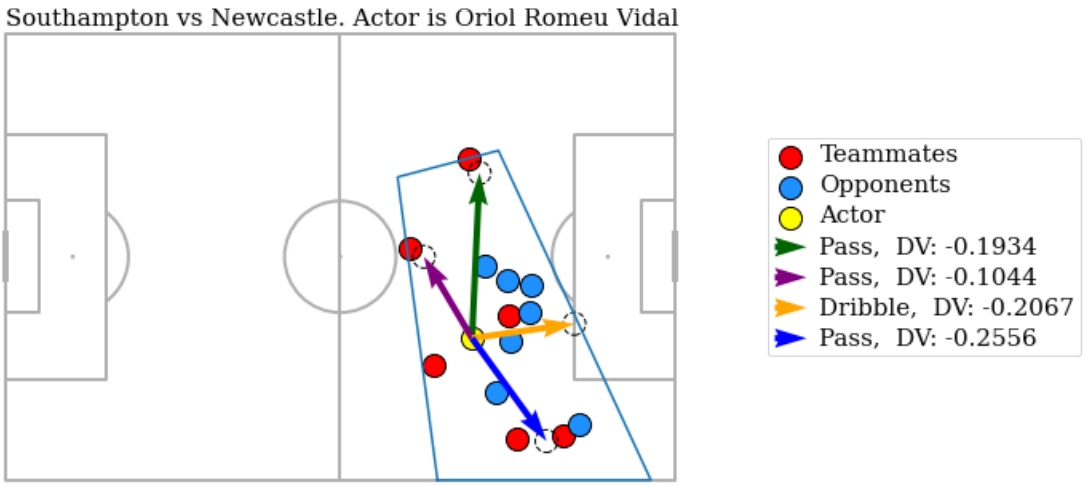


Figure 5.22: Scenario 2

In Figure 5.22, Oriol Romeu has possession of the ball halfway between the middle of the pitch and the edge of the opposition penalty area. The simulated actions include three passes and a carry take-on action. The worst action in this case is indicated in blue. The pass receives a low DV value since the opposition players are in the vicinity. Since DV is the expected return of performing the action within the particular scenario, the lower DV represents the fact that the model expects this action to pose a greater threat to possession loss, and that there is less of an expectation for the action to have a positive outcome. This is in line with expectations when considering the situation.

When compared to the xT model for this particular action, the source of the action in yellow obtains a value of 0.037, and the destination indicated by the edge of the blue arrow obtains a value of 0.051. Thus, the action is considered to be a positive action, since it is moving possession into a more 'threatening' area, and increasing the xT of possession by 0.014. This shows the advantage of the DV model when compared with the xT model, which does not make any considerations of the tracking data. The second worst action in this case according to the DV model is the take-on action carried directly into the box, where the larger negative reward indicates the expectation that performing the action will result in possession loss. The best action in this case, by far, is the pass marked in purple. This is in line with what we would expect, as the ball is being passed into a relatively safe zone, whilst also allowing for the team to possess the ball in a more useful area, as the receiving player would then be in a much better position to pass the ball to his teammate on the left side. In comparison with xT, the action marked in purple would be considered a negative action, since it is moving the ball into an area with xT 0.017, from an area with xT of 0.037, thereby marking a negative difference of 0.02.

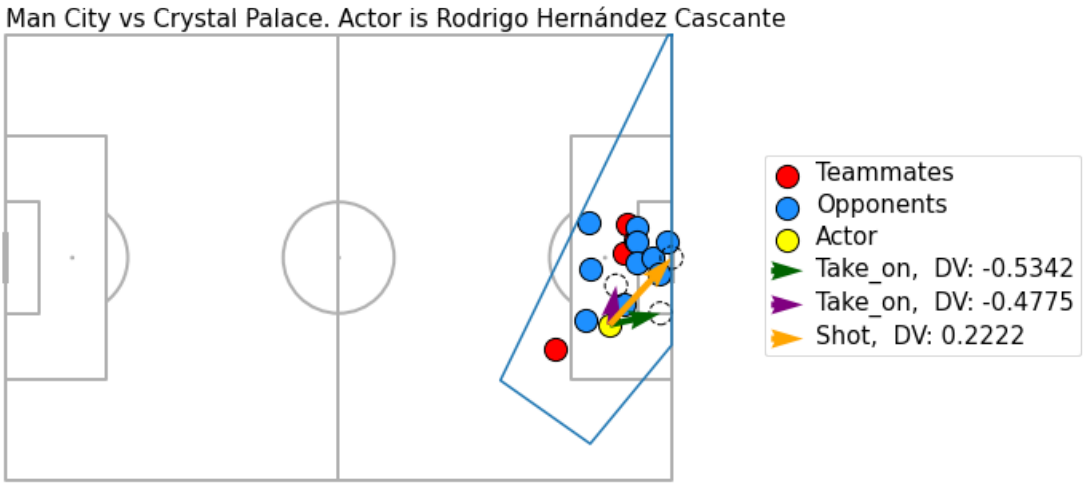


Figure 5.23: DV Evaluation in Shot Action

In the scenario shown in Figure 5.23, three actions have been simulated, two of which being Take-On actions, and the other being a Shot action. A Take-On action refers to the player in possession of the ball to try to directly beat an opposition in a 1-on-1 scenario. In this case, both take on actions present a significant level of risk. Here, the DV for a shot action is significantly higher. This indicates that the expected return for performing a Take-On in either direction is significantly lower than the expected return for shooting from this position. This is also in line with expectations, as performing a Take-On in this scenario is likely to lead in a loss of possession. Performing a shot is also likely to result in possession loss, however the expected return is higher, as it has an associated probability of resulting in a goal, which can be seen in the DV that is much higher. The xT model however would consider both take on decisions to be valuable since they would move possession into a more valuable area of the pitch.

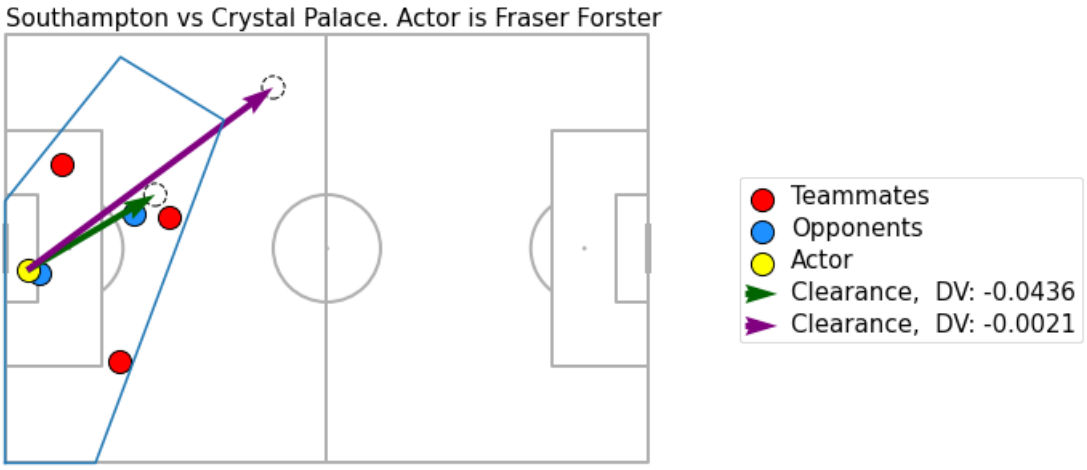


Figure 5.24: Goalkeeper Clearance Decision Analysis

In Figure 5.24, the goalkeeper can be seen under pressure from two opposition players. The simulated decisions in this case reflect the likely actions that the goalkeeper could take, which would be to clear the ball, or to attempt to get the ball away from the opposition players that are pressing him, imminently. The first clearance marked in green would place the ball close to an opposition player, whereas the second clearance marked in purple would place possession of the ball closer to the halfway line, further away from danger. This is reflected in the corresponding DV values, as the shorter clearance receives a much higher DV value than the action. This shows that the DV model is able to capture the contextual information present within the scene, as surrendering possession of the ball that close to the opposition players would present a higher risk than simply clearing the ball further away.

5.3.4 Analysis by pitch section

In this section, the pitch was split into several zones, shown in Figure 5.25.

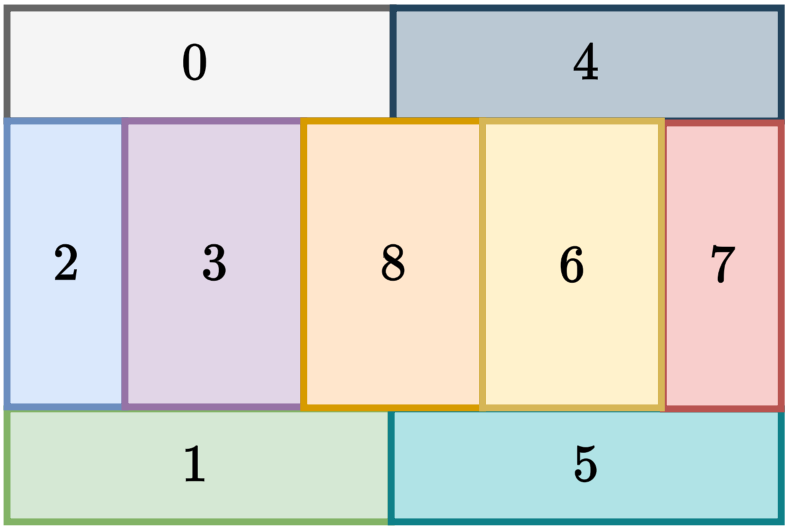


Figure 5.25: Pitch Zones

The mean DV obtained by each team in the Premier League was calculated, for each zone within Figure 5.25. The mean DV per zone was then calculated for individual teams, and the difference between the league average and team average was found per zone. The data in Figure 5.26 shows which regions of the pitch each team performs at a higher level than the league average. Each chart is relative to itself, thus the darkest zone of each chart indicates the zone that is worst in relation to the league average for the particular team, and the lightest part indicates the area of the pitch that is the best in relation to the rest of the league average.

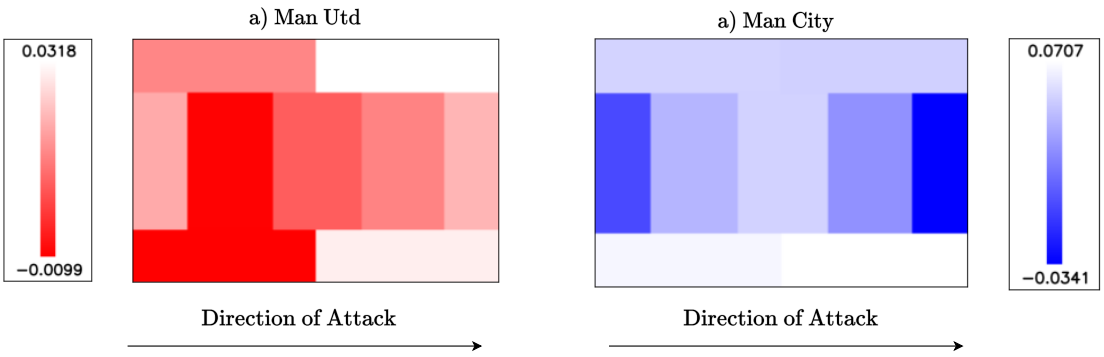


Figure 5.26: DV Performance over average for Man Utd and Man City

The results in Figure 5.26 show that Manchester United's darkest regions correspond with the positions typically occupied by the centre-back, right-back, central defensive midfield and midfield positions. In turn, Manchester United purchased defender Lisandro Martinez, defensive midfielder Casemiro and central midfielder Christian Eriksen. All of the purchases were made after the of the 2021/22 season [5]. Casemiro has won the Champions League five times, the Spanish league three times, and also won the Copa America with his native Brazil. Lisandro Martinez was part of the Argentina squad that won the World Cup in 2022, as well as winning the Copa America with Argentina. He has won the Dutch league title twice, the latter of which he was voted his team's Player of the Year for. Cristian Eriksen was voted the Danish player of the year for three consecutive years between 2013 and 2015. He was also a part of the team that won the Italian league in the 2020/21 season, and has also been included in the Premier League's Team of the Year. This indicates that gaps in quality that can be seen in Figure 5.26a was addressed directly by Manchester United by signing high-quality players. United also focused on the development of their right-back spot, which could be seen by Diogo Dalot's improvement within the 2022/23 season.

Manchester City's is mostly covered with lighter colours, which is positive and indicates that they tend to achieve higher DVs across most areas of the pitch when compared with the league average. The darkest region of the pitch is in the position typically occupied by the striker, which is also where they invested most heavily at the start of the 2022/23 season by purchasing Erling Haaland from Borussia Dortmund. The striker is widely recognised as one of the best attacking players in the world, achieving a total of 218 goal contributions (goals + assists) in only 220 games. He is listed as the second most valuable player in the world, currently valued at €170 million [6]. Thus, the chart shows how real world purchases are in line with the weak areas highlighted by the DV model.

Figure 5.27 shows the areas of the pitch where Chelsea and Liverpool needed to improve their decision making the most. To address this, Chelsea purchased Marc Cucurella. The purchase of the left back was surprising to most fans, however the chart suggests that the decision was a valid one that would improve their decision making within a relatively weak area. To address their other shortcomings, Chelsea also purchased two forwards in Raheem Sterling and Pierre-Emerick Aubameyang, as the areas typically occupied by the strikers were also found to have relatively low DV. Both Sterling and Aubameyang were notorious for their attacking output, with Sterling having achieved 312 goal contributions

[5] https://www.transfermarkt.com/manchester-united/transfers/verein/985/plus/?saison_id=2022&pos=&detailpos=&w_s=s

[6] https://www.transfermarkt.com/spieler-statistik/wertvollstespieler/marktwertetop

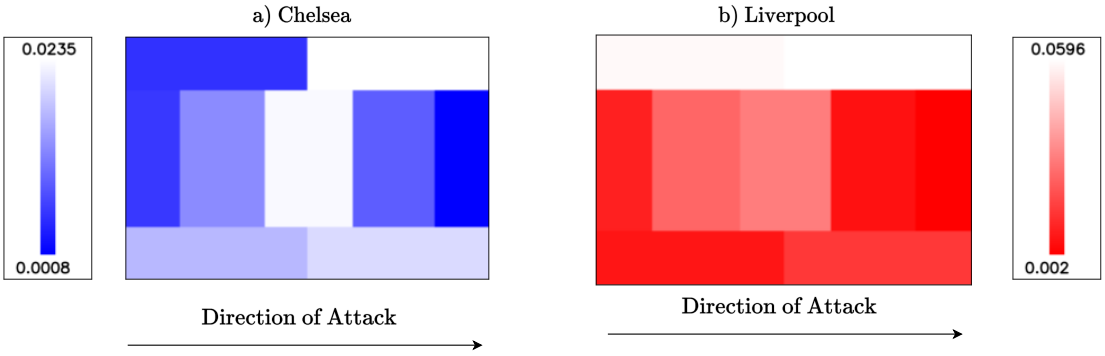


Figure 5.27: DV Performance over average for Chelsea and Liverpool

in 490 games for Manchester City and Liverpool, and Aubameyang having made 404 goal contributions in 609 games.

Figure 5.27 also shows that for Liverpool, their average decision making on the left side of the pitch was significantly better than other areas. The club chose to sell their best performing left-winger Sadio Mane, and have not yet found a like for like replacement, which might be a reason for their struggling form in the 2022/23 season, as the chart shows that his DV was exceptionally high. In other areas of the pitch, Liverpool purchased strikers Darwin Nunez and Cody Gakpo, as well as attacking Midfielder Artur on loan from Juventus for the 2022/23 season. Both Nunez and Gakpo are some of the most highly rated youth attacking prospects, rated at a combined total of €130 million[7,8]. These purchases are in line with the areas of the pitch that need reinforcing according to the DV model. The darker zones on the right hand side might also indicate that Trent-Alexander Arnold tends to make risky actions that would yield a much lower rate of success were they to be attempted by players with a lower skill level.

5.4 Summary

Within this chapter, both quantitative as well as qualitative analysis was carried out. The mean DV obtained by each team within the Premier League was shown to correlate highly with the actual rankings achieved by the teams, achieving a higher correlation than using both of the metrics that were used to train the model (xG and OBV). This shows that the combination of the two metrics, alongside the addition of the context within which the actions were provided allows for a more holistic PVM that has better explanatory power. Several real world scenarios were then shown to allow for the qualitative analysis of the

[7] https://www.transfermarkt.co.uk/darwin-nunez/profil/spieler/546543
[8] https://www.transfermarkt.co.uk/cody-gakpo/profil/spieler/434675

decisions that that could have been made by players, to show that the DV model's outputs are in line with footballing intuition. The scenarios were then modified to show how the model's output depends on the context within which the actions were performed. Finally, the DV model was used to analyse the weaker areas identified for a number of Premier League teams, the results of which were compared with the subsequent purchases made by those teams. The results showed that the average DV achieved when compared with the other leagues in the team for each particular zone was a good indicator as to the areas that received heavy investment, further justifying the model's applicability within a footballing context, and thereby addressing the third objective O3 from Section 1.2.

6 Conclusion

In this concluding chapter, the achievements made throughout this work will be re-visited at a high level. The limitations of the study will also be discussed. Finally, future work that can be carried out to extend the work made within this research will be discussed.

6.1 Revisiting Aims and Objectives

By ensuring that all the objectives set within Section 1.2 were met, the aim of the research was also met successfully. That being to research and implement a PVM that can take the context into account when rating player decisions throughout a football game.

O1: Generate an augmented dataset suited for Deep Reinforcement Learning using existing football tracking and event analysis:
The first, and possibly most important task within this research was to obtain a dataset that could be used to perform Deep Reinforcement Learning on. The data is a crucial part of any research carried out within the field of Machine Learning that aims to learn from expert behaviour, as without a dataset that contains the required information necessary to make the desired predictions, learning would not be possible. In this research, several football datasets were considered and experimented with. The research into the best possible datasets was concluded with the choice of the StatsBomb Open Data dataset, as it contained 50 games of data that contained paired event and tracking data. The Open Dataset provided by StatsBomb has since been amended to also contain all games carried out within the 2022 Fifa World Cup. However, the dataset used was provided by StatsBomb themselves after presenting the work carried out within this research at Wembley Stadium for the 2022 StatsBomb Conference.

The data contained confidential paired event and tracking data obtained from the 2020/21 and 2021/22 English Premier League seasons. The data was augmented to be in the format of an image, as this allowed for the highest level of flexibility considering the constraints of the dataset itself. Each channel of the image was used to represent a

different key aspect of the image, being the actor, teammates and the opposition. The details of the event itself were encoded as a vector of length 7 that contains the details required to describe the action that was carried out, thereby encoding the decision that was made by the player at that moment in time within the context that it was made. Further steps were required to obtain a dataset that is optimised for DRL, as the reward function and the terminal flags were also required to be set correctly. Defining a balanced reward function that is both conservative enough to reduce risk, whilst also taking the reward associated with shooting proved to be a challenging task. The terminal flag was also defined meticulously to ensure that any noise in the dataset did not punish decisions unfairly. Thus, through the creation of a dataset that contains paired observations, actions, rewards and terminal flags, the first objective (O1) was met satisfactorily.

O2: Research and implement a Deep Reinforcement Learning model to learn to value football player decisions:
After obtaining a dataset suited for DRL that sufficiently encoded actions and the context that they were performed in, the next objective was to train a DRL model that could converge onto an optimal behavioural policy based on this dataset. To achieve this, a model that makes use of a continuous state representation and a continuous action space was used. Offline DRL was used since the agent would not be experimenting withing an interactive environment. Instead it would be learning based on the actions made by elite level football players, whilst also being guided by the reward function. The `d3rlpy` library was used to train the models, as it contains ready made implementations that allowed for the work within this section to be focused on selecting the right algorithm rather than the replication of the algorithms in literature. These implementations are continuously verified by the community of researchers. Thus, all of the algorithms that supported the paradigm required for this work were attempted. The algorithms that were computationally cheap enough were then elected as candidates for use within the final model.

Hyper-parameter tuning was then carried out using the Optuna library on each algorithm. Several factors were then taken into account when considering the final model to be used within this work, such as the Actor Loss, Critic Loss and the TD error obtained. The stage within which over-fitting starts to take place, as well as the alignment of the best performing model from each hyper-parameter tuning experiment with expected real world results was also performed. The result of this process was the IQL model being chosen as the ideal algorithm within this work, and the results showed that the model was able to converge onto an optimal behavioural policy as defined by the augmented dataset. Thus, the second objective (O2) was also achieved adequately.

O3: Utilise this RL model to develop a football player decision making evaluation metric that can also be used to evaluate team performance:
The third and final objective within this research was to use the DV model within real world scenarios to analyse its outputs and compare it with PVMs that seek to achieve similar goals. Since the data provided by StatsBomb as a part of the research competition contained data from two separate seasons, it was possible to use one season for training, whilst using the second season for evaluation. Thus, the second season could be analysed without any of the data having been seen previously by the model. The results showed that using the average DV obtained by each team allowed the model to predict the average league table order with an 87% Spearman correlation with regards to the ordering obtained when using the points obtained by the teams throughout the season. This was found to have a higher correlation than using either the xG or the OBV model used for the non-shot rewards, showing that the context provided to the model allowed it to obtain a higher explanatory power than using either component by itself.

The results also show that the model was able to adequately value decisions within scenarios based on the location of the opposition players, and could also be used for a variety of different tasks such as obtaining the best performing players within each main position category, as well as identifying the main areas that teams need to reinforce in with summer transfers. The model was also compared with the xT model to show how the added context is crucial when considering the expected value of possession, as not considering the location of opposition players would lead to highly overvalued decision valuations. Thus, through the quantitative and qualitative analysis carried out with the DV model, the third objective (O3) was met sufficiently.

6.2 Critiques and Limitations

6.2.1 Limited Context Awareness

The dataset used within this research, provided by StatsBomb was crucial in order to represent the notion of context to the DV model. However, the model makes use of limited tracking data as it is extracted from broadcast footage, using a mixture of computer vision and manual annotation. This is a common approach used by multiple companies within the world of football [1] [2]. Using tracking data obtained from the broadcast camera allows the company to extract the same standard of data across several different leagues, meaning that the same insights that can be drawn from elite level football can be drawn

[1] https://skillcorner.com/#tracking-data
[2] https://www.statsperform.com/opta-vision

from low level, obscure divisions. The model could also be improved if complete tracking data was provided instead of only having access to the location of the players visible to the broadcast camera, as this would allow for the development of more accurate and advanced pitch control models, that could include the location off all of the players.

Another issue that arises with the state representation that we have chosen is that whilst the model is given information about the visible players' location, it is not given any information about their velocity and bearing of the players at that moment. If two players are equidistant from the ball, but one player is already facing the ball and running towards it while the other is in a high velocity sprint in the other direction, then the former player will have a much higher chance of retaining possession than the latter even though a velocity-less model would say otherwise. A dataset that includes this data could be used to generate a better performing model.

6.2.2 Rule Violations

Within such decision valuation frameworks, it is also important to consider that the valuation of the decisions is in line with the rules of the game. For example, within this work there is no consideration taken for the offside rule, thus any passes that are made that violate this rule are not valued accordingly by the DV model.

6.2.3 Limited Action Space

Within this work, the actions considered as a part of the DV model were the Pass, Dribble, Take On, Clearance and Shot actions. However, several more actions exist that can be taken that were not included within this work. Direct free kicks, throw ins and penalties were not included for example. The notion of a third dimension was also not encoded within the action vector, thus the model does not differentiate between low driven passes, or high crosses, and treats them equally. This could lead to imperfect valuations for certain actions, since the pass height is not included. Actions that cause an own goal were also excluded from this work.

6.3 Future Work

Whilst the work carried out within this research achieves the objectives that were set out within Section 1.2, it also opens up several avenues for future research. These improvements mostly stem from the limitations faced given the dataset. Thus, future work could include using data that contains the actual location of all 22 players at the same time,

thereby providing the model with a complete observation of the state rather than only being able to see what the broadcast camera can see. Similarly, work that includes the velocity, bearing, and identification of the players could lead to a more accurate model. Further work can also be carried out to take the identity of the players into account. Considering a scenario where a player has the option to pass to two players who each have goal scoring opportunities, the model does not take into account that one of them could be an elite finisher, whilst the other could be a defender that has somehow ended up in this valuable position.

Similarly, if a pass is being played to a player that is known to shoot with their right foot, but the pass is played such that it only facilitates shooting with the left foot, the difference between the passes will not be captured by the current model. Further research could be carried out to include the strengths and weaknesses of each player in the model.

6.4 Concluding Remarks

This research has explored the concept of evaluating player decisions by making use of both event and tracking data using Deep Reinforcement Learning. A custom dataset was created to allow for the problem to be tackled using RL, which was then used to successfully train an actor-critic model using the IQL algorithm. The model obtained was then evaluated both quantitatively and qualitatively and provided results in line with expectations. This allows the model to directly address the scenario described in the Motivation within Section 1.1, as by solely utilising data obtained from broadcast data, the approach proposed will allow teams of various budgets to obtain a high quality model that can value decisions with respect to the context that they were performed in.

References

T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 7 2019. [Online]. Available: https://arxiv.org/abs/1907.10902v1

B. S. Baumer, S. T. Jensen, and G. J. Matthews, "OpenWAR: An open source system for evaluating overall player performance in major league baseball," *Journal of Quantitative Analysis in Sports*, vol. 11, no. 2, pp. 69–84, 6 2015. [Online]. Available: https://www.degruyter.com/document/doi/10.1515/jqas-2014-0098/html

B. Burriel and J. M. Buldú, "The quest for the right pass: Quantifying player's decision making," in *StatsBomb Innovation in Football Conference, London, United Kingdom*, 2021.

S. Caicedo-Parada, C. Lago-Peñas, and E. Ortega-Toro, "Passing Networks and Tactical Action in Football: A Systematic Review," *International Journal of Environmental Research and Public Health 2020, Vol. 17, Page 6649*, vol. 17, no. 18, p. 6649, 9 2020. [Online]. Available: https://www.mdpi.com/1660-4601/17/18/6649/htmhttps://www.mdpi.com/1660-4601/17/18/6649

R. Chia, "The concept of decision: A deconstructive analysis," *Journal of management studies*, vol. 31, no. 6, pp. 781–806, 1994.

C. Collet, "The possession game? A comparative analysis of ball retention and team success in European and international football, 2007–2010," *Journal of Sports Sciences*, vol. 31, no. 2, pp. 123–136, 2013. [Online]. Available: https://doi.org/10.1080/02640414.2012.727455

T. Decroos, L. Bransen, J. Van Haaren, and J. Davis, "Actions Speak Louder Than Goals: Valuing Player Actions in Soccer," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 11, pp. 1851–1861, 2 2018. [Online]. Available: http://arxiv.org/abs/1802.07127http://dx.doi.org/10.1145/

3292500.3330758

——, “VAEP: an objective approach to valuing on-the-ball actions in soccer,” in *IJCAI*, 2020, pp. 4696–4700.

Y. Duan, Q. Liu, and X. H. Xu, “Application of reinforcement learning in robot soccer,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 7, pp. 936–950, 10 2007.

C. J. Efthimiou, “The voronoi diagram in soccer: a theoretical study to measure dominance space,” *arXiv preprint arXiv:2107.05714*, 2021.

H. Eggels, R. van Elk, and M. Pechenizkiy, “Expected goals in soccer: Explaining match results using predictive analytics,” in *The machine learning and data mining for sports analytics workshop*, vol. 16, 2016.

D. Ernst, P. Geurts, and L. U. A. Be, “Tree-Based Batch Mode Reinforcement Learning Louis Wehenkel,” *Journal of Machine Learning Research*, vol. 6, pp. 503–556, 2005.

M. Fatemi, M. Wu, J. Petch, H. Health, S. W. Nelson, A. Benz, A. Carnicelli, and M. Ghassemi, “Semi-Markov Offline Reinforcement Learning for Healthcare,” pp. 119–137, 4 2022. [Online]. Available: https://proceedings.mlr.press/v174/fatemi22a.html

J. Fernández, L. Bornn, and D. Cervone, “Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer,” in *13th MIT Sloan Sports Analytics Conference*, 2019.

S. Fujimoto and S. S. Gu, “A Minimalist Approach to Offline Reinforcement Learning,” in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 20 132–20 145. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/a8166da05c5a094f7dc03724b41886e5-Paper.pdf

S. Fujimoto, H. Van Hoof, and D. Meger, “Addressing Function Approximation Error in Actor-Critic Methods,” *35th International Conference on Machine Learning, ICML 2018*, vol. 4, pp. 2587–2601, 2 2018. [Online]. Available: https://arxiv.org/abs/1802.09477v3

A. García-Aliaga, M. Marquina, J. Coterón, A. Rodríguez-González, and S. Luengo-Sánchez, “In-game behaviour analysis of football players using machine learning techniques based on player statistics,” *International Journal of Sports Science and Coaching*, vol. 16, no. 1, pp. 148–157, 2 2021. [Online]. Available: https://www.researchgate.net/publication/344431117_In-game_behaviour_analysis_of_football_players_using_machine_learning_techniques_based_on_player_statistics

S. J. Gershman and L. Lai, "The reward-complexity trade-off in schizophrenia," *bioRxiv*, p. 2020.11.16.385013, 12 2020. [Online]. Available: https://www.biorxiv.org/content/10.1101/2020.11.16.385013v2https://www.biorxiv.org/content/10.1101/2020.11.16.385013v2.abstract

J. F. Gréhaigne, P. Godbout, and D. Bouthier, "The Teaching and Learning of Decision Making in Team Sports," *https://doi.org/10.1080/00336297.2001.10491730*, vol. 53, no. 1, pp. 59–76, 2 2012. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00336297.2001.10491730

D. Gronow, B. Dawson, J. Heasman, B. Rogalski, and P. Peeling, "Team movement patterns with and without ball possession in Australian Football League players," *International Journal of Performance Analysis in Sport*, vol. 14, no. 3, pp. 635–651, 2014. [Online]. Available: https://doi.org/10.1080/24748668.2014.11868749

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft Actor-Critic Algorithms and Applications," *arXiv preprint arXiv:1812.05905*, 12 2018. [Online]. Available: https://arxiv.org/abs/1812.05905v2

L. Higgins, T. Galla, B. Prestidge, and T. Wyatt, "Measuring the pitch control of professional football players using spatiotemporal tracking data," *Journal of Physics: Complexity*, 2023.

S. Huang, W. Chen, L. Zhang, S. Xu, Z. Li, F. Zhu, D. Ye, T. Chen, and J. Zhu, "TiKick: Towards Playing Multi-agent Football Full Games from Single-agent Demonstrations," *arXiv preprint arXiv:2110.04507*, 10 2021. [Online]. Available: https://arxiv.org/abs/2110.04507v5

M. Hussing, J. A. Mendez, C. Kent, and E. Eaton, "Robotic Manipulation Datasets for Offline Compositional Reinforcement Learning," in *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.

J. Jara-Ettinger, "Theory of mind as inverse reinforcement learning," *Current Opinion in Behavioral Sciences*, vol. 29, pp. 105–110, 10 2019.

A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J. M. Allen, V. D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 8248–8254, 5 2019.

A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar, "Graph Policy Gradients for Large Scale Robot Control," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and

K. Sugiura, Eds., vol. 100. PMLR, 3 2020, pp. 823–834. [Online]. Available: https://proceedings.mlr.press/v100/khan20a.html

J. Kim, N. James, N. Parmar, B. Ali, and G. Vučković, "The Attacking Process in Football: A Taxonomy for Classifying How Teams Create Goal Scoring Opportunities Using a Case Study of Crystal Palace FC," *Frontiers in Psychology*, vol. 10, pp. 1–8, 3 2019.

S. Kim, "Voronoi Analysis of a Soccer Game," *Nonlinear Analysis: Modelling and Control*, vol. 9, no. 3, pp. 233–240, 7 2004. [Online]. Available: https://www.journals.vu.lt/nonlinear-analysis/article/view/15154

K. Konyushkova, K. Zolna, Y. Aytar, A. Novikov, S. Reed, S. Cabi, and N. de Freitas, "Semi-supervised reward learning for offline reinforcement learning," 2020.

I. Kostrikov, A. Nair, and S. Levine, "Offline Reinforcement Learning with Implicit Q-Learning," *arXiv preprint arXiv:2110.06169*, 10 2021. [Online]. Available: https://arxiv.org/abs/2110.06169v1

A. Kumar and I. Kuzovkin, "Offline Robot Reinforcement Learning with Uncertainty-Guided Human Expert Sampling," 2022.

A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," *Advances in Neural Information Processing Systems*, vol. 2020-December, 6 2020. [Online]. Available: https://arxiv.org/abs/2006.04779v3

K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, and S. Gelly, "Google Research Football: A Novel Reinforcement Learning Environment," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 4501–4510, 7 2019. [Online]. Available: https://arxiv.org/abs/1907.11180v2

S. Kusmakar, S. Shelyag, Y. Zhu, D. Dwyer, P. Gastin, and M. Angelova, "Machine Learning Enabled Team Performance Analysis in the Dynamical Environment of Soccer," *IEEE Access*, vol. 8, pp. 90 266–90 279, 2020.

N. Lambert, M. Wulfmeier, W. Whitney, A. Byravan, M. Bloesch, V. Dasagi, T. Hertweck, and M. Riedmiller, "The Challenges of Exploration for Offline Reinforcement Learning," 2022.

B. Larrousse, "Improving decision making for shots," in *StatsBomb Innov. Footb. Conf*, 2019, pp. 1–15.

Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp.

436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

J. Lee, R. Kim, Y. Koh, and J. Kang, "Global stock market prediction based on stock chart images using deep q-network," *IEEE Access*, vol. 7, pp. 167 260–167 277, 2019.

J. Lee, S. Kim, S. Kim, W. Jo, and H.-J. Yoo, "GST: Group-Sparse Training for Accelerating Deep Reinforcement Learning," *arXiv preprint arXiv:2101.09650*, 1 2021. [Online]. Available: http://arxiv.org/abs/2101.09650

S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," *arXiv preprint arXiv:2005.01643*, 5 2020. [Online]. Available: https://arxiv.org/abs/2005.01643v3

R. Liessner, C. Schroer, A. Dietermann, and B. Bäker, "Deep Reinforcement Learning for Advanced Energy Management of Hybrid Electric Vehicles," in *ICAART (2)*, 3 2018, pp. 61–72.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

D. Linke, D. Link, and M. Lames, "Football-specific validity of TRACAB's optical video tracking systems," *PLOS ONE*, vol. 15, no. 3, p. e0230179, 2020. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0230179

G. Liu and O. Schulte, "Deep Reinforcement Learning in Ice Hockey for Context-Aware Player Evaluation," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, pp. 3442–3448, 5 2018. [Online]. Available: http://arxiv.org/abs/1805.11088http://dx.doi.org/10.24963/ijcai.2018/478

G. Liu, Y. Luo, O. Schulte, and T. Kharrat, "Deep soccer analytics: learning an action-value function for evaluating soccer players," *Data Mining and Knowledge Discovery 2020 34:5*, vol. 34, no. 5, pp. 1531–1559, 7 2020. [Online]. Available: https://link.springer.com/article/10.1007/s10618-020-00705-9

B. Macdonald, "An Expected Goals Model for Evaluating NHL Teams and Players," *MIT Sloan Sports Analytics Conference*, 2012.

J. G. March, *Decisions and organizations*. Blackwell, 1989.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-

level control through deep reinforcement learning," *Nature 2015 518:7540*, vol. 518, no. 7540, pp. 529–533, 2 2015. [Online]. Available: https://www.nature.com/articles/nature14236

K. Muelling, A. Boularias, B. Mohler, B. Schölkopf, and J. Peters, "Inverse reinforcement learning for strategy extraction," in *ECML PKDD 2013 workshop on machine learning and data mining for sports analytics (MLSA 2013)*, 2013, pp. 1–9.

R. B. Myers, M. Burns, Q. B. Coughlin, and E. Bolte, "On the Development and Application of an Expected Goals Model for Lacrosse – The Sport Journal," 12 2021. [Online]. Available: https://thesportjournal.org/article/on-the-development-and-application-of-an-expected-goals-model-for-lacrosse/

B. T. Naik, M. F. Hashmi, Z. W. Geem, and N. D. Bokde, "DeepPlayer-Track: Player and Referee Tracking With Jersey Color Recognition in Soccer," *IEEE Access*, vol. 10, pp. 32 494–32 509, 2022.

A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

A. V. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine, "Visual Reinforcement Learning with Imagined Goals," *Advances in Neural Information Processing Systems*, vol. 31, 2018. [Online]. Available: https://sites.google.com/site/

T. L. Paine, C. Paduraru, A. Michi, C. Gulcehre, K. Zolna, A. Novikov, Z. Wang, and N. de Freitas, "Hyperparameter selection for offline reinforcement learning," *arXiv preprint arXiv:2007.09055*, 2020.

F. Pan, Q. Cai, P. Tang, F. Zhuang, and Q. He, "Policy Gradients for Contextual Recommendations," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1421–1431. [Online]. Available: https://doi.org/10.1145/3308558.3313616

J. Perl and D. Memmert, "Soccer analyses by means of artificial neural networks, automatic pass recognition and voronoi-cells: An approach of measuring tactical success," *Advances in Intelligent Systems and Computing*, vol. 392, pp. 77–84, 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-24560-7_10

D. Pizarro, A. Práxedes, B. Travassos, F. del Villar, and A. Moreno, "The effects of a nonlinear pedagogy training program in the technical-tactical behaviour of youth futsal players," *International Journal of Sports Science and Coaching*, vol. 14, no. 1, pp.

15–23, 2 2019. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1747954118812072

M. Poloczek, J. Wang, and P. Frazier, "Multi-information source optimization," *Advances in neural information processing systems*, vol. 30, 2017.

R. F. Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *arXiv preprint arXiv:2203.01387*, 2022.

M. Pulis and J. Bajada, "Reinforcement Learning for Football Player Decision Making Analysis," *StatsBomb Conference*, 2022.

P. Rahimian and L. Toka, "Inferring the Strategy of Offensive and Defensive Play in Soccer with Inverse Reinforcement Learning," *Communications in Computer and Information Science*, vol. 1571 CCIS, pp. 26–38, 2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-02044-5_3

P. Rahimian, A. Oroojlooy, and L. Toka, "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)*, 2021, pp. 1–12.

P. Rahimian, J. Van Haaren, T. Abzhanova, and L. Toka, "Beyond action valuation: A deep reinforcement learning framework for optimizing player decisions in soccer," in *16th Annual MIT Sloan Sports Analytics Conference. Boston, MA, USA: MIT*, 2022, p. 25.

R. Rein and D. Memmert, "Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science," *SpringerPlus 2016 5:1*, vol. 5, no. 1, pp. 1–13, 8 2016. [Online]. Available: https://springerplus.springeropen.com/articles/10.1186/s40064-016-3108-2

M. Riedmiller, T. Gabel, R. Hafner, and S. Lange, "Reinforcement learning for robot soccer," *Autonomous Robots*, vol. 27, no. 1, pp. 55–73, 7 2009.

P. Robberechts and J. Davis, "How data availability affects the ability to learn good xG models," *Communications in Computer and Information Science*, vol. 1324, pp. 17–27, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-64912-8_2

B. Robson and P. Hayward, *Bobby Robson: Farewell but not Goodbye - My Autobiography: The Remarkable Life of a Sporting Legend*. Hodder & Stoughton, 2006. [Online]. Available: https://books.google.com.mt/books?id=m2j5ZKj993QC

S. Rudd, “A framework for tactical analysis and individual offensive production assessment in soccer using markov chains,” in *New England symposium on statistics in sports*, 2011.

T. Seno, M. Imai, K. University, and S. Ai, “d3rlpy: An Offline Deep Reinforcement Learning Library,” 11 2021. [Online]. Available: https://arxiv.org/abs/2111.03788v1

D. Shah, A. Bhorkar, H. Leen, I. Kostrikov, N. Rhinehart, and S. Levine, “Offline Reinforcement Learning for Visual Navigation,” 2022.

A. F. Silva, D. Conte, and F. M. Clemente, “Decision-Making in Youth Team-Sports Players: A Systematic Review,” *International journal of environmental research and public health*, vol. 17, no. 11, 6 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32471126/

D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic Policy Gradient Algorithms,” pp. 387–395, 1 2014. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html

K. Singh, “Introducing Expected Threat (xT),” 2018. [Online]. Available: https://karun.in/blog/expected-threat.html

S. Sinha, A. Mandlekar, and A. Garg, “S4RL: Surprisingly Simple Self-Supervision for Offline Reinforcement Learning in Robotics,” pp. 907–917, 1 2022. [Online]. Available: https://proceedings.mlr.press/v164/sinha22a.html

H. Sotudeh, “Potential Penetrative Pass (P3),” in *StatsBomb Conference*. [Online]. Available: http://statsbomb.com/wp-content/uploads/2021/11/Hadi-SotudehStatsBomb-Conference-2021-Research-Paper.pdf

W. Spearman, A. Basye, G. Dick, R. Hotovy, and P. Pop, “Physics-Based Modeling of Pass Probabilities in Soccer,” in *MIT Sloan Sports Analytics Conference 2017*, 3 2017.

StatsBomb, “StatsBomb Release Expected Goals with Shot Impact Height | StatsBomb,” 2020. [Online]. Available: https://statsbomb.com/2020/07/statsbomb-release-expected-goals-with-shot-impact-height/

——, “Introducing On-Ball Value (OBV) - StatsBomb | Data Champions,” 2021. [Online]. Available: https://statsbomb.com/articles/soccer/introducing-on-ball-value-obv/

D. Sumpter, “Decentralised football is more effective than focusing on one or two players | by David Sumpter | Medium,” 2017. [Online]. Available: https://soccermatics.medium.com/decentralised-football-is-more-effective-than-focusing-on-one-or-two-players-c197216ac2b8

R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

R. Takayanagi, K. Takahashi, M. Wataba, K. Ohkawara, and T. Sogabe, "Decision Making in American Football under State Uncertainty by Stochastic Inverse Reinforcement Leaning," *Bulletin of Networking, Computing, Systems, and Software*, vol. 11, no. 1, pp. 25–29, 1 2022. [Online]. Available: http://ww.w.bncss.org/index.php/bncss/article/view/158

R. Vaeyens, M. Lenoir, A. M. Williams, L. Mazyn, and R. M. Philippaerts, "The effects of task constraints on visual search behavior and decision-making skill in youth soccer players," *Journal of sport & exercise psychology*, vol. 29, no. 2, pp. 147–169, 2007. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/17568064/

M. Van Roy, P. Robberechts, T. Decroos, and J. Davis, "Valuing on-the-ball actions in soccer: a critical comparison of XT and VAEP," in *Proceedings of the AAAI-20 Workshop on Artifical Intelligence in Team Sports. AI in Team Sports Organising Committee*, 2020.

M. Van Roy, P. Robberechts, W.-C. Yang, L. De Raedt, and J. Davis, "Learning a markov model for evaluating soccer decision making," in *Reinforcement Learning for Real Life (RL4RealLife) Workshop at ICML 2021*, 2021.

M. Vohra and G. S. D. Gordon, "Markov Cricket: Using Forward and Inverse Reinforcement Learning to Model, Predict And Optimize Batting Performance in One-Day International Cricket," *arXiv preprint arXiv:2103.04349*, 3 2021. [Online]. Available: https://arxiv.org/abs/2103.04349v1

R. Wang, Y. Wu, R. Salakhutdinov, and S. Kakade, "Instabilities of offline rl with pre-trained neural representation," in *International Conference on Machine Learning*, 2021, pp. 10 948–10 960.

Z. Wang, G. Liao, X. Shi, X. Wu, C. Zhang, Y. Wang, X. Wang, and D. Wang, "Learning List-Wise Representation in Reinforcement Learning for Ads Allocation with Multiple Auxiliary Tasks," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 3555–3564. [Online]. Available: https://doi.org/10.1145/3511808.3557094

Z. Wang, A. Novikov, K. Zolna, J. S. Merel, J. T. Springenberg, S. E. Reed, B. Shahriari, N. Siegel, C. Gulcehre, N. Heess, and others, "Critic regularized regression," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7768–7778, 2020.

R. J. Williams, "Simple statistical gradient-following algorithms for connectionist

reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

C. T. Woods, A. J. Raynor, L. Bruce, and Z. McDonald, "Discriminating talent-identified junior Australian football players using a video decision-making task," *https://doi.org/10.1080/02640414.2015.1053512*, vol. 34, no. 4, pp. 342–347, 2 2015. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/02640414.2015.1053512

C. Yanai, A. Solomon, G. Katz, B. Shapira, and L. Rokach, "Q-Ball: Modeling Basketball Games Using Deep Reinforcement Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, pp. 8806–8813, 6 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20861

H. Yoo, B. Kim, J. W. Kim, and J. H. Lee, "Reinforcement learning based optimal control of batch processes using Monte-Carlo deep deterministic policy gradient with phase segmentation," *Computers & Chemical Engineering*, vol. 144, p. 107133, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098135420307912

C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 4 2019.