(Wallpaper Flare., n.d.)

Optimization of Running Paths in American Football

Using a Mathematical Approach

Author
Steijn Heijerman

Educational Institution
Amsterdam University of Applied Sciences

Education
Applied Mathematics

Date
12-06-2024

Optimization of Running Paths in American Football

Using a Mathematical Approach

Author	Steijn Heijerman
Student number	500759701
Company	Beyond Sports
Department	Data Team
Business supervisor	Stan Zutt
Role	Data Scientist
Educational Institution	Amsterdam University of Applied Scientist
Faculty	Engineering
Education	Applied Mathematics
Docent supervisor	Michelangelo Vargas Rivera
Second examiner	Miriam Loois
Date	12-06-2024

Hogeschool van Amsterdam

Management Summary

The research in this report is conducted for Beyond Sports, an innovative visualization company specializing in AI technologies to create unique perspectives and views of real sports footage. Beyond Sports collaborates with leading broadcasters such as Disney and ESPN, offering advanced player positioning and limb-tracking technologies to transform real-time sports matches into various animation styles, such as Toy Story, SpongeBob, and Ninja Turtles.

Beyond Sports is dedicated to enhancing the sports viewing experience and the informational value of sports broadcasts. The company aims to provide deeper insights into sports performance, particularly in American football, where every action on the field is crucial. The primary task involves creating a model that can predict the optimal running path to the endzone for an American football player with the ball. The optimal path minimizes the total cost, calculated as the sum of certain weights (to be discussed later). Ultimately, this path with the minimal total cost is chosen as the optimal path. These predictions can help analysts provide more detailed insights into what a player could have done better during the game, thereby increasing the knowledge and engagement of viewers.

This model takes into account the influence of opponents and teammates, as well as the distances between different points on the field. The influence of opponents and teammates, as well as the distances, are expressed as weights distributed across the field. The end zone is the target towards which the path is chosen. The Dijkstra algorithm is used as the model to find the minimal total cost on the field.

The influence weights (of opponents and teammates) are calculated using the space control model, which assigns a weight to each location on the field, indicating which team is likely to have ball possession at that spot. Additionally, the model takes into account the speed of players, the fact that a player cannot make sharp turns at full speed, the reaction time of players to the actions of the player with the ball, and the rules that the defensive team can tackle the player with the ball while the offensive team can only block. This model is visually implemented, as shown in figure 1.

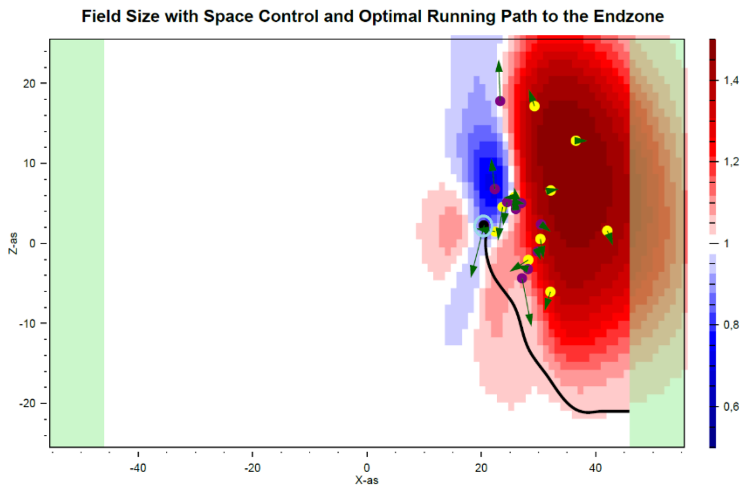


Figure 1: The prediction of the optimal running path to the endzone for an American football player with the ball (circled in light blue). The yellow dots represent the opponents with a red heatmap represents their dominant space control (influence weights) on the field. The purple dots represent the teammates with a blue heatmap represents their dominant space control (influence weights) on the field.

The requirements from Beyond Sports, along with the conclusion that the result meets these requirements, are as follows:

- The processing time of the prediction algorithm must not exceed 5 seconds.
- The predicted running path should be realistic compared to what happens in practice.
- The calculations of the prediction algorithm must work on any data frame of a loaded American football game.

After conducting the research, several recommendations have been formulated to improve this study:

- Simplify the programming code of the model in Beyond Sports' software to speed up the processing time of the prediction algorithm.
- Test other algorithms besides the Dijkstra algorithm used, to achieve a faster processing time.
- Evaluate the predicted running path in a programmed simulation to improve the evaluations of the predicted paths.
- Further investigate the added turning angle condition to make the paths even more realistic.
- Consider other targets besides the end zone, such as other distances, to predict paths to different strategic goals.
- Assign a success score to a predicted path to provide a success probability for the predicted running path.

The report concludes with an encouragement for further research into the method described in the article "A Reinforcement Learning Based Approach to Play Calling in Football" (Biro, P. & G. Walker, 2021). This research utilizes a Markov decision process to determine optimal play choices based on collected data and outcome probabilities. By applying this methodology, a model can be developed that predicts the best actions for the player with the ball, such as throwing, running, or handing off the ball. This is recommended because Beyond Sports is interested in developing a model that can make these predictions. If the prediction from this model is that the player should run, the model from the current research can then be used to determine in which direction to run.

Contents

1 Introduction

American football, a strategic and dynamic sport, captivates millions of fans worldwide (Afi, 2023). Teams of eleven players compete for victory within 60 minutes, where tactics, strength, and precision are key. In the past, analysts and so fans had limited data available, mostly basic statistics like yards run or passes completed. However, the introduction of sensors and tracking cameras has not only changed the analysis but also enhanced the viewing experience (Sports Player Tracking | Zebra, n.d.). These technologies collect detailed data on speed, position, and distances covered by players, providing fans with insights and a more intense game experience. These developments have introduced new methods for analyzing and experiencing matches, continuously evolving the sport (Stables, 2014).

1.1 The Client

Beyond Sports, a visualization company specializing in AI technologies, is the client for this project. The company, which has about 50 employees including six interns, provides broadcasters, teams, clubs, and brands with unique perspectives and renderings of real sports footage using player position and limb-tracking technologies. Beyond Sports is divided into several departments: the Data team, Unity team, Art/Design team, Software Development team, Marketing team, HRM (Human Resource Management) team, Machine Learning/AI team, Finance team, and the Board. The research is conducted within the Data department, which consists of four people. This team is tasked with enhancing and validating incoming data for visualizations. Their responsibilities include implementing event detection and adding animations, such as correcting unrealistic events in sports footage, for example, the ball with American football has no height yet so it is moved along the ground. These tasks require programming to ensure that the data reflect realistic scenarios, thereby making the visualizations representations of the reality.

Beyond Sports has achieved a breakthrough in the visualization of live sports events. The company is now capable of converting real-time games into various animation styles, such as that of Toy Story. As seen in figure 2, the real game is displayed on the left side while the Toy Story-style visualization is on the right. This screenshot, taken during a game, illustrates the capability to not only follow the action in real-time but also to replay it in the chosen animation style. Beyond Sports collaborates with broadcasters like Disney and ESPN to deliver these animated visualizations. They create content in various styles, including those of Toy Story, SpongeBob, and Ninja Turtles, for a wide range of sports such as American football, ice hockey, and soccer. The visualizations are made with live data, allowing viewers to watch the game in animation style simultaneously on Disney or ESPN platforms. A key aspect of this service is the ability for viewers to watch the game from different camera angles. Beyond Sports' revenue model is based on contracts with these broadcasters, for whom they produce simulations in animation style that are then broadcasted on their platforms. This provides a unique viewing experience that transforms the way fans experience sports (Zachary, 2023).

Figure 2: On the left, the real American football game, and on the right, the simulation by Beyond Sports.

Note: Adapted from: Sports (2024)

1.2 The Desired Situation

Beyond Sports is committed to enriching the sports experience and the informational value of sports broadcasts. The company is interested in providing insights into sports performance, particularly in American football, because this sport is one of the most popular sports in America (Jones, 2024). Beyond Sports aims to develop a model that can predict the best action for a player with the ball: whether to throw, run, or hand off the ball. If the player opts to throw, in which direction or to which player? If the player chooses to run, what is the best route they can take? If the player decides to hand off the ball, to which player? With these predictions, analysts can provide more explanations of what a player could have done better during the game, which not only improves athletic performance (Ambler, 2024) but also enhances the knowledge and engagement of viewers (Power, 2024).

1.3 The Main Assignment

After consultation with the client and based on the time available for this study, it was decided to contribute to a contribution of the desired situation.

Research goal: Developing a model that can predict the optimal running path to the endzone for an American football player with the ball.

These paths are optimized based on both the shortest distance to the endzone (the area where a touchdown can be scored) and the positions and movements (also referred to as a certain influence) of opponents and teammates on the field. The influence of opponents and teammates, as well as the distances, are expressed in weights. The model will propose paths where the sum of the distances and weights to the endzone is minimal. This is considered the most effective progression to the endzone for a player with the ball. Additionally, the optimal running path must be realistic and correspond to what happens in practice. The optimal paths will eventually be visually represented to evaluate the paths and demonstrate them to Beyond Sports.

1.4 Sub Questions

- What is required to develop a model that predicts the optimal running path to the endzone?
    - What are the criteria for defining an optimal running path?
    - What data is needed to determine the optimal running path?

    - What model is needed to determine the optimal running path?
    - How can the model account for realistic scenarios in American football?
    - What variables are necessary to determine the optimal running path?
    - How is a single point chosen, and thus a single route for an American football player with the ball determined?
- How will the optimal running path to the endzone be presented to Beyond Sports?

1.5 Delivery and Requirements from/to Beyond Sports

Beyond Sports will receive a C# implementation that will be integrated into their existing software. This existing software results in the simulation visible in figure 2. The implementation consists of a model, with corresponding calculations, that can predict the optimal running path to the endzone for an American football player with the ball. The predicted running path is ultimately visualized.

To achieve the desired outcome of the main assignment, Beyond Sports has set several requirements that must be considered in the development of the final result:

- **The processing time of the prediction algorithm must not exceed 5 seconds**. The client has added a requirement to the main assignment: the developed algorithm should be capable of functioning during a live match for the desired situation. This enables an analyst to display a visualization of the predicted optimal running path at any point during a live match. To achieve this, the processing time must not exceed 0.1 seconds. There is also a possibility that the algorithm will not function during a live match but will be used afterward, for instance in a game summary. In this scenario, the processing time can be longer, but must not exceed 5 seconds. This ensures that the analysis is still quick and efficient, suitable for rapid reviews and evaluations immediately after the game.
- **The outcome for the running path must be realistic in comparison to what occurs in practice.**
- **The calculations of the prediction algorithm work on any dataframe of a loaded American football game**. So this means at any time during the match.

1.6 Reading Guide

This report consists of several sections. Chapter 2 explains the workings of the game of American football, including definitions of game terms that will appear later in the report. Chapter 3 describes the data that is available. Chapter 4 covers the methodology and literature review. Following this, Chapter 5 presents the evaluations of the model and the results. The report concludes with the conclusion, discussion, and recommendations.

2 The Game of American Football

American football is a team sport played by two teams of eleven players each. The goal of the game is to score points by getting the ball into the opponent's endzone, where a touchdown can be made. This can be achieved by either carrying or throwing the ball. A match is divided into four quarters, each lasting 15 minutes (Haddad, 2024).

Points can be scored in various ways: a touchdown is worth six points and is scored when a player brings the ball into the opponent's endzone; after a touchdown, the scoring team has the opportunity to score an extra point by kicking the ball between the goal posts, or two points by making another touchdown from the two-yard line. Additionally, a team can score three points with a field goal, where the ball is kicked between the goal posts from the playing field (Haddad, 2024).

The game begins with a kick-off, where one team kicks the ball to the other team. The receiving team then tries to kick the ball as far towards the opponent's endzone as possible. The offense of the team with the ball then tries to advance at least ten yards in a series of four attempts (called downs) to earn a new set of four downs and continue until they score or the ball is turned over to the opponent. The defence tries to stop the offense and take possession of the ball (Haddad, 2024).

A 'snap' is the moment the game starts (after a kick-off), the ball is passed from the centre to the quarterback. The quarterback leads the game, deciding whether to throw, run, or hand off the ball. The line to gain (LTG) is an imaginary line that the team must reach to extend their set of four downs and get closer to scoring. The LOS (line of scrimmage) is the starting line for each action. In short, the snap initiates the game, the quarterback sets the strategy, and the LTG marks the success target for the offense (Martin, 2022).

A down in American football ends when the player with the ball is tackled, steps out of bounds, or when a pass is incomplete (not caught). After the end of a down, the ball is repositioned on the line where the previous action ended, and a new down begins, from a midfield area of the field. The midfield area is a strip that runs from front to back across the middle of the field, giving the offensive team a chance to try to reach the line of gain (LTG) again and continue their series of four downs to get closer to a score (Martin, 2022a).

3 Data

In this chapter "Data," the dataset from Beyond Sports is described in detail. Section 3.1 addresses the complete dataset, while Section 3.2 specifically focuses on the variables used for the research. The dataset employed for the study has already been cleaned and made user-friendly; there are no missing values.

3.1 Data exploration

Beyond Sports has comprehensive data from 60 different American football games in the NFL (National Football League). This data is stored in JSON files that contain ten frames per second. With an average duration of three hours per game (including total time with game stoppages), this results in approximately 108,000 frames per game. The dataset used is an enhanced and cleaned version of the raw data, including information such as the height of the ball, the speed of the players, ball possession, and various events (more on this later). This dataset is immediately usable in simulations, enabling the visual analysis and review of a match. As depicted in figure 3, the simulation displays the positions of the players, the ball, and various game-defining lines such as the Line of Scrimmage (LOS) and Line To Gain (LTG) after loading the data. The simulation also indicates the team affiliation of each player, the current score, the period, and the time. The field is oriented with the centre spot at the coordinates (0,0,0), where the y-axis represents height, the x-axis length, and the z-axis the width of the field.

Figure 3: A moment from Beyond Sports' simulation room where the data from a match is loaded.

Note: Adapted from: The simulation of Beyond Sports

Table 1 contains information about the subjects and variables included in the frames, and this table refers to the appendices where the variables within the subjects are described.

Table 1: Overview of variables and all subjects from the dataset of an American football game.

Variables or subjects names	**Example**	**Definition**
FrameCount	64452	The frame number
TimestampUTC	1652366216500	This is a specific value when the observation is made. It has a Unix timestamp and is expressed with millisecond precision (Unix Time Stamp - Epoch Converter, n.d.).
Persons	It is a list, see Appendix 1, table 8 for an overview of these variables in the list.	Player information of every player on the field.
Ball	It is a list, see Appendix 1, table 9 for an overview of these variables in the list.	Ball information e.g. location and speed of the ball.

FootballContext	It is a list, see Appendix 1, table 10 for an overview of these variables in the list.	Game information for example where the yard line is from the snap, what play minute and what down it is.
DownMarkersContext	It is a list, see Appendix 1, table 11 for an overview of these variables in the list.	Markers information on the field e.g. LOS and LTG lines.
GameClockContext	It is a list, see Appendix 1, table 12 for an overview of these variables in the list.	Game time information for example what quarter they are in and what time it is. In addition, it is given per second.
MatchScoreContext	HomeScore: 30, AwayScore: 6	Match score information, gives the score of the match. In addition, it is given per second.
GameEventContext	IdfGameEvents: {Id: 1497, Name: kickoff_play}	Provides information about a particular action taking place on the field. In addition, this data is only detected if there is an event. For all events contained in the dataset, Appendix 2, table 13.

3.2 Data description for the main assignment

The data variables that are essential for developing the optimal running path for a player to the endzone are detailed in table 2. These data are characterized by a high level of detail and accuracy. This is evident as, with the simulation and the datasets used for the research, an American football game can be played out exactly as it would in real life within the simulation. Within this dataset, specific events are also included, representing important actions within American football. The events used for the main task are 'Run' and 'Touchdown'. 'Run' refers to an action where the player with the ball runs, while 'Touchdown' indicates that a player has scored a touchdown following a running play.

Table 2: The data variables used to develop the optimal run path to the endzone, with example and definition per variable.

Variables	**Example**	**Definition**
Timestamp	1699225574100	This is a specific value when the observation is made. It has a Unix timestamp and is expressed with millisecond precision (Unix Time Stamp - Epoch Converter, n.d.).
Player_Id	151	This is a player's ID, each player has them own ID in the dataset.

Ball_Id	2	This is an ID of the ball, the ball also has its own ID in the dataset.
Player_position	(3.045, 0, 19.818)	This is x, y and z coordinate. Where x is the length, y is the height and z is the width. The unit is in meters.
Speed	0.204	The magnitude of a player's velocity in m/s.
TeamSide	2	Which team the player plays for.
HasBallPossession	True	This is whether the player is holding the ball (True is fixed and False is not fixed).
Ball_position	(3.044, 2.088, 19.818)	This is x, y and z coordinate. Where x is the length, y is the height and z is the width. The unit is in meters.

In figure 4, a frame from a game is shown, with the variables (table 2) that are relevant for the main task. The positions of the players are visible through red and blue dots, while the position of the ball is marked with a black dot. This clearly indicates which players belong to the red team and which to the blue team. The frame is plotted over the dimensions of the field, which is 110 meters long along the x-axis and 50 meters wide along the z-axis, with an extra margin of 1 meter on the width to accommodate variations in field dimensions between different games. Central in this image is the middle of the field, indicated by the coordinates (0,0). Although more variables from table 2 are used for the models and calculations, they are not visible in figure 4.

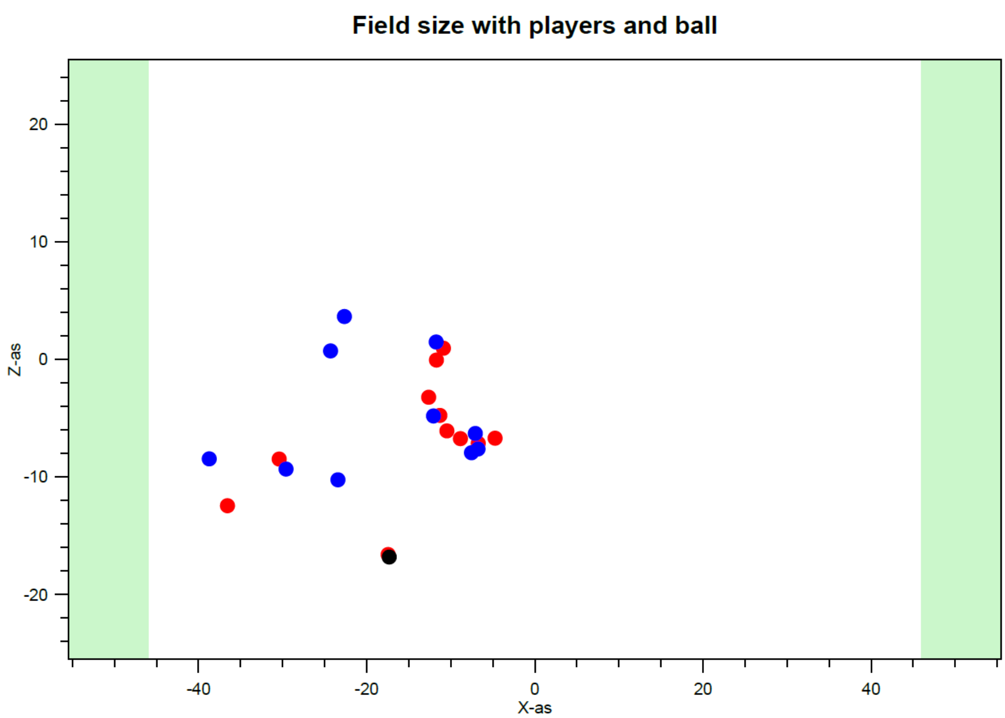


Figure 4: The data of player positions (red and blue) and ball positions (black) plotted on an x and z axis, in the dimensions of a field of American football. With endzones is light green.

4 Method

This chapter outlines the methodology of this research. First, the literature review is discussed. The review is divided as follows. Section 4.1 addresses previous research on this topic. Section 4.2 discusses the chosen model, including the selection criteria and the selection process. Finally, section 4.3 explains and discusses the applied method of the model on the main task.

4.1 Previous Research

The study "A Reinforcement Learning Based Approach to Play Calling in Football" explores a reinforcement learning-based method for making play decisions in American football. The research employs a Markov decision process to determine optimal choices at every level of the game, taking into account the collected data and probabilities of different game outcomes. This process aids in optimizing decisions in American football by anticipating the expected utility of various actions, such as running or passing, in different game situations. The paper presents a methodology for calculating these expectations and demonstrates how this can assist in making better decisions during the game, with the goal of maximizing the score. The approach is analysed with various examples and scenarios within the game, providing insights into how data analysis and machine learning can contribute to strategic decision-making in sports like American football (Biro, P. & G. Walker, 2021).

The decision-making process described in the article determines whether it is optimal to run or pass based solely on the current game state (down, distance to the first down, and distance to the end zone). However, it does not take into account the direction in which a pass should be made or the specific route that players should follow. In paragraph 1.2, the desired situation is described, which presents an interesting research question posed by Beyond Sports: "a model that can predict what a player with the ball should ideally do: throw, run, or hand off the ball." However, Beyond Sports asks for more; specifically, the direction in which the ball should be thrown or run. Given the limited time for this research and a lesser degree of expertise in reinforcement learning, it has been decided to focus on what the optimal running path to the endzone is for an American football player.

4.2 The Algorithms

Determining the optimal running path in American Football combines theory and data to establish the shortest route to the endzone, taking into account the positions and movements of both opponents and teammates. This process is described as a shortest path problem, with section 4.2.1 first explaining the general concept of the shortest path problem. Section 4.2.2 discusses the examined properties on which the algorithm was chosen.

The choice of the algorithm and the model for calculating influence weights are further elaborated in the following subsections. In section 4.2.3, the final choice of the algorithm is detailed, and in section 4.2.4, the model for calculating influence weights at the field is explained.

4.2.1 The Shortest Path Problem

The shortest path problem involves finding the fastest or least costly route between two points in a network, which consists of nodes connected by roads or links. Each connection has a 'weight' that can represent distance, time, or cost. This problem is commonly encountered in practical applications. For example, in navigation systems like GPS, which calculate the fastest route to a destination (W3Schools, n.d.).

4.2.2 The Examined Properties for the Algorithms

In the search for algorithms to approach the shortest path problem, with the goal of predicting the optimal running path in American football, the following properties have been examined: type of graph, optimality, efficiency, and implementation.

- A graph is a mathematical structure consisting of nodes (also called "vertices") and edges that connect these nodes. In the context of shortest-path algorithms, a graph is used to model networks, such as road networks (GeeksforGeeks, 2024). Graphs can be static or dynamic (type):
    - Static graphs: The structure and weights of the graph do not change during the calculation (Madkour et al., n.d.).
    - Dynamic graphs: The structure or weights of the graph can change during the calculation, for example, by adding new edges or nodes, or by changes in the weights (Madkour et al., n.d.).
- Optimality refers to the extent to which an algorithm is capable of finding the best solution. For shortest-path algorithms, this usually means finding the path with the least total distance (or cost) between two nodes:
    - Global optimum: The algorithm always finds the shortest path, regardless of the circumstances (Arora, 2004).
    - Heuristic: The algorithm uses estimates (heuristics) to speed up the search process. This can sometimes be at the expense of optimality, resulting in not always finding the absolute best path, but rather a very good path in a shorter time (Khodadadi et al., 2023).
- Efficiency pertains to the amount of computational power (time) and memory (space) that an algorithm requires to accomplish its task. The degree of efficiency: very efficient, highly efficient, efficient, less efficient (R & Ahmed, 2021).
- Implementation refers to the practical aspects of programming and executing an algorithm. The degree of implementation: simple, moderate, and complex (GeeksforGeeks, 2023).

The choice to evaluate algorithms based on these properties is based on the following reasons. The type of graph affects the difficulty of implementation; static graphs are simpler to implement than dynamic graphs. Choosing an easily implementable algorithm accelerates development time and reduces the risk of errors, leading to reliable and time-efficient solutions.

For this research, the main properties when selecting an algorithm are optimality and efficiency. Optimality is crucial because this research aims to find the optimal running path, which means always finding the best running path under the given conditions. Efficiency pertains to the processing time in combination with the size of the graph. Beyond Sports aims to run algorithms on live data, requiring processing time to stay within 0.1 seconds. For post-game predictions aimed at analysts, the processing time can be up to 5 seconds. Therefore, the goal is to use algorithms with high or very high efficiency.

4.2.3 The Choice of Algorithm

The following four algorithms, often used for solving the shortest path problem, are compared. Each is optimized for different types of graphs and scenarios:

- Dijkstra's Algorithm: Suitable for graphs with non-negative weights (Algorithm Examples, 2024).
- Bellman-Ford Algorithm: Can handle negative weights and detects negative cycles (Algorithm Examples, 2024).
- A* Algorithm: Uses heuristics for more efficient searches in large networks (R & Ahmed, 2021).
- D* Lite Algorithm: Ideal for dynamic environments, such as robotics (Jin et al., 2023).

In table 3, the chosen properties for each algorithm are displayed (GeeksforGeeks, 2023) (R & Ahmed, 2021) (Jin et al., 2023). When evaluating the two most important properties, efficiency and optimality, Dijkstra's algorithm and the D* Lite Algorithm emerge as the best options. Ultimately, Dijkstra's algorithm is chosen because its implementation is significantly simpler compared to the D* Lite Algorithm.

Table 3: The chosen algorithms are displayed in the table, along with the properties on which they are selected, for each individual algorithm.

Algorithm	**Type of Graph**	**Optimality**	**Efficiency**	**Implementation**
Dijkstra	Static	Global optimum	Highly efficient	Simple
Bellman-Ford	Static	Global optimum	Less efficient	Simple
A* Algorithm	Static	Heuristic	Very efficient	Moderate
D* Lite Algorithm	Dynamic	Efficient in dynamic	Efficient with dynamic changes	Complex

It is still possible to test other algorithms if the processing time of Dijkstra's algorithm proves to be too slow. This also applies to algorithms that have not been included in this paragraph. Additionally, various options can be considered to make Dijkstra's algorithm itself faster.

4.2.4 The Weight of the Nodes

In addition to the distance weights between nodes considered by Dijkstra, an extra weight is added to a node. This is to implement a model that takes into account the positions of opponents who can tackle the player with the ball, as well as teammates who can block these opponents. This model, called "space control," was developed by Fernández and Bornn (2018). It was originally applied to soccer and describes the area of influence of a team on the field.

Spearman (2016) defines "space control" as the probability that a team has possession of the ball at position x when the ball moves to position x. This concept can also be applied to American football, based on the principles on which the model is founded and explained.

An example of an outcome from the space control model by Fernández and Bornn, applied to soccer, is shown in figure 5. The probabilities and colours in figure 5 represent the chances for the red team.

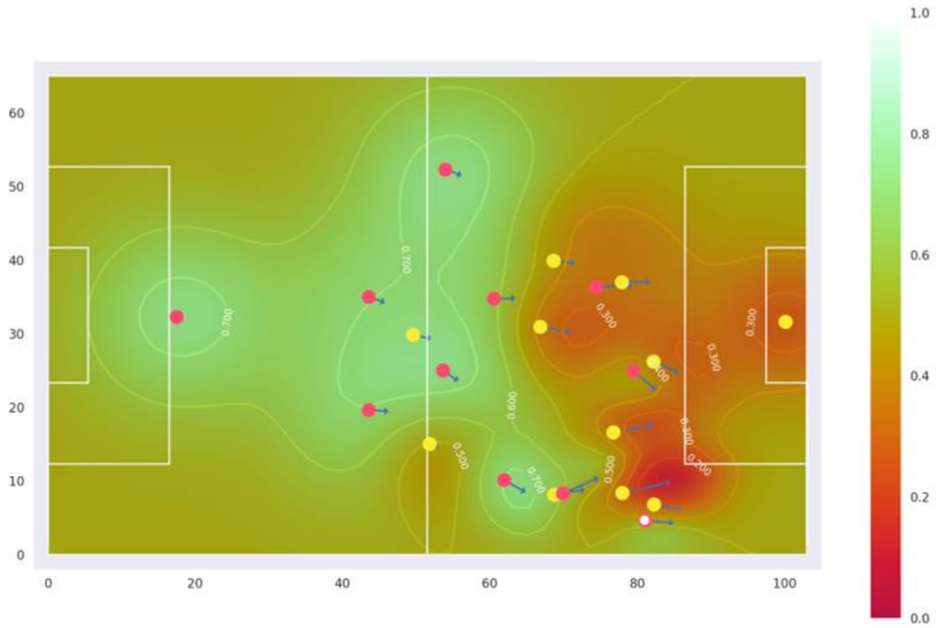


Figure 5: Example of a possible outcome from the space control model in soccer.

Note: Adapted from: Wide Open Spaces: A statistical technique for measuring space creation in professional soccer, by J. Fernandéz & L. Bornn, p. 6.

Before "space control" can be calculated, the "player influence area" for each individual player on the field is first determined. This is the probability that a player will have possession of the ball at position x when the ball moves to position x. In short, space control is the sum of all player influence areas on the field (Fernandéz & Bornn, 2018).

The exact method of calculating these values will be explained later. First, the principles on which the player influence area is based will be discussed. Player influence areas are based on the following reasoning (Fernandéz & Bornn, 2018):

- The player influence areas of players vary depending on their position relative to the ball. Players who are farther from the ball exert influence over a larger area because they have more time to anticipate the ball's movements. Conversely, players closer to the ball have less time for anticipation, resulting in a smaller player influence area.
- A player's speed is also crucial in determining the player influence area. A running player can exert greater influence in areas in the direction they are moving compared to when they are walking or jogging.
- Additionally, a player's influence is stronger in nearby spaces than in more distant areas.

The proportion between the weights of space control and the distance weights in Dijkstra's grid, and how this proportion is determined, will be discussed in the next paragraphs.

4.3 Application of the Models

This paragraph outlines the methodology of the models used for this assignment. First, a general explanation of the Dijkstra algorithm is provided (section 4.3.1). Next, the application of the Dijkstra algorithm in American football is discussed (section 4.3.2). Then, the method for the Space Control model is explained, including the transformations performed (section 4.3.3). Section 4.3.4 describes how the values of Space Control are integrated into the Dijkstra algorithm grid. Following this, section 4.3.5 discusses how the running path is visualized in a plot. Section 4.3.6 covers the turn limit constraint in the Dijkstra grid, and finally, section 4.3.7 explains how the optimal running path is evaluated.

4.3.1 General Functioning of Dijkstra's Algorithm

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may or may not have weights on the edges (GeeksforGeeks, 2024b). An example of a chosen path through the Dijkstra algorithm is shown in figure 6. In the Dijkstra algorithm, the search for the shortest path begins from starting point A with an initial distance of 0. At each step, the route with the minimal total distance to the next point is chosen, until the endpoint G is reached with the shortest possible route.

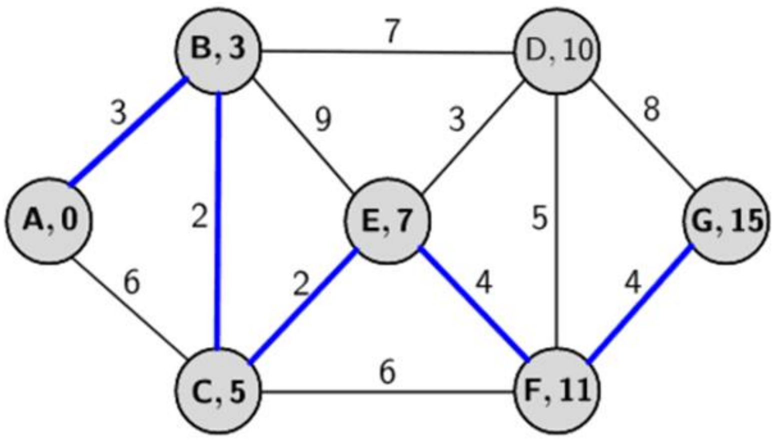


Figure 6: An example of a shortest path from point A to point G, chosen by the Dijkstra algorithm.

Note: Adapted from: Cambre, C. (n.d.).

The algorithm uses a priority queue to process the node with the lowest distance to the start point. The process works as follows:

1. **Initialization**: Set the distance to the starting point to 0 and to all other points to infinity. Mark all points as 'unvisited'.

2. **Choose the node with the lowest distance**: Initially, this is the starting point because the distance to itself is 0. Mark this point as 'visited'.

3. **Update distances**: For each directly connected, unvisited node of the currently processed node, calculate the total distance from the start point to this node via the currently processed node. If this distance is lower than the currently recorded distance, update the distance.

4. **Repeat**: Repeat steps 2 and 3 until all nodes have been visited or the minimum distance in the priority queue is infinity (which means that the remaining unvisited nodes are not reachable from the start point).

5. **Conclusion**: After processing all nodes, the distance record for each node contains the shortest distance from the start point to that node.

4.3.2 Dijkstra algorithm applied in American football

For this application of the Dijkstra algorithm, a grid is used as input, along with a pre-determined starting point and the endzone with a number of endpoints. Below is a more detailed description of each of these components:

For the application of the Dijkstra algorithm within the context of American football, a grid model has been developed that is placed over the playing field. This grid can be divided into various sizes, with smaller cells providing more accurate route determination but also increasing the number of nodes and thus the processing time of the algorithm. An example of a scaled grid model over an American football field can be seen in figure 7. The specific end zones are indicated in red and blue. The American football field is equipped with a grid, where each cell has a vertical or horizontal distance of 1 meter. Diagonal connections between the nodes have a distance of √2 meters. These measurements are based on the average stride length of a person, which ranges between 0.7 (walking stride length) and 1.5 meters (jogging stride length) (Henkny, 2014). A diagonal step is larger than a step taken straight forward or sideways. Each corner of the squares in the grid serves as a node. This can be seen in figure 8, where the dimensions are indicated by the lines and the blue dots represent the nodes (and the corner points). Figure 7 shows a smaller grid than what is actually used for the main task. For the main task, a grid is used that corresponds to the standard dimensions of an American football field: 110 meters long and 50 meters wide. The grid contains nodes, and these nodes are also located at the edges of the field, resulting in 111 by 51 nodes. This means there are a total of 5661 nodes within the Dijkstra grid. The Dijkstra grid can be represented as a matrix of 5661 by 5661, where each element in the matrix indicates the weight of the connection between each pair of nodes in the grid.

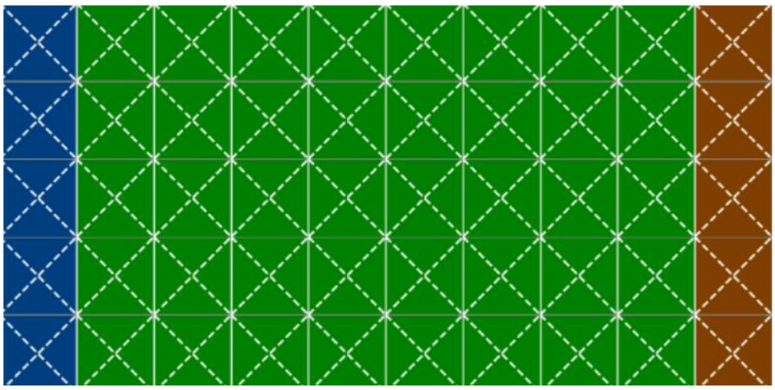


Figure 7 : American football field with the possible running lines for a player on the field, the red and blue squares are the endzones.

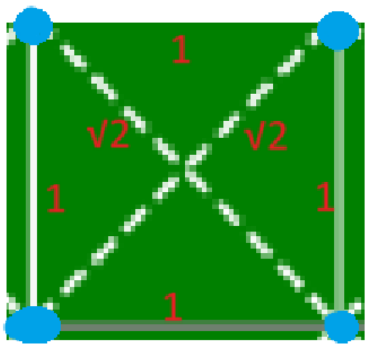


Figure 8 : A zoomed-in cell on the field with the dimensions per running line.

The starting point for the Dijkstra algorithm is the coordinate of the player in possession of the ball. This coordinate is translated into a node since only nodes exist within the Dijkstra grid. If the coordinate of the player with the ball falls between two nodes, the node closest to the player is chosen as the starting point.

Finally, the Dijkstra algorithm is used to calculate a route to a specifically indicated area on the playing field. This area, the endzone, is crucial because reaching or crossing this line results in a touchdown. As shown in figure 9, there are several nodes directly on the endzone line. The Dijkstra algorithm calculates an optimal route from the starting point for each of these nodes. The algorithm ensures that the routes of the player with the ball are directed to the correct indicated area. This is the area where the player with the ball needs to score at that moment. In the game, after halftime, sides are switched, causing the touchdown area to change.

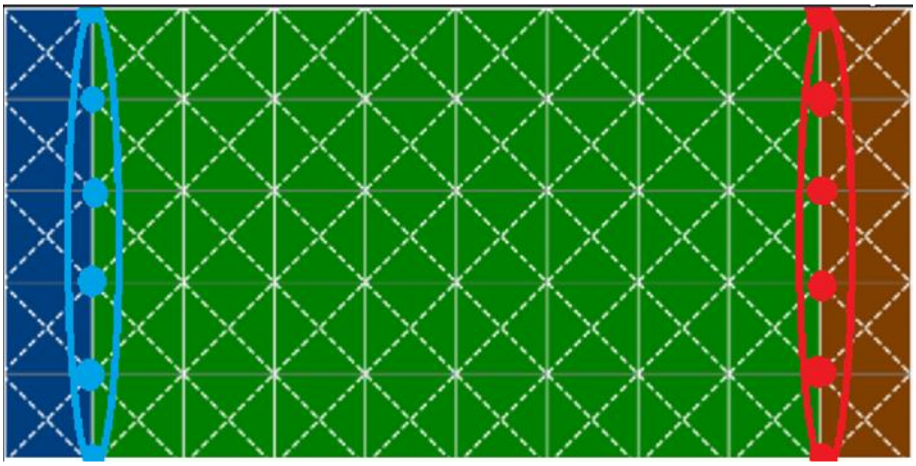


Figure 9: American football field with the possible running lines for a player on the field, the red and blue squares are the endzones. With the specified areas where the end nodes are located are indicated on the field.

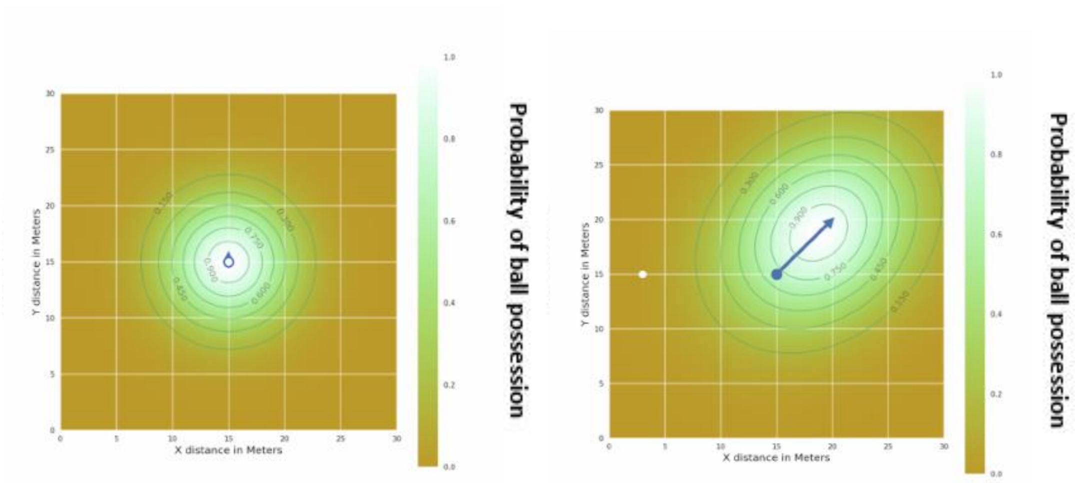
After determining the starting point, the Dijkstra algorithm calculates the shortest routes from this single point to all pre-specified nodes on the endzone. Although multiple routes are calculated, the goal of the algorithm is to select only one optimal route. Each calculated route is assigned a score, which in this case is based solely on the distance (additional weights will be added later). These scores are compared, and the route with the minimal score, is chosen as the optimal route. The result of the algorithm is a list of nodes that form this optimal route, making it clear which path the player should take to reach the endzone.

4.3.3 Space Control

In this section, the space control model is discussed. This model assigns a specific weight to each location on the field, indicating which team is likely to have ball possession at that particular spot. Initially, this model determines the player influence area of each player on the field.

The player influence area is defined by a multivariate normal distribution, the shape of which is adjusted by the player's location, speed, and relative distance to the ball, as shown in figure 10 (these are examples of player influence areas). The detailed method (and formulas) for this are described in the Space Dominance report (Heijerman et al., 2024, pp. 21-24) and the Fernández and Bornn (2018) report (pp. 18-19).

First, the player influence area function is explained (section 4.3.3.1), followed by how these player influence values are converted into Space Control values (paragraph 4.3.3.2). Next, the transformations performed in the Space Control model are discussed (paragraph 4.3.3.3), and finally, how the magnitude of these transformations was determined (paragraph 4.3.3.4).



(a) Player influence area of a player in possession of the ball, without speed.

(b) Player influence area of a player, 15 meters away from the ball, with a speed of 6.36 m/s.

Figure 10: An example of a player's player influence area with different speeds and ball distances.

Note: Adapted from: Wide Open Spaces: A statistical technique for measuring space creation in professional soccer, door J. Fernandéz & L. Bornn, p. 5.

4.3.3.1 Player influence area function

The player influence area is based on three principles: speed, distance to the ball, and nearby locations having a higher probability value (paragraph 4.2.4). The player influence area function, Formula 1, is designed to model the influence of a player's position on the field using the multivariate normal distribution. This function utilizes the variables $\mu(t)$ and $COV(t)$, which respectively determine the mean and the covariance matrix of the distribution of the player's position at a specific timestamp (t) (Heijerman et al., 2024, pp. 21).

Formule 1: $I_c(\mu\,, COV, L_c, p) = \dfrac{PDF(\mu, COV, L_c)}{PDF(\mu, COV, p)}$

$$I = The\ player\ influence\ area$$

$$PDF = \textit{Probability density function}$$

$$COV = Covariance\ matrix\ of\ the\ multivariate\ normal\ distribution\ of\ player\ positions$$

$$\mu = Mean\ of\ the\ multivariate\ normal\ distribution\ of\ player\ positions$$

$$p = The\ current\ coordinate\ of\ the\ player$$

$$L = Matrix\ of\ every\ coordinate\ on\ the\ field$$

$$c = A\ coordinate\ on\ the\ field$$

The formula takes as input L, which is a matrix of every coordinate on the field where the influence of a single player is assessed. This is a 111 by 51 matrix, corresponding to the number of nodes in the length and width of the field used in this study. L_c represents one coordinate on the field. In Formula 1, p is the current coordinate of the player.

First, a multivariate normal distribution is created with the variables $\mu(t)$ and $COV(t)$. This multivariate normal distribution is used to calculate the probability density at the given coordinate L_c, expressed as $PDF(\mu\ , COV, L_c\ )$, where "PDF" stands for the probability density function. The numerator of Formula 1 is given by $PDF(\mu\ , COV, L_c\ )$, representing the degree of influence at the specified locations according to the multivariate normal distribution. The denominator of the formula is given by $PDF(\mu\ , COV, p)$, representing the probability density of the player at his current coordinate. This formula provides a standardized measure of the influence (I_c) of the specified coordinate L_c relative to the current coordinate of the player $p(t)$, where the modeling is determined by the multivariate normal distribution with variables $\mu(t)$ and $COV(t)$. This is done for each coordinate (c) on the field, creating the player influence area (I), as well as for each player (Heijerman et al., 2024, pp. 21).

To calculate the values of the probability density functions, the mean $\mu(t)$ and covariance matrix $COV(t)$ of the multivariate normal distribution must first be determined. This can be found in the Space Dominance report (Heijerman et al., 2024, pp. 21).

In the calculation of the values of the covariance matrix $COV(t)$, the player influence area radius is included (source). The article by Fernández & Bornn (2018) assumes that the range of the influence radius in soccer is between 4 and 10 meters. This range has been adjusted because this assignment focuses on running paths. The ball only moves because a player is running, which means that the ball moves slower compared to when it is thrown or kicked. Therefore, the further a player is from the ball, the greater their anticipation time and thus their area of influence becomes. This range has been adjusted to an interval between 10 and 40 meters.

The influence radius depends on the distance to the ball. Thus, the distance between the ball and player must first be calculated for a given moment t. Figure 11 shows the relationship between the distance to the ball and the influence radius. With this function, the distance to the ball can be converted into the influence radius. Then, the influence area is determined with the influence radius and subsequently corrected by the speed fraction and direction of movement. This is explained in detail in the Space Dominance report (Heijerman et al., 2024, pp. 21-23).

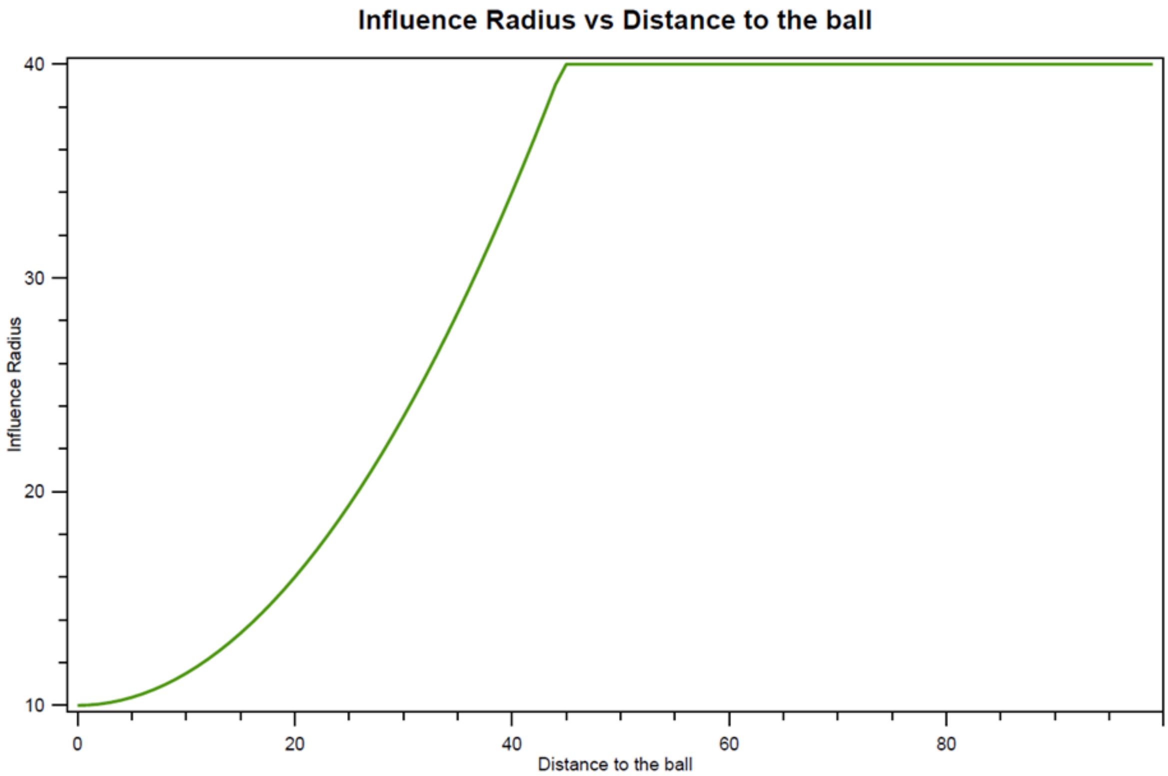


Figure 11: The transformation function is plotted; with the distance to the ball, the influence radius of the player influence area is determined.

4.3.3.2 Space Control Values

In paragraph 4.3.3.1, the determination of the player influence area is described. The player influence area (I) is represented as a 51 by 111 matrix, containing 5661 values. Each value indicates the probability of ball possession for a player at a specific location c on the field at a moment t during the match. Such a value is also referred to as an influence value of a player. With the influence values for location c from all players on the field, the space control for location c is determined for one of the teams, namely the defending team. In the match, there is an attacking team, the team with the ball, and a defending team, the team without the ball. Formula 2 is used to determine the space control ($SC\_Defence_c$) for the defending team for a location c on the field. The letter i under the first summation sign represents the players from the defending team. The letter j under the other summation sign represents the players from the attacking team. Once the space control for the defending team for each location c is determined, the output is placed in a 51 by 111 matrix. This represents, for each location c, the probability of ball possession for the defending team at time t (Heijerman et al., 2024, pp. 23).

The logistic transformation, without considering the yellow and blue marked factors, ensures that the value of space control falls within the range [0, 1]. This was done to create a probability value for space control. Subsequently, transformations were performed to convert these probability values into weights (or score) of a certain magnitude for the Dijkstra grid.

Formule 2: $SC\_Defence_c = \frac{1}{1 + e^{-1.3(1.3\sum_i I_c - \sum_j I_c)}} + 0.5$

$$SC_{Defence} = Space\ Control\ for\ the\ defending\ team$$

$$c = A\ coordinate\ (loacation)\ on\ the\ field$$

$$I = The\ player\ influence\ area$$

$$i = Represents\ the\ players\ from\ the\ defending\ team$$

$$j = Represents\ the\ players\ from\ the\ attacking\ team$$

The first transformation is a translation of +0.5 (marked in blue) in the formula. This adjustment changes the range to [0.5, 1.5]. The reason for this translation lies in how the space control matrix is combined with the matrix (the grid) of the Dijkstra algorithm. The values of space control are multiplied by the corresponding distance weights (the values in the Dijkstra matrix). These two matrices have different structures: one is a 51 by 111 matrix, while the other is a 5661 by 5661 matrix. How these multiplications take place will be explained later in this chapter.

The translation of +0.5 is done to equate the minimum values of the Dijkstra grid (which only contains the distance weights before multiplication) with the neutral value of the space control matrix. With this translation, the neutral number is now 1, meaning there is no prevailing influence for either team in the range of space control output values.

In the Dijkstra algorithm grid, the vertical and horizontal distances are 1, which are the minimum values in the grid. This means that these numbers are now equal, and the entire field starts with a weight of 1. Anything above 1 in the Dijkstra grid is additional weight and an addition to the distances. This is because the player with the ball (who belongs to the attacking team) must avoid the defending team as much as possible to avoid being stopped. Hence, the weights for the positions on the field where defensive players are located increase in the Dijkstra grid.

Additionally, there can be values that fall within the range of [0.5, 1], which means a decrease in weight. This indicates that the attacking team has more influence on that specific part of the field. For the Dijkstra algorithm grid, the space control values of the teammates (the attacking team), which lie between [0.5, 1], are brought to 1. This is done because otherwise, the Dijkstra algorithm would prefer this node due to the lower weight than the minimum distance weight. A route past a teammate should not be more advantageous than a route where no one (neither a teammate nor an opponent) is present; this should not make a difference. A weight between [0.5, 1] can occur, for example, if a teammate is alone in that part of the field or has a larger influence area in that part of the field than the opponent.

4.3.3.3 Transformations of Scale Factors in the Formula

Two important transformations have been applied in formula 2, namely a multiplication with two scale factors (highlighted in yellow in Formula 2). These transformations were added to generate more realistic routes. Here is an explanation of both scale factors.

The first scale factor pertains to the values within the parentheses. The influence values of the defending team are increased by 30%. This is because the attacking team is allowed to block the defending team but not tackle them, while the defending team can tackle the player with the ball (International Federation of American Football, 2023). The rationale for the magnitude of this number will be further explained in the next paragraph. As a result, the player influence values for the defending team carry more weight than those for the attacking team.

The second transformation is the addition of a scale factor of 1.3 to the logistic transformation (outside the parentheses in formula 2). This factor affects the rate at which the function increases or decreases. Specifically, the factor 1.3 ensures that the rise of the logistic function is faster than with the standard formula without the factor. This means that the transition from 0.5 to 1.5 is steeper and occurs over a smaller range of the sum of the player influence values (I_c), as shown in figure 12. This adjustment ensures that the space control values ($SC\_Defence_c$) reaches their maximum or minimum weight more quickly.

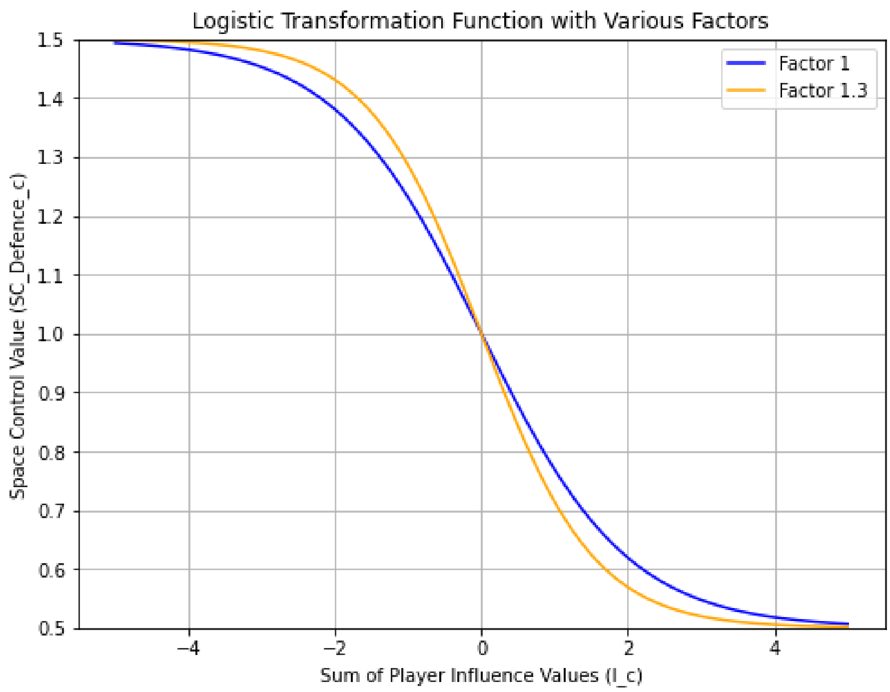


Figure 12: The logistic transformation plotted, to get the sum of player influence values within an interval of space control value. With the difference when a factor is added to this transformation.

4.3.3.4 Determining the Magnitude of the Transformations

For the evaluation of the optimal running path, with input from experts at Beyond Sports who specialize in American football, the routes were made realistic by adjusting the magnitude of the scale factor transformations, the range of the influence radius, and the space control values. In figure 13, a real game situation from a match is shown in the Beyond Sports simulation. Here, the player receives the ball, and the black line in figure 13 represents the successful touchdown run. The model predicts the optimal running path to the endzone by adjusting the magnitude of the transformations and the range to match the prediction with the successful touchdown run.

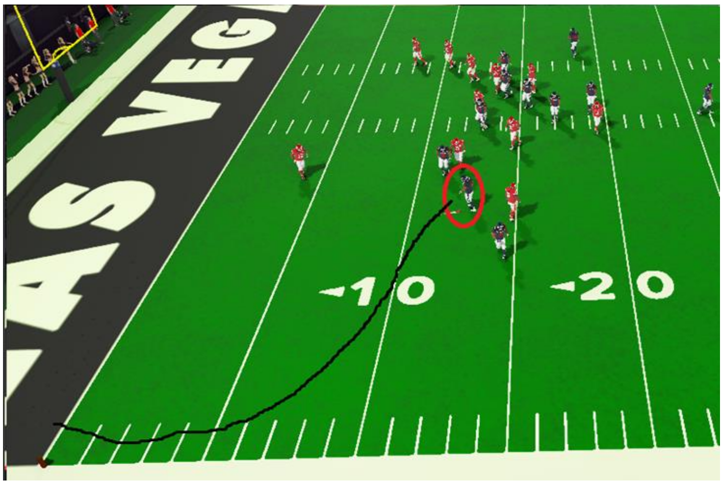
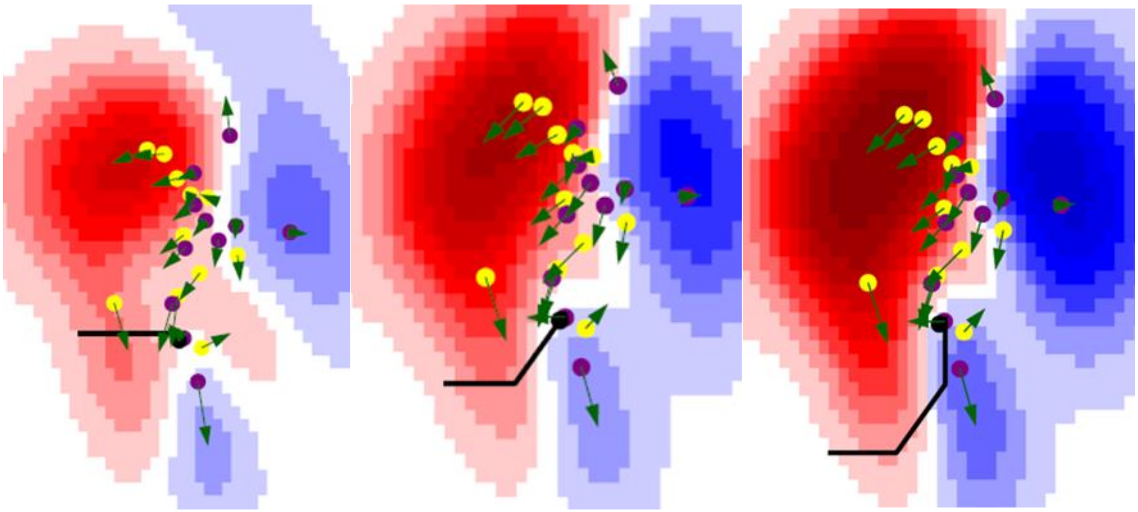


Figure 13: The successful touchdown run (black line) for the player with the ball (circled in red) in a real American football game.

Note: Adapted from: The simulation of Beyond Sports

This process is illustrated in figure 14, where the red heatmap represent the dominant areas of space control for the defending team and the blue heatmap represent the dominant areas of space control for the attacking team. The yellow dots indicate the players of the defending team, while the purple dots indicate the players of the attacking team in this case. The black line represents the predicted running path and starts at the player with the ball (black dot). Figure 14a shows the outcome of the initial step without any transformations or changes in the range of the space control model. Figure 14b shows the outcome of a slight change in the range and the height of the scaling factor of formula 2 (the space control formula). Figure 14c shows the outcome of the adjustments in range and scaling factor as explained in this chapter. Multiple successful touchdown runs were analysed, and the transformations were adjusted accordingly, with the condition that previously analysed situations did not deteriorate but remained the same or improved. This approach ensures that the model's predictions are not only realistic but also consistent with real game situations, as evaluated by experts.

Through this process, the range of the space control values for the defending team was adjusted. These values were increased from the range [1, 1.5] to [1, 4]. This range pertains to the values that are added as extra weight, in addition to the distance weights, to the grid of the Dijkstra algorithm.



a) Shows the outcome of the initial step without any transformations or changes in the range of the space control model.

b) Shows the outcome of a slight change in the range and the height of the scaling factor of formula 2 (the space control formula)

c) Shows the outcome of the adjustments in range and scaling factor as explained in this chapter.

Figure 14: Three figures, where the running path (black line) to the endzone is plotted on the field with the defending team (yellow dots) and attacking team (purple dots) for the player with the ball (purple dot with the black dot). The red heatmap represent the dominant areas of space control for the defending team and the blue heatmap represent the dominant areas of space control for the attacking team.

4.3.4 Converting Space Control-Matrix to Dijkstra Algorithm Grid

This paragraph explains how the summed and transformed Space Control weights (also known as $SC\_Defence$) and distance weights are combined. As previously mentioned, the $SC\_Defence_c$ values are multiplied by the corresponding distance weights. The $SC\_Defence_c$ values are in an $SC\_Defence$-matrix of 51 by 111, while the distance weights are already incorporated into a grid (or matrix) of the Dijkstra algorithm in a 5661 by 5661 matrix.

To illustrate, here is an example with smaller matrices; the principle is the same for the larger matrices in this study. The $SC\_Defence$-matrix (table 4) is a 2 by 3 matrix, where the rows represent points on the x-axis and the columns represent points on the z-axis, with a step size of 1. In this case, the field is 2 meters long (x-axis) and 1 meter wide (z-axis), with 6 nodes.

In table 5, the distance matrix, which is the grid format of the Dijkstra algorithm, is shown. In table 6, these two are combined. For example: node 1, row 1 column 0 in table 4, has a value of 1.3. This value is multiplied by the distance when the player moves from node 0 to node 1.

Table 4: Small-scale example of a $SC\_Defence$ $matrix$.

1	1.2
1.3	3.2
2.2	1.8

Table 5: A small-scale example of the Dijkstra algorithm grid with only the distance weights between the nodes.

From node ↓/to node →	**0**	**1**	**2**	**3**	**4**	**5**
0	0	1	0	1	√2	0
1	1	0	1	√2	1	√2
2	0	1	0	0	√2	1
3	1	√2	0	0	1	0
4	√2	1	√2	1	0	1
5	0	√2	1	0	1	0

Table 6: A small-scale example of the Dijkstra algorithm grid with the distance weights and the $SC\_Defence_c$ values combined between the nodes. Here, the $SC\_Defence_c$ $values$ are added to the grid.

From node ↓/to node →	**0**	**1**	**2**	**3**	**4**	**5**
0	0	1.3	0	1.2	√2*3.2	0
1	1	0	2.2	√2*1.2	3.2	√2*1.8
2	0	1.3	0	0	√2*3.2	1.8
3	1	√2*1.3	0	0	3.2	0
4	√2*1	1.3	√2*2.2	1.2	0	1.8
5	0	√2*1.3	2.2	0	3.2	0

4.3.5 Simple Visualization of Optimal Running Path

To evaluate the optimal running path to the endzone calculated by the Dijkstra algorithm, this path is visualized on the field as shown in Section 3.2, figure 4. A heatmap is plotted over the field. This heatmap provides a visual representation of the areas on the field controlled by a team based on space and positions (space control).

Each player is associated with a team, with one team represented by a red heatmap colour (the defensive team) and the other team by a blue heatmap colour (the offensive team). The heatmap is based on the values in the space control matrix (paragraph 4.3.3.2), which show the range [0.5, 1.5], before it is adjusted for the Dijkstra algorithm grid.

The optimal running path to the endzone, calculated by the Dijkstra algorithm, is then plotted as a black line in the visualization. The plot of this line uses space control values in the range of [1, 4]. This path is initially given by the Dijkstra algorithm in nodes, which are then converted into coordinates on the x,z-coordinate system so that the path can be plotted. Finally, a smooth function called Savitzky Golay filter (Gallagher et al., 1964) is applied to the optimal running path to obtain a smoother visualization.

4.3.6 Turn Limit Condition to the Dijkstra Grid

The purpose of these conditions in this sub-paragraph is that it works in the algorithm to get more realistic paths. The Dijkstra algorithm now calculates a route for the player with the ball from a still image, as shown in figure 15. In figure 15, the player with the ball is represented as the yellow dot with the black dot (the ball). The green arrow represents the velocity vector in the direction of movement, indicating the speed at which the player is running at that moment, expressed in meters per second. In figure 15, the player with the ball has a speed of 6.6 meters per second. It is unrealistic for a player at this speed to make a turn of 90 to 180 degrees in one go, without causing any delay in their declaration (Dos'Santos et al., 2021). Therefore, a condition has been added to the grid of the Dijkstra algorithm.

For this conditions, a simple approach was chosen. This means that for a given running speed, from the moment the running path is calculated, a turning limit is introduced, as shown in table 7. The data below are derived from source BSc (2023) indicating that the average adult jogger has a speed of approximately 1.67-2.22 m/s. For this approach, it is assumed that about 1 meter is needed if a player wants to make a turn greater than 90 degrees. This interval from the source (BSc, 2023) has been rounded, as shown in table 7, and from there, a new interval has been determined step by step with a logical condition.

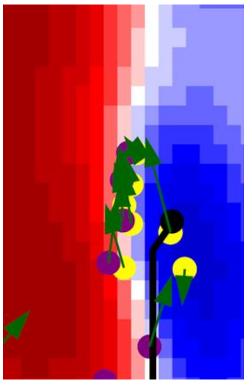


Table 7: Turn Limit Table, indicating which speed intervals have a turn angle limit and the angle within which the player can still turn. For the first three nodes.

Speed interval (m/s)	Turn Angle Condition (degrees)
1.5 - 2.5 m/s	<90
2.5 - 3.5 m/s	<67.5
3.5 - 4.5 m/s	<45
>4.5 m/s	<= 22.5

Figure 15: A running path prediction without a turning limit relative to speed (the speed is here 6.6 m/s).

The steps for the turning angle condition are set at 22.5 degrees. This is because the distances between different nodes, as chosen in the method, can place a player's speed vector in a certain direction precisely between two possible paths. This is clearly visible in figure 16: the player in purple, at the bottom right corner, is moving faster than 4.5 m/s and has a certain speed vector (red). This player is therefore between two possible paths between two nodes. If the turning angle condition in table 7. had been chosen to be less than 22.5 degrees, the algorithm would not be able to choose the next node.

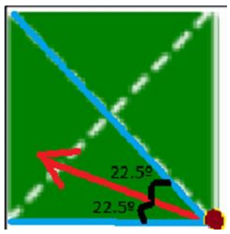


Figure 16: The two paths (indicated in light blue) that a player (purple) can choose when moving at high speed (red vector). The direction of movement (red vector) can fall exactly between these two paths, forming an angle of 22.5 degrees.

These conditions listed in table 7 was implemented for the first three nodes, which amounts to 3 meters from the starting point. For the following three nodes, a newly calculated speed is used. This is done because the player must slow down during the first meter, the first two meters or the first three meters if the player wants to turn and go the other way. The basic formula from kinematics (Toppr, 2019) has been rewritten to calculate the new speed for node 1, 2 or 3, resulting in Formula 3. The value -5.7 in the formula represents a constant deceleration, and d represents the distance travelled. This constant deceleration is taken from the following source STATSports (2021), which is based on research on professional soccer players but remains a simple approach in this study. For the speed at node 1, d is equal to 1 meter, for node 2, d is equal to 2 meters, and for node 3, d is equal to 3 meters. All negative results in the formula are set to 0, as a negative number is not possible under a square root.

Formule 3: $new_{Speed} = \sqrt{InitialSpeed^2 + 2 * (-5.7) * d}$

$$new_{speed} = The\ new\ speed\ when\ a\ player\ decelerates\ with\ this\ rate$$

$$InitialSpeed = The\ speed\ of\ the\ player\ at\ the\ moment\ of\ the\ prediction$$

$$d = The\ distance\ covered$$

4.3.7 Evaluation Method of the Optimal Running Path

This section describes how the optimal running path is evaluated. First, the researcher and experts from Beyond Sports, who understand American football, evaluated whether the routes are realistic compared to real game situations. Then the successful touchdown runs from different games are compared with the model's predicted paths. It is assumed that the successful touchdown run is always the most advantageous route to endzone. A total of 20 different touchdown runs from different games are analysed and compared to the predicted routes, resulting in a certain prediction percentage based on these 20 runs. This number of 20 was chosen because larger numbers take more time to analyse and, in addition, at this number can still yield a significant prediction percentage.

5 Results

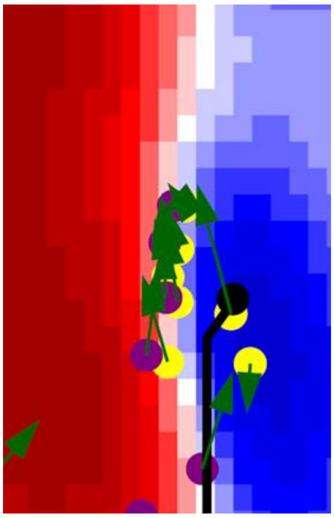
In this chapter, the result of the predicted running path to the endzone is compared with successful and unsuccessful touchdown runs in an actual game. Additionally, a hypothetical scenario is outlined, followed by a discussion of the requirements for the final product and a presentation of the final product.

5.1 Evaluation

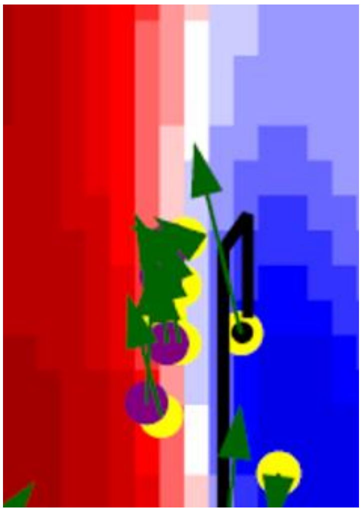
In this paragraph, we first examine whether the turning limit condition results in a more realistic path. Then, we compare the predicted running path with a successful touchdown, and finally, we compare the predicted running path with an unsuccessful touchdown.

5.1.1 Evaluation Turn Limit Condition

A turning angle condition has also been implemented, as described in paragraph 4.3.6 of the Method chapter. The reason for this is to predict more realistic paths. In figure 17, two situations are shown that illustrate this condition. In figure 17a, the turning angle condition has not been applied. As a result, the turn is too sharp to be made in one go at the given speed (6.6 m/s), resulting in an unnatural path. In figure 17b, the turning angle condition has been applied. Here, due to the high speed preventing an immediate turn, a straight movement is made first. This demonstrates that the turning angle condition is effective, as the movement becomes more realistic, and a player cannot make a sharp turn at high speed in one go.



a) Turning angle condition not applied in the prediction of the running path. The speed of the player with the ball (yellow dot with the black dot) is 6.6 m/s.



b) Turning angle condition applied in the prediction of the running path. The speed of the player with the ball (yellow dot with the black dot) is 6.6 m/s.

Figure 17: Two figures, where the running path (black line) is plotted on the field, with opponents and teammates, for the player with the ball (yellow dot with the black dot). Figure a is without turn condition and figure b is with turn condition.

5.1.2 The Predictive Model vs Successful Runs

After determining the magnitude of the scale factors, the range of the influence radius, and the space control values transformations based on various successful touchdown runs from real games (subsection 4.3.3.4), multiple successful runs were analysed. These analyses aimed to determine how many successful touchdown runs were correctly predicted by the model. Here, as previously mentioned, it is assumed that the successful touchdown run is always the most advantageous route to the endzone.

In the figure 18, example of a successful touchdown run in an American football game can be seen, with the black line representing the run. To the right of this figure is the model prediction for the game on the left, shown in figure 19. In figure 19, the yellow dots represent the defensive team, and the purple dots represent the offensive team. The red heatmap always represents the dominant areas of space control for the defensive team and the blue heatmap always represents the dominant areas of space control for the attacking team. In this situation, the model predicted the same run that occurred in the game.

For this analysis, 20 different successful touchdown runs from various games were examined. Of these runs, 16 closely matched the model's predictions, as determined by visual inspection. This indicates that the model accurately predicted 80% of the successful touchdown runs analysed.

In Appendices 3, 4 and 5, three additional situations (like shown in figures 18 and 19) can be seen that have been compared with each other. These scenarios have been added to illustrate the predicted running paths from various distances to the end zone or different positions on the field (such as from the sidelines or from the centre).

Figure 18: The successful touchdown run (black line) for the player with the ball (circled in red) in a real American football game.

Note: Adapted from: The simulation of Beyond Sports

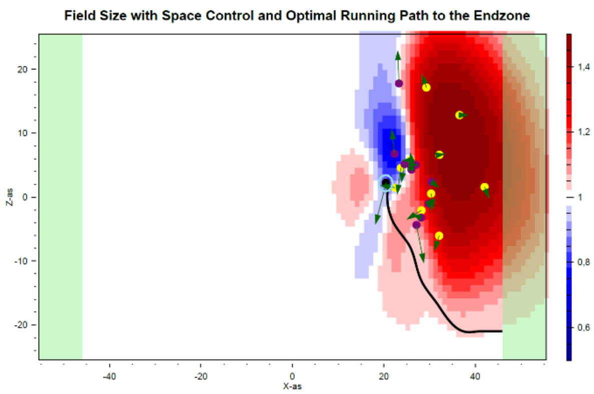


Figure 19: The prediction of the optimal running path to the endzone for an American football player with the ball (circled in light blue). The yellow dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The purple dots represent the attacking team with a blue heatmap represents their space control on the field.

5.1.3 The Predictive Model vs Unsuccessful Runs

Finally, runs were also analysed where the player was intercepted on the way to the 10-yard line or the endzone, classified as a failed run, while the predictive model predicted a different running path. Below, two situations of a failed run in real life are discussed, providing insight into the model's behaviour and the decisions it makes.

Most successful touchdowns that have been viewed go around the outside. However, when the outside is too crowded, these attempts often fail, indicating that the player should have chosen a different route. The model was also analysed for this, as it is important that it not only suggests routes along the outside but identifies where the most space is. An example of such a situation is shown in figure 20.

In figure 20, the player with the ball (circled in light blue) is shown, and the black line indicates what the model chose as the optimal route to the endzone. The pink cross shows where the player went and was stopped. In this situation, the yellow dots are the attacking team, and the purple dots are the defensive team. This comparison shows that the model does not always choose the outside. In some situations, the model advises going to the middle, as this may offer a better chance of advancing. This is illustrated by an example where a player in a real situation wanted to go around the outside, but the model suggested going to the middle, which could ultimately be a more successful option.

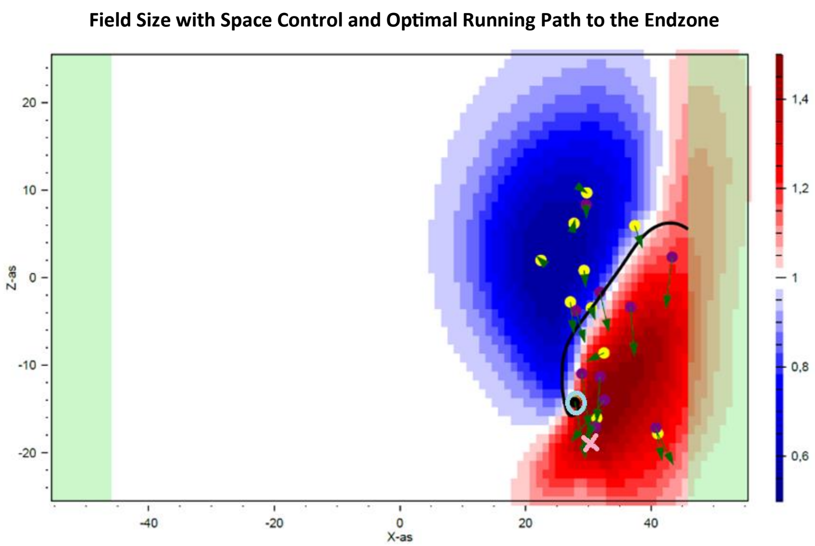


Figure 20: The prediction of the optimal running path to the endzone for an American football player with the ball (circled in light blue). The purple dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The yellow dots represent the attacking team with a blue heatmap represents their space control on the field. The pink cross on the field marks where the player was stopped in the actual game.

In figure 21, the situation is reversed. Here, the yellow dots are also the attacking team, and the purple dots are the defensive team. The player with the ball (circled in blue) starts in the middle of the field. In this situation, the player chooses to go through the middle, where it is crowded, and is stopped at the location of the pink cross. However, the model predicted a path via the outside, where the chances might have been greater for the player to advance towards the end zone.

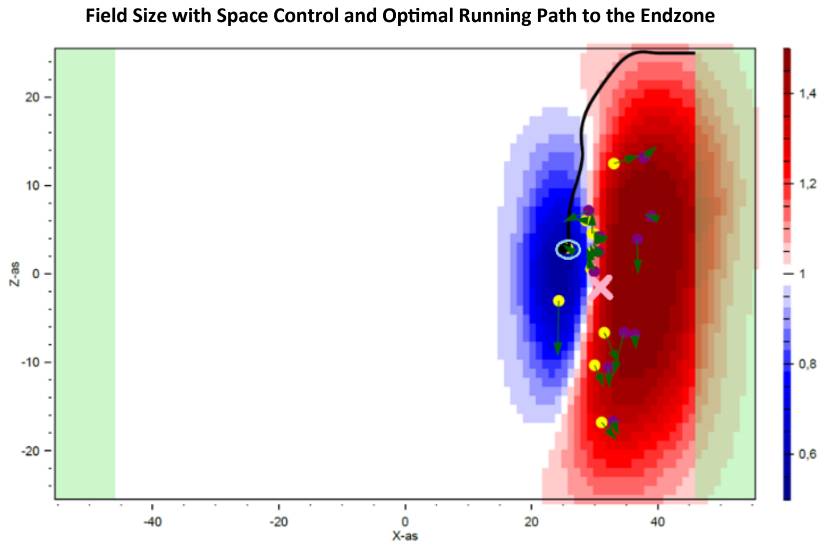


Figure 21: The prediction of the optimal running path to the endzone for an American football player with the ball (circled in light blue). The purple dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The yellow dots represent the attacking team with a blue heatmap represents their space control on the field. The pink cross on the field marks where the player was stopped in the actual game.

5.2 The Hypothetical Scenario

In this paragraph, a hypothetical scenario is outlined (figure 22) to demonstrate to Beyond Sports how this model can be applied in the desired situation described in the introduction. The desired situation: a model that can predict the best action for a player with the ball: to throw, run, or pass the ball. If the player chooses to throw, in which direction or to which player?

In the actual game, the player with the ball (blue circle) decides to run. However, the player is quickly stopped, as indicated by the light-green cross in the diagram. In the outlined scenario, the player would have been better off throwing the ball (light-green line) to the teammate indicated by the orange circle. From this player, an optimal running path is predicted (black line), increasing the likelihood of scoring a touchdown, as this player has much more space to advance.

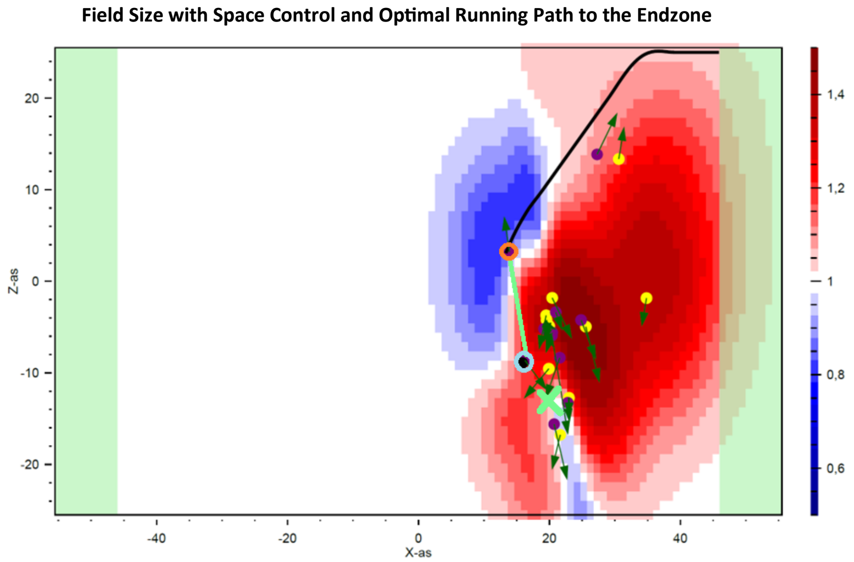


Figure 22: This is the hypothetical scenario; it shows the player with the ball as a blue circle. The light green line represents the hypothetical throw to the player circled in orange, and the black line indicates the optimal running path. The green cross marks the spot where the player was stopped.

5.3 Final Product

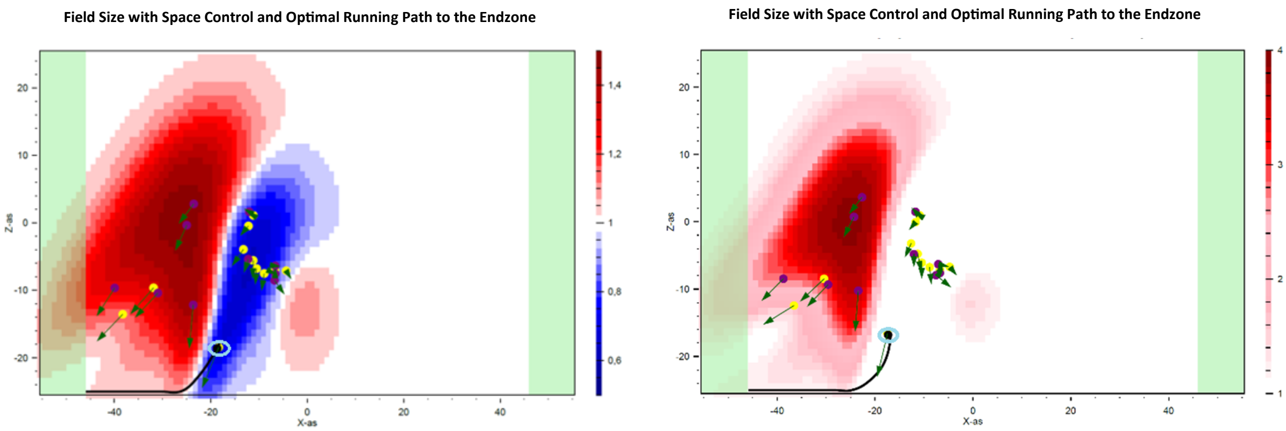
The final product is a model that can predict the optimal running path to the endzone for an American football player with ball. Figure 23 shows how the same optimal running path for the same frame is plotted in two different figures.

As explained in paragraph 4.3.5, there is a space control matrix for the heatmap with a range of [0.5, 1.5]. This shows which team has control and how much control the team has over a specific part of the field. For both plots in figure 23, the running path uses a range of space control values from [1, 4]. These values, within the range of [1, 4], represent the actual weights used for space control in the Dijkstra grid. Figure 23a shows the optimal running path with space control values for the heatmap within the range of [0.5, 1.5]. This visualization displays space control in colour for both teams. Conversely, figure 23b shows the same optimal running path but with space control values in the range of [1, 4]. This is done to demonstrate how the space control values in the Dijkstra grid (used to select the optimal running path) appear when plotted in a heatmap. These visualizations are kept simple to facilitate the analysis and presentation of the model to the client. It is up to the client to implement the calculation of the prediction model into the Beyond Sports simulation.

The model had to meet several specific requirements, which were thoroughly checked. The first requirement from the client was that the processing time of the predictive algorithm should not exceed five seconds. Additionally, the client envisioned a scenario where the developed algorithm could function during a live match. This would enable an analyst to display a visualization of the predicted optimal running path at any given moment during a live match. To achieve this, the processing time should not exceed 0.1 seconds. On average, the processing time of this model is 0.212 seconds per frame, which means the model meets the specified requirement but not the desired scenario for live use.

The second requirement from the client was that the outcome for the running path had to be realistic compared to what happens in practice. To verify this, experts from the client reviewed various visualizations of the predicted optimal running path. As described in the previous paragraph, 20 different successful touchdown runs were compared to the model's predictions. This approach revealed that the model correctly predicted approximately 16 out of 20 routes, indicating that the model has a degree of accuracy in predicting realistic running paths.

The third requirement from the client was that the calculations of the predictive algorithm must work on any data frame. This was verified by loading a full match and ensuring that the software did not produce any error messages. Additionally, various situations from different matches in the dataset were visualized, each time displaying a predicted running path. In every visualized situation, no errors in the software code were discovered.



a) Space control values where the range is between [0.5, 1.5] for the heatmap. Here, the space control per team is visible.

b) Space control values (weights) where the range is between [1, 4]. This is the space control values in the Dijkstra grid.

Figure 23: The prediction of the same optimal running paths for the same situation is illustrated, where figure 23a shows the space control of both teams, and figure 23b displays the space control values in the Dijkstra grid.

6 Conclusion

In the introduction, the following main task has been formulated:

Developing a model that can predict the optimal running path to the endzone for an American football player with the ball.

To carry out this main task, sub-questions have been formulated. In this chapter, we will first answer the formulated sub-questions. Finally, the end result will be presented, and the requirements of Beyond Sports will be discussed.

6.1 Optimal Running Path Model

To determine what is needed for developing a model to predict the optimal running path to the endzone, six sub-questions have been formulated. By answering these sub-questions, this main question can ultimately be answered.

What are the criteria for defining an optimal running path? To define an optimal running path to the endzone for an American football player with the ball, various criteria are used. The goal is to find a path that, taking into account the influence of opponents (defensive team) and teammates (attacking team) on the field and the distances between different points, has the lowest total cost. The influence of opponents and teammates, as well as the distances, are expressed as weights distributed across the entire field. The total cost is calculated as the sum of these weights. The end zone is the target towards which the path is chosen. The optimal path minimizes the total cost, and ultimately, this path with the lowest total cost is chosen as the optimal path.

What data is needed to determine the optimal running path? The required dataset contains the following data, which will be used to predict the optimal running path. The x and z coordinates of players and the ball per timestamp, as well as the speed variables per player and a variable indicating which player has the ball. Additionally, a variable is used to indicate which team a player belongs to. The timestamps are recorded at 10 frames per second.

What model is needed to determine the optimal running path? To determine the most optimal running path for an American football player, a model is needed that finds the shortest path in a field with various weighted weights. This problem can be solved using the Dijkstra algorithm, which determines the path with the lowest total cost by taking into account both the influence weights of the players on the field and the distance weights between nodes.

The influence weights for the players on the field are calculated by the space control model. This model assigns a specific weight to each location on the field, indicating which team can have ball possession at that specific spot. Ultimately, the space control model calculates these scores for every location on the field. First, the player influence area of each player on the field is determined, which is used to calculate the influence weight of each individual player. The player influence areas of all players are then combined, forming the space control model.

How can the model account for realistic scenarios in American football? The model takes into account realistic scenarios in American football by using the player influence area variable, which considers the distance of players from the ball and their speed. Players farther from the ball have a larger player influence area because they have more time to anticipate the movement of a player running with the ball. Players closer to the ball have a smaller player influence area due to less reaction time. Running players exert more influence in their direction of movement compared to walking or jogging players.

Additionally, the space control model assigns a heavier weighting factor to the player influence area of the defensive team. This is because the offensive team is allowed to block the defensive team but not tackle them, while the defensive team is allowed to tackle the player with the ball. Therefore, the influence weights of the defensive team carry more weight in the model, providing a more realistic representation of game influences.

Moreover, a turn limit condition has been introduced, which stipulates that for a certain running speed, from the moment the running path is calculated, a turn limit is applied. This specifically applies to the first 3 meters from the player's starting point. It is unrealistic to expect a player who is already at full speed to abruptly turn 90 to 180 degrees. By integrating this condition, abrupt turns are limited for certain speeds in the first 3 meters of the path prediction, resulting in more realistic running paths.

What variables are needed to determine the optimal running path? The variables required for the Dijkstra algorithm model are as follows:

- **Nodes:** Represent the different positions on the field, including the starting point, possible intermediate points, and the endzone.
- **Connections between nodes:** Represent the possible paths along which the player can move. These connections have distance weights, which indicate the distances between the nodes.
- **Player influence area variable:** Indicates the influence weight of an opponent or teammate, with higher weights indicating greater influence (and thus more resistance).
- **Source:** The starting point of the player with the ball.
- **Target:** The endzone where the player needs to reach.

How is a single point chosen, and thus a single route for an American football player with the ball determined? The goal in the model is the endzone, where the player can make a touchdown. Various nodes are located on the line of the endzone. The Dijkstra algorithm calculates the route with the lowest total cost to these points. These costs are then compared, and the route with the lowest cost is chosen as the optimal route.

6.2 Final Product

The assignment involved the development of a model, with corresponding calculations, that can predict the optimal running path to the endzone for an American football player with the ball. The results of the implementation needed to be visualized, as shown in figure 24. The red heatmap represents the dominant areas of space control for the defensive team (yellow dots) and the blue heatmap represents the dominant areas of space control for the attacking team (purple dots). Beyond Sport had several requirements that needed to be met.

The first requirement was that the processing time of the prediction algorithm should not exceed 5 seconds. The average processing time of this model is 0.212 seconds per dataframe, thus meeting this requirement.

The second requirement was that the outcome for the running path should be realistic compared to what occurs in practice. Experts from the client reviewed various visualizations of the predicted optimal running path. Twenty successful touchdown runs were compared with the model's predictions. The model correctly predicted approximately 16 out of these 20 routes, demonstrating a level of accuracy in predicting realistic running paths.

The third requirement was that the prediction algorithm works on any dataframe of a loaded American football game. This was verified by loading an entire game without any error messages. No software errors were detected.

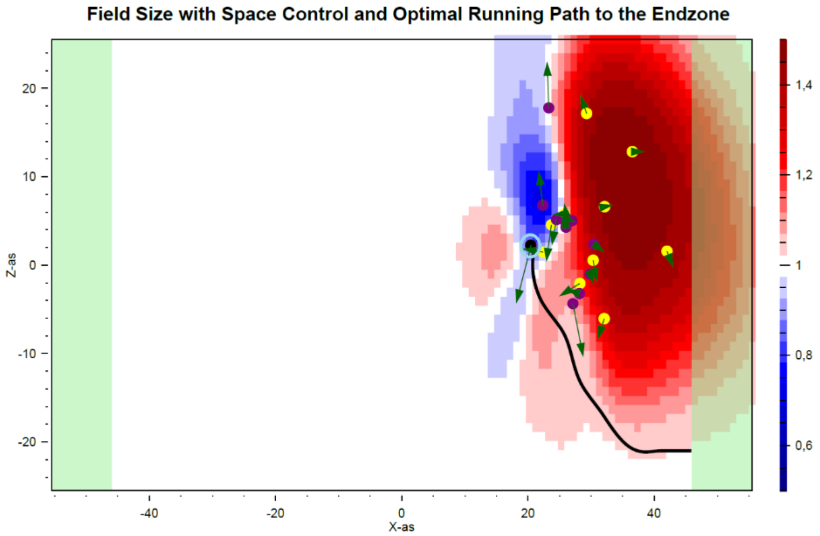


Figure 24: The prediction of the optimal running path to the endzone for an American football player with the ball (circled in light blue). The yellow dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The purple dots represent the attacking team with a blue heatmap represents their space control on the field.

7 Discussion

During this chapter, certain assumptions and limitations of the model are discussed, particularly in the context of repeating this research.

In this study, the optimal running paths were evaluated based on successful touchdown runs. The predicted running paths were assessed on 20 different successful touchdown runs. This approach revealed that the model correctly predicted about 16 out of 20 routes, indicating a certain degree of accuracy in forecasting realistic running paths. However, this number, 16 correctly predicted paths, could vary if a different set of 20 successful touchdown runs were used. To obtain a more accurate estimate of the model's performance, the number of observations should be increased. Statistical methods such as confidence intervals and sample size calculations can help determine how many touchdown runs are needed to achieve the desired accuracy (Mcleod, 2023). For example, to obtain a 95% confidence interval with an accuracy of ±5%, a larger sample than the current 20 runs is required (Mcleod, 2023). This would likely reduce the variability in the predictions and provide a more precise picture of the model's performance.

Additionally, the correctness of the predictions was visually inspected, meaning that in most cases, the predicted route did not exactly match the successful touchdown run. The RMSE (Root Mean Square Error) measures the average deviation between predicted and actual values. At selected points, the differences between predictions and measurements are calculated, squared, averaged, and then the square root of this average is taken. RMSE thus represents the average error of a model. A deviation of 1 meter may be acceptable, but it is better to discuss this with the client or a American football specialist. For more information on the calculation of RMSE, see the source Bobbitt (2021).

It is assumed that the successful touchdown run is always the most advantageous route to the end zone. This assumption can be questioned in some situations, as it is possible that there was an even more advantageous route to the end zone than the successful touchdown run.

The routes were made more realistic by transformations of scale factors, the range of the influence radius, and the space control values based on successful touchdown runs. It is possible that using other successful touchdown runs for these transformations would slightly change the magnitude of the scale factors or the range.

In the space control model, speed affects the player influence area, both in direction and how quickly the player moves towards something. However, the model does not account for different types of players, which can lead to situations where a player is close to the predicted running path but lacks speed at that moment, reducing their influence towards the path. This player, however, might be extremely fast and have strong acceleration, meaning their influence on the field towards the path should be greater than calculated. Ultimately, a player with this acceleration is more likely to stop the player with the ball.

8 Recommendations

After conducting the research and completing the assignment, several recommendations should be considered for future research or further investigation.

1. **Simplifying the Programming Code:** The first recommendation is to simplify the programming code of the model in Beyond Sports' software. This research was conducted by a mathematician with basic programming knowledge, with the focus being on getting the model to work rather than on efficient coding. Although the model currently functions, the code is not optimized for simplicity. By simplifying the programming code, the processing time per dataframe can be reduced, leading to faster processing times.

2. **Testing Alternative Algorithms:** The second recommendation is to test other algorithms for solving the shortest path problem. This should include both the algorithms discussed in this research and those not yet considered. The current processing time does not yet meet the desired 0.1 seconds per dataframe, which is necessary for live match applications. The D* Lite algorithm (Jin et al., 2023) is recommended for further testing, despite its complex implementation. D* Lite uses incremental planning (De Swart, 2023), allowing it to adjust existing calculations to new circumstances without starting over. This saves time and computational power in dynamic environments (Jin et al., 2023). In contrast, Dijkstra's algorithm must completely recalculate every time changes occur, which takes more time. Additionally, there are several other algorithms (Chumbley, n.d.) (not discussed in this research) that can solve the shortest path problem, but these are often more complex to implement than Dijkstra's, which is known for its simple implementation.

3. **Evaluating Predicted Running Paths in Simulations:** The third recommendation is to evaluate the predicted running path in a programmed simulation. This involves creating a simulation where players react to situations similar to what happens in a game. For instance, an opponent tries to intercept the ball, and teammates help the player with the ball advance as far as possible. If such a simulation is created, the player with the ball can be programmed to run the predicted path. This would allow an assessment of whether the path is indeed the optimal running path to a specific point or where the player is stopped.

4. **Researching Maximum Turning Angles per Speed Interval:** Further research should be conducted on the maximum turning angles per speed interval used in the turning angle condition. These parameters are currently chosen with a simple approach. It is recommended to conduct independent research with athletes, examining the speeds at which a player can turn and the corresponding turning angles. Additionally, it is crucial to investigate how long a player needs to turn between 90 and 180 degrees at high speeds and how much they need to decelerate for this. Measuring the average deceleration rate of an elite athlete is also relevant for this research. This recommendation would make the running paths more realistic, as the parameters measured by this proposed research are more reliable compared to the simple approach used in the current research. However, the effect would not be significantly large since it only concerns the first few meters of the running path. There is already a turning radius (with the simple approach) in the model, but this radius might be adjusted, either larger or smaller, through further research in this area.

5. **Considering Other Distances as End Goals:** Other distances as end goals should be considered. Currently, only the end zone is chosen as the end goal. However, in some situations, a team aims to advance just 10 yards from their starting point on the first down to earn four new downs and continue until they score. Therefore, the 10-yard line from the starting point of the first down is also very important in American football. This can lead the model to predict a different path than when the end zone is the end goal. Furthermore, it may be necessary to adjust the scale factors and the range of the influence radius for this new end area. These paths should then be reevaluated. In which situation each end goal is used – the running path to the 10-yard line or the end zone – still needs to be determined. One possibility is to predict both options and leave the choice to an analyst. This analyst can use the predictions for game segments to support what a player could have done better during the game.

6. **Assigning a Success Score to a Predicted Path:** A success score should be assigned to a predicted path. Currently, a running path is predicted, but no probability or score of success is calculated. The model (Dijkstra's algorithm) also calculates the total cost per path and indicates how many nodes the path passes through. This allows the average weight per step to be calculated. Each step between nodes has a certain weight with an upper and lower limit. By comparing these limits, it can be determined whether the average weight per step is closer to the lower or upper limit, which enables a simple score for the probability of success. Additionally, it is interesting to explore other ways to measure the success score of the predicted path.

7. **Studying and Implementing Reinforcement Learning Approaches:** The final recommendation is to study and implement the research in section 4.1, "A Reinforcement Learning Based Approach to Play Calling in Football"(Biro, P. & G. Walker, 2021). This research focuses on a reinforcement learning-based approach to making play decisions in American football. It uses a Markov decision process to determine optimal choices at each play level, considering collected data and the probabilities of various play outcomes. This process helps optimize decisions by anticipating the expected utility of different actions, such as running, passing, or handing off the ball, in various game situations.

   The client should study and integrate this methodology to answer the primary question in the desired situation: What is the best action for a player with the ball—throw, run, or hand off the ball? If the answer is for the player to run, the model from the current research can then be used to determine which direction to run. The approach is extensively analysed in the document with various examples and scenarios within the game. This provides insights into how data analysis and machine learning can contribute to strategic decision-making in American football. By applying this methodology, a model can be developed that not only

Bibliography

Afi. (2023, September 21). *American Football's Global Reach: The Growth of the NFL Worldwide*. American Football International. Consulted on 4 April 2024, from https://www.americanfootballinternational.com/american-footballs-global-reach-the-growth-of-the-nfl-worldwide/

Algorithm Examples. (2024, February 13). *Dijkstra vs Bellman-Ford: a shortest Path Algorithms showdown - algorithm examples.* Consulted on 22 May 2024, from https://blog.algorithmexamples.com/graph-algorithm/dijkstra-vs-bellman-ford-a-shortest-path-algorithms-showdown/

Ambler, W. (2024, June 7). *Sports Analytics: What is it & How it Improves Performance? - Catapult*. Consulted on 8 June 2024, from https://www.catapult.com/blog/what-is-sports-analytics

Arora, J. S. (2004). *Optimum Design Concepts*. In Elsevier eBooks (pp. 83–174). Consulted on 20 May 2024, from https://doi.org/10.1016/b978-012064155-0/50004-5

Biro, P. & G. Walker (2021, March 15). *A Reinforcement Learning Based Approach to Play Calling in Football*. Pdf, University of Texas at Austin. Consulted on 30 April 2024, from *https://arxiv.org/pdf/2103.06939.pdf*

Bobbitt, Z. (2021, May 10). *How to interpret Root Mean Square Error (RMSE). Statology.* Consulted on 11 June 2024, from *https://www.statology.org/how-to-interpret-rmse/*

BSc, D. C. (2023, July 4). *Hoe Snel Rent een Mens Gemiddeld (km/u)? En Wat is het Wereldrecord? Human Nature.* Consulted on 30 May 2024, from https://gohumannature.com/hoe-snel-rent-een-mens/?utm_content=cmp-true

Cambre, C. (n.d.). *Dijkstra algoritme (1). GeoGebra*. Consulted on 9 May 2024, from https://www.geogebra.org/m/khkGTYvf

Chumbley, A (n.d.). *Shortest Path Algorithms | Brilliant Math & Science Wiki*. Consulted on 4 June 2024, from https://brilliant.org/wiki/shortest-path-algorithms/

De Swart, N. (2023, June 19). *Agile: incrementeel en iteratief*. Reaco Academy. Consulted on 3 June 2024, from https://www.reaco.nl/blog/agile-incrementeel-en-iteratief/#:~:text=Wat%20is%20incrementeel%3F,aan%20hetgeen%20er%20al%20was.

Dos'Santos, T., Thomas, C., McBurnie, A., Comfort, P., & Jones, P. A. (2021). *Change of Direction Speed and Technique Modification Training Improves 180° Turning Performance, Kinetics, and Kinematics.* Sports, 9(6), 73. Consulted on 10 June 2024, from https://doi.org/10.3390/sports9060073

Fernández, J., & Bornn, L. (2018). *Wide Open Spaces: A statistical technique for measuring space creation in professional soccer. ResearchGate.* Consulted on 20 April 2024, from https://www.researchgate.net/publication/324942294_Wide_Open_Spaces_A_statistical_technique_for_measuring_space_creation_in_professional_soccer

Gallagher, N. B., Eigenvector Research, Training, and Software, Eigenvector Research, Inc., Gallagher, N., Eilers, P., & Eigenvector Research, Inc. (1964). *Savitzky-Golay Smoothing and Differentiation filter* [White paper]. Eigenvector Research, Inc. Consulted on 30 May 2024, from https://eigenvector.com/wp-content/uploads/2020/01/SavitzkyGolay.pdf

GeeksforGeeks. (2023, November 23). *Shortest Path Algorithm Tutorial with Problems*. GeeksforGeeks. Consulted on 2 June 2024, from https://www.geeksforgeeks.org/shortest-path-algorithms-a-complete-guide/

GeeksforGeeks. (2024b, June 4). *How to find Shortest Paths from Source to all Vertices using Dijkstra s Algorithm*. GeeksforGeeks. Consulted on 7 June 2024, from https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/

Haddad, C. (2024, February 12). *The complete Beginners Guide to American Football - vIQtory Sports*. Consulted on 2 April 2024, from https://www.viqtorysports.com/how-to-understand-american-football-beginners-guide/

Heijerman, S., (500759701), Drost, S., (500807060), Feenstra, J., (500838122), Wellens, H., (500785541), & Wesselink, F., (500883260). (2024). *SPACE DOMINANCE*. In Forward Football & Hogeschool van Amsterdam, Forward Football [Report].

Henkny. (2014, 27 augustus). Stride Length and Running Stride Lenth for FitBit. MyFitnessPal.com. Consulted on 11 June 2024, from https://community.myfitnesspal.com/en/discussion/1415411/stride-length-and-running-stride-lenth-for-fitbit

International Federation of American Football. (2023, June 6). *A Guide to Tackle Football - International Federation of American Football*. Consulted on 6 April 2024, from https://www.americanfootball.sport/a-guide-to-tackle-football/#:~:text=Players%20on%20the%20offensive%20team,and%20pulled%20to%20the%20ground.

Jin, J., Zhang, Y., Zhou, Z., Jin, M., Yang, X., & Hu, F. (2023). *Conflict-based search with D* lite algorithm for robot path planning in unknown dynamic environments.* Consulted on 20 April 2024, from https://doi.org/10.1016/j.compeleceng.2022.108473

Jones, B. J. M. (2024, March 7). *Football retains dominant position as favorite U.S. sport. Gallup.com*. Consulted on 1 April 2024, from https://news.gallup.com/poll/610046/football-retains-dominant-position-favorite-sport.aspx#:~:text=Football%20Continues%20to%20Rank%20as%20Americans'%20Favorite%20Sport&text=Trend%20in%20Americans'%20favorite%20sport,%25%20and%209%25%2C%20respectively.

Khodadadi, N., Gharehchopogh, F. S., Abdollahzadeh, B., & Mirjalili, S. (2023). *Space truss structures' optimization using metaheuristic optimization algorithms.* In Elsevier eBooks (pp. 163–179). Consulted on 22 May 2024, from https://doi.org/10.1016/b978-0-323-91781-0.00009-0

Madkour, A., Aref, W. G., Faizan Ur Rehman, Mohamed Abdur Rahman, & Saleh Basalamah. (n.d.). *A Survey of Shortest-Path Algorithms.* Purdue University, West Lafayette, USA. Consulted on 10 May 2024, from https://arxiv.org/pdf/1705.02044

Martin, C. (2022, November 30). *What are Downs in Football? (Full Explanation)*. Football Advantage. Consulted on 15 April 2024, from https://footballadvantage.com/what-are-downs-in-football/

Martin, C. (2022, December 19). *What is the Line of Scrimmage in Football? (Explained)*. Football Advantage. Consulted on 5 April 2024, from https://footballadvantage.com/line-of-scrimmage/'

Mcleod, S., PhD. (2023). *Confidence Intervals Explained: Examples, Formula & Interpretation. Simply Psychology.* Consulted on 11 June 2024, from https://www.simplypsychology.org/confidence-interval.html

Power, R. (2024, May 24). *Transforming fan engagement in sports: Going beyond the game*. Consulted on 28 May 2024, from https://dolby.io/blog/transforming-fan-engagement-in-sports-going-beyond-the-game/

R, A. B. a. a. B., & Ahmed, A. S. A. (2021). *Designing and Implementing Shortest and Fastest Paths; A Comparison of Bellman-Ford algorithm, A*, and Dijkstra's algorithms.* Consulted on 23 May 2024, from https://doi.org/10.14445/22312803/ijctt-v69i5p102

Spearman, W. (2016). *Quantifying pitch control. ResearchGate*. Consulted on 20 April 2024, from https://doi.org/10.13140/RG.2.2.22551.93603

Sports, B. (2024, 4 juni). *NFL Toy Story Funday_Sidy-by-Side_1080* [Video]. Vimeo. Consulted on 3 April 2024, from https://vimeo.com/913021511/099182ac16?share=copy

Sports Player Tracking | Zebra. (n.d.). *Zebra MotionWorks Sport:Real-Time Player Tracking.* Consulted on 10 April 2024, from https://www.zebra.com/us/en/products/location-technologies/sports-player-tracking.html

Stables, J. (2014, August 6). *NFL to track players with wearable sensors*. Consulted on 11 April 2024, from https://www.wareable.com/sport/nfl-to-track-players-with-wearable-sensors

STATSports. (2021, July 12). *Maximum Acceleration and Deceleration – metric Considerations and Uses – STATSports*. Consulted on 26 May 2024, from https://pro.statsports.com/maximum-acceleration-and-deceleration-metric-considerations-and-uses/

Toppr. (2019, November 12). *Kinematics Formulas - Definition, Equations, Examples*. Consulted on 2 June 2024, from https://www.toppr.com/guides/physics-formulas/kinematics-formulas/

Unix Time Stamp - Epoch Converter. (n.d.). *What is the unix time stamp?* Consulted on 20 April 2024, from https://www.unixtimestamp.com/

Wallpaper Flare. (n.d.). *HD wallpaper: american football, running back, action, eluding, tacklers*. Consulted on 20 April 2024, from https://www.wallpaperflare.com/american-football-running-back-action-eluding-tacklers-wallpaper-zoqbw

W3Schools. (n.d.). *DSA Shortest Path.* Consulted on 8 May 2024, from https://www.w3schools.com/dsa/dsa_theory_graphs_shortestpath.php

Zachary. (2023, September 29). *An animated Behind-the-Scenes look at ESPN's "Toy Story Funday Football" - the Walt Disney Company*. The Walt Disney Company. Consulted on 20 April 2024, from https://thewaltdisneycompany.com/an-animated-behind-the-scenes-look-at-espns-toy-story-funday-football/

Appendices

Appendix 1

In this appendix, you can find the tables that provide an overview of the variables contained in the topics, which in turn are included in the dataframe. These topics are described in the chapter 'Data'. The topics Persons (table 8), Ball (table 9), FootballContext (table 10), DownMarkersContext (table 11), and GameClockContext (table 12) provide an overview of the variables.

Table 8: Overview of the variables from the dataset from topic 'Persons'.

Variables	Example	Definition
Id	151	This is a player's ID, each player has them own ID in the dataset.
Timestamp	1699225574100	This is a specific value when the observation is made. It has a Unix timestamp and is expressed with millisecond precision (Unix Time Stamp - Epoch Converter, n.d.).
Position	(3.045, 0, 19.818)	This is x, y and z coordinate. Where x is the length, y is the height and z is the width. The unit is in meters.
Speed	0.204	The magnitude of a player's velocity in m/s.
TeamSide	2	Which team the player plays for.
JerseyNumber	52	What number a player wears on his jersey.
PersonContext	MovementOrientation: 178,93 HasBallPossession: false	MovementOrientation is how it is rotated and is in degrees. HasBallPossession is whether the player is holding the ball (True is fixed or False is not fixed).

Table 9: Overview of the variables from the dataset from topic 'Ball'.

Variables	Example	Definition
Id	0	This is an ID of the ball, the ball also has its own ID in the dataset.
Timestamp	1699225574100	This is a specific value when the observation is made. It has a Unix timestamp and is

		expressed with millisecond precision (Unix Time Stamp - Epoch Converter, n.d.).
Position	(3.044, 2.088, 19.818)	This is x, y and z coordinate. Where x is the length, y is the height and z is the width. The unit is in meters.
Speed	0.303	The speed of a ball in m/s.

Table 10: Overview of the variables from the dataset from topic 'FootballContext'.

Variables	Example	Definition
ToGO	10	How many yards left to go for the LTG.
BallOn	36	The yard line, where the snap is.
PlayClockTime	25	What playing minute in the game.
Down	2	Which down it is.
HomeTimeoutsLeft	1	Time outs left for the home team.
GuestTimeoutsLeft	1	Time outs left for the away team.

Table 11: Overview of the variables in the dataset from topic 'DownMarkersContext'.

Variables	Example	Definition
InitialDown	27.432	It is starting point from the first down. It’s in meters.
LineOfScrimmage	27.432	The LOS in meters
LineToGain	18.287	The LTG, the line they must achieve to get 4 new downs, in yards.

Table 12: Overview of the variables from the dataset from topic 'GameClockContext'.

Variables	Example	Definition
Period	1	What quarter the match is in.
Minute	14	What minute the match sits
Second	55	Which second the match sits
InjuryTime	0	The injury time that may be above the regular time
IsCLockRunning	True or False	Indicates whether time is stopped (i.e., pause moment).

Appendix 2

In this appendix, an overview of the events included in the dataset is provided, with a given definition (table 13). The "Data" chapter refers to this appendix.

Table 13: Overview of events that are going to be potentially useful for this project, with a definition attached.

Events names	**Definition**
Pass_forward	Forward pass is thrown by the player with ball.
Run	The action of player with ball is running.
Touchdown	It became touchdown after running.
Pass_outcome_touchdown	The forward pass outcome has become a touchdown, so other player caught the ball in the endzone.
Pass_outcome_caught	The forward pass outcome is caught by a fellow player.
Pass_outcome_incomplete	The forward pass was intercepted by an opponent or the ball was caught on the ground.
Handoff	Handing the ball off to a teammate.
Fumble	Fumble occurs when a player loses the ball before it is marked as down or when he throws or drops the ball.
Lateral	Passes backwards or sideways.
Tackle	Player with ball is tackled by opponent.
Out_of_bounds	The player has run over the sideline with the ball. After this, a new down starts.
Qb_sack	The quarterback has the ball and is tackled; this is the end of the down.
Qb_kneel	The quarterback has the ball and kneels; this is the end of the down.

Appendix 3

In this attachment, the optimal running path, as predicted in figure 26, is compared to the actual running path of the player during a real American football game (figure 25). The comparison focuses on the running path through the middle from their own half.

Figure 25:The successful touchdown run (black line) for the player with the ball (circled in red) in a real American football game.

Note: Adapted from: The simulation of Beyond Sports

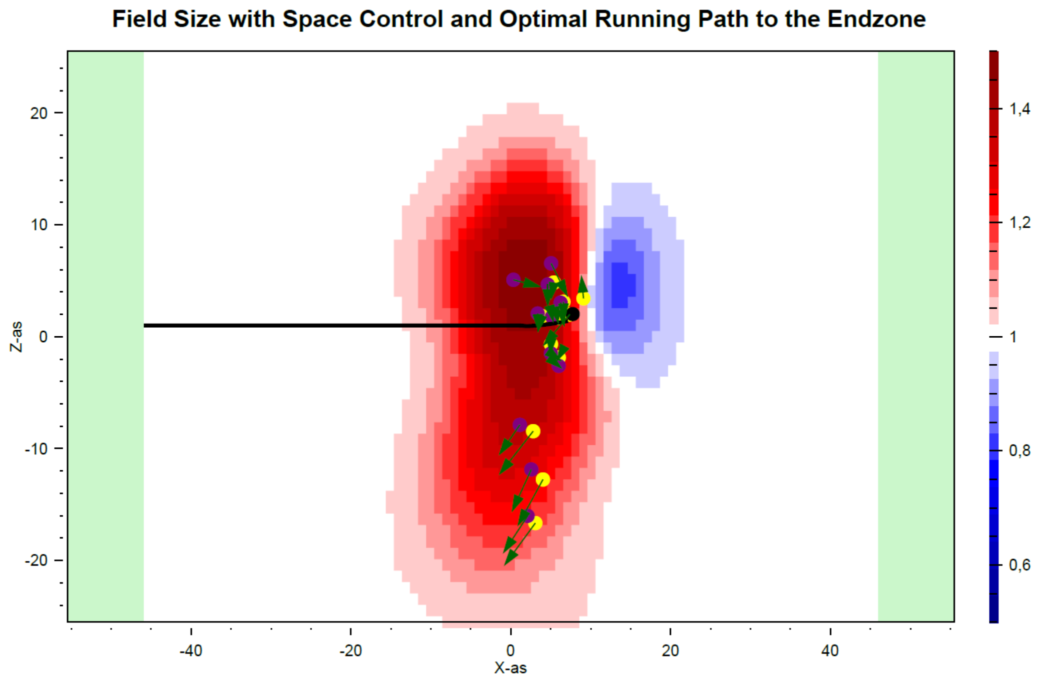


Figure 26: The prediction of the optimal running path to the endzone for an American football player with the ball (black dot). The purple dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The yellow dots represent the attacking team with a blue heatmap represents their space control on the field.

Appendix 4

In this attachment, the optimal running path, as predicted in figure 28, is compared to the actual running path of the player during a real American football game (figure 27). The comparison focuses on the running path from the side of their own half.

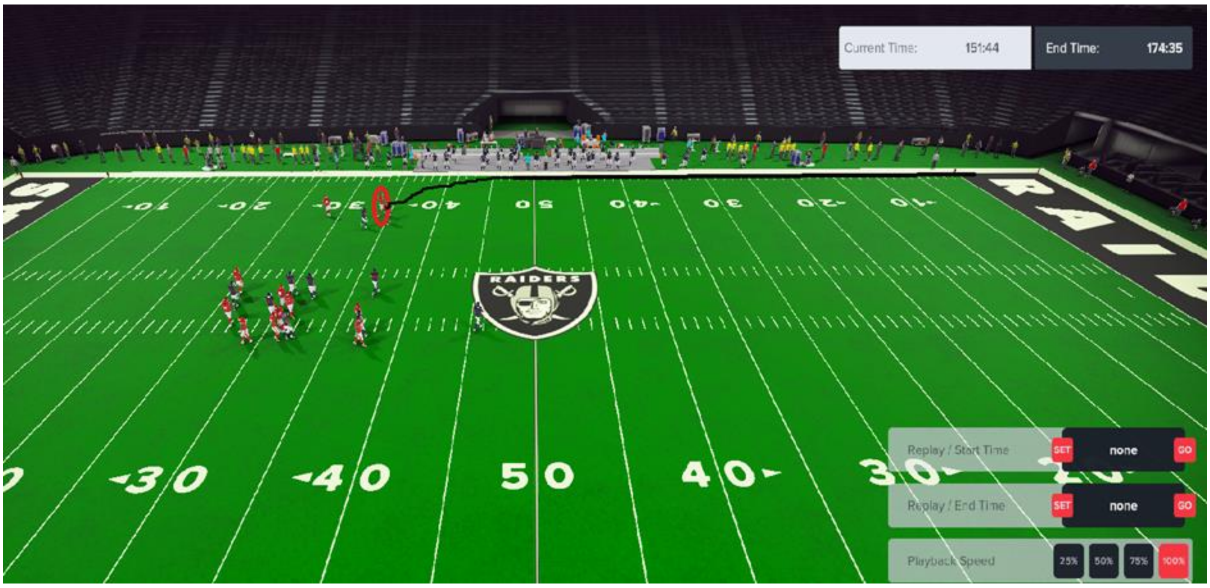


Figure 27: The successful touchdown run (black line) for the player with the ball (circled in red) in a real American football game.

Note: Adapted from: The simulation of Beyond Sports

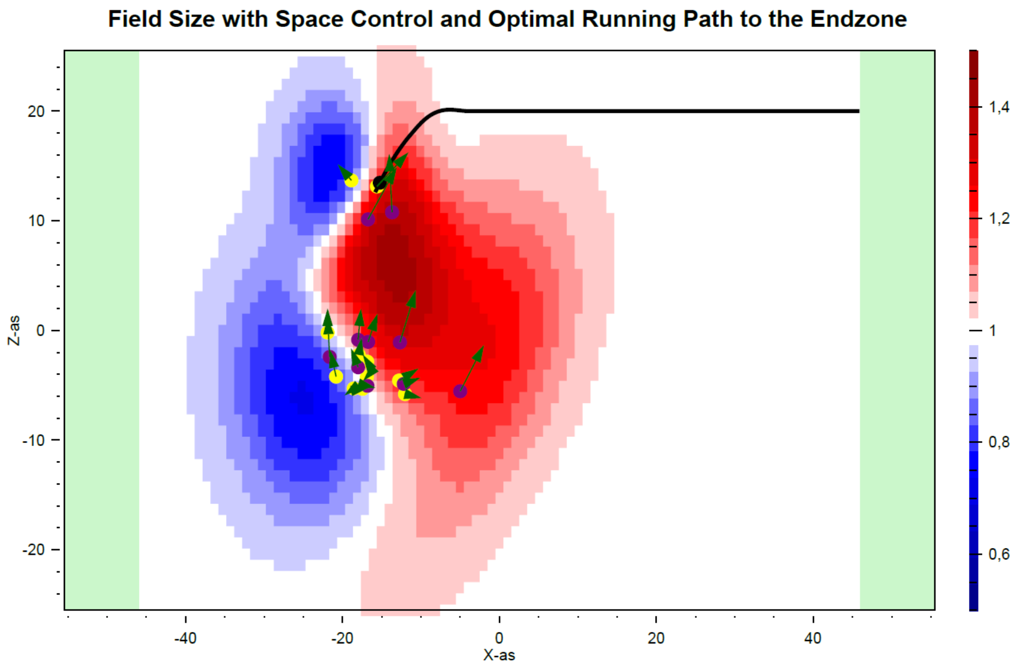


Figure 28: The prediction of the optimal running path to the endzone for an American football player with the ball (black dot). The purple dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The yellow dots represent the attacking team with a blue heatmap represents their space control on the field.

Appendix 5

In this attachment, the optimal running path, as predicted in figure 30, is compared with the actual running path of the player during a real American football game (figure 29). Here, the running path is compared from the side, starting from near the endzone.

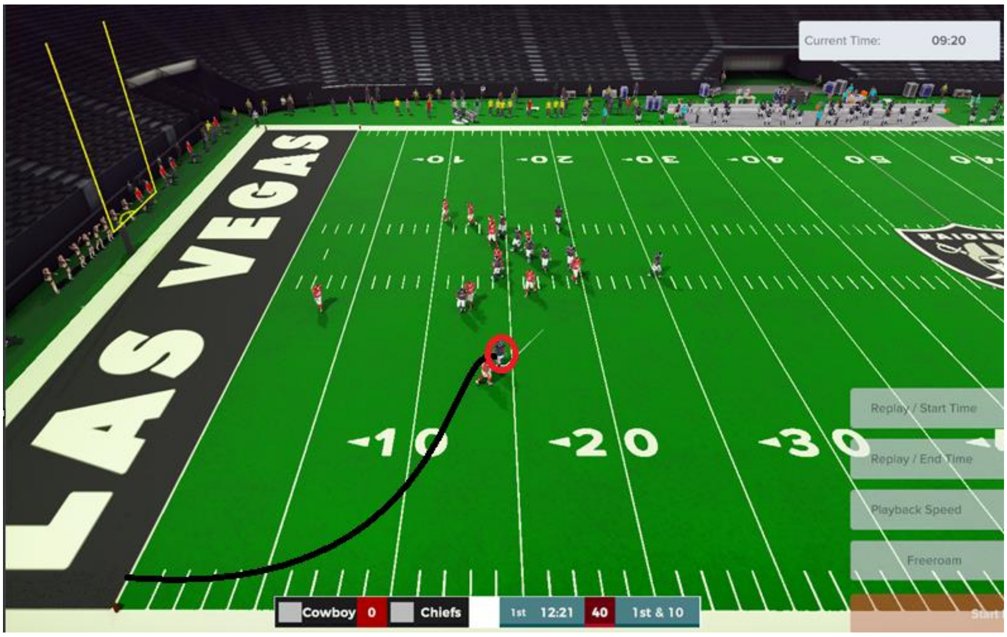


Figure 29:The successful touchdown run (black line) for the player with the ball (circled in red) in a real American football game.

Note: Adapted from: The simulation of Beyond Sports

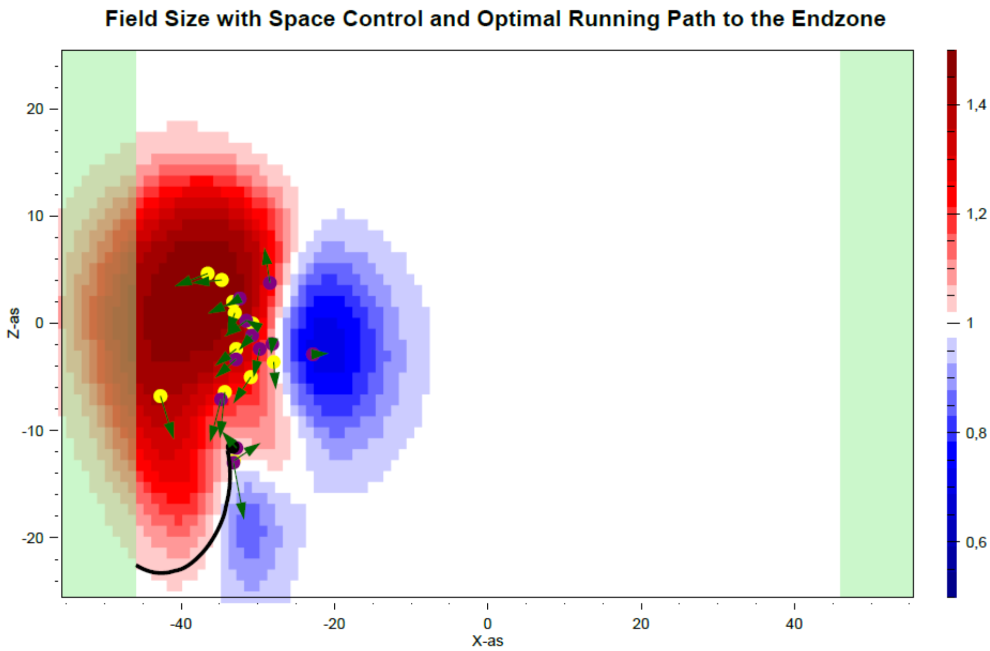


Figure 30: The prediction of the optimal running path to the endzone for an American football player with the ball (black dot). The yellow dots represent the defensive team with a red heatmap represents the dominant areas of space control on the field. The purple dots represent the attacking team with a blue heatmap represents their space control on the field.