Cooperative play classification and analysis in team sports with machine learning

Zhang Ziyi

Contents

Abstract

Recent advancements in measurement technologies, such as camera-based systems, give us an opportunity to improve our understanding of multi-agent behaviors in various fields. However, due to their complexity, modeling the intricacies of multi-agent behaviors remains a challenge. This has led to a shift towards model-free, data-driven approaches. Data-driven modeling, especially in machine learning with complex structures like neural networks, plays an important role in extracting insights and making predictions from real-world data. While these models offer enhanced expressiveness and predictive capability, the interpretability of their results poses a significant challenge. This is crucial in practical applications, such as team sports, where understanding the rationale behind actions and plays is essential for coaches and players. In particular, in team sports, the intricacies of multi-agent behaviors also lead to extremely high labor costs for manual labeling. These factors make the implementation of data-driven classification and analysis methods difficult in the field of team sports.

To address these problems, utilizing machine learning techniques, two approaches are proposed to classify and analyze cooperative play in team sports. In the first study, a classification approach based on semi-supervised learning methods is proposed for cooperative play classification in team sports. I examine this approach for classifying strategic cooperative plays called screen-play in basketball using a smaller labeled dataset and a larger unlabeled dataset. In the experiment, the classification performance of the semi-supervised learning approaches improved upon the conventional supervised approach for minor types of screen-plays. For the interpretability, we found that self-training obtained similar or higher contribution of some features than the baseline.

In the second study, for cooperative play analysis in team sports, a deep learning-based comparative analysis method to analyze multi-agent trajectories in basketball games is proposed. A neural network approach based on an attention mechanism using multi-agent motion characteristics (e.g., the distances between agents and objects) as the input is adopted, designed to detect distinct segments in trajectories of given classes. This enables us to understand differences between classes by highlighting segmented trajectories and which variables correlate with the given labels.

In this thesis, these approaches are validated by comparing them with other baseline methods, and the second approach is also validated by analyzing the attacking plays in an NBA dataset. In addition, these methods also reveal the relationship between some behaviors and certain cooperation plays, which can provide coaches and athletes with more information or guidance about the game.

1 Introduction

1.1 Background

Advancements in measurement technologies, such as global/local positioning and camera-based systems, have played a crucial role in multi-agent behavior analysis in the real world. Recent progress in this field has enhanced our comprehension of the fundamental principles governing multi-agent behaviors, a critical concern across diverse scientific and engineering domains, e.g., human behavioral science [1, 2], and robotics [3, 4].

Modeling, analysis, and understanding real-world multi-agent behavior often becomes a big challenge because of the lack of physical links of multi-agent systems. Employing mathematical models based on fundamental rules can provide a means to comprehend the dynamics of multi-agent interactions, such as social force models [5], which have found extensive application in pedestrian dynamic behavior analysis. Moreover, these models can be extended to address intricate multi-agent behaviors in team sports [6, 7, 8].

However, due to the high complexity of multi-agent behaviors in various aspects, multi-agent behavior modeling is a mathematical challenge. Thus, a model-free (or equation-free) and data-driven approach becomes essential to enhance our comprehension of these behaviors [9, 10].

Utilizing data-driven modeling represents a potent approach capable of extracting valuable insights and making predictions from intricate real-world datasets. Especially in machine learning, there has been a dedicated exploration of the learning dynamics inherent in models featuring complex, nonlinear structures like neural networks [11]. While these models can deliver heightened expressiveness and predictive prowess, the interpretability of their results poses a significant challenge, establishing a delicate balance between interpretability and predictability. This dilemma becomes particularly pivotal in practical scenarios, such as sports games, where coaches and players rely on understanding the rationale behind goals and the characteristics evident in subsequent plays.

1.2 Challenges of This Thesis

In general, various cooperative plays in team sports can be manually labeled by experts, and it requires much labor costs. In previous work, supervised learning methods in machine learning have been used for automatic classification of labeled cooperative plays [12, 13, 14, 15, 16]. They used the complete pairs of the features as the input and labels of team play for the classifier. However, it requires much labor costs, and a large amount of unlabeled data is not utilized if it exists.

To solve this issue, semi-supervised learning [17] is one of the approaches utilizing a large amount of unlabeled data, which is conceptually situated between supervised and unsupervised learning and shows their strength when labeled data is scarce. I use a semi-supervised approach for cooperative play classification in Study I. Unsupervised learning and self-supervised learning are also effective methods for such issues, which are discussed in Chapter 2. Moreover, a large amount of official data with some popular labels can be obtained from official websites and used for analysis, such as score and turnover. Other kinds of popular labels, such as shooting percentages, are computable by these labels directly. In this case, a large dataset with labels can be obtained, which is used in Study II.

Another challenge of this thesis is that nonlinear machine learning models have difficulty in interpretability. In other words, it is often hard to find out how the nonlinear machine learning models got the results. It prevents the researchers from understanding the underlying rules of cooperative play. To solve this problem, earlier work proposed some model-free interpretable methods, e.g., Local Interpretable Model-Agnostic Explanations (LIME) [18] and SHapley Additive exPlanations (SHAP) [19], which have been widely used as effective methods. I use such a model-free interpretable method in Study I. For deep learning methods, the condition is different and complex. Besides the approaches mentioned above, several unique approaches were proposed to enhance the performance of interpretability, such as self-explaining neural network [20], attention-based deep neural network [21]. I use an attention mechanism to interpret the model prediction in Study II.

In this thesis, some assumptions are made at first for the cooperative play classification problem in team sports. Although multi-agent trajectory labels are complex and not clear compared to image labels in general, I assume that the data with the same labels indicate that they have some similar and unique features among their trajectory data. The data with different labels indicate that they have different features among their trajectory data. These differences can be distinguished between different labels. With these assumptions, I conducted two studies about team play classification and analysis in this thesis. Fig. 1.1 provides a comprehensive overview of this thesis.

1.2.1 Team play classification

In this thesis, to solve the challenge of labor costs of labeling for multi-agent trajectory data and interpretability of machine learning, a team play classification method based on semi-supervised learning is studied in Study I (Chapter 3). Screen play is selected as the target team play and classified by the proposed method. This is because screen play is a basic, common, but essential team play in basketball, which appears in most half-court attacks, and most team tactics in basketball are based on it. Furthermore, SHAP [19], which utilizes an interpretable approximate model of the original nonlinear prediction model, is used to improve interpretability. For practical significance, this study can help coaches and athletes, whether professional or unprofessional, to automatically label their play data with team play labels and analyze their play with these labels.

1.2.2 Team play analysis

Also, to solve these challenges of labor costs of labeling for multi-agent trajectory data and interpretability of machine learning, a team play analysis method based on deep learning is proposed in Study II (Chapter 4). The effective attack, which is based on the concept of wide-open shots, is chosen as a classification label and an evaluation criterion in this study. This is because compared with goal/non-goal, which is affected by the effect of the shooting skills of shooters and randomness of shooting, the effective attack evaluates whether a player makes an effective shot attempt and can directly evaluate the tactics of attacks. And it can be calculated by multi-agent trajectory data directly through its definition (For details, please see Section 4.2.3). Furthermore, comparative analysis with deep learning with attention mechanisms [21] are used in Study II to highlight the segments with handcrafted features to improve interpretability. For practical significance, Study II can help coaches and athletes automatically analyze and evaluate their play data and find a better way to make an effective attack and obtain a score.

1.3 Thesis Overview

The thesis is organized as follows. In Chapter 2, related work on cooperative play classification and analysis in team sports are discussed. In Chapter 3, a cooperative play classification method based on semi-supervised learning is described. Chapter 4 discusses multi-agent trajectory comparative analysis using deep learning. In Chapter 5, the contributions of this thesis are summarized, and future work is discussed.

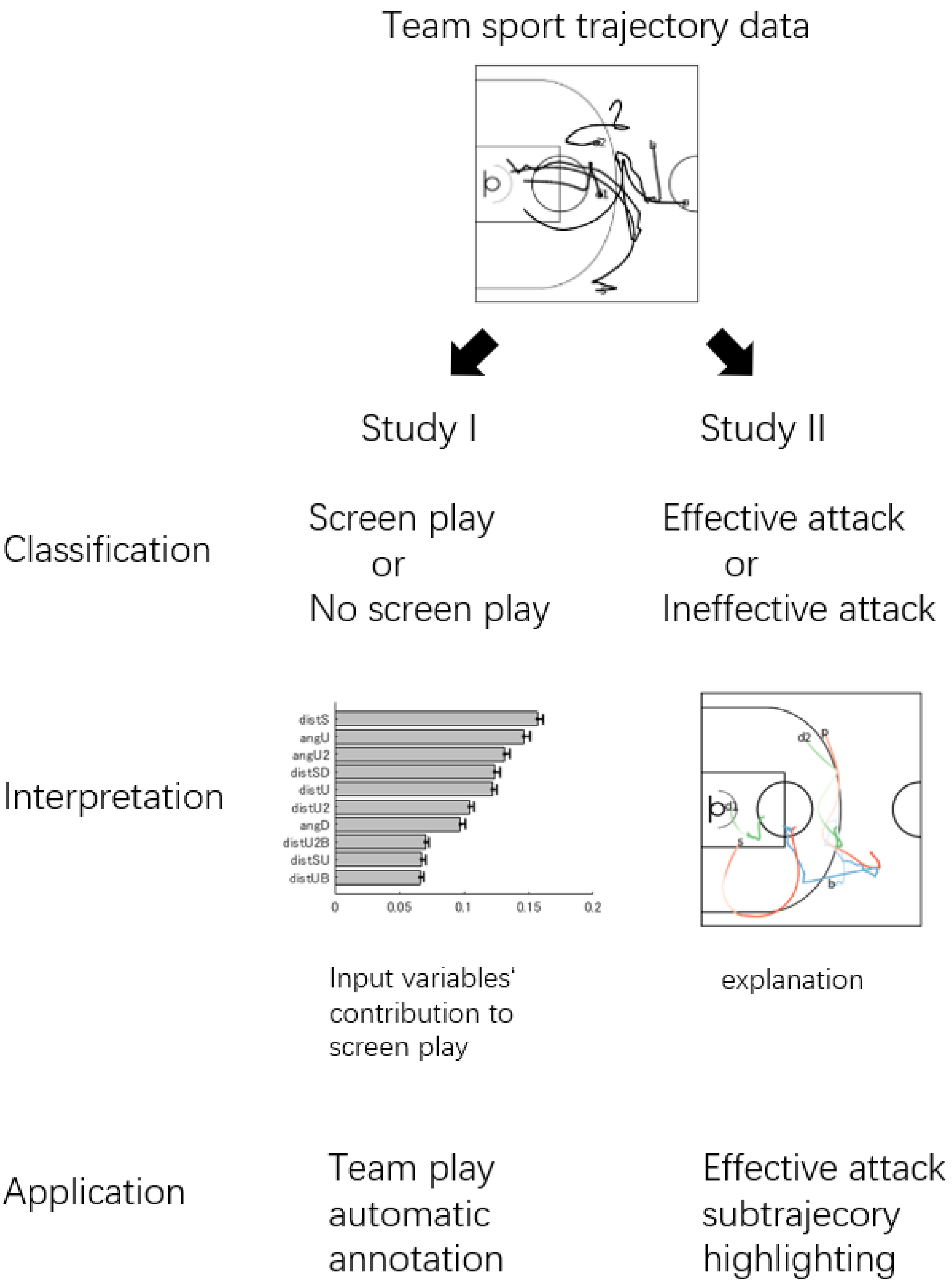

Figure 1.1: **Overview of the thesis.** *Study I focuses on the classification of cooperative plays, while Study II focuses on the analysis of cooperative plays. Both of them can provide coaching insights into cooperative plays.*

2 Literature Review

Studies related to this thesis can be divided into three categories. The first is research on traditional analysis methods of team sport analysis. The second is research on machine learning methods for team sport analysis. The third is interpretability problems in machine learning for team sports. Here, I introduce related work on these topics.

2.1 Traditional analysis in team sports

Traditional methods without machine learning in various fields typically rely on researchers' experience and established theories to evaluate the characteristics of multi-agent behaviors. In the team sports field, for example, researchers have calculated the distances and relative phases of two athletes [22, 23, 14], speeds of movements [24], frequencies and angles of actions (e.g., shots [25] and passes [26, 27, 28]), as well as their representative values (e.g., average and maximum values). Some advanced measurement systems like motion capture systems and force platforms have the ability to analyze skillful maneuvers [29, 30]. Using the output of representative values from these systems, specific hypotheses have been tested [14, 31]. In contrast, other studies have used more sophisticated mathematical approaches, such as Voronoi diagrams [32], network theory [33], and group theories [34], to compute representative values.

The above approach is inverse to obtaining insight from data, but there is a forward approach to considering models and performing simulations to understand the behaviors. In traditional approaches, mathematical models based on some simple rules are widely used to understand the underlying rules of multi-agent movements. For example, social forces models in pedestrians [5], similar rules in flocks of birds [35], and schools of fishes [36]. And those models also applied to team sports [6, 7, 8] under certain assumptions.

Traditional approaches indeed have their advantages, such as being easy to interpret and applicable to small datasets in specific sports. However, they may need to be more flexible to represent cooperative/competitive interactions in detail. Additionally, due to multi-agent inherently higher-order social interactions, cognition, and body dynamics, these multi-agent modeling methods can sometimes be mathematically difficult. To overcome these limitations, data-driven approaches, including machine learning, have been developed to extract, classify, and regress automatically.

2.2 Machine learning approaches in team sports

Learning-based approaches that utilize positional data of players can be broadly categorized into unsupervised, supervised, semi-supervised, self-supervised, and reinforcement learning approaches. Here, I introduce these approaches in this order and the merits and demerits for each approach is described in Table 2.1.

2.2.1 Unsupervised learning methods

Unsupervised learning is a type of machine learning where an algorithm is trained on unlabeled data without explicit guidance on the desired output. Unlike supervised learning, where the algorithm learns from labeled examples, unsupervised learning involves finding patterns, relationships, or structures within the data without predefined categories or target labels. Popular unsupervised methods include clustering and dimensionality reduction.

Table 2.1: Comparison of Learning Approaches

Learning Approach	Merits	Demerits
Supervised	High accuracy, clear objectives, easier to measure performance	Requires large amount of labeled data
Unsupervised	No need for labeled data, can discover hidden patterns in data	Less accurate, harder to measure performance, results can be ambiguous
Semi-Supervised	Requires fewer labeled examples, can leverage large amounts of unlabeled data	Performance dependent on the quality and amount of labeled data
Self-Supervised	Does not require external labels, learns representations by predicting parts of its input	May not directly optimize for the task at hand, requires careful design of pretraining tasks
Reinforcement	Ideal for decision-making tasks, learns through trial and error, flexible to environment changes	Complex to implement, requires a lot of computational resources, can be unstable during training

In clustering, the algorithm aims to group similar data points based on inherent patterns or similarities. The goal is to discover natural groupings within the data. There are various clustering algorithms, such as hierarchical clustering, centroid-based clustering, density-based clustering, and distribution-based clustering.

In team sports, hierarchical clustering [37, 38] based on similarity [39, 40, 41] and distribution-based clustering using a Gaussian mixture model [42] has been studied. However, clustering time-series data can be challenging because it is hard to directly compute the similarity between data when the data does not fix the time length. In that case, Fréchet distance [43] and dynamic time warping (DTW) [44] have been used to measure the similarity between trajectories in basketball [39, 40] and soccer [39]. However, these methods have high computational costs and are inappropriate for big data scales.

Dimensionality reduction transfers high-dimensional data to low-dimensional meaningful data. Because of the importance of time-series structure in multi-agent movement, time-series structure should be considered in dimensionality reduction in team sports. A proposed method uses neural networks [45, 46] and self-organizing maps [47, 48] to transfer trajectory data into images. Another approach for extracting physically-interpretable dynamical properties is a method called dynamic mode decomposition (DMD) [49, 50], which was applied to basketball score prediction [51, 9] and screen-play and zone defense classifications [16].

2.2.2 Supervised learning

Supervised learning is a machine learning paradigm in which an algorithm learns from labeled training data to make predictions or decisions without explicit programming. In this approach, the algorithm is provided with a dataset consisting of input-output pairs, where the inputs are the features or attributes of the data, and the outputs are the corresponding labels or desired outcomes. The goal of supervised learning is for the algorithm to generalize from the provided examples and accurately predict the output for new, unseen data. Supervised learning can be categorized into two main types: classification and regression.

In classification tasks, the algorithm is trained to assign input data to predefined categories. The regression algorithm is trained to output continuous values for the input data. In team sports, supervised learning methods are applied to some classic problems by inputting features of the original data, such as screen-play classification in basketball [12, 13, 15], score prediction in basketball [52, 53, 54], team strategy assessing in [55]. These features can be obtained by unsupervised learning methods, or they are hand-crafted static features that were introduced in Chapter 2.1.

Due to the intricate multi-agent behaviors in team sports, time-series data features should be considered in various problems. In this case, inputting dynamic features to models, which can be obtained from unsupervised learning, is a straightforward strategy for supervised learning. To illustrate, using the aforementioned dynamic mode decomposition (DMD) and assessing similarity enables the classification of various tactics (defensive or offensive) [16] and classification and prediction of scoring probability [51, 9].

Moreover, end-to-end approaches, which use a single model to handle the entire process, from raw input data and extracting features to generating the final output, without explicitly breaking down the task into sub-tasks, are also applied to team sports problems. For instance, end-to-end deep learning approaches were used to classify NBA offensive plays [45], team activity analysis [56], and micro-action evaluating [57].

However, end-to-end approaches are usually hard to interpret. To address this, some approaches that can balance predictability and interpretability have been proposed, such as employing matrix [58], multi-resolution tensor learning [59], Poisson point processes [60], and trajectory mining method [61].

As regression problem, simulating short-term multi-agent trajectories have been considered, which usually predict trajectories for several seconds in team sports, e.g. basketball and soccer, based on recurrent neural networks (RNN) [62, 63, 64], hierarchical variational RNN [65, 66], transformer [67] and conditional generative adversarial networks [68]. Recent developments in graph neural networks [69, 70, 71, 72] have addressed the permutation problem.

2.2.3 Semi-supervised and self-supervised learning

Supervised learning provides a potent strategy when a large amount of labeled data is available, yet the associated disadvantages include big labor costs and often limited size of labeled datasets. In response, semi-supervised learning combines both labeled and unlabeled data for training machine learning models, integrating aspects of supervised and unsupervised learning.

In semi-supervised learning, the model is trained on a dataset that contains labeled data and unlabeled data, which are usually much bigger than labeled data. The key idea behind semi-supervised learning is to use the limited labeled data along with a more abundant pool of unlabeled data to enhance the model's performance. In team sports, various semi-supervised methods have been applied, ranging from generative models predicting annotated trajectories [73] to the extraction of tactical patterns in soccer using graph neural networks (GNN) [74].

Addressing annotation cost concerns, self-supervised learning provides an alternative by using unlabeled data and deriving supervisory signals through diverse preprocessing techniques. Notably, transformer-based approaches, such as those applied in basketball for recognizing group behaviors [75] and learning trajectory representations [67], stand out as representative examples.

While there are few studies on semi- and self-supervised learning in team sports at present, the potential necessity for such approaches may grow in the future, particularly if access to larger amounts of (unlabeled) data becomes available.

2.2.4 Reinforcement learning

Reinforcement learning (RL) is a machine learning method where an agent learns to make decisions by interacting with an environment. The goal of the agent is to learn a strategy called a policy that maximizes the cumulative reward over time.

Some planning-based methods that are focused on long-term movement objectives primarily formulate agents' policies and paths to reach these objectives, usually related to reinforcement learning. And these planning-based methods can be roughly divided into inverse and forward planning approaches.

The term "inverse planning" refers to the process of working backward from desired outcomes to determine the actions or strategies necessary to achieve those outcomes. In the context of machine learning, specifically reinforcement learning, inverse planning involves the use of statistical learning techniques to estimate action models or reward functions based on observed data.

In team sports, inverse planning approaches are like implementing a reinforcement learning framework in practical situations, where the focus is on assessing the actions and states of agents to accomplish specific goals without features extracting. In the basketball field, inverse planning approaches were utilized in estimating possession outcomes [52, 76], double teaming (a kind of defensive alignment) [77] with deep reinforcement learning, and state-action values to evaluate players and teams [78]. In other sports, to value on-ball actions, several studies have estimated Q-function or other policy functions [79, 80, 81, 82, 83]. In terms of inverse reinforcement learning, there has been research on estimating reward functions [84, 85].

Forward planning approaches center on anticipating future states and actions to achieve the desired outcomes. In the context of reinforcement learning, forward planning involves creating a sequence of actions that leads to optimal results based on predefined criteria. In team sports, forward planning approaches involve devising algorithms aimed at achieving a victory in competitions. For example, 3-vs-3 basketball simulator [86], Google research football [87], and a soccer game with robot teams [88]. However, there are few studies on the integration of inverse and forward planning methods, and closing this gap should be a central focus for future research [89].

2.3 Interpretability in machine learning for team sports

Interpretability in machine learning refers to the ability to understand and explain the results made by a model. In the context of team sports, interpretability becomes crucial for various reasons, including gaining insights into player performance, informing coaching strategies, and ensuring transparency in decision-making processes. In this section, I introduce some aspects of interpretability in machine learning for team sports. Among various interpretability aspects, first, I introduce transparent models, which inherently offer clarity in their decision-making processes. Moving beyond inherent transparency, I describe model-agnostic methods, which provide insights into models regardless of their complexity. Lastly, the role of attention mechanisms in deep learning is explained to derive meaningful insights for team sports trajectories.

2.3.1 Transparent models

In team sports, it is essential for players, coaches, and analysts to comprehend the rationale behind machine learning model predictions, recommendations, and other outputs. Transparent models, such as linear models and decision trees, inherently provide explanations. However, purely linear models are rarely utilized in team sports. On the contrary, decision trees find widespread use in various team sports such as in basketball analysis [90], hockey analysis [91], and soccer [92].

2.3.2 Model-agnostic methods for interpretability

Nevertheless, within team sports, numerous commonly employed machine learning approaches lack transparency. It becomes essential to utilize model-agnostic methods to enhance the interpretability of models. In this case, several model-agnostic methods, such as SHapley Additive exPlanations (SHAP) [19] and Local Interpretable Model-agnostic Explanations (LIME) [18], are widely applied in various domains.

SHAP [19] is a popular and powerful framework for interpreting the output of machine learning models. It is based on cooperative game theory and is designed to allocate the contribution of each feature to the prediction made by the model. The formula for calculating SHAP values is as follows. For a given feature i and model f, the SHAP value ϕ_i is calculated using the following formula:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} \cdot [f(S \cup \{i\}) - f(S)] \tag{2.1}$$

Here, N is the set of all features, S is a subset of features excluding i, $|\cdot|$ is the size of set, $f(S)$ is the model output with features in S, and $f(S \cup \{i\})$ is the output when i is added to S. Each term in the formula represents the marginal contribution of feature i when added to a specific combination of features S. The contributions across all possible combinations are weighted and summed to obtain the SHAP value for feature i. The weighting factor $\frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!}$ ensures a fair distribution of contributions.

In team sports, various studies have applied SHAP, e.g., football team performance prediction [93, 94], volleyball match outcomes prediction [95], and hockey player injury prediction [96].

LIME [18] is another popular technique for interpreting machine learning models, with a focus on providing local explanations for individual predictions. LIME aims to create simple and interpretable models that approximate the behavior of more complex models in a specific local region of the feature space. For example, [97] utilized LIME to deduce and assess the precise reasoning behind the model predictions on NBA outcomes.

2.3.3 Attention mechanism in deep learning

The attention mechanism in deep learning, particularly in the context of neural networks, enhances the model's ability to focus on specific parts of the input when generating the output. It is akin to how human attention works: when we perceive a scene, we focus on a certain part while perceiving others with less focus. In this case, the attention mechanism could enhance the interpretability of the deep learning network. The formula for calculating attention values is as follows. Assume we have an input sequence consisting of n vectors:

$$X = \{x_1, x_2, \ldots, x_n\} \tag{2.2}$$

Each x_i is a d-dimensional vector, typically representing word embeddings or hidden states at a certain time step in the sequence. Attention mechanisms typically introduce three matrices W^Q, W^K, and W^V to map each input vector x_i into a query vector q_i, key vector k_i, and value vector v_i:

$$q_i = W^Q x_i, \quad k_i = W^K x_i, \quad v_i = W^V x_i \tag{2.3}$$

where W^Q, W^K, and W^V are learnable weight matrices. For a given query vector q_i, we calculate its similarity with all key vectors k_j to obtain the attention scores:

$$\text{score}(q_i, k_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}} \tag{2.4}$$

Here, $\cdot$ denotes the dot product, and d_k is the dimensionality of the key vectors, typically equal to the dimensionality of the query vectors. The term $\sqrt{d_k}$ is a scaling factor to mitigate the issue of growing dot product values with increasing dimensions. We convert the attention scores into attention weights (a probability distribution) using the softmax function:

$$\alpha_{ij} = \frac{\exp(\text{score}(q_i, k_j))}{\sum_{j=1}^{n} \exp(\text{score}(q_i, k_j))} \tag{2.5}$$

α_{ij} represents the attention weight on input x_j when computing the representation of input x_i. The output of the attention mechanism is the weighted sum of all value vectors:

$$z_i = \sum_{j=1}^{n} \alpha_{ij} v_j \tag{2.6}$$

where z_i is the final output representation, integrating information from all inputs x_j that are relevant to x_i.

In the field of multi-agent analysis, the attention mechanism is already used in some works to interpret the result. For example, a self-explaining neural network [20] was applied in multi-agent trajectory analysis [98].

Moreover, a multi-head attention mechanism is used in the transformer [99]. That allows the model to jointly attend to information from different representation subspaces at different positions. In team sports, a transformer-based approach [67] for learning team sports trajectory representation was proposed recently.

2.3.4 Technical contributions

In Study I of this thesis, a semi-supervised learning-based method is used to classify cooperative plays in basketball games due to the high labor costs of cooperative play labeling. The proposed method utilizes a small labeled dataset with a large unlabeled dataset to train the semi-supervised learning model. SHAP is used to show the relationship between the input features and the predicted results.

In Study II of this thesis, a neural network approach based on an attention mechanism is adopted for classifying the multi-agent trajectory in basketball and highlighting distinct segments. The attention values are utilized to show the correlation between input variables and the labels. The distinct segments in trajectories can be detected by analyzing attention values.

3 Team play classification via semi-supervised learning in basketball games

3.1 Introduction

Team sports represent a form of coordinated physical activity where participants organize into teams, collectively pursuing common objectives such as scoring, winning matches, or outperforming opponents. The research in this field spans various disciplines, including sports science, psychology, computer science, and management, aiming to comprehensively understand the essence of team sports and their impacts on different levels.

A notable challenge in this research landscape is the classification of complex behavioral data within team sports. Successfully addressing this challenge could substantially enhance our understanding of tactical cooperation or competition in these contexts. However, in such functional collective motions, such as human groups, including team sports, the cooperation often occurs locally and often changes their rules or objectives in a time-dependent manner according to various situations [30, 14]. Therefore, the large variance of behaviors even within the same functional formations makes it difficult to find similar and different motion structures using the same and different labels, respectively [15].

Human groups performing sports have been recently studied (see also [11]) to clarify the relationship between collective motion and an objective such as team plays [12, 13, 14, 15, 16] and achieving a score [51, 9]. However, these investigations mainly employed supervised learning methodologies and extensive datasets requiring manual labeling. This reliance resulted in substantial labor costs associated with the labeling process. Consequently, while these studies have contributed valuable insights, they also highlight the need for more efficient and scalable analytical methodologies capable of adapting to the dynamic and complex of team sports.

While classification methods based on semi-supervised learning [17] often work well when labeled data is scarce. In such cases, it may be difficult to construct a reliable supervised classifier, such as in computer-aided diagnosis, drug discovery, and part-of-speech tagging [100]. Similarly, for team plays in sports, since there have still been no public annotated datasets, semi-supervised learning approaches will have the potential to improve the detection performance of labeled cooperative behaviors. However, the effect of semi-supervised learning methods on classification performance in cooperative plays of a team sport is unknown.

This study examines semi-supervised learning methods for the classification of strategic cooperative plays (called screen plays) in basketball using a smaller labeled dataset and a larger unlabeled dataset, as illustrated in Fig. 3.1. After segmentation, I extract the input features of the classifier for each screen play candidate. Finally, I obtain predicted labels (i.e., screen play types) via semi-supervised learning models. I compare the classification performance of several basic semi-supervised learning methods and a supervised method and analyze the differences in the importance of the input features between both approaches. The purpose of this study is to investigate the effect of semi-supervised learning methods on the classification performance of cooperative plays in team sports.

To this end, I used screen play data to validate our method. Screen play is the play in which an offensive player (called a "screener") is standing on the course of a defensive player like a

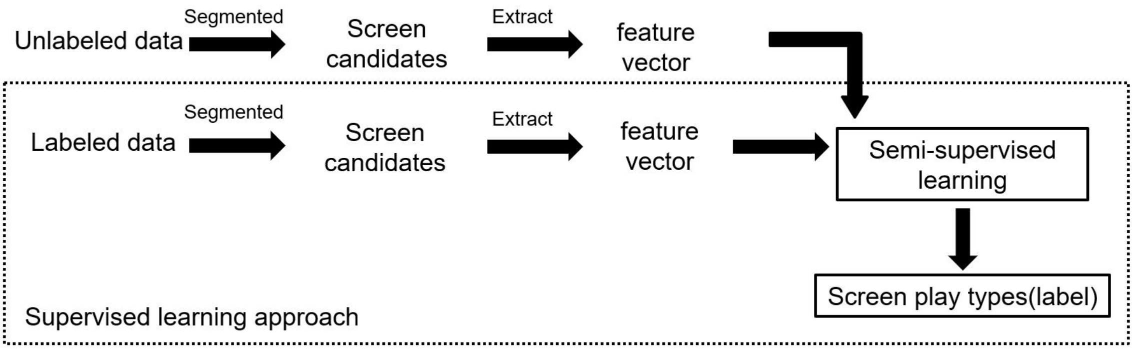

Figure 3.1: **Overview of our semi-supervised approach.** *I use both labeled and (larger amount of) unlabeled datasets. After the segmentation, I extract the input feature of the classifier for each screen play candidate. I then perform semi-supervised learning and finally obtain predicted labels (i.e., screen play types). The dashed rectangle is a supervised approach.*

wall and uses their body to prevent the defensive movement against another offensive player (called a "user") in a legal way (Fig. 3.2), allowing the teammate to move more freely and potentially create a scoring opportunity. Screen play is common and basic but important in basketball games, and various basketball tactics rely on screen play. Therefore, in this study, I choose screen play as the classification target. The main contributions of this work are as follows:

1. I perform the classification of cooperative plays in team sports via existing semi-supervised learning frameworks, including self-training, label-propagtion, and label-spreading.
2. Various types of screen plays, including minor types, are classified in completely automatic ways.
3. Results show that the classification performance of the semi-supervised learning approaches improved upon the conventional supervised approach for minor types of screen plays.

3.2 Materials and methods

A dataset of international games with labels

I used a dataset with labels of various screen plays [15]. Here, I describe how the data were collected. The positional data of players and the ball (25 frames per second) were obtained from men's Asian international-level practical games held in 2015 and preprocessed by the STATS SportVU system (Northbrook, IL, USA). Data acquisition was based on the contract between the teams and the company (STATS LLC.), not between the players and me. They are top-level players, and then the data was not anonymized. The company was licensed to acquire this data, and it was guaranteed that the use of the data would not infringe on any rights of players or teams. I analyzed 220 min of play (in four days) in which the two teams scored 746 points (386 vs. 360). For each day, players performed one and a half games (i.e., 60 minutes) except for one day (only one game). The positional data contained the XY position of each player on the court and the XYZ coordinates of the ball. I used this dataset as the labeled dataset for supervised classification in the semi-supervised classification framework.

A dataset of NBA games with no label

I also used the basketball dataset from the National Basketball Association (NBA) 2015-2016 season preprocessed by the STATS SportVU system (Northbrook, IL, USA). The dataset contains trajectories of basketball players and the ball in the same format above. Data acquisition was based on the contract between the league and the company (STATS LLC.), not between the players and us (the remaining part is the same as the above). I chose 100 games from the dataset because I obtained enough amount of data (for details, see below). I did not annotate the screen play labels and used them as the unlabelled dataset for the semi-supervised classification.

Data segmentation

Prior to data segmentation, I used an automatic individual play-detection system, such as a shot using the positional data. In addition, for our analyses and to obtain label information, all types of screens were labeled visually (for details, see [14, 15, 51]). In data segmentation, I segmented $2,038$ samples in the labeled dataset and $222,334$ samples in the unlabeled dataset (called "actions"), in which an offensive player moves to a defensive player and two offenses who might use the screen and their defenses (shown in Fig. 3.2) were automatically detected. At least two attackers are related to a screen play; a screener and a user. A screener is defined as the attacker to set the screen. A user is defined as the attacker who uses the screen to free from the defender. In legal screen plays, a screener sets the screen, and then the user starts to move. An off-ball screen play is a screen play without relation to the ball directly. In this study, an off-ball screen play was defined as a screen play in which the candidate screener and user do not possess the ball when the distance was the shortest (I call it the minimum distance in Fig. 3.2), whereas all other actions were defined as on-ball screen plays.

For the segmentation, first, all offensive players were considered to be candidate screeners, and for each candidate, two other offensive and three defensive players were defined as candidate users and candidate defenders for screeners and users, respectively. Then, a signal that screen play was likely to occur was defined if the players satisfied the following two conditions: (1) the distance between a candidate screener and a candidate user-defender was less than 1.2 m, and (2) the user-defender was the closest player to the candidate user. A user-defender is defined as the defender mainly defending the user. I define the segment of screen play candidates as 25 frames before and after the minimum distance in which the distance between the screener and the user-defender was the minimum. I refer to the first and last of the 25 frames as the start time and end time, respectively. Signals that were too short

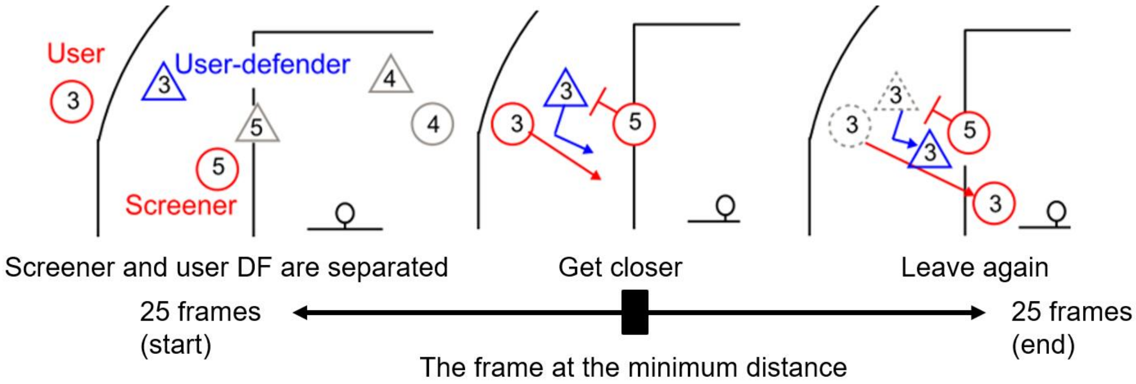

Figure 3.2: **Segmentation of screen play candidates.** *A signal that screen play was likely to occur was defined if the players satisfied the following two conditions: (1) the distance between a candidate screener and a candidate user-defender was less than 1.2 m, and (2) the user-defender was the closest player to the candidate user. I define the segment of screen play candidates as 25 frames before and after the minimum distance in which the distance between the screener and the user-defender was the minimum.*

(less than three successive frames) were excluded from the analysis, and temporary adjacent actions were jointed. This is because, in a legal screen play, a screener sets the screen, and then the user moves in, and the distance keeps short at a certain time. The authors in the previous work [15], who have experience playing basketball, labeled all actions (Table 3.1).

Feature vectors

I then computed three types of feature vectors for all classifiers (see the subsections below) according to the previous work [15]: the inter-agent distance (40 dimensions), individual geometric information (50 dimensions), and individual moving distance (4 dimensions). I first computed the inter-agent distance employed in a previous study [12]. They used eight distances between two of the screeners, the user, the user-defender, and the ball. Here, I consider the following distances: S-D, S-U, D-U, S-U2, D-U2, S-B, U-B, and U2-B, where

Table 3.1: **Types of screen plays in the labeled dataset.** *All types of screen plays based on the previous study (Hojo et al., 2018) are described, except for the flex screen because of fewer data (only 8 samples).*

category	screen type	times	description
off-ball	down	104	A screen usually sets a screen relatively near the basket ring and the user uses it to move toward the passer.
off-ball	flare	68	A screener sets a screen far from the basket ring, and it allows the user to move away from the passer and the ring.
off-ball	pin	23	A screener sets a screen near the basket and baseline, and the user uses it to move toward the corner.
off-ball	back	31	A screener sets a screen on the back side of a user-defender far from the ring, and the user uses it to move toward the corner.
off-ball	cross	51	A screener moves parallel to the baseline toward the user, sets the screen, and the user moves toward the screener ’ s past position.
on-ball	pick	140	A screener sets a screen for the user who has the ball (usually dribbling).
on-ball	hand-off	42	A screener has the ball, makes a handoff pass, and sets the screen for the user.
no screen	no screen	1427	No screen play.

S, D, U, U2, and B denote a screener, a user-defender, the first user candidate, the second user candidate, and the ball, respectively. For each distance, I computed five features. Here, I denote the time T1, T2, and T3 as the start, the minimum distance, and the end of the analyzed interval, respectively. For each distance, I computed the distance at time T1, T2, and T3, and the average in distance from T1 to T2 and that from T2 and T3. Thus, the inter-agent distance had 40 dimensions in total. Next, the moving distance and geometric information of the individual player were considered based on the previous work [15]. The moving distances include the distances of four players (the screener, the defense of the user, and two candidate users) from the start to the end of the action. The geometric information includes the distances and angles from the midpoint of the endline to each of the four players and the ball. For each distance and angle, I computed five features in the same way as above (i.e., the distance at time T1, T2, and T3, and the average distance from T1 to T2 and that from T2 and T3).

Supervised classification method as a baseline

Here, I describe a supervised classification method as a baseline. This study employed a multi-class support vector machine (SVM) to classify the detailed six types of off-ball screen plays, two types of on-ball screen plays, and no screen play. SVM is widely used for discriminative classification to find the optimal hyper-plane between two classes [101]. A soft margin SVM using a Gaussian kernel was employed. Since SVM can not directly solve multi-class classification problems, there are two strategies to solve this problem: one-to-all SVM and one-to-one SVM. For the one-to-all SVM, k 2-class SVM models (k is the number of classes. In this study, $k = 9$) were constructed. In the training of each SVM model in one-to-all SVM, one class is labeled as positive, and the remaining classes are labeled as negative. For the one-to-one SVM, $k(k-1)/2$ two-class SVM models were constructed. In the training of each SVM model in one-to-one SVM, one class is labeled as positive, and another class is labeled as negative, then the other classes are ignored. In general, for a few classes' classifications, one-to-all SVM will lead to a high accuracy [102] (11 classes). On the other hand, as the number of classes increases, the imbalance of the number of samples will lead to a decrease in accuracy. One-to-one SVM can solve the problem of imbalance to some extent. In this study, the imbalance problem occurred in both strategies (see Table 3.1), and the classification problem only considered nine classes; thus, I used the one-to-all SVM, similarly to the previous study [15].

Semi-supervised learning

Here I introduce three classical semi-supervised learning methods: self-training (e.g., [103]), label-propagation (e.g., [104]), and label-spreading (e.g, [105]). Although there have been many semi-supervised variants, for interpreting the results and having fewer data, I used the representative two methods.

The first is self-training, which uses labeled data to train the classifier. The algorithm is shown as follows.

1. $(Initial\ Training)$ Self-training starts with a small amount of labeled data. This data is used to train a basic model.
2. $(Prediction\ on\ Unlabeled\ Data)$ The trained model is then used to make predictions on the unlabeled data.
3. $(Confidence\ Thresholding)$ The model's predictions are evaluated based on a confidence score. Predictions with confidence scores above a certain threshold are considered reliable. In this study, the threshold was set as 0.75 of prediction probability.
4. $(Labeling\ Unlabeled\ Data)$ The highly confident predictions are used as pseudo-labels for the previously unlabeled data.
5. $(Re-training\ the\ Model)$ The model is re-trained on the expanded dataset, which now includes both the original labeled data and the new pseudo-labeled data.
6. $(Iteration)$ Steps 2 through 5 are often repeated multiple times until there are no changes in the dataset. With each iteration, the model should ideally become more accurate as it's being trained on progressively more data.

The second is label-propagation. The algorithm is shown as follows.

1. $(Graph\ Construction)$ The algorithm begins by constructing a graph where each data

point, whether labeled or unlabeled, is represented as a node. Edges between nodes are created based on the similarity between data points. This similarity is often calculated using metrics like Euclidean distance.

2. (*Propagation of Labels*) The algorithm then propagates these labels through the graph. The idea is to spread the label information from the labeled nodes to the unlabeled nodes based on the probabilistic transition matrix.
3. (*Iterative Process*) This propagation is typically an iterative process. In each iteration, every unlabeled node updates its label based on the labels of its neighboring nodes.
4. (*Convergence*) The process continues until the labels converge, which means that the labels of the unlabeled nodes do not change significantly between successive iterations.
5. (*Final Label Assignment*) Once the labels have converged, each unlabeled node is assigned the label that it most frequently received during the propagation process.

As a similar graph-based method, label-spreading (e.g., [105]) has been proposed, but I used label-propagation method for simplicity (I obtained similar results between label-propagation and label-spreading methods).

To compare with the one-to-all SVM, I use SVM as the base classifier for the self-train algorithm. For label-propagation algorithm, I use the *fitsemigraph* function in Matlab 2021a with the default hyper-parameters (e.g., Euclidean distance was used for the distance).

Statistical analysis

The F1 score was mainly used to validate the classifier. This is because the accuracy presented issues when there were significantly more negative cases than positive ones, as was the case in this study. As an intuitive example, accuracy scores are good even when all negative cases are predicted from the data with only 10% positive cases. Instead, I mainly used the F1 score to evaluate whether the true positives can be classified without considering the true negatives. The F1 score is expressed as

$$\mathrm{F1score} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{(\mathrm{Precision} + \mathrm{Recall})}, \tag{3.1}$$

where the Recall is equal to the true-positive rate, and the Precision is defined as the ratio of the sum of true positives and true negatives to false positives. In this index, only true positives are evaluated, not true negatives. Furthermore, the contribution of the input variables to the prediction of the method was calculated by SHAP (SHapley Additive exPlanations) [19], which utilizes an interpretable approximate model of the original nonlinear prediction model.

3.3 Results

First, the classification performances among the methods are shown. Three methods are compared: SVM, self-training, and label-propagation. These methods classify eight types of screen plays, including minor types (i.e., flare, pin, back, cross, and hand-off). As described above, the F1 score was used as the classification performance because the datasets were imbalanced, and true positives should be appropriately evaluated. As shown in Table 3.2 using a 95% confidence interval with the normal distribution, overall, classification performances in semi-supervised methods were higher than those in the supervised method (SVM). For major types of screen plays (i.e., down, pick, and no-screen), self-training shows similar or better classification performance than other methods. For down screen, self-training outperformed SVM, but for pick and no-screen, the performances in self-training were similar to those of SVM. Label-propagation shows the worst performance for the major types of screen plays. For minor types of screen plays, three types of semi-supervised methods show better clarification performance than the supervised method. For flare, back, and hand-off screens, self-training outperformed SVM, and label-propagation outperformed self-training.

For pin and cross screens, the performances of self-training and SVM were similar, but label-propagation outperformed both.

	Supervised method			Semi-supervised methods					
	SVM			S-train			Label-propagation		
	Mean	95% CI (lower)	95%CI (higher)	Mean	95%CI (lower)	95%CI (higher)	Mean	95%CI (lower)	95%CI (higher)
down	0.394	−0.017	0.017	**0.422**	−0.014	0.014	0.361	−0.009	0.009
flare	0.003	−0.037	0.003	0.037	−0.015	0.015	**0.173**	−0.010	0.010
pin	0.003	−0.003	0.004	0.005	−0.005	0.005	**0.039**	−0.009	0.009
back	0.114	−0.025	0.025	0.173	−0.031	0.031	**0.224**	−0.015	0.015
cross	0.314	−0.018	0.018	0.322	−0.019	0.019	**0.379**	−0.014	0.014
pick	0.613	−0.009	0.009	**0.615**	−0.008	0.008	0.510	−0.007	0.007
hand-off	0.031	−0.013	0.013	0.069	−0.020	0.020	**0.228**	−0.013	0.013
no screen	**0.879**	−0.015	0.015	**0.879**	−0.015	0.015	0.825	−0.017	0.017

Table 3.2: **Classification performance of two methods for eight classification tasks.** F1 score and accuracy are indicated. Overall, classification performances in the semi-supervised method were higher than those in the supervised method.

Next, I show the contribution of the input variables to the prediction of the SVM and self-training method by SHAP [19] in Fig. 3.3. For variable names, here I denote dist as the distance between the two players or individual moving distance among screener (S), screener-defender (D), a first user candidate (U), a second user candidate (U2), and the ball (B). I also denote the ang as the angle between the two players with the goal as the center or the individual position angle relative to the center and a sideline.

For down screen and cross screen, the results of orders and SHAP values were very close. For down screen, distSD, distD, distU, distU2B, distSU, distUB, and distU2 show high and close SHAP values in both SVM and self-training. For cross screen, distS, angU, angU2, distSD, distU, distU2, and angD indicate high and SHAP values score in both SVM and self-training, but the SHAP values of these input variables in self-training were slightly lower than those in SVM. For back screen, the results of orders and sharpley values were quite different. Most input variables show much higher SHAP values in self-training than in SVM.

Higher SHAP values suggest that features have a more significant impact on model predictions, making these effects easier to observe and understand. Self-training shows higher SHAP values for most features compared to SVM, which indicates that the contributions of features to the predictions are more pronounced and significant in self-training. This is often interpreted as the model having higher interpretability because the impact of each feature on the predictions is easier to identify and explain.

3.4 Summary

This study proposed a semi-supervised learning-based method for classifying cooperative play, in particular, various types of screen plays, including minor types, in completely automatic ways. This approach utilizes a large unlabeled dataset and significantly reduces the labor costs associated with data labeling. The proposed method first roughly filters potential segments through the definition of screen play. Then, based on previous related research [15], feature vectors for the semi-supervised learning model are extracted from these segments. These feature vectors, combined with a small amount of labeled data, are used to train three types of semi-supervised learning models, including self-training, label-propagation, and label-spreading. The effectiveness of the proposed method is validated by comparing it with supervised learning models trained on the same labeled dataset. Additionally, the relevance of various feature values to the prediction results is quantified by calculating the SHAP values of the input features.

In future work, the increase in the dataset amount could be considered. From Table 3.2, for some minor types of screen plays, even though the semi-supervised learning obtained higher F1 scores than SVM, the results were still not good. Major types of screen plays got much better results than minor types of screen plays. To improve the classification performances, the increase of data amount, especially minor types of screen plays, will be needed.

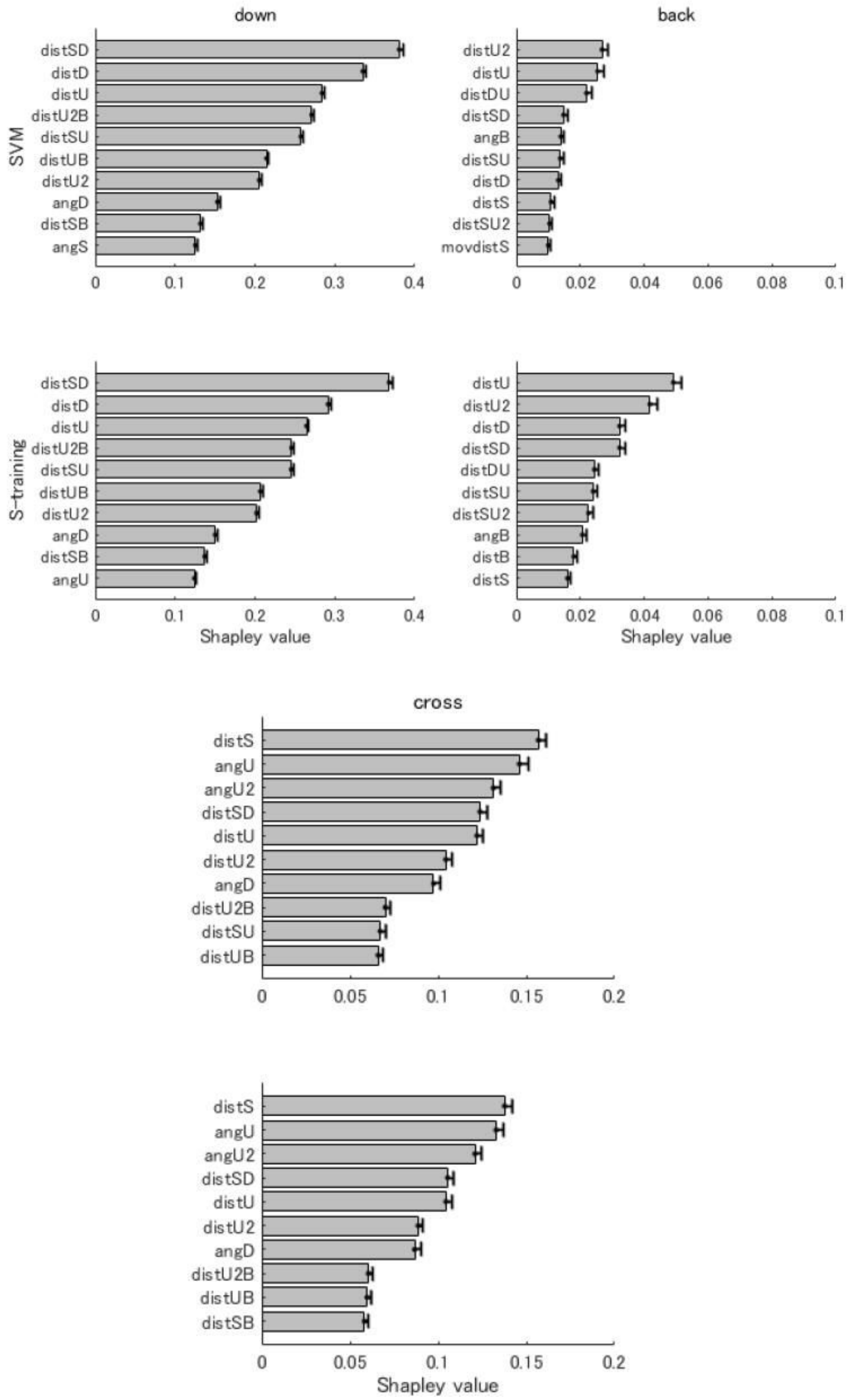

Figure 3.3: **Contribution of the input variables to the prediction of the SVM and self-training method.** *Since the SVM (upper, supervised) and the self-training (lower, semi-supervised) can be fairly compared, here I show the SHAP values in down, back, and cross screen plays, of which classification performances improved in the self-training. Of the top 10 features, those at the top had greater contributions than those at the bottom.*

4 Multi-agent comparative analysis of team sport trajectories

4.1 Introduction

Data-driven modeling is a powerful tool that can extract information and make predictions from complex real-world data e.g., multi-agent trajectories. Machine learning has actively studied the learning process of models with complex, nonlinear structures such as neural networks [11]. Although these models can offer higher expressiveness and predictive performance, their results can be challenging to interpret, creating a trade-off between interpretability and expressiveness (or predictability). This challenge is particularly important for practical applications in actual sports games, where coaches and players need information about why a goal was scored and the characteristics observed in subsequent plays.

Currently, the trajectory data of players and the ball in professional sports (e.g., basketball or soccer) can be accessed. For trajectory analysis, combining with labels (e.g., good or bad attacks in ball games) can provide insights compared to only trajectory data. There have been many approaches for supervised and unsupervised learning of multi-agent trajectory data (see Section 2.2). Compared with previous approaches, the analysis of multiple player trajectories by highlighting the difference between labels to understand multi-agent behaviors has not been considered. In a different research field, that of animal trajectory analysis, comparing two data classes to obtain useful insights is referred to as comparative analysis using DeepHL [21]. However, this deep-learning-based method used only single-agent animal trajectories; multi-agent motion characteristics (e.g., the distances between agents) were not considered.

In this study, a comparative analysis method to analyze multi-agent trajectories for team play in a ball game is proposed, called Multi-Agent Deep-learning based Comparative Analysis (MADCA, Fig 4.1). A neural network approach based on an attention mechanism is adopted, which is the combination of a convolutional neural network (CNN) and a recurrent neural network (RNN), to detect distinct segments in trajectories of given classes (Fig 4.1c). Compared with the previous method [21], to apply the model to the multi-agent modeling task, I input additional multi-agent motion features (e.g., distance between the agents) to show the interaction between agents. Then, because of the significant increase in the training time for multi-agent tasks using MADCA, I made some modifications to the structure of the DeepHL model to improve training speed without significantly impacting performance. Finally, a new interpretable and simple label, effective attack, was defined to evaluate the performance of a team.

This method enables us to understand the differences between classes by highlighting segmented trajectories and identifying which variables correlate with the labels (Fig 4.1d). Our approach was verified by comparing various ablated models and demonstrated its effectiveness through use cases that analyze the difference between effective and ineffective attacks in US National Basketball Association (NBA) games. For example, based on the correlation between attention values and the handcrafted features, the distance between the shooter and the shooter defender was selected and the histograms of the selected feature were clearly different between the effective and ineffective attack classes.

The main contributions of this section are as follows:

1. MADCA is proposed, a comparative analysis method to analyze multi-agent trajectories in ball games, the goal of which is to understand the differences between classes by

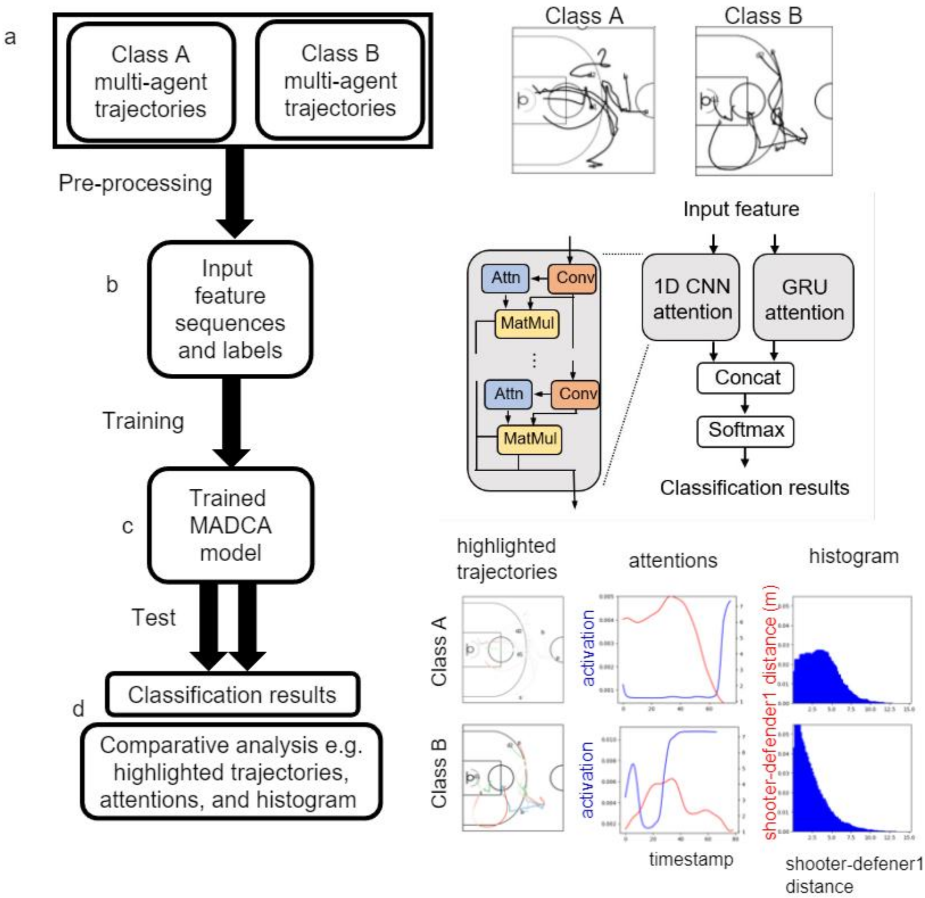

Figure 4.1: **Our framework of multi-agent trajectory comparative analysis using MADCA.** *(a) Multi-agent trajectories of classes A and B are given. (b) Input feature sequences are computed by pre-processing. (c) A MADCA model called MADCA-net comprises a 1D convolutional neural network (CNN) and gated recurrent unit (GRU) with attention mechanism and outputs classification results. (d) The main outputs are classification results, highlighted trajectories (left), attentions for each layer and time (middle), and attended features (right). For details, see Sections 4.2.2 and 4.2.5.*

highlighting segmented trajectories and which variables correlate with the labels.

2. A neural network approach based on an attention mechanism is adopted that uses multi-agent motion characteristics (e.g., the distances between agents and objects) as input and detects distinct segments in trajectories of given classes.
3. Our approach is verified by comparing various ablated models with effective/ineffective attack labels and goal/non-goal labels, using different sizes of the dataset. We also demonstrate the effectiveness of our method through use cases that analyze the difference between effective and ineffective attacks in the NBA dataset.

The remainder of this section is organized as follows. First, our method is described in Section 4.2. The experimental results are presented and discussed in Sections 4.3 and 4.4, respectively.

4.2 Materials and Methods

In this section, the dataset is first described, then our machine learning model, preprocessing, and analysis procedures.

4.2.1 Dataset

A basketball dataset from the NBA 2015-2016 season, preprocessed by the STATS SportVU system (Northbrook, IL, USA), was used, which contains the positional data of players and the ball (at a frequency of 25 frames per second). 600 games from the dataset were chosen because we deemed that this represented sufficient data. The positional data contained the (x,y) positions of each player on the court and the (x,y,z) coordinates of the ball. The dataset was divided beforehand into attack segments from the start of the attack (ball possession of the team or already divided segmentation in the raw data) to the transition to the next attack. The end of the attack segment is defined as being when a shot is made or the ball is lost (known as a turnover in basketball). The data contained a total of 45,307 attacks, sub-sampled at 10 Hz, i.e., the time between each point coordinate is always 0.1 seconds. In our dataset, there were 18,021 shot successes, 27,286 shot failures (including 7131 turnovers), 22,159 effective attacks, and 23,148 ineffective attacks (the definitions of effective/ineffective attacks are described in Section 4.2.3). The probabilities of scoring, given the attack was effective and ineffective, were 0.466 and 0.333, respectively, which indicates that the effective attack indicator is valid in terms of scoring on average. However, in a strict sense, scoring and effective attack are different (for further detail, see Section 4.2.3).

In our analysis, the trajectories of five agents (Fig 4.1a) are considered. These five agents comprise the ball and four players: the shooter (or the last player who was on the ball, including a ball lost case), the defender of the shooter at the last frame (called DF1), the last passer to the shooter (called passer), and the defender of the last passer at the last frame (called DF2). These agents were selected because the verification of our approach is focused on, and all trajectories may be too diverse for the model to learn a good representation for highlighting the differences between the two labels. In general, this is a multi-agent role assignment problem for an unsorted diverse dataset (see e.g., [40]). This problem is avoided by using only predetermined roles about four players and the ball. It is considered that it is more difficult to determine the roles in a fixed manner as the number of players increases, and fewer players may be less informative in this analysis. Then, the interval from the ball-receiving time of the passer to the end of the above attack segment was analyzed.

4.2.2 Proposed Model

In general, neural network approaches are flexible in the field of team sports (e.g., trajectory prediction [62, 106, 107, 108, 65, 109]), but they sometimes lack interpretability. To obtain interpretable spatial representations, several approaches have been developed such as using matrix [58] and tensor [59, 110] factor models, and Poisson point processes [60] that focus on on-ball behaviors. Compared with these approaches, the analysis of the trajectories of multiple players to understand multi-agent behaviors using a neural network with attention mechanism is focused on in this study.

The proposed approach is designed to understand the differences between classes of multi-agent trajectories in sports by highlighting segmented trajectories and which variables correlate with the labels. To this end, a comparative analysis method is proposed to analyze multi-agent trajectories called MADCA, which extends the single-agent trajectory DeepHL framework [21] to the multi-agent trajectory problem. Similar to [21], we assume that there are two classes of trajectory data with different characteristics such that each trajectory with either class A or B (e.g., scored or not). Here, the neural networks for MADCA are explained, which are shown in Fig. 4.1c. The pipeline of MADCA is briefly introduced such that:

1. Our network (hereafter called MADCA-Net) is first trained using the trajectory data of two classes, which is the combination of CNN and RNN.
2. The attention mechanism in MADCA-Net computes the attention values of each time stamp of the trajectory.
3. After obtaining the attention, distinctive parts of the trajectories in two classes are highlighted using the attention in a particular layer. To find such a layer (hereafter referred to as a “distinctive layer”), the score is computed for each layer using the attention value.
4. MADCA also extracts the highlighted segments with handcrafted features, which is

based on the correlation between the attention values and the features (Fig. 4.1d middle).

Next, the input of the model is described. The input is a time series of features, an $l_{MAX} \times N_f$ matrix, where l_{MAX} is the maximum length of the input trajectories and N_f is the dimension of the features. The features in DeepHL [21] include basic features used in locomotion analysis (e.g., position and speed). In this study, multi-agent motion features such as the distances between the K agents and an object (e.g., a goal called a ring in basketball) are used. For further details of the input features, see Subsection 4.2.3. Since the lengths of the trajectories are not identical to each other, the missing elements are masked when training the network.

MADCA-Net classifies a trajectory into either class and outputs the segments of a trajectory to which the distinctive layer pays attention. Figure 4.1c shows the architecture of MADCA-Net, consisting of four stacks of 1D convolutional layers and a gated recurrent unit (GRU) [111] layer, which is one of the RNN architectures. As used in DeepHL [21], the 1D convolutional layers (the orange blocks in Fig. 4.1c) extract short-term features. To compute features at different levels of scale, in each 1D convolutional layer, we extract features using a kernel size or filter width F_t, which are $3\%, 6\%, 9\%$, and 12% of l_{MAX} in the four convolutional stacks. A step size or stride of one sample is used in terms of the time axis. Padding is employed to ensure that the length of the outputs of a given layer is corresponded with that of the inputs to the layer. The convolutional stacks are constructed to compute features at different levels of scale by utilizing different filter sizes across the different stacks.

In contrast, the GRU layers compute features reflecting long-term dependencies (the configuration of the attention mechanism is the same as in the 1D convolutional layers). Compared with DeepHL [21], which uses long short-term memory (LSTM), we used a GRU, which has a smaller number of parameters than LSTM, since in preliminary experiments, the original DeepHL models took a long time to train. In addition, because we believe last-moment information (e.g., shot) will be important, and the different levels of abstraction may not be important for multi-agent trajectory data in sports, a simple two-layer GRU is implemented rather than four stacks of LSTMs with different layer sizes [21].

In order to identify which segments of the trajectories are significant for each layer, an attention mechanism [112] is incorporated into the model like the previous work [21] as illustrated in Fig. 4.1c. The output of each 1D convolutional/GRU layer for an input trajectory is used to calculate the attention vector such that

$$\mathbf{a} = \mathrm{softmax}\left(\tanh(\mathbf{W}_{\mathrm{a}}\mathbf{Z}^{\top} + \mathbf{b}_{\mathrm{a}})\right), \tag{4.1}$$

where $\mathbf{a} \in \mathbb{R}^{1\times l_{\mathrm{MAX}}}$. This indicates the importance of each time stamp in the trajectory, which is used to highlight parts of the trajectory. An attention vector has the same length as the trajectory, where l_{MAX} is the maximum length of the input trajectories. Matrix $\mathbf{Z} \in \mathbb{R}^{l_{\mathrm{MAX}}\times N}$ is an output of the 1D convolutional/GRU layer, where N is the number of nodes in the convolutional/GRU layer. $\mathbf{W}_{\mathrm{a}} \in \mathbb{R}^{1\times N}$ and $\mathbf{b}_{\mathrm{a}} \in \mathbb{R}^{1\times l_{\mathrm{MAX}}}$ are the weight matrix and its bias, respectively. The softmax function ensures that the sum of all output values is equal to one, while the tanh function constrains the output value of its input to a range between -1 and 1. The attention mechanism is implemented as a neural network in MADCA-Net, specifically using layer-wise attention as indicated by the aqua blocks in Fig. 4.1c. The attention vector $\mathbf{a}$ is multiplied by the outputs of the 1D convolutional/GRU layer using matrix multiplication (MatMul), as shown by the khaki blocks in Fig. 4.1c. The outputs of all layers are concatenated and used to estimate the class (Class A or B) in the final layer of MADCA-Net, which is a densely connected output layer using the softmax function, as shown in Fig. 4.1c. Again, the model is designed to compute attention information at different levels of scale using 1D convolutional/GRU layers.

It is noteworthy that the parameters in Eq. (4.1) for each layer, including $\mathbf{W}_{\mathrm{a}}$, $\mathbf{b}_{\mathrm{a}}$, and the parameters of the convolutional and GRU layers, are estimated in the training. The tanh activation function is introduced into Eq. (4.1) to smooth the output attention. Without the tanh function, if an outlying large value is included in $\mathbf{W}_a\mathbf{Z}^\top + \mathbf{b}_a$ at time t, attention values other than those at time t become extremely small. This causes only one data point to be colored in red when visualizing a trajectory using such attention values, making it difficult to identify important segments.

For processing the layers in MADCA-Net, a scoring system is employed, similar to that used in DeepHL [21], using the following equation:

$$s(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}) = s_{\mathrm{fc}}(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}) + s_{\mathrm{it}}(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}), \tag{4.2}$$

where $\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}$ and $\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}$ with the ith layer are sets of attention vectors calculated from trajectories belonging to class A and B, respectively. Since an attention vector from a distinctive layer is expected to exhibit high values within a restricted range of segments, $s_{\mathrm{fc}}(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}})$ computes the average variance of the attention values, which is normalized based on the average trajectory length, as follows:

$$\begin{aligned} &s_{\mathrm{fc}}(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}) \\ &= \sqrt{\frac{1}{|\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}} \cup \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}| \cdot l(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}} \cup \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}})} \sum_{\mathbf{a} \in \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}} \cup \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}} \mathrm{Var}(\mathbf{a})}, \end{aligned} \tag{4.3}$$

where $\mathrm{Var}(\cdot)$ computes the variance and $l(\cdot)$ computes the average trajectory length. Using $l(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}} \cup \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}})$, the computed variance is normalized. To prevent a larger variance for longer trajectories because the softmax function in Eq. (4.1) ensures that all values sum to one, the average variance is normalized using the average trajectory length.

Additionally, to assess the difference in attention value distributions between the two classes, the score $s_{\mathrm{it}}(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}})$ is computed as follows:

$$s_{\mathrm{it}}(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}, \mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}) = (1 - \mathrm{Intersect}(h(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{A}}), h(\mathbf{A}_{\mathbf{i},\mathbf{C}_\mathrm{B}}))). \tag{4.4}$$

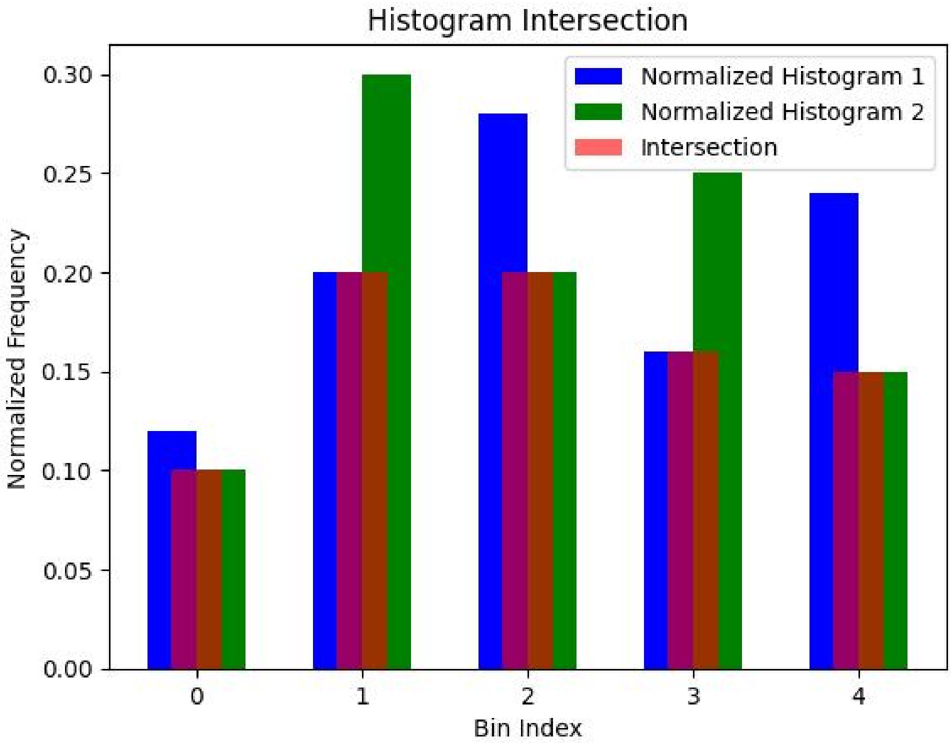

Figure 4.2: **Example of histogram intersection.** *In this fictitious example, 5 bins for normalized histograms 1 and 2 are considered. Equation 4.5 computes the sum of the intersection histogram.*

Based on DeepHL [21], $h(\cdot)$ computes a normalized histogram of attention values with 200 bins, and $\text{Intersect}(\cdot,\cdot)$ computes the area of overlap between the two histograms as illustrated in Figure 4.2. Specifically, the computation is given by the equation:

$$\text{Intersect}\left(H_1, H_2\right) = \sum_i \min(H_1(i), H_2(i)), \tag{4.5}$$

where $H_1(i)$ denotes the normalized frequency of the ith bin of histogram H_1.

4.2.3 Processing Procedure

Here, the steps taken in this study are described to compute the input features and target labels for MADCA-net. For the input features, the variables used in DeepHL [21] are first computed: position, speed, and distance from the initial location for each agent. The location data have two dimensions, while the speed (the norm of two-dimensional velocity) and distance from the initial location have one dimension for each agent. In this study, as multi-agent features, various distances among the agents and the ring (a fixed object) were added: shooter-DF1, shooter-DF2, passer-DF1, passer-DF2, shooter-ring, passer-ring, and ball-ring. Furthermore, the moving average and moving variance of the above variables as per DeepHL [21] are computed.

Next, the steps are described to compute target labels. Two types of labels are considered: goal/no-goal and effective/ineffective attacks. The two MADCA-Nets were trained separately using the two types of labels. The goal/no-goal label can be straightforwardly defined based on the results of the attacks. However, since goal predictions are difficult in general (e.g., [51, 9]), another label was defined to be based on whether or not a particular play was an "effective attack," rather than whether a goal was scored in a particular play.

The tactics and strategy of a coach and team may be most influential up until the point at which there is a good scoring opportunity to make a shot, and it is then the skills/form of the individual player that determines whether this opportunity is actually converted into a goal. A good scoring opportunity in basketball is considered to be a shot being attempted in a context in which there is a high expected probability of scoring based on historical attempted and successful shots. Therefore, an interpretable and simple indicator is computed from available statistics (i.e., based on the frequency) to evaluate whether a player makes an effective shot attempt, rather than using a label based on whether a goal was scored or a learning-based score prediction model.

From the available NBA statistics, two basic factors were focused on for effective attacks at an individual player level: the shot zone on the court and the distance between a shooter and the nearest defender. This is based on the previous work [109] that only uses shooter and DF1 information (not all four players). These two factors are considered important for basketball successful shot prediction [14, 51, 9]. In the NBA advanced stats [113], probabilities of successful shots attempted in each zone and distances for each player can be accessed. The shot zones are separated into four areas: the restricted area, in-the-paint, mid-range, and 3-point area. The restricted area is defined as the area within a radius of 2.44 m (the distance between the side of the rectangle and the ring) from the ring. The in-the-paint area is defined as the area within a radius of 5.46 m (the distance between the ring and the farthest vertex of the rectangle) from the ring. The three-point area is defined as the area that is outside of the 3-point line. The mid-range area is the remaining area. The shooter's distance from the nearest defender is categorized into four ranges: 0-2 feet, 2-4 feet, 4-6 feet, and more than 6 feet.

An effective attack is defined as one that meets the following criteria:

- The shooter's position in the restricted area is effective at any distance (because a defender is often located near the shooter).
- The shooter's position in the paint and mid-range is effective at a distance of 6 feet or more from the nearest defender (this range is regarded as "open" in the NBA advanced stats).
- The shooter's position in the 3-point area is effective when a player with a shot success probability of at least 0.35 attempts a shot at a distance of 6 feet or more from the nearest defender (because some players do not shoot tactically).

Based on the statistics in the 2014/2015 season and the tracking data, The probabilities of

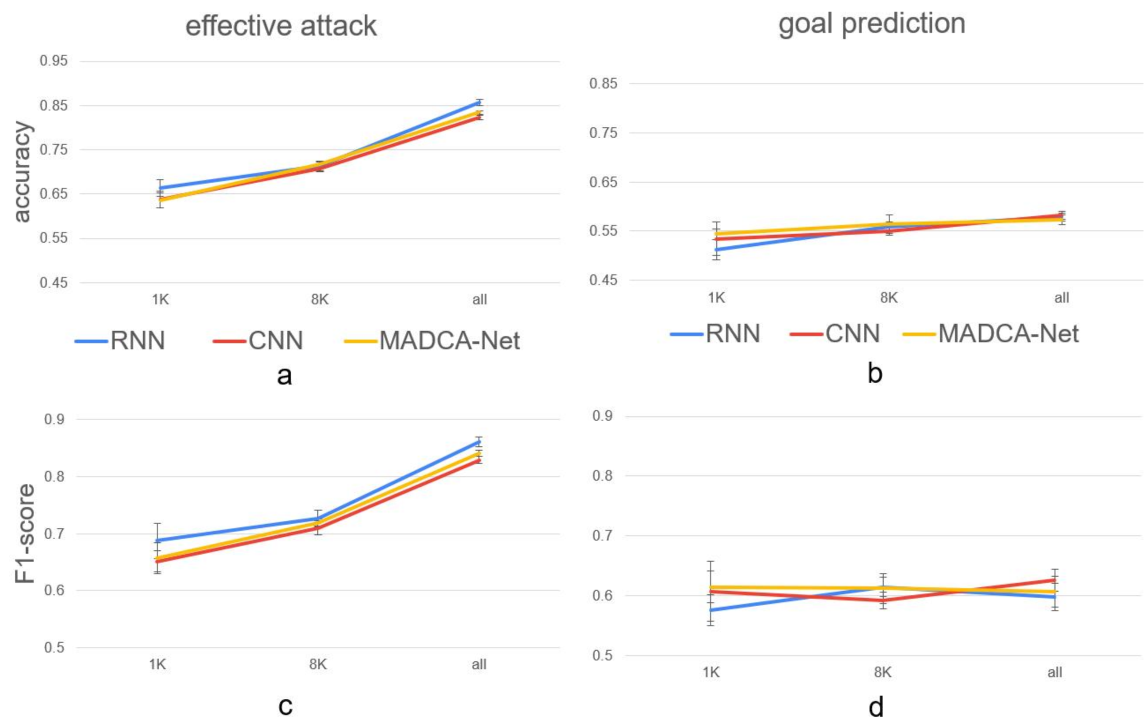

Figure 4.3: **Prediction performances of all models.** *The effective/ineffective attack prediction and goal/non-goal prediction accuracies (a,b) and F1-scores (c,d) are shown.*

successful shots were computed for each zone and the distances for each player. The probability of the player who attempted shots less than 10 times was computed as the probability of the player that is of the same position (i.e., guard, forward, center, guard/forward, and forward/center based on their registration in the NBA 2014/2015 season). It should be noted that certain characteristics of a good shot can differ depending on the court location and context, for example, for 2-point and 3-point shots. Note that, unfortunately, those for only two areas (the 2-point and 3-point areas), with four distance categories, could be accessed. Thus, the shot success probabilities in the restricted, in-the-paint, and mid-range areas were computed using those from the 2-point area.

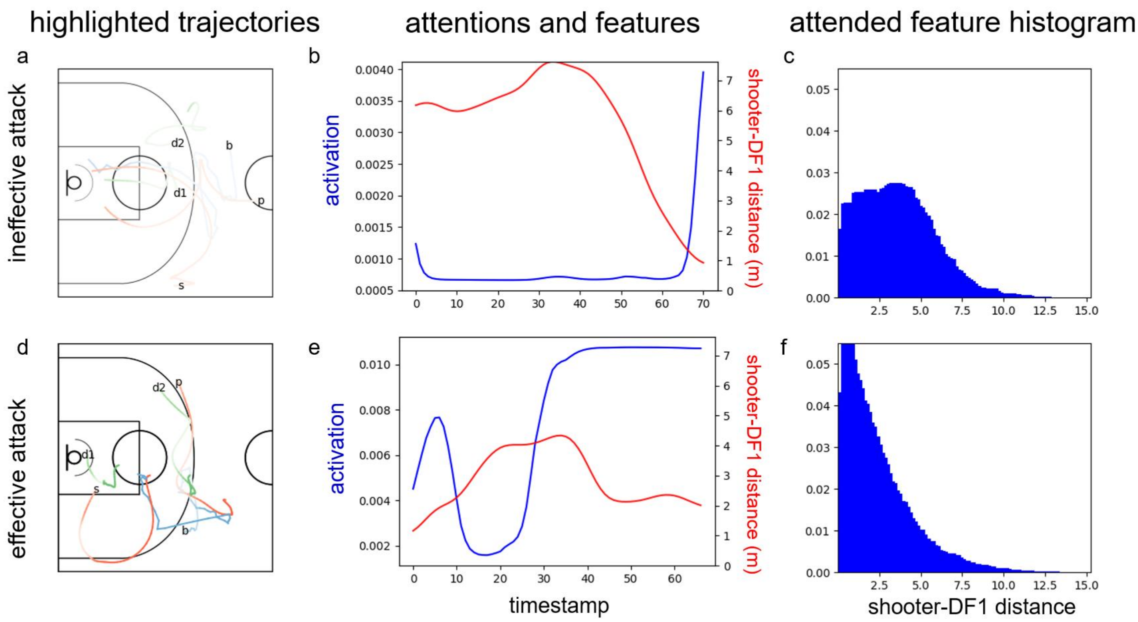

Figure 4.4: **Example results of MADCA in effective and ineffective attacks.** *(a,d) Example of highlighted trajectories on a basketball court, (b, e) Example of attention values and feature sequences, and (c,f) Distinctive attended feature histogram for a test dataset. In the highlighted trajectories, blue, red, and green represent the ball, attacker, and defender when the features are distinctive between the labels (otherwise, they are white). Attention sequences (blue) are presented with a specific feature (red). The attended feature histogram is based on the distinctive features between the labels during the specific (highlighted) interval.*

4.2.4 Training

MADCA-Net was trained on 80% of the trajectories, which were randomly selected, to minimize the binary classification error on the training data, employing backpropagation based on the Adam optimizer. Then, the trained MADCA-Net was tested using the remaining 20% of trajectories to compute the classification accuracy. All models were trained for 50 epochs, and there were 128 neurons in each convolutional/GRU layer. To reduce overfitting, dropout was used with a rate of 0.5.

4.2.5 Analysis

In our analysis, our methods were first validated in terms of classification performance. Then, example visual analyses were shown using our framework. Lastly, analyses of team performances were demonstrated.

To validate our methods in terms of classification performance, we used accuracy and F1-score metrics. When comparing our full model, CNN-RNN (MADCA), to two ablated models, separating 1D CNN and GRU (RNN) models was considered. Our approach was verified by comparing the models with effective/ineffective attack labels and goal/non-goal labels using different dataset sizes (1,024, 8,192, and all 45,307 samples). It is speculated that 1K is the minimum size of the training, and 8K is roughly an intermediate size between the minimum and full sample sizes in a log scale. Note that it is not obvious that more data provides a better result in this dataset and these tasks if a classification task is inherently difficult or some aspect of the model or input features is wrong. To examine these possibilities, we verified our approach using different dataset sizes. With five different random seeds, when splitting the data into training and testing sets, the mean and standard deviation of the classification performances were evaluated.

To understand the meaning of the highlights, as per DeepHL [21], the Pearson correlation coefficients between the attention values of each layer and handcrafted features were computed as shown in Fig. 4.1d middle. Based on the correlation coefficients, the highlighted trajectories (Fig. 4.1d left) were plotted. For feature analysis, the differences between the distributions of each handcrafted feature for two classes within the highlighted segments were computed [21] as follows:

$$\mathrm{diff}(A_{i,C_\mathrm{A}}, F_{j,C_\mathrm{A}}, A_{i,C_\mathrm{B}}, F_{j,C_\mathrm{B}}) = 1 - \mathrm{Intersect}\left(h(m(A_{i,C_\mathrm{A}}, F_{j,C_\mathrm{A}})), h(m(A_{i,C_\mathrm{B}}, F_{j,C_\mathrm{B}}))\right), \tag{4.6}$$

where F_{j,C_A} is a set of time series of the jth handcrafted feature, calculated from trajectories belonging to class A. In addition, $m(\cdot,\cdot)$ is a masking function that extracts feature values within the highlighted segments. Distinctive features were selected based on this value and plotted a histogram to understand the attended (or highlighted) features (Fig. 4.1d right).

Finally, for team analysis, the average number of effective attacks and goals was analyzed to examine the effective attack label as a team evaluation metric. Pearson correlation coefficients (r-value) were computed between statistical results (e.g., actual goals and effective attacks) and the 2015-16 NBA season results (field goal percentage and field goal scores), which were obtained from the official NBA website (nba.com). For season results, it was confirmed that field goal percentage was very highly correlated with the season ranking ($\tau = -0.990$ using Kendall's τ), which suggests the field goal percentage reflects the team winning, whereas the field goal scores reflect more offensive aspects. Since the sample size was small ($N = 30$) in the correlation analysis, the r value was used as an effect size for evaluation rather than the p-value. As described in a previous study [114], correlation coefficients less than 0.20 can be interpreted as *slight, almost negligible relationships*, between 0.20 and 0.40 as *low correlation*; between 0.40 and 0.70 as *moderate correlation*; between 0.70 and 0.90 as *high correlation*, and correlation greater than 0.90 as *very high correlation*.

4.3 Results

The purpose of our experiments was to validate our methods for application to real-world team sports data. To this end, our methods were first validated in terms of classification performance. Then, example visual analyses were presented using our framework. Lastly, quantitative analyses of team performance were shown using our approaches.

4.3.1 Model validation

First, our methods were validated in terms of classification performance. Our approach was verified by comparing two baselines with effective/ineffective attack labels and goal/non-goal labels, using different sizes of the dataset (1,024, 8,192, and all 45,307 samples), as shown in Fig 4.3. First, as the size of the datasets increased in effective/ineffective attack prediction, the prediction performances increased in all models and predictions, indicating that all models would benefit from a greater amount of data. Compared with the goal/non-goal prediction models, effective/ineffective attack prediction models show better performance, which seems reasonable because goal/non-goal prediction is inherently more difficult than effective/ineffective attack prediction. Thus, we basically used the effective/ineffective attack prediction model was basically used with all data for the following analysis. Among the three models, in the effective/ineffective attack prediction, the performance was better in descending order of RNN, CNN-RNN (MADCA-Net), and CNN particularly when sufficient data is available. In the goal/non-goal prediction, the differences among the models were similar. The prediction performance of the RNN was better than that of MADCA-Net and CNN, but to find the distinctive (highlighted) part of trajectories was aimed, which can be modeled by a 1D CNN. Then, MADCA-Net was used for the following analysis, which combined the interpretability of the 1D-CNN model and the predictability of an RNN model.

4.3.2 Example analysis

Next, example visual analyses were shown using our framework. In this subsection, example results of effective/ineffective attacks were presented in Fig 4.4, and then those of goal/no-goal attacks were presented in Fig 4.5.

First, distinctive layers were found by computing a score for each layer by providing a

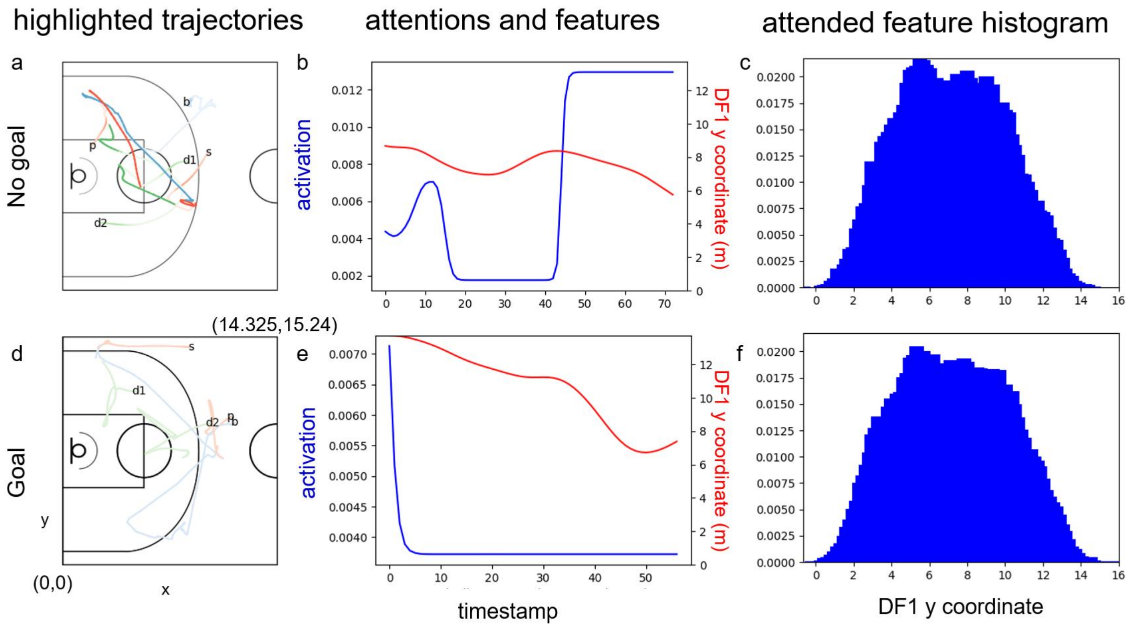

Figure 4.5: **Example results of MADCA in goal/no-goal attacks.** *(a,d) Example highlighted trajectories on a basketball court, (b, e) Example attentions and feature sequences, and (c,f) Distinctive attended feature histograms for a test dataset are shown. Color configurations are the same as those in Fig 4.1.*

ranking of the layers based on the calculated scores. In both effective/ineffective and goal/no-goal attacks, the last layer of the fourth 1D-CNN is the distinctive layer, which means that the longest filter width in the 1D CNN layers was selected.

Next, trajectories colored by the identified distinctive layer were compared. In the example of Fig. 4.4d (colored by a distinctive layer with the highest score), the start and end segments of effective attack trajectories were highlighted in color. On the other hand, in the example of Fig. 4.4a, almost no trajectory segments in an ineffective attack were highlighted. Qualitatively, the latter (Fig. 4.4a) may be a usual attacking play from the top position (and, therefore, not distinctive), while the former (Fig. 4.4d) may be an effective shooter movement for creating a scoring opportunity.

Lastly, we tried to understand the meaning of the highlighted segments. MADCA offers two methods for comprehending the rationale behind the attention drawn to a specific segment through a distinctive layer. First, a correlation is computed between the time series of attention values and each of the pre-computed handcrafted features. In effective/ineffective attacks, the distance between the shooter and DF1 (the shooter defender at the last frame) was selected, which seems reasonable because the shooter-DF1 distance is related to shot performance. In Figs 4.4b and e, activation in attention and the shooter-DF1 distance were negatively correlated, which indicates that MADCA-Net focused on the scoring opportunities with larger shooter-DF1 distances. Second, the difference was provided in the distributions of each handcrafted feature among the two classes within the highlighted segments. In Figs 4.4 c and f, the histograms of the attended feature (in this case, the shooter-DF1 distance) were different between effective and ineffective attacks, which indicates that the attended feature can distinguish between the effective and ineffective attacks.

Beside the feature of the distance between the shooter and DF1, the highest 5 diff values (which was mentioned in Equation 4.6) of features in MADCA and RNN were shown in Fig. 4.6. Since the results for CNN and MADCA were similar, the histogram of features with high diff values in MADCA and RNN were shown in Figs. 4.7 and 4.8 respectively. Through Figs. 4.6, Fig. 4.7 and Fig. 4.8, it is clear that higher diff values indicate more distinct histogram distributions. In Fig. 4.6, the highest diff value in MADCA was significantly higher than RNN. When the attention values of a feature show significant differences across different classes, it indicates that the model is clearly focusing on these features to distinguish between categories. This noticeable distinction helps us understand which features the model relies on to make decisions, thereby enhancing the model's interpretability. And in this study, that means the MADCA model has a better performance on interpretability.

Next, example MADCA results of goal/no-goal attacks were shown in Fig 4.5. In the example of Fig. 4.5a, the start and end segments of effective attack trajectories were highlighted

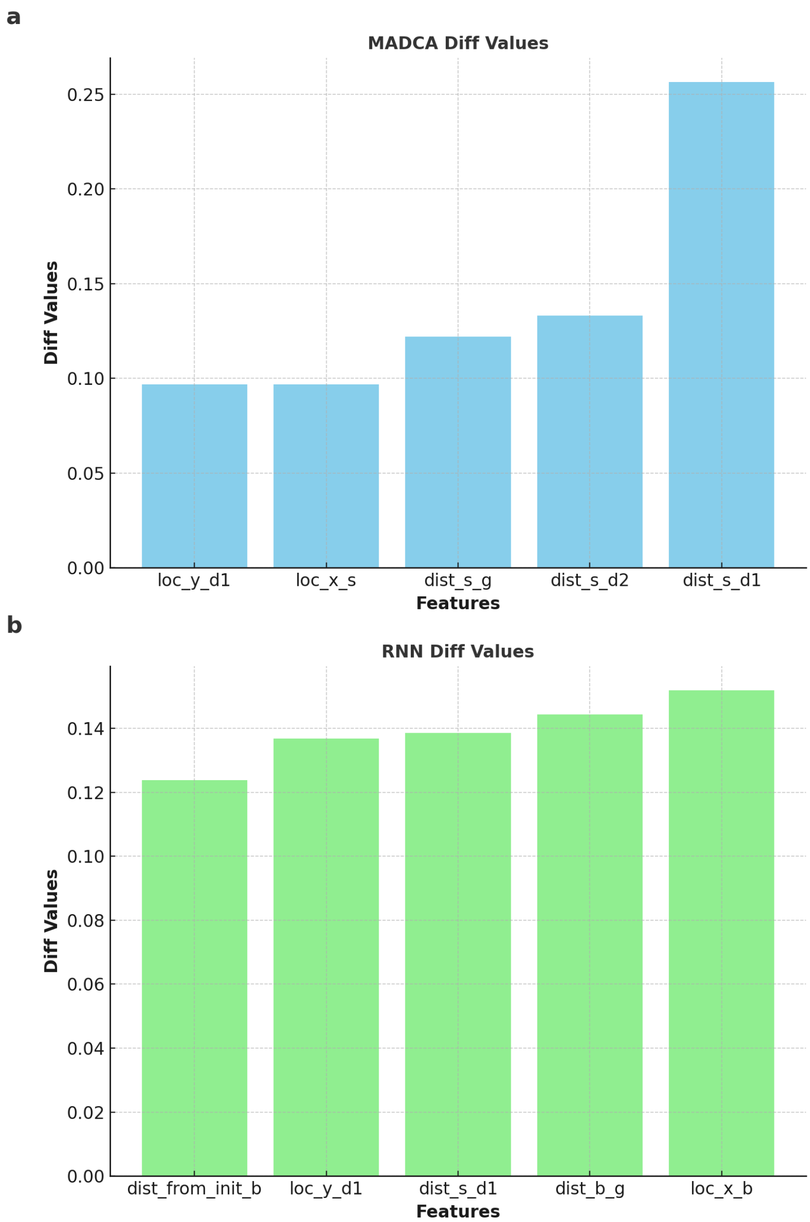

Figure 4.6: **Highest diff values in MADCA and RNN.** *Here I show the highest 5 diff values of features in MADCA and RNN respectively.In this figure, d1 means DF1. d2 means DF2. s means shooter. g means ring. b means ball. loc_y means y coordinate. loc_x means x coordinate. Dist means the distance between the following 2 agents. dist_from_init means the distance between the following agent and its initial position.*

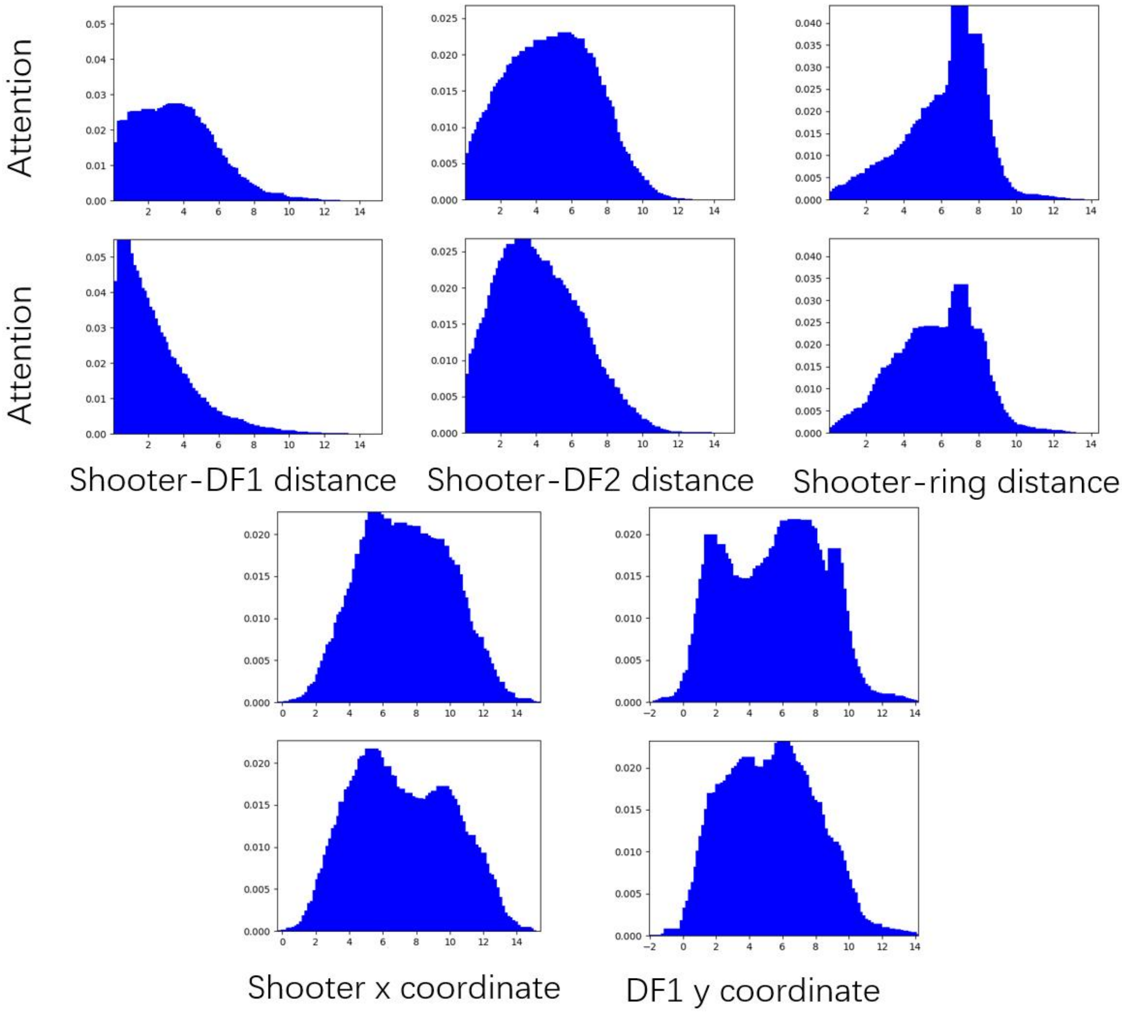

Figure 4.7: **Histograms of top-5 features in MADCA.** *Here I show the histograms of features with high diff values in MADCA for a test dataset.*

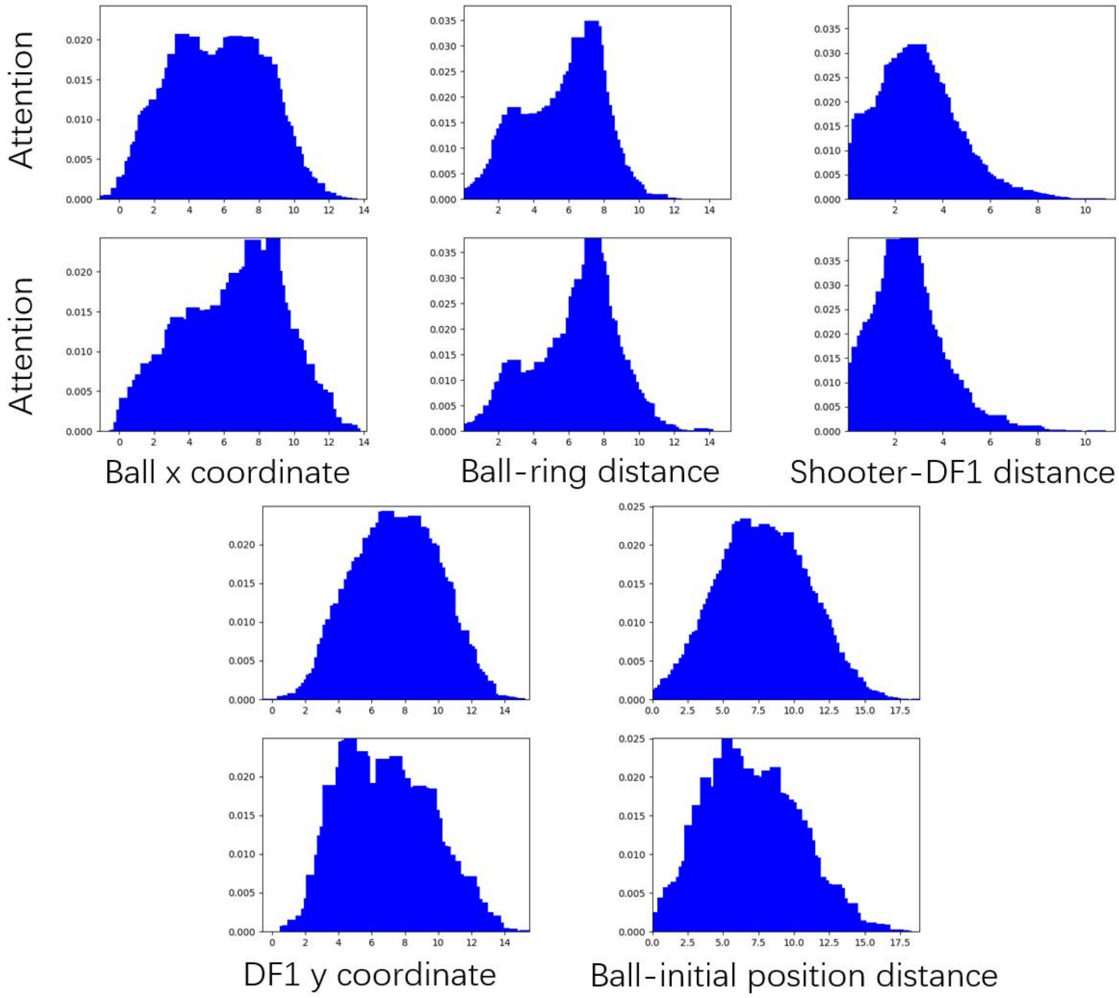

Figure 4.8: **Histograms of top-5 features in RNN.** *Here I show the histograms of features with high diff values in RNN for a test dataset.*

in color. On the other hand, in the example of Fig. 4.5d, almost no trajectory segments were highlighted in a goal play. Qualitatively, this may not provide useful information because we want to know the highlighted trajectories (i.e., movements) in goal plays rather than no-goal plays. As a distinctive feature, in goal/no-goal attacks, the DF1 y-coordinate was selected, which seems somewhat reasonable but not essential information about multi-agent interaction because this feature indicates that a shooter defender retreated toward the end-line of the court, but there is no information about the shooter. In fact, there was almost no correlation between activation in attention and the DF1 y-coordinates in Figs 4.5 b and e. In Figs 4.5 c and f, the histograms of the attended feature (in this case, DF1's y-coordinate) were almost the same in goal and no-goal plays, which indicates that it would be difficult to distinguish between the two types of plays using the attended feature. Note that, in the above two cases, the information of all features (e.g., passer and DF2) were considered in this analysis. According to the procedure in Section 4.2.5, we selected and showed the distinctive features.

4.3.3 Team analysis

Lastly, quantitative analyses of team performance are shown to examine the effective attack metric. Table 4.1 shows the 2015-2016 rankings and season field goal percentages and points, as well as statistical results (e.g., actual scores and effective attacks) for each team. The season performance (1,230 games) was estimated using tracking data from a subset of the season's games (600 games).

To gauge the importance of these metrics, using the results from Table 4.1, Pearson correlation coefficients were computed between the statistical results (actual goals and effective attacks) and 2015-16 NBA season results (field goal (FG) percentage and points). Note that the Golden State Warriors was excluded, who had a record win-to-loss ratio (73 wins and 9 losses) at the time, from the analysis because the Warriors had, by far, the highest season FG

Table 4.1: Rankings and statistics of teams in the 2015-2016 NBA season, including season field goal (FG) percentages and points, as well as statistical results (mean actual goals and effective attacks from our data).

Team Name	Rank	Season FG %	Season FG pts	Mean goal	Mean eff. att.
Golden State Warriors	1	48.7	8055	0.413	0.478
San Antonio Spurs	2	48.4	7148	0.443	0.512
Oklahoma City Thunder	3	47.6	7422	0.417	0.463
Miami Heat	4	47.0	6798	0.416	0.493
Milwaukee Bucks	5	46.7	6730	0.400	0.501
Los Angeles Clippers	6	46.5	7079	0.395	0.468
Minnesota Timberwolves	7	46.4	6645	0.398	0.466
Sacramento Kings	8	46.4	7226	0.390	0.532
Washington Wizards	9	46.0	7185	0.383	0.505
Cleveland Cavaliers	10	46.0	7222	0.398	0.490
Atlanta Hawks	11	45.8	7151	0.398	0.588
Orlando Magic	12	45.5	7120	0.396	0.522
Brooklyn Nets	13	45.3	6803	0.433	0.466
Houston Rockets	14	45.2	7066	0.387	0.517
Toronto Raptors	15	45.1	6720	0.403	0.483
Portland Trail Blazers	16	45.0	7198	0.403	0.453
Indiana Pacers	17	45.0	6947	0.386	0.460
Utah Jazz	18	44.9	6608	0.389	0.496
New Orleans Pelicans	19	44.8	7008	0.408	0.533
Dallas Mavericks	20	44.4	6934	0.387	0.506
Denver Nuggets	21	44.2	6842	0.399	0.505
Chicago Bulls	22	44.1	6981	0.387	0.460
Memphis Grizzlies	23	44.0	6542	0.386	0.517
Boston Celtics	24	43.9	7149	0.387	0.491
Detroit Pistons	25	43.9	6962	0.369	0.477
New York Knicks	26	43.9	6654	0.422	0.388
Charlotte Hornets	27	43.9	6945	0.414	0.501
Phoenix Suns	28	43.5	6840	0.398	0.505
Philadelphia 76ers	29	43.1	6704	0.364	0.502
Los Angeles Lakers	30	41.4	6399	0.359	0.407

Table 4.2: **Pearson's r-values with each of the quantitative metrics from our data and team performance in the 2015-2016 season (excluding the Warriors).**

Quantitative Metric	Season FG (%)	Season FG points
Mean goals	0.595	0.296
Mean effective attacks	0.237	0.349

points (8,055) in the league (the second highest was Oklahoma City Thunder with 7,422). From Table 4.2, the results show that mean actual goals had moderate and low positive relationships with season FG percentage ($r = 0.595$) and points ($r = 0.296$), respectively. Mean effective attacks, on the other hand, had low positive correlations with season FG percentage ($r = 0.237$) and points ($r = 0.349$), respectively. From these results, the mean FG goal from our data can estimate season FG percentage, which seems reasonable given it is the same data, and the mean effective attacks can estimate season FG points better than the mean FG goals from our data. The results may be related to the effective attack considering the shot area (including 2- and 3-points). Note that, from the correlation results, it is difficult to directly examine the effectiveness of an effective attack because there is no ground truth of an effective attack without considering the scoring results. In other words, the correlation is examined with scoring results, but a higher correlation does not mean higher reliability. These results are discussed in the next section.

4.4 Summary

In this study, a comparative analysis method called MADCA was proposed, which analyzes multi-agent trajectories in ball games. The proposed method utilized a deep neural network which was combined with CNN layers and GRU layers, and this network was trained by multi-agent features that were extracted from the dataset. Moreover, an attention mechanism was applied in the deep neural network. To evaluate the performance of each attack, the concept of effective attack was adopted, and its rationality as an evaluation criterion was demonstrated by analyzing NBA game data. In addition, the proposed is compared with various ablation models with effective/ineffective attack labels and goal/non-goal labels. Furthermore, the proposed method can extract distinctive layers in MADCA-Net and features in each prediction task to highlight the segmented trajectories and show which variables were correlated with the labels. This could provide insight into the relationship between attack efficiency and certain attack characteristics.

This approach can extract distinctive layers in MADCA-Net and features in the effective/ineffective attack prediction task, as shown in Fig 4.4. However, specifically, for goal/non-goal prediction, when the trajectories of ball sports are dealt with, note that all trajectories may not have the characteristics of a specific class. For example, the shooting skills of a shooter and the randomness of the successful shot affect the goal/non-goal prediction performance. In addition, it can be speculated that in top-level teams (e.g., Warriors), players can score even in ineffective situations because of their superior shooting skills. Although it is inherently difficult to validate the effective/ineffective attack metric as mentioned above, we believe it would be more plausible than the metric based on goal/non-goal prediction from these prediction performances.

5 Conclusion

5.1 Summary of the Thesis

This study proposed methods to classify various team plays and analyze multi-agent trajectories in team sports. This thesis addressed the following problems. The first is how to utilize large amounts of unlabeled data in cooperative play classification. The second is how to understand the differences between different trajectories of team play and the correlation between variables and labels of team play.

Chapter 2 introduced the literature review of previous related works in three aspects. Firstly, I introduced the traditional classification and analysis methods in team sports, including inverse approaches and forward approaches, both relying on researchers' experience and established theories, easy to interpret but hard to model. Secondly, I introduced machine learning methods, including unsupervised learning, supervised learning, self-supervised learning, semi-supervised learning, and reinforcement learning, easy to model but hard to interpret. Finally, interpretability in machine learning for team sports was introduced.

In Chapter 3, to classify team plays in basketball utilizing large amounts of unlabeled data, a semi-supervised learning framework was adopted (Study I). In this study, large amounts of unlabeled data were labeled by prediction models, and various types of screen-plays were classified in a completely automatic way. To provide insights into features used in classifiers, SHAP [19] was also applied in this method to show the relationship between each feature and prediction results.

In Chapter 4, to analyze multi-agent trajectories of team play in team sports, a multi-agent deep learning-based comparative analysis method in basketball was proposed (Study II). This study provided a way to understand the differences between given classes by highlighting segmented trajectories using the attention mechanism-based neural network and correlation between variables and labels.

Overall, these two studies of this thesis all aim to discover insights of team sports. Even though the two studies focus on different directions, they can both provide some more unique insights for coaches' tactical guidance of team sports.

5.2 Future Work

Although our proposed approaches in this thesis could provide some insights in cooperative play classification and analysis without much labor cost for labeling, a number of challenges still remain.

Limited number of considered agents. In this thesis, including Study I and II, only a part of the agents were considered to improve interpretability and avoid the role assignment problem. In more general cases, all agents (in basketball, 11 agents including a ball) should be considered to provide a more accurate insight for team sports in future work. For example, a graph neural network [69, 108, 109] could be applied to improve predictability and a Gaussian mixture model [106, 65] to address the role assignment problem in order to improve the interpretability.

Limitations of a single kind of team sport. The two studies mentioned in this thesis both focus on basketball and primarily use the same dataset from the NBA. Therefore, whether the methods proposed in this thesis can be effectively applied to basketball games at other levels or even to other types of ball sports remains unknown. Moreover, the features that need to be extracted from the trajectories may vary for different ball sports. Accordingly, as future work, it could be considered to test the methods on basketball leagues at different levels and to incorporate expertise from other ball sports into the proposed methods to expand their applicability.

Introduce other machine learning methods. In future work, introducing other machine learning methods into the study of this thesis could be considered, which may help us further improve the performance. For example, some studies in graph neural networks [69, 70, 71, 72] can improve representation ability in team sports trajectory. Transformer-based approach [67] can also be expected to learn better trajectory representation in basketball games to evaluate players and teams. However, more complicated models will often decrease interpretability. Thus, the trade-off between predictability and interpretability should be considered. As another approach to directly evaluate the actions of players, a deep reinforcement learning-based approach for basketball games [78] can be considered.

Practical application. The practical application of studies in this thesis should also be considered in future work. In the study I's framework, to a given basketball game's trajectory, the framework can classify which kind of screen play appeared among it in an automatic way after the games. Due to this functionality, this framework could be applied in automatic annotation in team play for basketball games instead of manual annotation. It not only reduces the labor cost but also provides the annotation faster.

In the study II's framework, with a large amount of training data, the MADCA model is trained, and then to the given basketball game's trajectory data, which the users are interested in, MADCA can classify effective/ineffective attacks and highlight distinct subtrajecory for effective attack. It could be useful for players and coaches to analyze players' performance in attacks.

Acknowledgement

First and foremost, my heartfelt thanks go to Prof. Kazuya Takeda at Nagoya University for his steadfast supervision and motivation during my doctoral journey. His invaluable advice in my research endeavors and prompt support during crucial times have been instrumental. This work could not have been accomplished without the guidance of Prof. Takeda.

I am also immensely grateful to Prof. Fujii Keisuke of Nagoya University, whose suggestions, support, and patience have been the cornerstone of my doctorate life. His expertise and insight have been pivotal in shaping my research and academic pursuits. His mentorship extended beyond academic instruction but also to students' physical and psychological health. I am immensely thankful for the countless hours he has dedicated to discussing research, reviewing the progress, and providing constructive feedback. The doctorate journey would not have been the same wonderful without him, and I am deeply appreciative of the patience he has had on these days.

I am deeply grateful for the opportunity to be a part of this lab and to work under the exceptional guidance of Prof. Takeda and Prof. Fujii. Their relentless support and mentorship have been fundamental to the success of my research. And besides my advisors, I also would like to my heartfelt thanks to Prof. Tomoki Toda of Nagoya University for his advice and suggestions throughout my doctoral course. I would also like to express my gratitude to Associate Prof. Ming Ding for his kind support and suggestions. Additionally, I extend my sincere appreciation to Assistant Prof. Kazushi Tsutsui of Nagoya University for his kind support and advice.

Special thanks go to Mr. Rory Bunker, Mr. Calvin Yeung, as well as other members of Takeda Lab. This acknowledgment would be incomplete without mentioning the supportive and collaborative atmosphere fostered by each one of you.

Last but not least, I would like to express my love and gratitude to my family for their unconditional love, understanding, and support over these years.

References

[1] T. L. Dickinson and R. M. McIntyre, "A conceptual framework for teamwork measurement," Team performance assessment and measurement, pp. 19–43, 1997.

[2] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone, "Evidence for a collective intelligence factor in the performance of human groups," Science, vol. 330, no. 6004, pp. 686–688, 2010.

[3] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tâche, I. Saïd, V. Durier, S. Canonge, J. M. Amé et al., "Social integration of robots into groups of cockroaches to control self-organized choices," Science, vol. 318, no. 5853, pp. 1155–1158, 2007.

[4] J. Werfel, K. Petersen, and R. Nagpal, "Designing collective behavior in a termite-inspired robot construction team," Science, vol. 343, no. 6172, pp. 754–758, 2014.

[5] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," Physical Review E, vol. 51, no. 5, p. 4282, 1995.

[6] K. Yokoyama, H. Shima, K. Fujii, N. Tabuchi, and Y. Yamamoto, "Social forces for team coordination in ball possession game," Physical Review E, vol. 97, no. 2, p. 022410, 2018.

[7] W. Spearman, A. Basye, G. Dick, R. Hotovy, and P. Pop, "Physics-based modeling of pass probabilities in soccer," in Proceeding of the MIT Sloan Sports Analytics Conference, 2017.

[8] F. P. Alguacil, P. P. n. Arce, D. Sumpter, and J. Fernandez, "Seeing in to the future: using self-propelled particle models to aid player decision-making in soccer," in Proceedings of the MIT Sloan Sports Analytics Conference, 2020.

[9] K. Fujii, T. Kawasaki, Y. Inaba, and Y. Kawahara, "Prediction and classification in equation-free collective motion dynamics," PLoS Computational Biology, vol. 14, no. 11, p. e1006545, 2018.

[10] K. Fujii, N. Takeishi, B. Kibushi, M. Kouzaki, and Y. Kawahara, "Data-driven spectral analysis for coordinative structures in periodic human locomotion," Scientific Reports, vol. 9, no. 1, pp. 1–14, 2019.

[11] K. Fujii, "Data-driven analysis for understanding team sports behaviors," Journal of Robotics and Mechatronics, vol. 33, no. 3, pp. 505–514, 2021.

[12] A. McQueen, J. Wiens, and J. Guttag, "Automatically recognizing on-ball screens," in Proceedings of the MIT Sloan Sports Analytics Conference, 2014.

[13] A. McIntyre, J. Brooks, J. Guttag, and J. Wiens, "Recognizing and analyzing ball screen defense in the NBA," in Proceedings of the MIT Sloan Sports Analytics Conference, 2016, pp. 11–12.

[14] K. Fujii, K. Yokoyama, T. Koyama, A. Rikukawa, H. Yamada, and Y. Yamamoto, "Resilient help to switch and overlap hierarchical subsystems in a small human group," Scientific Reports, vol. 6, 2016.

[15] M. Hojo, K. Fujii, Y. Inaba, Y. Motoyasu, and Y. Kawahara, “Automatically recognizing strategic cooperative behaviors in various situations of a team sport,” PLoS One, vol. 13, no. 12, p. e0209247, 2018.

[16] K. Fujii, N. Takeishi, M. Hojo, Y. Inaba, and Y. Kawahara, “Physically-interpretable classification of network dynamics for complex collective motions,” Scientific Reports, vol. 10, no. 3005, 2020.

[17] X. J. Zhu, Semi-supervised learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, 2005.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?" explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[19] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems 30, 2017, pp. 4765–4774.

[20] D. Alvarez Melis and T. Jaakkola, “Towards robust interpretability with self-explaining neural networks,” Advances in neural information processing systems, vol. 31, 2018.

[21] T. Maekawa, K. Ohara, Y. Zhang, M. Fukutomi, S. Matsumoto, K. Matsumura, H. Shidara, S. J. Yamazaki, R. Fujisawa, K. Ide et al., “Deep learning-assisted comparative analysis of animal trajectories with deephl,” Nature Communications, vol. 11, no. 1, pp. 1–15, 2020.

[22] J. Bourbousson, C. Sève, and T. McGarry, “Space–time coordination dynamics in basketball: Part 2. the interaction between the two teams,” Journal of Sports Sciences, vol. 28, no. 3, pp. 349–358, 2010.

[23] B. Travassos, D. Araújo, R. Duarte, and T. McGarry, "Spatiotemporal coordination behaviors in futsal (indoor football) are guided by informational game constraints," Human Movement Science, vol. 31, no. 4, pp. 932–945, 2012.

[24] J. Sampaio, T. McGarry, J. Calleja-González, S. Jiménez Sáiz, X. Schelling i del Alcázar, and M. Balciunas, "Exploring game performance in the national basketball association using player tracking data," PLoS One, vol. 10, no. 7, p. e0132894, 2015.

[25] K. Goldsberry, "Courtvision: New visual and spatial analytics for the nba," in 2012 MIT Sloan sports analytics conference, vol. 9, 2012, pp. 12–15.

[26] V. Correia, D. Araujo, C. Craig, and P. Passos, "Prospective information for pass decisional behavior in rugby union," Human Movement Science, vol. 30, no. 5, pp. 984–997, 2011.

[27] L. Vilar, D. Araújo, K. Davids, V. Correia, and P. T. Esteves, "Spatial-temporal constraints on decision-making during shooting performance in the team sport of futsal," Journal of Sports Sciences, vol. 31, no. 8, pp. 840–846, 2013.

[28] K. Fujii, Y. Yoshihara, Y. Matsumoto, K. Tose, H. Takeuchi, M. Isobe, H. Mizuta, D. Maniwa, T. Okamura, T. Murai et al., "Cognition and interpersonal coordination of patients with schizophrenia who have sports habits," PLoS One, vol. 15, no. 11, p. e0241863, 2020.

[29] K. Fujii, D. Yamashita, S. Yoshioka, T. Isaka, and M. Kouzaki, "Strategies for defending a dribbler: categorisation of three defensive patterns in 1-on-1 basketball," Sports Biomechanics, vol. 13, no. 3, pp. 204–214, 2014.

[30] K. Fujii, T. Isaka, M. Kouzaki, and Y. Yamamoto, “Mutual and asynchronous anticipation and action in sports as globally competitive and locally coordinative dynamics,” Scientific Reports, vol. 5, 2015.

[31] P. Power, J. Hobbs, H. Ruiz, X. Wei, and P. Lucey, “Mythbusting set-pieces in soccer,” Proceedings of the MIT Sloan Sports Analytics Conference, 2018.

[32] T. Taki and J.-I. Hasegawa, “Visualization of dominant region in team games and its application to teamwork analysis,” in Proceedings Computer Graphics International, 2000, pp. 227–235.

[33] Y. Yamamoto and K. Yokoyama, “Common and unique network dynamics in football games,” PLoS One, vol. 6, no. 12, p. e29638, 2011.

[34] K. Yokoyama and Y. Yamamoto, “Three people can synchronize as coupled oscillators during sports activities,” PLoS Comput Biol, vol. 7, no. 10, p. e1002181, 2011.

[35] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” in Proceedings of the 14th annual Conference on Computer Graphics and Interactive Techniques, 1987, pp. 25–34.

[36] I. Aoki, “A simulation study on the schooling mechanism in fish,” Bulletin of the Japanese Society of Scientific Fisheries, vol. 48, no. 8, pp. 1081–1088, 1982.

[37] J. Hobbs, P. Power, and L. Sha, “Quantifying the value of transitions in soccer via spatiotemporal trajectory clustering,” in Proceedings of the MIT Sloan Sports Analytics Conference, 2018.

[38] J. Hobbs, M. Holbrook, N. Frank, L. Sha, and P. Lucey, “Improved structural discovery and representation learning of multi-agent data,” arXiv preprint arXiv:1912.13107, 2019.

[39] T. Decroos, J. Van Haaren, and J. Davis, “Automatic discovery of tactics in spatio-temporal soccer match data,” in ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 223–232.

[40] L. Sha, P. Lucey, Y. Yue, P. Carr, C. Rohlf, and I. Matthews, “Chalkboarding: A new spatiotemporal query paradigm for sports play retrieval,” in International Conference on Intelligent User Interfaces, 2016, pp. 336–347.

[41] S. Kanda, K. Takeuchi, K. Fujii, and Y. Tabei, “Succinct trit-array trie for scalable trajectory similarity search,” in Proceedings of the 28th International Conference on Advances in Geographic Information Systems, 2020, pp. 518–529.

[42] M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, and J. Perš, “A trajectory-based analysis of coordinated team activity in a basketball game,” Computer Vision and Image Understanding, vol. 113, no. 5, pp. 612–621, 2009.

[43] H. Alt and M. Godau, “Computing the Fréchet distance between two polygonal curves,” International Journal of Computational Geometry & Applications, vol. 5, no. 01n02, pp. 75–91, 1995.

[44] D. J. Berndt and J. Clifford, “Using Dynamic Time Warping to Find Patterns in Time Series,” in Proceedings of AAAI workshop, 1994.

[45] K.-C. Wang and R. Zemel, “Classifying nba offensive plays using neural networks,” in Proceedings of the MIT Sloan Sports Analytics Conference, 2016.

[46] A. Nistala, “Using deep learning to understand patterns of player movement in basketball,” Ph.D. dissertation, Massachusetts Institute of Technology, 2018.

[47] A. Grunz, D. Memmert, and J. Perl, “Tactical pattern recognition in soccer games by means of special self-organizing maps,” Human Movement Science, vol. 31, no. 2, pp. 334–343, 2012.

[48] M. Kempe, A. Grunz, and D. Memmert, “Detecting tactical patterns in basketball: comparison of merge self-organising maps and dynamic controlled neural networks,” European Journal of Sport Science, vol. 15, no. 4, pp. 249–255, 2015.

[49] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, “Spectral analysis of nonlinear flows,” Journal of Fluid Mechanics, vol. 641, pp. 115–127, 2009.

[50] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” Journal of Fluid Mechanics, vol. 656, pp. 5–28, 2010.

[51] K. Fujii, Y. Inaba, and Y. Kawahara, “Koopman spectral kernels for comparing complex dynamics: Application to multiagent sport plays,” in European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD’17). Springer, 2017, pp. 127–139.

[52] D. Cervone, A. D’Amour, L. Bornn, and K. Goldsberry, “Pointwise: Predicting points and valuing decisions in real time with nba optical tracking data,” in Proceedings of the MIT Sloan Sports Analytics Conference, 2014.

[53] Y.-H. Chang, R. Maheswaran, S. J. J. Kwok, T. Levy, A. Wexler, and K. Squire, “Quantifying shot quality in the nba,” in Proceedings of the MIT Sloan Sports Analytics Conference, 2014.

[54] P. Lucey, A. Bialkowski, M. Monfort, P. Carr, and I. Matthews, “quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data,” in Proceedings of the MIT Sloan Sports Analytics Conference, 2014.

[55] P. Lucey, D. Oliver, P. Carr, J. Roth, and I. Matthews, "Assessing team strategy using spatiotemporal data," in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 1366–1374.

[56] N. Mehrasa, Y. Zhong, F. Tung, L. Bornn, and G. Mori, "Deep learning of player trajectory representations for team activity analysis," in Proceedings of the MIT Sloan Sports Analytics Conference, 2018.

[57] A. Sicilia, K. Pelechrinis, and K. Goldsberry, "Deephoops: Evaluating micro-actions in basketball using deep feature representations of spatio-temporal data," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2096–2104.

[58] Y. Yue, P. Lucey, P. Carr, A. Bialkowski, and I. Matthews, "Learning fine-grained spatial models for dynamic sports play prediction," in 2014 IEEE International conference on Data Mining. IEEE, 2014, pp. 670–679.

[59] S. Zheng, R. Yu, and Y. Yue, "Multi-resolution tensor learning for large-scale spatial data," arXiv preprint arXiv:1802.06825, 2018.

[60] J. Mortensen and L. Bornn, "From markov models to poisson point processes: Modeling movement in the nba," in Proceedings of the 2019 MIT Sloan Sports Analytics Conference, 2019.

[61] R. Bunker, V. N. Le Duy, Y. Tabei, I. Takeuchi, and K. Fujii, "Multi-agent statistical discriminative sub-trajectory mining and an application to nba basketball," arXiv preprint arXiv:2311.16564, 2023.

[62] S. Zheng, Y. Yue, and J. Hobbs, "Generating long-term trajectories using deep hierarchical networks," in Advances in Neural Information Processing Systems 29, 2016, pp. 1543–1551.

[63] S. Hauri, N. Djuric, V. Radosavljevic, and S. Vucetic, "Multi-modal trajectory prediction of nba players," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1640–1649.

[64] T. Seidl, A. Cherukumudi, A. Hartnett, P. Carr, and P. Lucey, "Bhostgusters: Realtime interactive play sketching with synthesized nba defenses," in Proceedings of the MIT Sloan Sports Analytics Conference, 2018.

[65] K. Fujii, N. Takeishi, Y. Kawahara, and K. Takeda, "Policy learning with partial observation and mechanical constraints for multi-person modeling," arXiv preprint arXiv:2007.03155, 2020.

[66] E. Zhan, S. Zheng, Y. Yue, L. Sha, and P. Lucey, "Generating multi-agent trajectories using programmatic weak supervision," in International Conference on Learning Representations, 2019.

[67] M. A. Alcorn and A. Nguyen, "baller2vec: A multi-entity transformer for multi-agent spatiotemporal modeling," 2021.

[68] C.-Y. Chen, W. Lai, H.-Y. Hsieh, W.-H. Zheng, Y.-S. Wang, and J.-H. Chuang, "Generating defensive plays in basketball games," in Proceedings of the 26th ACM International Conference on Multimedia, 2018, pp. 1580–1588.

[69] R. A. Yeh, A. G. Schwing, J. Huang, and K. Murphy, "Diverse generation for multi-agent sports games," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4610–4619.

[70] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, “Neural relational inference for interacting systems,” in International Conference on Machine Learning, 2018, pp. 2688–2697.

[71] C. Sun, P. Karlsson, J. Wu, J. B. Tenenbaum, and K. Murphy, “Predicting the present and future states of multi-agent systems from partially-observed visual data,” in International Conference on Learning Representations, 2019.

[72] C. Graber and A. G. Schwing, “Dynamic neural relational inference,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[73] D. Fassmeyer, P. Fassmeyer, and U. Brefeld, “Semi-supervised generative models for multiagent trajectories,” Advances in Neural Information Processing Systems, vol. 35, pp. 37 267–37 281, 2022.

[74] G. Anzer, P. Bauer, U. Brefeld, and D. Faßmeyer, “Detection of tactical patterns using semi-supervised graph neural networks,” in MIT Sloan Sports Analytics Conference, vol. 16, 2022, pp. 1–3.

[75] S. Hauri and S. Vucetic, “Group activity recognition in basketball tracking data–neural embeddings in team sports (nets),” in ECAI 2023. IOS Press, 2023, pp. 1012–1019.

[76] D. Cervone, A. D ’ Amour, L. Bornn, and K. Goldsberry, “A multiresolution stochastic process model for predicting basketball possession outcomes,” Journal of the American Statistical Association, vol. 111, no. 514, pp. 585–599, 2016.

[77] J. Wang, I. Fox, J. Skaza, N. Linck, S. Singh, and J. Wiens, “The advantage of doubling: a deep reinforcement learning approach to studying the double team in the nba,” arXiv preprint arXiv:1803.02940, 2018.

[78] C. Yanai, A. Solomon, G. Katz, B. Shapira, and L. Rokach, "Q-ball: Modeling basketball games using deep reinforcement learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 8, 2022, pp. 8806–8813.

[79] G. Liu and O. Schulte, "Deep reinforcement learning in ice hockey for context-aware player evaluation," in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3442–3448.

[80] G. Liu, Y. Luo, O. Schulte, and T. Kharrat, "Deep soccer analytics: learning an action-value function for evaluating soccer players," Data Mining and Knowledge Discovery, vol. 34, no. 5, pp. 1531–1559, 2020.

[81] O. Schulte, M. Khademi, S. Gholami, Z. Zhao, M. Javan, and P. Desaulniers, "A markov game model for valuing actions, locations, and team performance in ice hockey," Data Mining and Knowledge Discovery, vol. 31, no. 6, pp. 1735–1757, 2017.

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

[83] H. Nakahara, K. Tsutsui, K. Takeda, and K. Fujii, "Action valuation of on-and off-ball soccer players based on multi-agent deep reinforcement learning," IEEE Access, vol. 11, pp. 131 237–131 244, 2019.

[84] Y. Luo, O. Schulte, and P. Poupart, "Inverse reinforcement learning for team sports: Valuing actions and players," in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 3356–3363.

[85] P. Rahimian and L. Toka, “Inferring the strategy of offensive and defensive play in soccer with inverse reinforcement learning,” in Machine Learning and Data Mining for Sports Analytics: 8th International Workshop, MLSA 2021, Virtual Event, September 13, 2021, Revised Selected Papers. Springer, 2022, pp. 26–38.

[86] H. Tang, J. Hao, T. Lv, Y. Chen, Z. Zhang, H. Jia, C. Ren, Y. Zheng, Z. Meng, C. Fan et al., “Hierarchical deep multiagent reinforcement learning with temporal abstraction,” arXiv preprint arXiv:1809.09332, 2018.

[87] K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet et al., “Google research football: A novel reinforcement learning environment,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 4501–4510.

[88] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, “Robocup: The robot world cup initiative,” in Proceedings of the First International Conference on Autonomous Agents, 1997, pp. 340–347.

[89] K. Fujii, K. Tsutsui, A. Scott, H. Nakahara, N. Takeishi, and Y. Kawahara, “Adaptive action supervision in reinforcement learning from real-world multi-agent demonstrations,” in 14th International Conference on Agents and Artificial Intelligence (ICAART’ 24), 2024.

[90] P.-F. Pai, L.-H. ChangLiao, and K.-P. Lin, “Analyzing basketball games by a support vector machines with decision tree model,” Neural Computing and Applications, vol. 28, pp. 4159–4167, 2017.

[91] S. Morgan, M. D. Williams, and C. Barnes, "Applying decision tree induction for identification of important attributes in one-versus-one player interactions: A hockey exemplar," Journal of Sports Sciences, vol. 31, no. 10, pp. 1031–1037, 2013.

[92] J. Marynowicz, M. Lango, D. Horna, K. Kikut, and M. Andrzejewski, "Predicting ratings of perceived exertion in youth soccer using decision tree models," Biology of Sport, vol. 39, no. 2, pp. 245–252, 2022.

[93] S. Moustakidis, S. Plakias, C. Kokkotis, T. Tsatalas, and D. Tsaopoulos, "Predicting football team performance with explainable ai: Leveraging shap to identify key team-level performance metrics," Future Internet, vol. 15, no. 5, p. 174, 2023.

[94] K. Toda, M. Teranishi, K. Kushiro, and K. Fujii, "Evaluation of soccer team defense based on prediction models of ball recovery and being attacked: A pilot study," Plos one, vol. 17, no. 1, p. e0263051, 2022.

[95] A. Lalwani, A. Saraiya, A. Singh, A. Jain, and T. Dash, "Machine learning in sports: A case study on using explainable models for predicting outcomes of volleyball matches," arXiv preprint arXiv:2206.09258, 2022.

[96] B. C. Luu, A. L. Wright, H. S. Haeberle, J. M. Karnuta, M. S. Schickendantz, E. C. Makhni, B. U. Nwachukwu, R. J. Williams III, and P. N. Ramkumar, "Machine learning outperforms logistic regression analysis to predict next-season nhl player injury: an analysis of 2322 players from 2007 to 2017," Orthopaedic journal of sports medicine, vol. 8, no. 9, p. 2325967120953404, 2020.

[97] Y. Wang, W. Liu, and X. Liu, "Explainable ai techniques with application to nba gameplay prediction," Neurocomputing, vol. 483, pp. 59–71, 2022.

[98] K. Fujii, N. Takeishi, K. Tsutsui, E. Fujioka, N. Nishiumi, R. Tanaka, M. Fukushiro, K. Ide, H. Kohno, K. Yoda, S. Takahashi, S. Hiryu, and Y. Kawahara, "Learning interaction rules from multi-animal trajectories via augmented behavioral models," in Advances in Neural Information Processing Systems, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 11 108–11 122. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/5c572eca050594c7bc3c36e7e8ab9550-Paper.pdf

[99] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[100] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," Machine Learning, vol. 109, no. 2, pp. 373–440, 2020.

[101] V. Vapnik, The nature of statistical learning theory. Springer science & business media, 2013.

[102] J. Milgram, M. Cheriet, and R. Sabourin, " " one against one " or " one against all " : Which one is better for handwriting recognition with svms?" in Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft, 2006.

[103] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in Proceedings of the 33rd annual meeting on Association for Computational Linguistics. Cambridge, Massachusetts: Association for Computational Linguistics, 1995, pp. 189–96.

[104] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," CMU CALD, Tech. Rep. CMU-CALD-02-107, 2002.

[105] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," Advances in neural information processing systems, vol. 16, 2003.

[106] H. M. Le, Y. Yue, P. Carr, and P. Lucey, "Coordinated multi-agent imitation learning," in Proceedings of the 34th International Conference on Machine Learning 70, 2017, pp. 1995–2003.

[107] M. Teranishi, K. Fujii, and K. Takeda, "Trajectory prediction with imitation learning reflecting defensive evaluation in team sports," in IEEE 9th Global Conference on Consumer Electronics (GCCE 2020), pp. 124–125.

[108] M. Teranishi, K. Tsutsui, K. Takeda, and K. Fujii, "Evaluation of creating scoring opportunities for teammates in soccer via trajectory prediction," in 9th Workshop on Machine Learning and Data Mining for Sports Analytics 2022 (MLSA'22) co-located with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECML-PKDD'22), 2022, pp. 53–73.

[109] K. Fujii, K. Takeuchi, A. Kuribayashi, N. Takeishi, Y. Kawahara, and K. Takeda, "Estimating counterfactual treatment outcomes over time in complex multi-agent scenarios," arXiv preprint arXiv:2206.01900, 2022.

[110] J. Y. Park, K. T. Carr, S. Zhang, Y. Yue, and R. Yu, "Multiresolution tensor learning for efficient and interpretable spatial analysis," arXiv preprint arXiv:2002.05578, 2020.

[111] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.

[112] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[113] NBA.com, "Nba advanced stats," https://www.nba.com/stats/players/shots-closest-defender/, 2015, accessed: 2023-03-18.

[114] J. P. Guilford, Fundamental statistics in psychology and education. McGraw-Hill, 1950.

List of Publications

Journal Papers

[1] Z. Zhang, K. Takeda, and K. Fujii, “Cooperative play classification in team sports via semi-supervised learning,” International Journal of Computer Science in Sport, Volume 21, Issue 1, pp. 111-121, 2022, DOI: 10.2478/ijcss-2022-0006

[2] Z. Zhang, R. Bunker, K. Takeda, and K. Fujii, “Multi-agent deep-learning based comparative analysis of team sport trajectories,” IEEE Access, vol. 11, pp. 43305-43315, 2023, DOI: 10.1109/ACCESS.2023.3269287

Conferences

[3] Z. Zhang, K. Takeda, and K. Fujii, “Automatic screen-play classification in basketball via semi-supervised learning,” in Proceedings of the 9th MathSport International Conference, Reading, UK, July 2022.

[4] Z. Zhang, R. Bunker, K. Takeda, and K. Fujii, “Multi-agent deep-learning based comparative analysis in basketball,” in Proceedings of the 37th Annual Conference of the Japanese Society for Artificial Intelligence, Kumamoto, Japan, June 2023, pp. 3U1IS304–3U1IS304.

List of Figures

List of Tables