Applied Intelligence (2023) 53:3783–3803
https://doi.org/10.1007/s10489-022-03631-z

Graph representations for the analysis of multi-agent spatiotemporal sports data

Dominik Raabe[1,2] · Reinhard Nabben[2] · Daniel Memmert[1]

Accepted: 12 April 2022 / Published online: 2 June 2022

Abstract

Analyzing tactical patterns in invasion games using multi-agent spatiotemporal data is a challenging task at the intersection of computer and sports science. A fundamental yet understudied problem in this area is finding an optimal data representation for processing athlete trajectories using machine learning algorithms. In the present work, we address this gap by discussing common representations in use and propose *Tactical Graphs*, an alternative graph-based format capable of producing integrative, contextualized models for machine learning applications. We provide an in-depth, domain-specific motivation of the proposed data representation scheme and show how this approach exploits inherent data traits. We propose *Tactical Graph Networks* (TGNets), a light-weight, hybrid machine learning architecture sensitive to player interactions. Our method is evaluated with an extensive ablation study and the first comprehensive state of the art comparison between standard feature, state vector, and image-based methods on the same dataset. Experiments were conducted using real-world football data containing short sequences of defensive play labelled according to the outcome of ball winning attempts. The results indicate that TGNets are on par with state-of-the-art deep learning models while exhibiting only a fraction of their complexity. We further demonstrate that selecting the right data representation is crucial as it has a significant influence on model performance. The theoretical findings and the proposed method provide insights and a strong methodological alternative for all classification, prediction or pattern recognition applications in the areas of collective movement analysis, automated match analysis, and performance analysis.

Keywords Spatiotemporal sports data · Multi-agent modeling · Graph neural networks · Sports analytics · Match and performance analysis

1 Introduction

Spatiotemporal data such as trajectories of moving objects evolving over time and space have become ubiquitous in domains such as human mobility [55, 59], ecology [26, 41], multi-agent modeling [25, 28], and many more [58]. This development has recently also reached invasion games such as football [17, 35], basketball [4], or hockey [37]. Current tracking technologies allow to capture athletes' trajectories at high temporal resolution with satisfactory accuracy [34, 38]. Albeit the early scientific interest in athletes' trajectories had predominantly been in terms of physiological loads and time motion analyses [6], increasingly tactical aspects and organizational principles of underlying sports trajectory generation are investigated [47].

This development has been a game changer for sports-related tasks such as the recognition of movement patterns, analysis of collective behaviors, automated match analysis, or performance analysis [39]. A major reason for this is the fine granularity of athletes' position data. Position data provide a comprehensive, low-level description of the game play process, in contrast to conventional data sources

✉ Dominik Raabe
dominik.raabe@dshs-koeln.de

Reinhard Nabben
nabben@math.tu-berlin.de

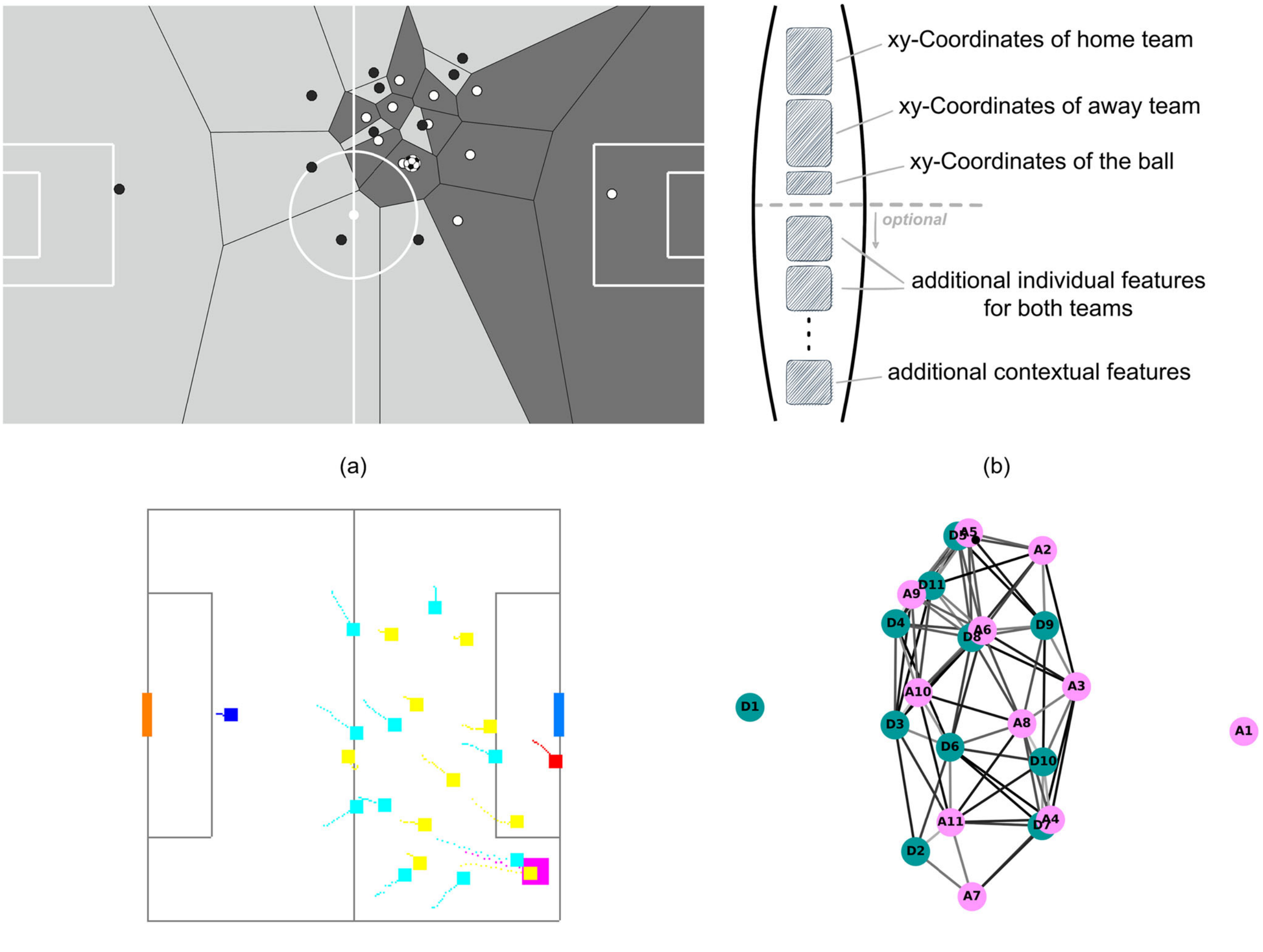
Daniel Memmert
d.memmert@dshs-koeln.de

[1] Institute of Exercise Training and Sport Informatics, German Sport University Cologne, Am Sportpark Müngersdorf 6, Cologne, 50933, Germany

[2] Institute of Mathematics, Technical University Berlin, Straße des 17. Juni 136, Berlin, 10623, Germany

such as video recordings or hand-annotated match reports. However, extracting useful information in terms of game play insights from this data has shown to be a difficult task [19]. One potential reason for this is the big semantic gap between raw trajectory data and high-level descriptions of behavioral patterns such as certain player constellations, individual well defined plays, or team specific playing styles [19, 37]. To connect the low-level description of athletes' trajectories to these more abstract representations, the modeling process requires several layers of abstraction, complexity reduction, and other critical design choices [10, 19, 45]. Additional domain-specific challenges are, on the one hand, restricted sample sizes due to restricted data availability and population sizes [10, 47]. On the other hand, researchers from sport science have repeatedly stressed the demand for approaches that are context-sensitive and integrative [16, 50].

A major unanswered question regarding the processing of spatiotemporal sports data thus is a methodological one: What are the technological prerequisites for leveraging this novel data source in order to better understand sports performance? Current approaches for invasion games tracking data typically derive and evaluate domain-specific performance metrics [40] (Fig. 1(a)). In this


xy-Coordinates of home team
xy-Coordinates of away team
xy-Coordinates of the ball
optional
additional individual features for both teams
additional contextual features
D5
A5
A2
A9
D11
D4
A6
D8
D9
D1
A10
A3
D3
A8
A1
D6
D10
A11
D7
A4
D2
A7

(a)

(b)

(c)

(d)

Fig. 1 Typical data representations used for spatiotemporal sports data applied to sample scenes in football. **(a) Hand-engineered features with domain expertise**: Rein et al. [48] calculated the space control of each player of both teams (black and white dots) by using a Voronoi-tesselation (areas shaded in light and dark gray). **(b) State Vector**: A state vector representation concatenates player and ball positions for a fixed time point into a single vector. This vector can be optionally extended with additional individual or contextual features. **(c) Image**: Wagenaar et al. [57] created color-images from raw data showing basic pitch markings (gray) including goals (orange, light blue), outfield players of both teams (cyan, yellow), goalkeepers (dark blue, red) and the ball (magenta). Colors were inverted for printer-friendly display. **(d) Graph**: We propose a graph representation where each player of both teams is encoded as one node (colored by team affiliation) and edges are included and weighted according to player distances

case, features and performance metrics are hand-engineered involving domain expert knowledge and subsequently used for (statistical) processing. An alternative set of techniques becoming increasingly popular are machine learning approaches such as Neural Networks (NN) [10, 21]. These methods, in contrast, often work end-to-end, i.e., they deliberately use little to no prior assumptions in the processing pipeline and thus leave the construction of appropriate features to the method [2]. Yet, to exploit the full potential of machine learning algorithms for spatiotemporal sports data processing, an appropriate data representation format has to be selected as part of the processing pipeline. Two popular data representation choices in sports are state vector representations of player positions (Fig. 1(b)) or images depicting the current game status (Fig. 1(c)). However, choosing the optimal data representation is a non-trivial task and specifies subsequent model selection steps. Representation as well as model selection affect sample size requirements and learning performance by imposing data invariances and inducing relational biases [2].

Researchers have previously emphasized domain-specific cases where it is beneficial that data representations are invariant towards permutation [31, 37], reflection [57], rotation, and translation [11] of input data. This is based on the assumption that applying these isometric transformations to player constellations changes their formal representation, but not their intrinsic characterization. These findings, however, are scattered among existing works. To our knowledge, no systematic evaluation of the relation between the characteristics of multi-agent spatiotemporal sports data and utilized data representations exists to date. A comparison of these data representation and corresponding machine learning methods in terms of computational performance is equally lacking. As a result, an architecture that recognizes not one but as many of these characteristics as possible has not been proposed. This is especially surprising given the domain-specific challenges highlighted above. Although major methodological issues for the analysis of multi-agent spatiotemporal data remain open, little attention is being paid to address these topics [21].

The present work aims to address this gap and proposes a novel data representation scheme that models players and their interaction as graphs (Fig. 1(d)). These *Tactical Graphs* exploit identified domain-specific characteristics of multi-agent spatiotemporal sports data and are capable of integrating features and contextual information as required. We correspondingly propose a light-weight, hybrid machine learning architecture called *Tactical Graph Networks* (TGNets) based on recently developed *Graph Neural Networks* (GNNs). This model framework is capable of processing individual player information with respect to their dyadic interactions, a property that has been missing in previous approaches. TGNet models learn parameterized filters sensitive to the overall graph connectivity encoding the structure of game play (Fig. 2). At the same time, they require only a fraction of the parameters used by comparable state-of-the-art architectures while allowing flexible feature engineering. We also conduct a state of the art comparison between different data representations and corresponding models on the same dataset, which we believe has not been done before. The experiments reveal that choosing the right representation and architecture has a strong influence on model performance, and that TGNets are as powerful as models of much greater complexity. We perceive our contribution as a theoretically grounded, domain-sensitive alternative for representing and processing spatiotemporal sports data. We argue that our contribution is less focused on a single application but instead tunable to the full range of supervised learning tasks. It may therefore be used for all classification, prediction or pattern recognition applications in the area of collective movement analysis, automated match analysis, and performance analysis.

In summary, the main motivation for the present paper is to evaluate how the choice of a certain data representation corresponds to the data idiosyncrasies and domain-specific demands in the area of spatiotemporal sports data analysis. The range of known representations is hereby extended by a novel graph-based representation with favorable properties, and the theoretical considerations are complemented by a systematic performance evaluation on a generic classification task. In that respect, our contribution is five-fold: (1) A collection of important characteristics of multi-agent spatiotemporal sports data; (2) Proposition of *Tactical Graphs*, an integrative, graph-based representation scheme for raw data; (3) Proposition of corresponding *Tactical Graph Networks* (TGNet), a GNN architecture suited for supervised learning tasks based on *Tactical Graphs*; (4) An extensive ablation study for the proposed architecture; (5) The first comparison of common data representations and corresponding methods on the same dataset and task, including simple baselines and the best TGNet variant.

A comprehensive overview over common data representations as well as Graph Neural Networks is provided in Section 2. We motivate and construct our data representation and method in Section 3. An ablation study as well as evaluation of all discussed representations on the same generic classification task is presented in Section 4. Section 5 discusses the results and concludes our work.

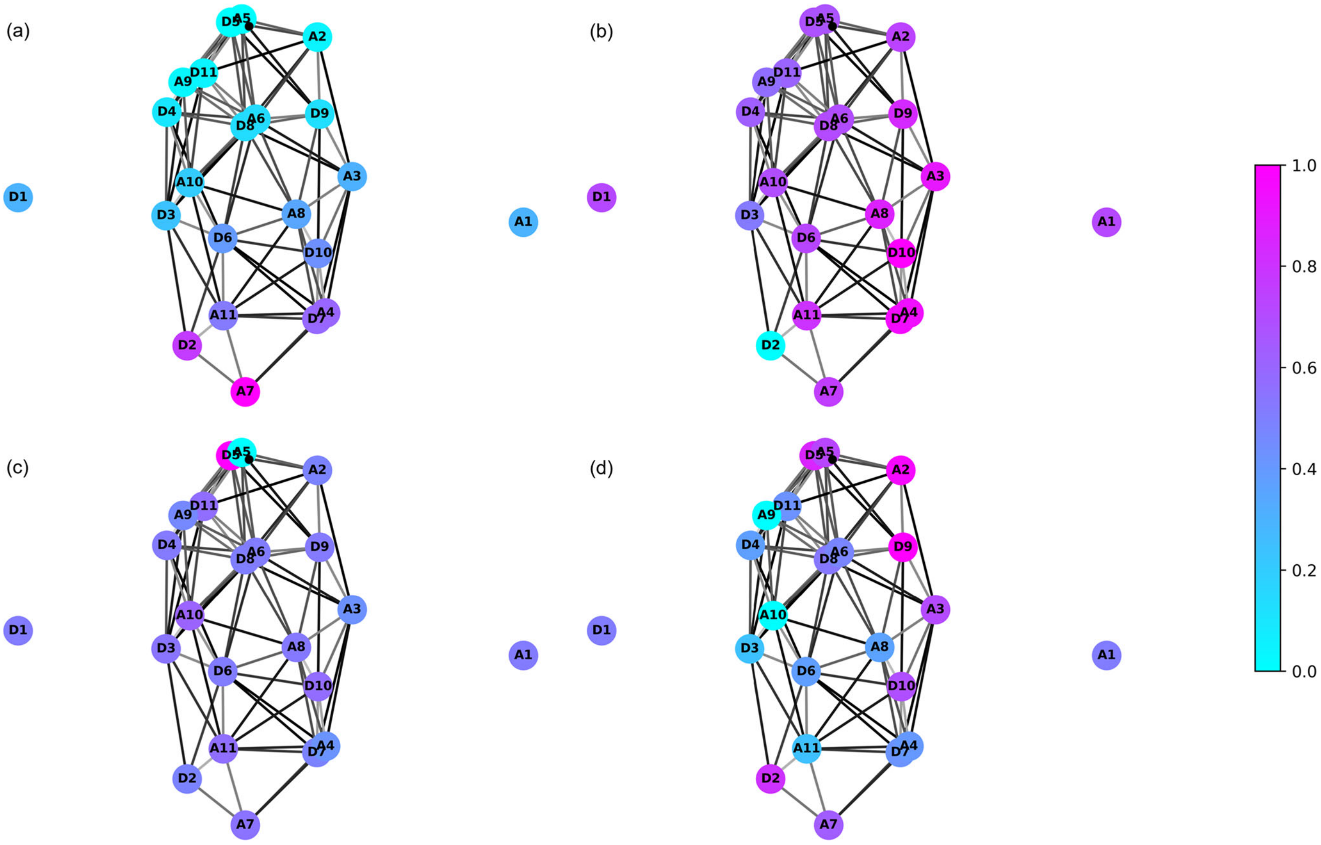


Fig. 2 Graph representation of a typical match situation with edges created between players that are less than 20 m apart, node colorings are based on different eigenvectors of the graph Laplacian and give an idea of the constructed frequency domain. **(a)** Lateral flow. **(b)** Longitudinal flow. **(c)** Two particularly close players. **(d)** Seemingly irregular structure

2 Related work

In this section, we review related work that aims to process spatiotemporal sports data with a special emphasis on the respective data representations used (Section 2.1). We also provide a brief introduction to recent developments within the area of Graph Neural Network learning and demonstrate how the resulting methods have shown great promise in the area of trajectory modeling (Section 2.2).

2.1 Representation of spatiotemporal sports data

The motivation and nature of scientific works utilizing spatiotemporal sports data are manifold and at times follow fundamentally different paradigms [14, 17]. They are nonetheless united by the initial challenge of choosing an appropriate data representation for subsequent processing. Three major options have emerged across disciplines to tackle this problem, which are reviewed in this section. These are manually constructing features (Section 2.1.1), concatenating raw data as state vectors (Section 2.1.2), and generating image snapshots of match play (Section 2.1.3).

2.1.1 Hand-engineered features

The majority of work concentrates on hand-engineering features from raw data by spatial and temporal aggregation [17]. These features are commonly referred to as *Performance Indicators* within the sport science domain and their calculation can be quite complex, often involving domain-expert knowledge. As an example, Rein et al. [48] calculated player- and event-related performance metrics such as space control gains based on Voronoi-tesselations (Fig. 1(a)) and outplayed opponents for passes in football. These were subsequently used in a cumulative link mixed model to test the influence of player performance on game outcome. A whole group of performance metrics are measures based on the team centroid, i.e., the geometric center of all players from one team [49]. These measures include the average distance of all players to their centroid (*Stretch Index*, [4]) or the distance between two team's centroid (*Inter-Team Distance*, [13]). Centroid based measures typically focus on the team structure as a whole, however more recently researchers have started to look at the summed individual distances of players to their nearest opponent (*Team Separateness*, [53]). According to a recent systematic review, these metrics are among the most prominent ones to study collective tactical behavior in football [35].

2.1.2 State vectors

An elementary approach to process spatiotemporal data consists of joining player and ball coordinates into state vectors. These vectors contain athlete's raw positions for a given time point (state) and may be enriched with additional contextual variables or features. Chronologically concatenating multiple state vectors results in multi-dimensional tensors containing full athlete's trajectories. This data structure can be consumed by a wide array of learning as well as non-learning algorithms.

In sports analysis, this representation has naturally been employed since spatiotemporal data itself has become available. In one of the earliest conducted studies, Intille and Bobick [23] used state vectors to classify plays in American football with Bayesian networks. Bialkowski et al. [3] have proposed an unsupervised algorithm to identify team formations in football. Their approach takes state vectors as inputs and subsequently calculates a player-to-role assignment. This intermediate step has previously been shown to compensate for a lack of ordering of player identities within state vector representation [37]. State vector representations have also been used for supervised learning algorithms. For example, Horton et al. [22] predicted the expert judgement of pass quality in football employing a variety of algorithms including support vector machines and logistic regression.

In the area of Neural Network learning, Grunz et al. [19] used state vectors to classify game initiations in football. The authors proposed self-organizing maps to reduce the dimensionality of the raw data and subsequently derive a set of prototypical player constellations. These player constellations were then utilized for classification in a hierarchical network setting. On the contrary, Le et al. [32] derived a continuous model of player behavior based on long short-term memory (LSTM) networks in an imitation learning setting.

Although methodologically quite different, all these works used state-vector representations for their algorithms. These vectors, however, differed in their configuration and inclusion of information exceeding raw positions. Whereas Grunz et al. [19] used Cartesian coordinates, Horton et al. [22] extended their data representation with player's angular displacement, their velocity or controlled space, and Lucey et al. [37] as well as Bialkowski et al [3] used a role-based instead of identity-based indexing. In these cases, state vectors were constructed problem-specific to form a collection of individual positions and features for a given state.

2.1.3 Images

Motivated by the great success of Convolutional Neural Networks (CNNs) in many computer vision tasks such as object classification [33], some authors have suggested an image format derived from raw spatiotemporal data [8, 57]. Wagenaar et al. [57] created RGB images (such as in Fig. 1(c)) and trained several networks to predict whether a short sequence of match play resulted in either a goal scoring opportunity or a loss of possession. Similar, Dick and Brefeld [8] created gray-scale images from position data and used a CNN in a reinforcement learning setting to rate player positioning. Although it seems unconventional at first to convert spatiotemporal data to a pure spatial representation, both works demonstrate a principal feasibility of the approach.

2.2 Graph neural networks

Research on learning algorithms working, in a Neural Network fashion, on graph domains dates back to Gori et al. [18] and Scarselli et al. [51]. Recently, graph learning algorithms have gained considerable attention for their capability of working on structured domains [9] as well as unstructured domains which are difficult to embed into Euclidean space such as molecules [15]. Graph Neural Networks (GNNs) have emerged from this line of research as a distinct class of learning algorithms capable of adapting to data from non-Euclidean domains such as graphs or manifolds [2, 5]. By fusing graph theoretic concepts with important results in harmonic analysis, these algorithms have a solid mathematical foundation which allows to analyze functions defined on manifolds and graphs [52]. This has enabled methods such as the *Graph Convolutional Network* (GCN) by Kipf and Welling [29] that transfer the concept of convolution to graph domains.

In terms of analyzing spatiotemporal data in general, GNN based approaches have recently shown great promise for modeling the behavior of multiple interacting objects and agents [1, 28]. Especially in the area of multi-agent trajectory prediction, such as the task of forecasting the path of a person walking in a crowd, recent advances have successfully utilized graph structures [25]. More specifically, these methods were able to predict future trajectories of athletes in football and basketball [28, 60].

When compared to more classical Neural Network architectures that digest Euclidean data such as state vectors, graph-based approaches offer a variety of advantages that may lead to more robust and general models when used with non-Euclidean input data. Battaglia et al. [2] argue that integrative approaches need to be contrived to achieve better combinatorial generalization within learning algorithms. The authors propose graph-based approaches as the most promising candidate in this regard as a graph representation of data enables to induce favourable relational biases into network architectures. According to the authors, the main benefits include

- the permutation-invariant representation of entities as sets,
- the ability to learn based on relationships and interactions between entities,
- the support of combinatorial generalization as filter parameters are reused across the entire domain,
- the possibility to construct hybrid models that are able to include a priori information, and
- the facilitation of network architectures that preserve data invariances under certain operations.

In summary, GNNs show a number of interesting properties and have been applied successfully in the area of multi-agent trajectory modeling. This makes them a potentially good fit for the processing of unstructured spatiotemporal sports data.

3 The tactical graph framework

To enable a GNN processing of spatiotemporal sports data, two things are required. First, a sensible graph representation of the input data needs to be derived. The representation should address domain-specific requirements and overcome shortcomings of existing data representation schemes like state vector or image representations. Second, a model architecture which can leverage this representation must be constructed and tested. This section provides the foundation of this work and tackles both steps. The section starts by presenting a suitable problem formulation (Section 3.1) and proposes *Tactical Graphs*, an integrative, graph-based data representation scheme as a potential alternative (Section 3.2). Subsequently, a motivation for the suggested approach by comparing the presented data representations regarding domain-specific characteristics of the underlying data is provided (Section 3.3). Finally, *Tactical Graph Networks*, a network architecture that is capable of leveraging the proposed data structure for learning tasks are introduced (Section 3.4).

3.1 Problem formulation

For a fixed $t \in \{0, 1, ..., T\}$, positions of a set $\mathcal{P} = \{1, 2, ..., N\}$ of N players in the Cartesian plane can be defined as a set of tuples[1]

$$\mathcal{C}_t = \{(x_{t,1}, y_{t,1}), (x_{t,2}, y_{t,2}), ..., (x_{t,N}, y_{t,N})\}.$$

Spatiotemporal sports data typically includes a playing device $b_t = (b_{t,1}, b_{t,2})$ such as a ball or puck. The union of these sets for multiple (discretized) timepoints $t_0 \leq t \leq t_1$ results in a set $\mathcal{C}_{t_0,...,t_1}$ of trajectories for a defined time segment. This set then includes the trajectory of an individual player i, i.e.,

$$\{(x_{t_0,i}, y_{t_0,i}), (x_{t_0+1,i}, y_{t_0+1,i}), ..., (x_{t_1,i}, y_{t_1,i})\} \subseteq \mathcal{C}_{t_0,...,t_1}.$$

Although these sets are chronologically ordered, they are spatially unstructured as athletes move freely around the pitch. We now would like to find a data representation scheme that assigns, for a given t and for each set of positions $\mathcal{C}_t$, a corresponding data point V_t. Consequently, for a set of trajectories $\mathcal{C}_{t_0,...,t_1}$, data in the form $V_{t_0,...,t_1}$ can be derived for game play unfolding over time.

This will allow us to formalize the data representations discussed in Section 2.1 (hand-engineered feature representations V^{feat}, state vector representations V^{sv}, and image-based representations V^{img}) as well as the proposed graph-based representation (V^{tg}). To describe these data representation schemes, expert-based, aggregated features (Section 2.1.1) as well as individual and contextual features that extend state vectors containing raw positions (Section 2.1.2) need to be defined. Thus, let $e_t^{(k)}$ specify the k-th of K total expert-based features at time point t, and $e_{t_0,...,t_1}^{(k)}$ the same feature spanning the temporal interval for $t_0 \leq t \leq t_1$. Furthermore, let $a_{t,i}^{(j)}$ specify the j-th of M total individual features for the i-th player at time point t. Finally, let $c_t^{(l)}$ specify the l-th of P total contextual features at time point t. This distinction between feature types is a technical rather than semantic one, intended to distinguish between data representation types. In this sense, expert-based features are characterized by an elaborate temporal aggregation (e.g. by summation or averaging over a time interval), whereas individual features are aggregated via concatenation. Individual features are additionally calculated per athlete, whereas contextual features are not.

A fully feature-based data representation over a given time interval can then be expressed as a vector

$$V_{t_0,...,t_1}^{feat} = (e_{t_0,...,t_1}^{(1)}, ..., e_{t_0,...,t_1}^{(K)})$$

[1] Assuming a description in 2D-space. Although less common, a 3D description is equally possible by adding a z-component to each tuple

with a total of K entries. A state vector representation for a given t containing raw positions of N players, the playing device and, optionally, M individual as well as P contextual features is defined as

$$V_t^{sv} = \underbrace{(x_{t,1}, y_{t,1}, ..., x_{t,n}, y_{t,n}}_{\text{player positions}},\ \underbrace{b_{t,1}, b_{t,2}}_{\text{ball position}},\ \underbrace{a^{(1)}_{t,1}, a^{(1)}_{t,2}, ..., a^{(2)}_{t,1}, ..., a^{(M)}_{t,N}}_{\text{individual features}},\ \underbrace{c^{(1)}_t, ..., c^{(P)}_t)}_{\text{contextual features}}\ .$$

This vector has a length of $|V_t^{sv}| = 2N + 2 + NM + P$, depending on the number of selected features. Joining multiple state vectors for a range of T time steps consequently results in matrices (or tensors)

$$V^{sv}_{t_0,...,t_1} = (V^{sv}_{t_0}, V^{sv}_{t_0+1}, ..., V^{sv}_{t_1})$$

of shape $T \times (2N + 2 + NM + P)$. Lastly, image representations can be created for a single time point, but may also incorporate trajectories of players such as the images constructed by Wagenaar et al. [57] which are also used in this study. Both variations thus conform to

$$V_t^{img}, V^{img}_{t_0,...,t_1} \in \mathbb{R}^{Width \times Height \times Channels}.$$

Whereas grayscale images only contain a single channel, this dimension increases for color images such as RGB representations.

The problem under investigation then reduces to three questions. (1) Are these known data representations appropriate for the analysis of spatiotemporal sports data? (2) Is a potential graph representation more appropriate given the nature of the data source at hand? And, (3) how do these representations perform in the same generic analysis task? Since the advent of machine learning algorithms such as Neural Networks, it remains unclear which of these data representations is most suitable as input for such methods. To our knowledge, no systematic comparison has been conducted on the same dataset. The three raised question are of particular importance in light of the domain-specific challenges presented in Section 3.3.

3.2 Construction of tactical graphs

Before we can compare and evaluate different data representations, we propose an alternative, graph-based representation scheme called Tactical Graphs. This scheme takes a snapshot of match play and produces a graph view of the data (see also Fig. 3 for a visual display). Players and player-related attributes are modelled as graph nodes and their interactions as weighted edges. The main motivation for choosing this representation is that it provides a structure for raw data that has interesting properties such as a strong emphasis on player relations (see Section 3.3 for an analysis of these properties). By incorporating player and interaction attributes, it also provides a mechanism to integrate task-specific information that exceed the expressiveness of the raw data, such as performance metrics, into a single and cohesive data structure. The representation thus facilitates hybrid models where processing is deliberately biased towards player interactions.

To this end, for a fixed time point t, we transform raw spatiotemporal data to an undirected, weighted graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})\ ,$$

where each player $i \in \mathcal{P}$ is identified by one node in a set of vertices $\mathcal{V} = \{1, ..., N\}$, and player relations are modeled by a set of edges $\mathcal{E}$. Edges represent relations between players, i.e., $e = (i, j) \in \mathcal{E}$ encodes the connection between the two players identified by node i and j. For each edge, we may then assign a non-negative weight $w_{ij} \geq 0$, which models an interaction between those players, such as their distance to each other on the pitch. The full set of edges translates to a weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ with $w_{ij} = 0$ if $w_{ij} \notin \mathcal{E}$. For our purposes, we can exclude self-loops, i.e., $w_{ii} = 0$ for $i \in \{1, ..., N\}$. Due to the undirectedness, $(i, j) \in \mathcal{E}$ further implies $(j, i) \in \mathcal{E}$. Hence, $w_{ij} = w_{ji}$ and $\mathbf{W}$ is symmetric. The *unnormalized graph Laplacian* [56] of a graph is then defined as

$$\mathbf{L} := \mathbf{D} - \mathbf{W},$$

where $\mathbf{D} = diag(d_{11}, ..., d_{NN})$ is the diagonal matrix of node degrees $d_{ii} = \sum_{j=1}^{N} w_{ij}$.

At this point we have constructed a graph that models a game situation by encoding players and their relations in a graph structure. Their relations are additionally encoded by an interaction property, such as their relative distance, as edge weights summarized by the Laplacian. We proceed from here and assign M individual player properties to the respective nodes by constructing a node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, where x_{ij} denotes the j-th feature of player i. These M features can be freely selected and correspond to the individual features $a^{(j)}_{t,i}$ used for constructing a state vector. The final Tactical Graph representation then summarizes to the graph Laplacian and node feature matrix fixed for a time point t, i.e.,

$$V_t^{tg} = (\mathbf{L}_t, \mathbf{X}_t),$$

and for a time interval they may be joined in a tuple, i.e.,

$$V^{tg}_{t_0,...,t_1} = (V^{tg}_{t_0}, V^{tg}_{t_0+1}, ..., V^{tg}_{t_1}).$$

Although this representation, compared to a state vector representation V_t^{sv}, might appear a little obfuscated at

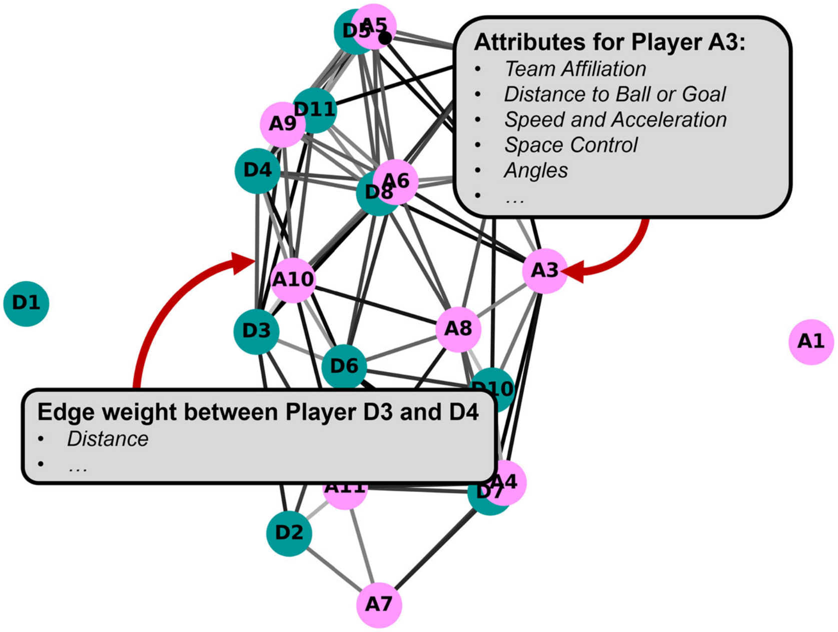


Fig. 3 Portrayal of a *Tactical Graph*. For each player of both teams we assign player attributes as node features (collected in a node feature matrix). Each edge weight between players is set as an interaction property (collected in the weighted adjacency matrix)

first, the two representation are in fact quite similar. All individual features $a_{t,i}^{(j)}$ of the state vector representations may be subsumed in the constructed node feature matrix. This includes raw (x, y)-coordinates of players if these are treated as an additional player property. These features are further complemented by a graph Laplacian that incorporates a compact encoding of the player interaction structure determined from the specific context. Most GNNs which follow the message passing scheme presented by Gilmer et al. [15] are also capable of including a set of global properties such as the ball position or contextual features [2]. The proposed representation can thus be easily extended with such features. For the present study, however, this possibility is omitted as the focus is put on the general model feasibility.

With regards to the computational costs, it should be noted that the construction of a graph Laplacian is generally not cheap, as it scales quadratically with the number of graph nodes. However, for sports applications, the number of players (and thus nodes) is typically very small, such as $N = 22$ in the present study. In these scenarios, the construction per time point can be performed with reasonable burden. The overall complexity of constructing $V_{t_0,\ldots,t_1}^{tg}$ then depends, on the one hand, on the number of analyzed frames. This is a linear relationship that can be mitigated by downsampling the temporal resolution of the data as desired. On the other hand, the complexity of the computed node features $\mathbf{X}$ and edge weights $\mathbf{W}$ further affects total computational cost. However, this is a general decision in model design that needs to be addressed by the analyst, irrespective of the chosen data structure.

3.3 Characteristics of spatiotemporal sports data

Spatiotemporal data themselves already prove to be challenging data source to work with. Data coming from invasion games, where players from two teams exert (mainly) organized collective movement behavior to achieve a common goal, pose additional challenges. In general, a major problem constitutes the mismatch between available sample sizes and data requirements of deep learning techniques in particular [10, 47]. Even if unlimited data were available, domain-specific research questions, e.g., regarding movement patterns of a particular group of players, naturally limit the number of usable samples.

Researchers from sports science have also stressed the demand for context-sensitive, integrative approaches which are able to incorporate information from multiple sources (such as individual physiological or technical abilities of athletes or general playing capabilities of teams) [16, 50]. This argument can be extended by anticipating future possibilities of including data from multiple data sources and modalities, such as biometric data from sensors worn by players or so-called event data capturing player actions.

Further, given the special nature of the data-generating process, several characteristics of spatiotemporal sports data can be identified. To address the challenges discussed above, it seems favourable to analyze these domain-specific data traits and incorporate the findings into future model construction. Machine learning architectures that are adjusted carefully to sports data should certainly require fewer training samples if they are able to exploit the intrinsic structure of the data. Additionally, models that recognize domain-specific demands have a much greater chance to eventually be useful for answering sports-related research questions. In this section, we motivate our proposed scheme by demonstrating how a graph representation applies particularly well to the analysis of spatiotemporal sports data by exploiting invariances (Section 3.3.1), relationality (Section 3.3.2) and compositionality (Section 3.3.3) of the data generating process.

3.3.1 Invariances

Consider the constellation of a 2-versus-1 overload situation in football, that could, e.g., result from an overlapping play on the wing (Fig. 4A). Similar to the argument presented by Feuerhake [11], we argue that this constellation (A) is characterized by player's relative positioning and is tactically equivalent to a translated (B), reflected (C), rotated (D), or permutated[2] (E) version. It would still remain an overlapping play if it occurred higher up the pitch (B), on the opposite flank (C), or in a different playing direction (D, E). We would also not be interested if player 2 overlaps player 1 or vice versa (E).

A data representation, such as a state vector representation, that is not invariant towards these operations is prone to accommodate semantically similar constellations in distant parts of the feature space, and needs to learn these mappings at the cost of increased sample size requirements. In fact, the role of player permutations has been frequently highlighted in the literature [3, 31, 32, 37]. With respect to formation detection, Lucey et al. [37] suggested that a role-based representation of players is more appropriate than an index-based representation and suggested a role-aligned state vector representation, a choice adopted by most further research concerned with this question. Knauf et al. [31] specifically designed their kernel to be invariant under permutations of trajectory components. Furthermore, Feuerhake [11] highlighted the importance of rotation and translation invariances in player constellations for pattern recognition. For image-based analyses, authors have included mirrored images in their datasets to account for a lack of reflection invariance [57]. In summary, we argue that it seems worthwhile to design data representations of player constellations to be invariant with respect to certain isometric transformations. This is certainly not the case for all situations as some tactical plays are depending on their absolute position[3]. Yet, this should be the case for only a subset of plays as invasion games playing surfaces are typically quite large and a major part of match play unfolds away from the sidelines. Accordingly, the absolute placement of a given constellation of players could still be encoded in a different manner.

Graph representations such as Tactical Graphs are invariant towards translations, reflections, and rotations [2], whereas this is not the case for state vector or image representations. Although researchers have proposed other solutions to partly circumvent this limitation (see discussion above), these invariances are inherent to graph representations by design. Furthermore, if we restrict computations performed on these graphs to be permutation invariant, the resulting models are also permutation invariant. This eliminates the need for the intermediate step of finding suitable role-assignments [37] as is the case for a state vector representation.

3.3.2 Relationality

Relationships and interactions between players occupy an inseparable and crucial part in dynamic game processes [47]. Especially dyadic player interactions have been extensively studied in recent years and play an important role in analyzing collective movement behavior from a sports science perspective (see [35] for an overview). This circumstance is further underlined by the many existing works that utilize network structures to investigate event-based player interactions such as passing networks in football [43, 46]. With regard to spatiotemporal data analysis, however, player interactions play a secondary role as data is typically analyzed for an entire team at once. GNNs, on the contrary, are explicitly designed to exhibit a strong sensitivity towards entity relations by inducing relational biases [2]. This is a favourable property in the present case as it provides a way to include player relations into the model building procedure.

3.3.3 Compositionality

It seems reasonable to assume that invasion games tactics further exhibit some sort of compositional and hierarchical structure. Especially on an organizational level of game play, collective player behaviors have been categorized into individual-, group-, and team-tactical levels [47]. Correspondingly, player interactions are often understood at a dyadic, group, or team level. Although this relationship is certainly bi-directional, a bottom-up composition of movement behavior might be assumed. Whereas only three players are involved in the overlapping play discussed in Section 3.3.1, this play could be embedded in a general attack strategy that involves overloading of one side of the pitch, which itself could be part of the overall positioning structure of the attacking team. Grunz et al. [19] exploited this trait in their work by designing a hierarchical Neural Network structure of multiple stacked layers performing

[2] and reflected for better perceptibility

[3] As an example, imagine a defensive situation that involves usage of the pitch sidelines.

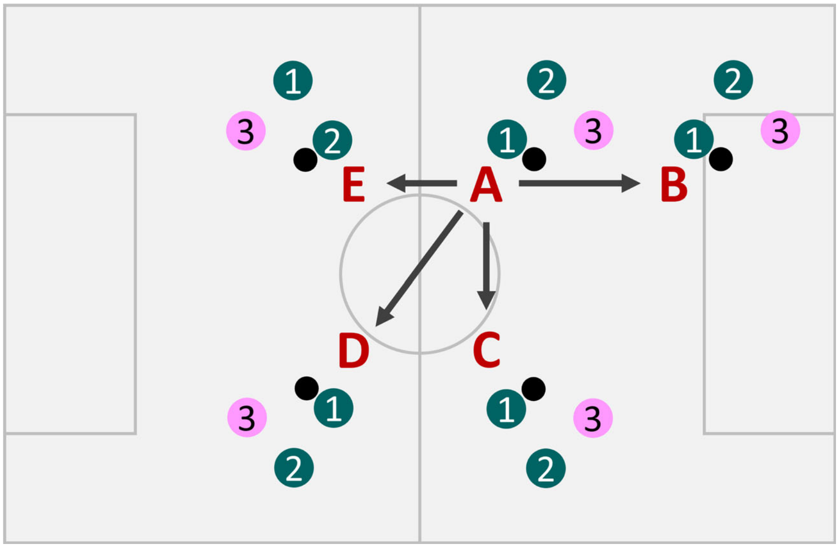


Fig. 4 Typical 2-versus-1 constellations in football between two attacking players (indexed as 1 and 2, petrol, with ball) and a single defender (indexed as 3, pink). The situation labelled **A** shows the original constellation. The other four variations can be derived by translation **(B)**, reflection **(C)**, rotation **(D)**, or permutation **(E)**

dimensionality reduction before final predictions were made. In a similar fashion, graph representations and consequently GNNs are particularly suitable to exploit compositionality within the data as per-node and per-edge functions are used across the domain [2]. Hierarchical designs could additionally be obtained in straightforward fashion by employing graph coarsening algorithms that subsume graph nodes. This approach reflects the typical design of CNNs, which effectively complement convolutional layers with pooling layers to achieve layer-wise abstraction and dimensionality reduction [33].

3.4 Tactical graph networks

Tactical Graphs allow to construct data structures that show beneficial properties and allow for flexible feature engineering. In this section, we propose a corresponding Neural Network architecture called *Tactical Graph Network* (TGNet). A schematic summary of the proposed architecture is presented in Fig. 5. Its basic architecture can be divided into multiple units performed in five blocks: construction of Tactical Graphs as input data (*Data Representation*), pre-processing of input data (*Pre-Processing*), a network layer performing multiple graph convolutions and embedding (*Convolution Block*), a recurrent layer for processing of subsequent embeddings (*Recurrent Block*) and finally a task-specific output layer (*Classification*).

The initial step consists of constructing Tactical Graphs by constructing selected features and composing the respective node feature matrices and Laplacian for each time step. As discussed in Section 3.2, feature engineering can be performed as desired and may be adapted with respect to the desired application. For pre-processing, batch normalization [24] was applied on the node feature matrix as different attributes range on independent scales of different magnitude. Next, we include a sparsification layer that

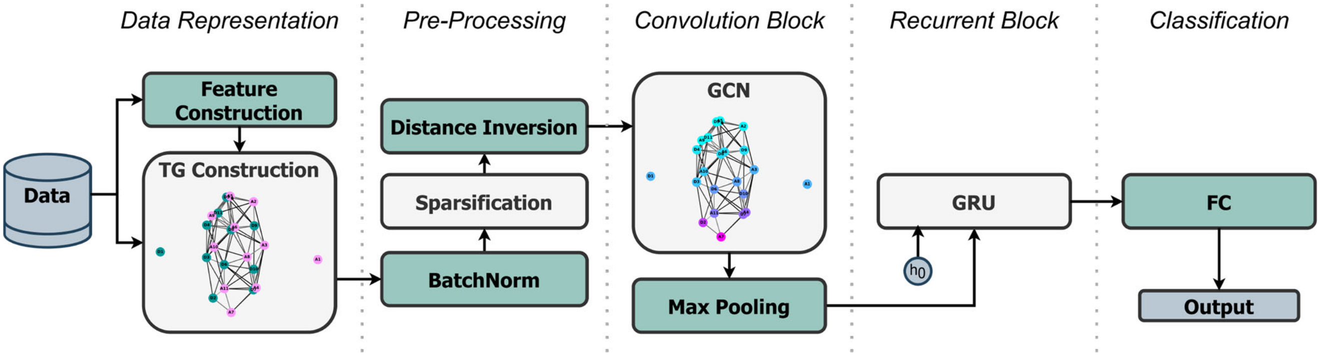


Fig. 5 The full Tactical Graph Network architecture in block structure. Each block can be assigned to one of five steps. Arrows indicate data flow within the model from raw data to a final, task-specific output (in the present study, this translates to a two-dimensional output vector). The four units colored in light gray are tested in an ablation study (see Section 4.4)

removes edges from the fully connected graph based on a hyperparameter threshold ϵ acting on edge weights. As we select player's distances as edge weights in the present study, this step can be understood as enforcing an ϵ-neighborhood graph [56] and is equivalent to removing edges between players that are separated further than the threshold distance. This operation assumes the dyadic interaction of players far apart to be of less relevance for the tactical constellation on the pitch. It also allows us to modulate the sparsity of the resulting graph Laplacian which improves computation speed and has previously shown to improve prediction accuracies [30]. The last (optional) pre-processing step involves inverting edge weights based on a fixed norm value. Thus, in our case, the distance between every pair of players was inverted based on the maximal distance two players can be apart on a standard sized football pitch of size $110\,m \times 68\,m$. Due to the inversion, two players standing at the same location will be connected by an edge of maximal weight (i.e. $129.32\,m$), whereas the weight converges towards zero as the players move apart. Distance inversion thus achieves a more meaningful encoding of edge weights, as a missing edge is represented as a zero within the weight adjacency matrix.

The convolution block consists of multiple *Graph Convolutional Network* (GCN) layers as proposed by Kipf and Welling [29]. GCNs have shown to be an effective, light-weight and scalable GNN layer, where each learnable filter is parameterized by only a single parameter per input channel. Furthermore, GCNs fully operate in the spatial domain as the propagation rule is spectrum-free, i.e., it does not depend on a spectral decomposition [5]. This is advantageous in terms of computational costs and also implies that the learned filters can be reused across different graphs. These filters are instead spatially localized by 1-hops, i.e., information is passed to and from every node within the respective 1^{st}-order neighborhood during convolution. This is another reason for the inclusion of a sparsification step, as a 1-hop filtering draws on a meaningful implementation of neighborhood.

After (multiple) convolutions, the resulting graph is embedded via max pooling. Max pooling coarsens the graph into a single node, where the node's output features are the maximum values of all respective features across all graph nodes. This is a permutation invariant operation, rendering the entire model invariant under permutation of the input data, i.e., changing the order of players in the graph by permuting node indices.

The resulting embeddings for each time step are then passed to a previously initialized Gated Recurrent Unit (GRU) [7]. This may be repeated for as long as a given sequence lasts, and each step produces an output as well as new hidden vector. The GRU output vectors are then used for generating a model output. This final layer may differ in its composition depending on the given task. The presented architecture is thus flexible and may adapt to a range of problem formulations. In the case of the present study, the output is used for a binary classification task, and therefore a fully connected layer is used to generate a two-dimensional output vector.

The overall model structure is reminiscent of similar approaches that attach a recurrent block to a convolutional block [55, 59]. The structure can be understood as a two-step procedure where the convolution is used for frame-by-frame feature construction, and the following recurrent block aims to detect temporal relations and patterns within sequences of the convolved input. The GCN layer in our model thus takes the task of feature construction and is of special importance for model performance. The used graph convolutions learn filter operations which process signals (encoded as the node feature matrix) based on the graph connectivity (encoded as the graph Laplacian). Practically speaking, this means that convolving a Tactical Graph amounts to a parameterized processing of player properties with respect to player interactions.

Although constructed spectrum-free, the convolution itself is dependent on the graph structure and has a clear mathematical interpretation in terms of the Laplacian spectrum. The eigenvalues of the graph Laplacian L carry a notion of frequency as discussed by Shuman et al. [52] and demonstrated by Bronstein et al. [5]. Eigenvectors of ascending eigenvalues for a sample graph Laplacian (Fig. 2) reveal the varying degree of oscillation exhibited throughout the graph domain and give an idea of which patterns the filters are sensitive to during learning. Low-frequency eigenvectors resemble isolated vertices disconnected from the main portion of the graph or lateral (Fig. 2(a)) and longitudinal (Fig. 2(b)) flow across the domain, respectively. With increasing frequencies, oscillations grow. The node colorings also demonstrate that the spectrum of the graph Laplacian produces a structurally dependent interpretation of the domain. In this sense, certain aspects of the graph are highlighted, which in some cases reflect an intuitive interpretation of the representation. For example, Fig. 2(c) discriminates between two particular close players. On the contrary, this does not always have to be the case as can be seen in Fig. 2(d). Note that the GCN propagation rule utilizes the symmetrically normalized adjacency matrix with added self-loops. This matrix differs slightly from the standard Laplacian, which we have used in the problem formulation and creation of Fig. 2 for the sake of clarity. The eigenvalues and -vectors of both matrices differ in scale and ordering, but demonstrate the same oscillating behavior.

4 Experiments

Two experiments were conducted. The first experiment optimized the proposed TGNet architecture by performing a four-step ablation study (Section 4.4). The second experiment conducted a state-of-the-art comparison between the discussed data representations and corresponding models as well as our contribution (Section 4.5). For both experiments, a generic classification task was designed (Section 4.1) and performed on the same dataset (described in Section 4.2). The used training routines are documented in Section 4.3. The results of both experiments are presented in Section 4.6.

4.1 Classification task

To evaluate the functionality of the discussed spatiotemporal data representations, we designed a generic binary classification task with real-world football data. As football is a low-scoring game, we choose an alternative target variable to ensure high occurrence of positive and negative cases. Researchers have previously used alternative targets for offensive success such as goal scoring opportunities [57] or entries to a defined danger zone [8]. We follow this approach of using sensible proxies and suggest ball wins, i.e., a regain in ball possession, as a measure for defensive success. The task then is to classify short sample sequences as successful (*possession win*) or unsuccessful (*no possession win*) from the defending team's perspective.

4.2 Dataset

Spatiotemporal data was acquired with a camera-based tracking system from 34 international top-flight games played between 2018 and 2019 with a sampling frequency of 25 Hz or 30 Hz. To ensure a uniform sampling frequency and remove temporal redundancy, the data was downsampled to a temporal resolution of 5 Hz. *Possession wins* or *transitions*, i.e., frames where possession changes between teams, were automatically extracted based on a supplied frame-by-frame possession flag. This possession information typically accompanies tracking data supplied by major data providers, where it is manually annotated and technically optimized for frame-accurate annotations.

Based on this information, samples were derived as short sequences of player and ball trajectories with a length of $T = 3\,s$, i.e., 15 frames. They were labelled *negative* if they did not end in a possession win for the defending team, and *positive* if they did. Positive sequences were supposed to capture the last three seconds before a possession win. For their construction, the respective sequence was chosen to start 3.5 s before and end .5 s before an identified possession win. This introduces a half-second gap between the last frame of the sequence and the transition moment to prevent the network from focusing on the exact player-to-ball distances at the moment of transition. Negative samples with the same length and buffer were randomly chosen from the remaining match play to match the number of positive samples. Subsequently, samples were excluded if

- the turnover occurred in combination with a dead ball situation (e.g. a goal kick after a missed attempt),
- possession changed within the sample sequence or
- game play was interrupted within the sample sequence.

The final dataset consists of $N_{samples} = 8366$ labeled sequences equally balanced between the two classes. All sequences contain uninterrupted, open play free of possession changes or dead ball situations by design.

For each sample, frame and player we calculated 13 individual data points *IF1 - IF13* summarized in Table 1. These features were selected to cover a broad variety of works both from the computer science as well as sport science community (see Section 2.1.1). As we will discuss more thoroughly in Section 4.5, these features will take the role of the individual features $a_{t,i}^{(j)}$ as well as, in aggregated form, the expert-based features $e_{t_0,\dots,t_1}^{(k)}$ used for constructing the models used in the state of the art comparison. These features are complemented by the (later inverted) player-to-player distances. This relational feature *RF1* encodes the dynamic interaction between players and will be used as edge weights during graph construction.

Additionally, for each sample, nine aggregated features were calculated which are summarized in Table 2. These features are taken as input for 'classical', feature-based models. They were selected to represent the current state of the art within the sport science community and were previously linked to defensive behavior [36]. Most of them can be derived by aggregation of the individual features (*IF1 - IF13*) to improve comparability between models.

4.3 Training

For both experiments, samples were divided into training, validation and test sets. The allocation was based on the chronological order the games took place to simulate a practical situation where a prediction for upcoming games is based on the previously played games. We aimed for a 70%-15%-15% split that translates to the (chronologically) first 24 games forming the training set, the subsequent 5 games forming the validation set and the final 5 games forming the test set. In terms of sample sizes, this resulted

Table 1 Individual features for each player, derived for each frame in each sample

Abbr.	Feature	Definition
IF1	X-Coordinate	Raw longitudinal position
IF2	Y-Coordinate	Raw lateral position
IF3	Team Affiliation	Player belonging to team in or out of possession
IF4	Distance to Ball	Distance in m
IF5	Speed	Average speed in ms^{-1}
IF6	Acceleration	Average acceleration in ms^{-2}
IF7	Distance to Centroid	Distance player to respective team's centroid in m
IF8	Space Control	Proportion of controlled space as by Voronoi approach
IF9	Angle to Ball	Angular displacement from ball in degree
IF10	Distance to Goal A	Distance in m
IF11	Distance to Goal B	Distance in m
IF12	Angle to Goal A	Reference angle between player, goal and horizontal mid
IF13	Angle to Goal B	Reference angle between player, goal and horizontal mid

in sets of 5906-1238-1222 samples (70.6%-14.8%-14.6%). All sets are balanced by construction. All Neural Networks were trained on the training set for 10 epochs. After each epoch, prediction accuracy on the validation set was calculated. After training, the model that achieved the highest validation accuracy was tested on the test set and common performance measured were determined. For Neural Network type models, a sigmoid activation was used for the output layer and hence *Cross Entropy Loss* was calculated for optimization.

All computations including data processing, dataset creation, network creation, and training were done in the *Python* programming language. Neural Network training was done using the *PyTorch* library [44] and for the implementation of the graph-dependend layers the *PyTorch Geometric* extension [12] was used. For graph visualizations the *NetworkX* package [20] was used. All models were trained and wall-clock times were measured on a mid-end consumer laptop computer (Intel Core i5 Processor with 2 × 2.60 GHz and 8 GB RAM) to simulate the lower bound of hardware used in practical settings.

4.4 Experiment 1: ablation study

To find an optimal TGNet architecture, we start with a minimal initial configuration and successively add modules or complexity while monitoring performance. The initial network configuration starts with Tactical Graphs constructed with two features ($a_{t,i}^{(1)} \widehat{=}$ *IF3* and $a_{t,i}^{(2)} \widehat{=}$ *IF4* from Table 1) and *RF1* as edge weights. Sparsification is omitted, i.e., data is passed with little pre-processing to the convolution block. The recurrent block is also omitted. Instead, the last frame of each sequence is passed exclusively to the network, and the classification is based directly on the feature representation that the convolutional layer produces. The convolution block itself consists of only a single GCN layer. The ablation is then performed in four successive steps:

- **Sparsification** A sparsification step is included as a pre-processing module, removing edges based on a hyperparameter threshold ϵ to reduce computational cost and improve performance. Different distances from

Table 2 Aggregated features constructed for each sample and their relation to individual features

Abbr.	Feature	Definition
AF1	Ball Distance Offense	Minimal *IF4* of all offensive players
AF2	Ball Distance Defense	Minimal *IF4* of all defensive players
AF3	Stretch Index Offense	Average *IF7* of all offensive players
AF4	Stretch Index Defense	Average *IF7* of all defensive players
AF5	Team Separateness Offense	Sum of minimal *RF1* for all offensive players
AF6	Team Separateness Defense	Sum of minimal *RF1* for all defensive players
AF7	Inter Team Distance	Distance between both team centroids
AF8	Space Control Offense	Sum of *IF8* of all offensive players
AF9	Space Control Defense	Sum of *IF8* of all defensive players

no dropping at all (resulting in a fully connected graph) to almost all edges being dropped are evaluated for test accuracy as well as training and testing times.

- **Convolution Complexity** The complexity of the convolutional block is mainly determined by the number of GCN layers and their size in terms of hidden neurons. To determine the optimal complexity of this block, both parameters were increased simultaneously, resulting in 6 levels of complexity ranging from one to six layer and 64 to 2048 neurons, respectively. All six configurations were evaluated regarding size, performance and training time.
- **Recurrent block** Once the pre-processing and convolution block are optimized, we add a recurrent block consisting of either a vanilla Recurrent Neural Network (RNN) or a GRU. In comparison to the first ablation steps, now all 15 frames of a given sample sequence are passed through the convolutional and successively recurrent block. The test accuracy of both modules is compared to a network without the recurrent block.
- **Number of features** Eventually, the number of features used for Tactical Graph construction is increased. This test evaluates if additional player properties increase learning performance as expected. Features are semantically grouped and added successively as follows:

  - Set 1: *IF3-IF4* (basic information, as in previous steps)
  - Set 2: *IF3-IF4, IF5-IF6* (adds player kinematic properties)
  - Set 3: *IF3-IF4, IF5-IF6, IF7-IF8* (adds individual tactical performance metrics)
  - Set 4: *IF3-IF4, IF5-IF6, IF7-IF8, IF9-IF13* (adds polar coordinates in various reference systems)
  - Set 5: *IF3-IF4, IF5-IF6, IF7-IF8, IF9-IF13, IF1-IF2* (adds raw positions)

  Notice that the last increment adds the raw player coordinates to the model and that they were omitted in all previous steps.

For all TGNet configurations, a ReLU activation function was used throughout the network except for the final output layer, where we used a Sigmoid function. Models were trained using an Adam optimizer [27] with an initial learning rate of $\alpha = .0003$. The learning rate was reduced by 30% after each epoch.

4.5 Experiment 2: state of the art comparison

After optimizing the TGNet architecture, we test the best model in a state of the art comparison. Unfortunately, no systematic comparison between models exists up to date [21]. Although some researchers have tested their model against simple baseline models (e.g. [57]), we are not aware of any performance evaluation between proposed advanced models on the same dataset. This is especially the case for models that perform on different data representations. The aim of this comparison is therefore to perform the first comparison between these different data representations on the same dataset. At the same time, we compare the optimized TGNet to evaluate if the benefits discussed in Section 3.3 lead to a better performance over models working with a different representation.

To this end, we implemented a naive baseline, four models that work on the data representations discussed in Section 2.1, and the best TGNet configuration. A logistic regression was used for classification based on aggregated expert features. We re-implemented the two most successful model types as used by Wagenaar et al. [57], a simple 3-layer CNN and a GoogLeNet [54] without pre-training. As both models are based on image representations, each sample was translated to a RGB image following the enhanced format presented by Wagenaar et al. [57]. See also Fig. 6 for examples of positive and negative samples. We also implemented a GRU operating on state vectors. All models are discussed in detail below.

- **Naive** A simple baseline approach to test the dependency between sample labels and ball location. This model principally follows the rule that whoever is closest to the ball at the end of the sequence will also be in possession at the time of evaluation, i.e., 0.5 seconds later. Therefore, the difference between both teams closest ball distance (*AF1-AF2*) is calculated. An accuracy-optimized decision threshold is then derived on the training set. Test cases are subsequently classified based on this decision threshold.
- **LogReg** A standard logistic regression is fitted on all aggregated features *AF1 - AF9* from the training set and subsequently evaluated on the test set.
- **SVGRU** A standard General Rectified Unit for state vector processing followed by a fully connected layer for classification. State vectors V_t^{sv} were constructed by joining all player's positions ($x_{t,i} \widehat{=}$ *IF1*, $y_{t,i} \widehat{=}$ *IF2* for $i = 1, ..., N$ and $N = 22$), $M = 11$ individual features, i.e., $a_{t,i}^{(1)} \widehat{=}$ *IF3*, ... $a_{t,i}^{(11)} \widehat{=}$ *IF13*, as well as the ball position $(b_{t,1}, b_{t,2})$. No contextual features were used in this study, resulting in state vectors of length $2N + 2 + NM = 288$ for a given t. The GRU was correspondingly constructed with 288 input, hidden, and output units, respectively. All $T = 15$ state vectors of a sequence were successively fed to the GRU followed by a *Rectified Linear Unit* (ReLU, [42]) activation function. The final GRU output was then

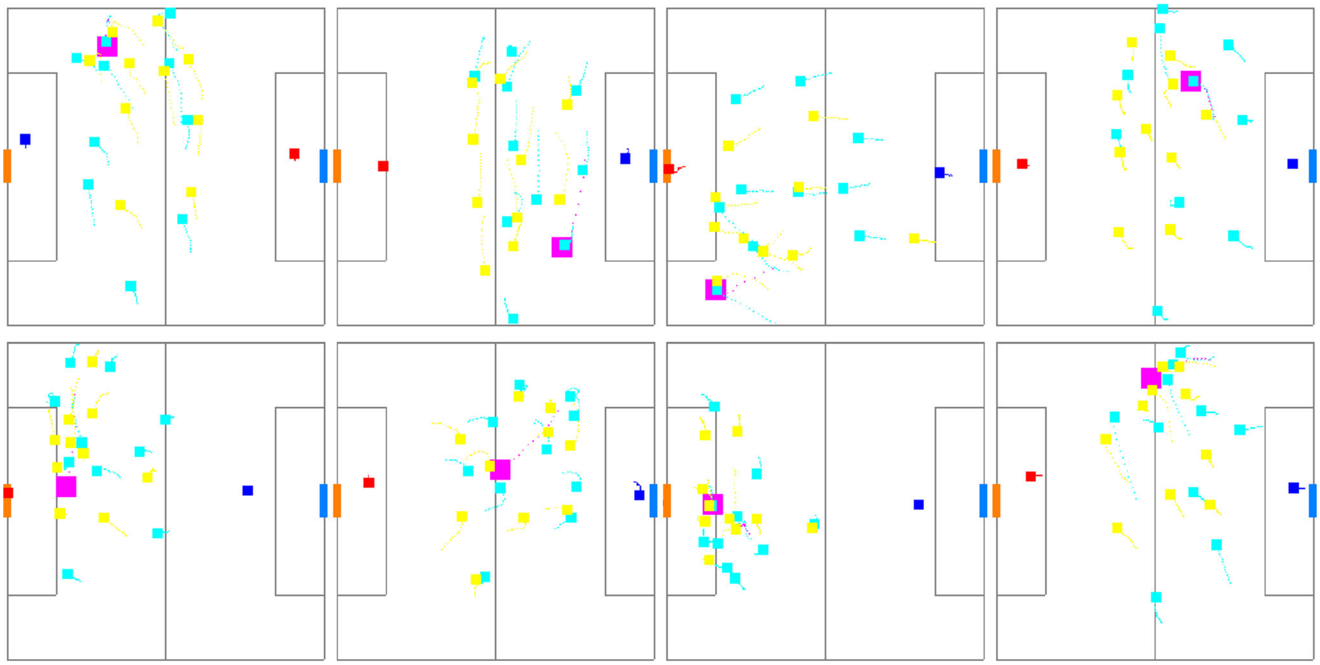


Fig. 6 Sample images as designed by Wagenaar et al. [57]. The top and bottom row display four randomly selected negative and positive samples, respectively. Colors were inverted for printer-friendly display

used for classification. The model was trained using an Adam optimizer, and a learning rate $\alpha = .00005$ that was reduced by a factor of .3 after each epoch.

- **CNN** A simple Convolutional Neural Network with three layers as constructed by Wagenaar et al. [57]. The convolutional layers consist of 48, 128, and 192 kernel with a size of 7×7, 5×5, and 3×3, respectively. All convolutional layer were followed by a ReLU activation and a max pooling layer of size 3×3 and a stride of 2. Trained with images identically to Wagenaar et al. [57], i.e., using a Nesterov optimizer with a learning rate $\alpha = .00005$, momentum of .9 and weight decay of $\lambda = .0005$. The number of epochs was increased from 2 to 10 compared to the original study.
- **GoogLeNet** A GoogLeNet [54] without pre-training as used by Wagenaar et al. [57]. Trained with images identically to Wagenaar et al. [57], i.e., using a Nesterov optimizer with a learning rate $\alpha = .001$, momentum of .9 and weight decay of $\lambda = .0005$. The number of epochs was increased from 8 to 10 compared to the original study.
- **TGNet** The TGNet used for the state of the art comparison was configured and trained in accordance with the ablation study results and all blocks and units as depicted in Fig. 5. More precisely, 11 individual features were used for constructing Tactical Graphs (Set 4 from Section 4.4). For sparsification, $\epsilon = 10$ was used. The convolutional block was constructed with two GCN layer each consisting of 128 filter parameters per input feature. A GRU was used for processing of the max-pooled embeddings. ReLU was used as the activation function throughout this block. All 15 frames of a sequence were treated as mini-batches (cf. [12]). Training used an Adam optimizer with a learning rate of $\alpha = .0003$ which was reduced by a factor of .3 after each of the 10 epochs.

4.6 Results

Model performance was measured on the test set and common performance metrics for binary classification tasks evaluated. Further, wall-clock times (measured in seconds) were recorded for training procedures (t_{train}) and inferences (t_{test}). The results from the ablation study are summarized in Section 4.6.1, and the state of the art comparison is presented in Section 4.6.2.

4.6.1 Ablation study

All four steps of the ablation were performed once on the described train, validation and test data sets. The examined configurations and, respectively, the chosen hyperparameter were investigated and optimized for prediction accuracy on the test set, with a secondary focus on training and inference times, as well as the number of model parameters.

The evaluation of the included sparsification is summarized in Table 3. The results show that decreasing the threshold distance ϵ increases model performance successively until an optimal value of $10\,m$ in terms of prediction accuracy. Further decreasing the threshold does not lead to better results, which conforms to an intuitive judgement regarding the importance of player relations with respect to their distance. Decreasing the threshold also leads to a drop in training and inference times. The faster computation times can be directly linked to the increased sparsity of the weight adjaceny matrix used for constructing the graph Laplacian (Fig. 7). For the subsequent steps a sparsification with $\epsilon = 10$ is chosen, which leads to an increase of 5.3 percentage points in accuracy with around 2.5 times faster training.

The evaluation of the convolution block complexity is summarized in Table 4. The results show that the number of model parameter and correspondingly training times increase together with model complexity. Model performance, however, is optimal for models containing two, three, or four layers. In light of these results, the model configuration consisting of two layer with 128 learnable filter parameters per input feature is chosen as it provides the optimal balance between the investigated evaluation metrics.

The evaluation of using various recurrent blocks is summarized in Table 5. Although adding a recurrent layer to the model increases both training and inference times, prediction accuracy on the test set also increases noticeably. Given that the difference in inference time is neglectable, the GRU configuration was chosen to integrated to the model for subsequent testing.

The evaluation of increasing the number of individual player properties used for constructing the node feature matrix during graph construction is summarized in Table 6. For the first four sets, accuracy and wall-clock times increase as more features are added. After that, accuracy stagnates. It should be noted that the first four sets do not contain any raw position data during graph construction, and that adding these positions does not increase performance in a model that already contains eleven features. This configuration with Set 4 of features is therefore selected for the state of the art comparison.

Table 3 Results for the first step of the ablation study testing the influence of different threshold distances ϵ on test prediction accuracy and time needed for training and inference (in seconds)

ϵ	Accuracy	t_{train}	t_{test}
None	.663	2,149	23.39
50	.708	1,978	22.34
30	.697	1,388	17.42
25	.691	1,235	15.75
20	.703	1,054	14.37
15	.705	917	13.44
10	**.718**	860	13.23
5	.711	831	12.81
2	.700	**827**	**12.44**

The best result for each category is presented in bold

4.6.2 State of the art comparison

The results of the state of the art comparison are summarized in Table 7. The naive baseline and logistic regression were run a single time on the described training, validation, and test set. All trainable models (SVGRU, CNN, GoogLeNet, and TGNet) were run five times on the same sets and performance metrics were averaged across all runs. Their average classification accuracy on the validation set and training loss during the five runs is summarized in Fig. 8.

The simple baseline model achieved a test accuracy of 74.5 % which gives a good estimation of the relation between sample label and player-to-ball distances. Thus, a solid prediction can be made by classifying samples based on the team affiliation of the closest player to the ball half a second before a potential turnover. It can be expected, however, that any classification performance above this baseline level can be attributed to the model taking a prediction based off more elaborate structures within the data. The logistic regression, fitted with additional expert-based features, is only marginally better than this baseline.

The SVGRU model operating on a state vector representation failed to meet the baseline, although composed of the second most parameters. This indicates that the model is inappropriate for the present learning task. The two image-based Neural Networks differ drastically in terms of performance, even to a greater extent than they did in the original study conducted by Wagenaar et al. [57]. In our experiment, the far more complex GoogLeNet achieves strong results above the baseline, wheres the CNN shows the worst performance among all tested models. The best results in terms of classification performance are achieved by the GoogLeNet and the proposed TGNet. Both models perform on par regarding the used performance metrics. It is noticeable, however, that the used TGNet is of significantly lower complexity than the voluminous deep learning model. It achieves the best prediction accuracy and clearly outperforms the baseline with only about one hundredth of the parameters used for the GoogLeNet model, and more than 15 times faster.

In terms of training, all models showed convergent behavior within ten epochs (Fig. 8). Although the loss curve of the GoogLeNet model still shows a slight downward trend after ten epochs, validation accuracy stagnated over

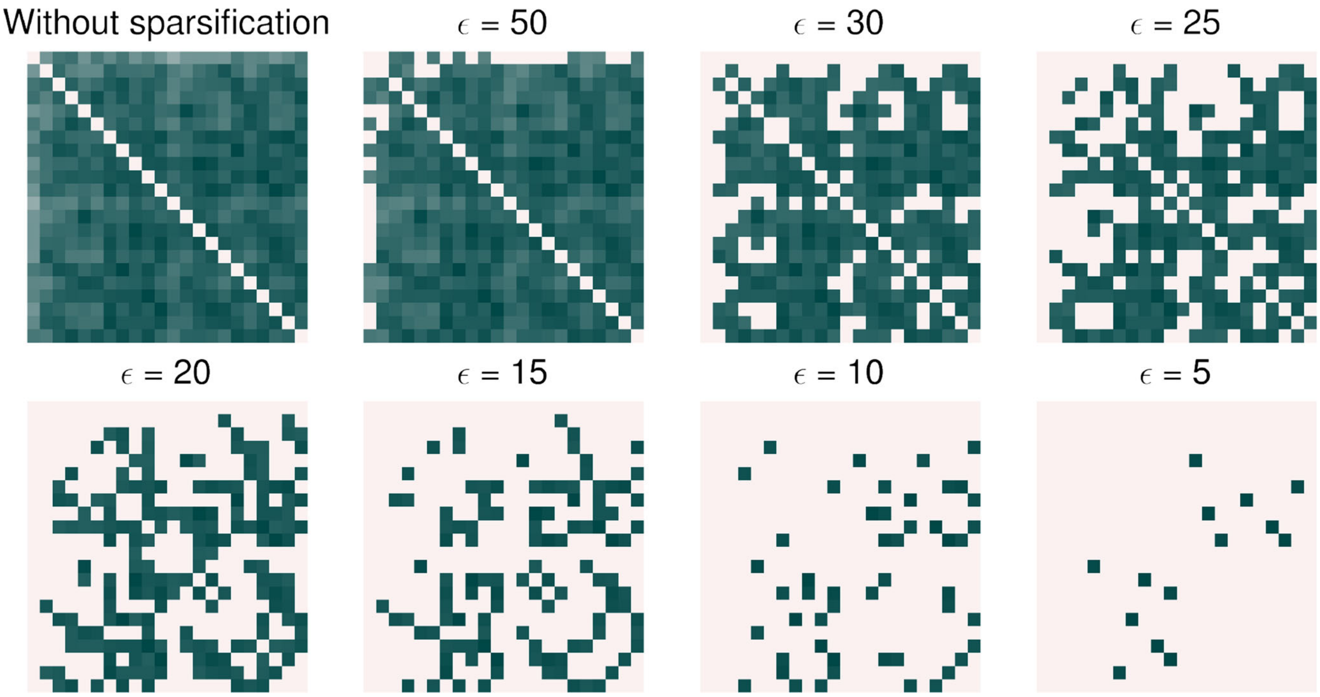


Fig. 7 Sparsity of adjacency matrices **W** after sparsification with different threshold distances. Matrix values are colored in petrol with respect to their magnitude, zero entries are colored in light pink

the last three epochs. The trends also show that a faster training is possible for the TGNet model which exhibits strong prediction accuracies on the validation set even after the first epoch.

5 Discussion and conclusion

As demonstrated by the experiments, the performance of models based on different representations of spatiotemporal sports data differ drastically when applied on the same dataset. As no such state of the art comparison was previously conducted, this is an important result that should increase the awareness of the methodological implications due to design choices during model construction. Previous research has begun to identify some of the challenges inherent to analyzing player position data. Yet, no comprehensive collection and exploitation of domain-specific data traits and requirements has been performed. In light of the considerations of Section 3.3, it seems highly likely that this lack has hindered previous applications from exploiting the full potential of machine learning algorithms when analyzing multi-agent tracking data.

Upon closer inspection, the detailed results give a first indication of the benefits and limitations of the different data representations and models. The logistic regression failed to improve the simple baseline although a range of additional expert-based features were used for the predictions. This result suggests that the temporal aggregation of features lead to ineffective regressors, potentially lacking the predictive power of a more fine-grained approach. In contrast, some trainable

Table 4 Results for the second step of the ablation study testing the influence of the convolution block size (number of layers and number of learned filter parameters per input feature) on test prediction accuracy and time needed for training and inference (in seconds)

Layer	Filter	Parameter	Accuracy	t_{train}	t_{test}
1	64	262	.714	**855**	**12.50**
2	128	16,384	.755	1,205	16.11
3	256	132,102	**.757**	2,573	27.36
4	512	788,486	.754	9,308	79.83
5	1024	4,198,406	.737	25,315	190.50
6	2048	20,979,718	.500	90,333	394.00

The best result for each category is presented in bold

Table 5 Results for the third step of the ablation study testing the influence of introducing a recurrent block to the model on test prediction accuracy and time needed for training and inference (in seconds)

Module	Accuracy	t_{train}	t_{test}
None	.750	**1,254**	**16.83**
RNN	.773	1,666	18.02
GRU	**.791**	1,999	18.27

The best result for each category is presented in bold

Table 6 Results for the fourth and last step of the ablation study testing the influence of increasing the number of individual player features on test prediction accuracy and time needed for training and inference (in seconds)

Feature Set	Features	Accuracy	t_{train}	t_{test}
Set 1	2	75.3	**2,234**	**19.59**
Set 2	4	76.3	2,316	20.53
Set 3	6	78.7	2,401	20.94
Set 4	11	**.805**	2,502	21.20
Set 5	13	**.805**	2,593	21.72

The best result for each category is presented in bold

models which do not rely solely on hand-engineered features, performed considerably better. The improved performance supports the effectiveness of more complex and less interpretable algorithms. However, only a non-exhaustive selection of state-of-the-art expert features was tested in our experiment and further studies are needed in the future.

The SVGRU, on the other hand, did not even achieve baseline results. Although not as extensively ablated as the remaining trainable models, this can nevertheless be partially attributed to the used data representation. The model contains a complex, state-of-the-art GRU component which has been shown to effectively find temporal structures existing in the data. However, the SVGRU purposely lacks any form of data augmentation or dimensionality reduction such as a convolution block for feature derivation. Previous studies have demonstrated that state vectors suffer from multiple problems such as a lack of invariances as discussed in Section 3.3. Thus, the results indicate that failing to compensate for these shortcomings results in poor overall model performance. At best, the missing performance related to inherent problems of the underlying data structure can be compensated for by increasing model complexity or sample size requirements.

The plain CNN model showed the worst overall performance, whereas the much more complex GoogLeNet operating on the same data representation achieved state-of-the-art performance. With almost ten million parameters, the GoogLeNet was by far the most complex and slowest of all tested models. As both models operate on images, they are usually optimized for image classification. It remains questionable if translating spatiotemporal data, and thereby largely reducing it to its spatial component, constitutes an optimal design choice. The results show, nevertheless, that an elaborated deep learning architecture can achieve strong results even when domain-specific properties of the underlying data are largely ignored. Comparing the huge difference in performance of both CNN and GoogLeNet, the question of how much of that difference can be attributed to the difference in model complexity remains.

The proposed TGNet architecture, on the other hand, was able to achieve better or similar results as the baseline and all other models based on Euclidean input data. This was achieved with the lowest complexity and the second-fastest inference times of all Neural Network type models tested. Especially with respect to the equally performing GoogLeNet, the difference in the number of used parameters is enormous. TGNet also shows the fastest convergence during training. Although this can partially be attributed to the number of parameters of each model, the fast convergence gives an indication about the ability of TGNets to quickly adapt to training data. As the TGNet operates on Tactical Graphs, the algorithm is completely invariant to permutation, translation, rotation and reflection and is able to exploit the compositionality and relationality of the input data. Exploitation of these properties can be seen as an essential property underlying the ability of the TGNet framework to compete with far more complex models. This is a crucial advantage given the domain-specific challenge of small available sample sizes.

A second insight from the conducted experiments concerns the number of used features. The TGNet framework, in contrast to image-based methods, allows for flexible feature engineering and feature integration. These features can be included depending on the specific task and with respect

Table 7 Results of the state of the art comparison between the different models and the corresponding data representation

Model	Data Repr.	Parameter	Acc.	Prec.	Rec.	F1	AUC	t_{test}
Naive	-	1	.745	.720	.804	.759	.829	1.63
LogReg	Features	10	.768	.742	.823	.780	.847	**0.11**
SVGRU	State Vector	499,970	.716	.694	.774	.732	.788	13.37
CNN	Image	437,506	.614	.621	.584	.602	.646	92.25
GoogLeNet	Image	9,936,038	.804	.772	**.862**	**.814**	.876	310.96
TGNet	Graph	109,212	**.805**	**.790**	.830	.810	**.879**	20.49

The performance metrics used are prediction accuracy on the test set (Acc.), precision (Prec.), recall (Rec.), the balanced F1-score (F1, calculated as the harmonic mean of precision and recall), the Area under the Curve (AUC) of the precision-recall-curve and inference times. For trainable models, metrics are averaged ofter a total of five runs. The best result for each category is presented in bold

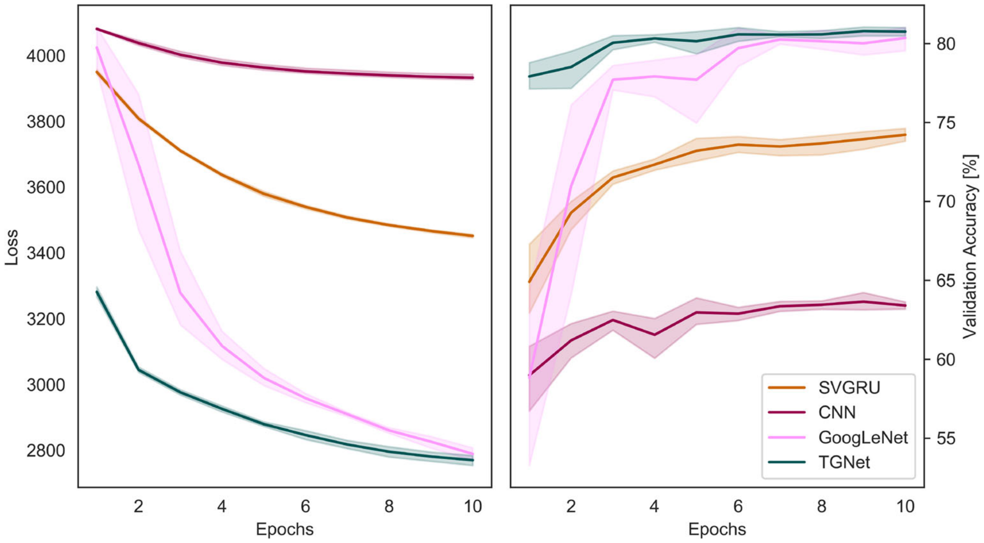


Fig. 8 Loss and validation accuracy curves for the four trainable models across ten epochs. Values are averaged over all five performed runs

to their ability to describe player characteristics, their interactions, or the global match context. This is a strong feature that helps to improve analyses by incorporating context-sensitivity more effectively. The ablation study showed, that inclusion of additional features from different areas such as player kinematics or tactical performance metrics, successively increase model performance in support of this view.

In summary, the present work sheds light on the relation between data representation and model performance when analyzing multi-agent spatiotemporal sports data. It also demonstrates the complexity of the matter and vastness of questions that still need to be answered. To fully leverage the potential of deep learning methodologies for this data source, we need to acknowledge, investigate and exploit the entangled relationship between the domain-specific data characteristics and the overall modeling process. These fundamental methodological topics are of great relevance, irrespective of the application area. Finding a suitable data representation as well as constructing powerful models is a uniting task in the areas of collective movement analysis, automated match analysis, and performance analysis.

The proposed Tactical Graphs and TGNets, although advantageous over current state of the art approaches, are just a single step in this direction. The comprehensive impact of data representations to model performance were not fully explored in this first state of the art comparison. The model itself, although tested in an initial ablation study, can still be improved. So far, spatial and temporal structures within the data are analyzed in two subsequent instead of a joint step. Furthermore, no suitable pooling module was found that could increase the effectiveness of graph convolutions. Model refinement thus remains one of the next steps to be taken. In more general terms, however, all future work needs to pay more attention towards domain-specific requirements, data representation and model construction as well as extensive baseline testing.

Acknowledgements The authors would like to thank Eric Müller-Budack, Robert Rein, Henrik Biermann, and Lana Blaschke most sincerely for their support during preparation of this work and comments on the manuscript.

Author Contributions Conceptualization: D.R.; Methodology: D.R., R.N.; Formal analysis and investigation: D.R.; Writing - original draft preparation: D.R.; Writing - review and editing: D.R.; Funding acquisition: D.M.; Resources: D.R., D.M.; Supervision: R.N., D.M.

Funding Open Access funding enabled and organized by Projekt DEAL. This project has received funding from the German Federal Ministry of Education and Research (BMBF – Bundesministerium für Bildung und Forschung) under 01IS20021A-C.

Declarations

Conflict of Interests The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Battaglia P, Pascanu R, Lai M et al (2016) Interaction networks for learning about objects, relations and physics. In: Advances in neural information processing systems (NeurIPS), pp 4509–4517
2. Battaglia PW, Hamrick JB, Bapst V et al (2018) Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261
3. Bialkowski A, Lucey P, Carr P et al (2014) Large-scale analysis of soccer matches using spatiotemporal tracking data. In: IEEE international conference on data mining (ICDM), pp 725–730. https://doi.org/10.1109/ICDM.2014.133
4. Bourbousson J, Sève C, McGarry T (2010) Space-time coordination dynamics in basketball: Part 2. The interaction between the two teams. J Sports Sci 28(3):349–358. https://doi.org/10.1080/02640410903503640
5. Bronstein MM, Bruna J, Lecun Y et al (2017) Geometric deep learning: going beyond euclidean data. IEEE Signal Process Mag 34(4):18–42. https://doi.org/10.1109/MSP.2017.2693418
6. Castellano J, Alvarez-Pastor D, Bradley PS (2014) Evaluation of research using computerised tracking systems (amisco® and prozone®) to analyse physical performance in elite soccer: A systematic review. Sports Med 44(5):701–712. https://doi.org/10.1007/s40279-014-0144-3
7. Cho K, van Merrienboer B, Gulcehre C et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078
8. Dick U, Brefeld U (2019) Learning to rate player positioning in soccer. Big Data 7(1):71–82. https://doi.org/10.1089/big.2018.0054
9. Dornaika F (2021) Flexible data representation with feature convolution for semi-supervised learning. Appl Intell. https://doi.org/10.1007/s10489-021-02210-y
10. Dutt-Mazumder A, Button C, Robins A et al (2011) Neural network modelling and dynamical system theory: Are they relevant to study the governing dynamics of association football players? Sports Med 41(12):1003–1017. https://doi.org/10.2165/11593950-000000000-00000
11. Feuerhake U (2016) Recognition of repetitive movement patterns–the case of football analysis. ISPRS Int J Geo-Inf 5(11):208–208. https://doi.org/10.3390/ijgi5110208
12. Fey M, Lenssen JE (2019) Fast graph representation learning with PyTorch geometric. arXiv:1903.02428
13. Frencken W, de Poel H, Visscher C et al (2012) Variability of inter-team distances associated with match events in elite-standard soccer. J Sports Sci 30(12):1207–1213. https://doi.org/10.1080/02640414.2012.703783
14. Fujii K (2021) Data-driven analysis for understanding team sports behaviors. J Robot Mechatron 33(3):505–514. https://doi.org/10.20965/jrm.2021.p0505
15. Gilmer J, Schoenholz SS, Riley PF et al (2017) Neural message passing for quantum chemistry. In: International conference on machine learning (ICML), pp 2053–2070
16. Glazier PS (2010) Game, set and match? Substantive issues and future directions in performance analysis. Sports Med 40(8):625–634. https://doi.org/10.2165/11534970-000000000-00000
17. Goes FR, Meerhoff LA, Bueno MJ et al (2021) Unlocking the potential of big data to support tactical performance analysis in professional soccer: a systematic review. Eur J Sport Sci 21(4):481–496. https://doi.org/10.1080/17461391.2020.1747552
18. Gori M, Monfardini G, Scarselli F (2005) A new model for learning in graph domains. In: Proceedings of the international joint conference on neural networks, pp 729–734. https://doi.org/10.1109/IJCNN.2005.1555942
19. Grunz A, Memmert D, Perl J (2012) Tactical pattern recognition in soccer games by means of special self-organizing maps. Hum Mov Sci 31(2):334–343. https://doi.org/10.1016/j.humov.2011.02.008
20. Hagberg AA, Schult DA, Swart PJ (2008) Exploring network structure, dynamics, and function using NetworkX. In: Python in science conference (SciPy), pp 11–15
21. Herold M, Goes F, Nopp S et al (2019) Machine learning in men's professional football: Current applications and future directions for improving attacking play. Int J Sports Sci Coach 14(6):798–817. https://doi.org/10.1177/1747954119879350
22. Horton M, Gudmundsson J, Chawla S et al (2017) Classification of passes in football matches using spatiotemporal data. ACM Trans Spatial Algorithms Syst 3(2):1–30. https://doi.org/10.1145/3105576
23. Intille SS, Bobick AF (2001) Recognizing planned, multiperson action. Comput Vis Image Underst 81(3):414–445. https://doi.org/10.1006/cviu.2000.0896
24. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning (ICML), pp 448–456
25. Ivanovic B, Pavone M (2019) The Trajectron: Probabilistic multiagent trajectory modeling with dynamic spatiotemporal graphs. In: IEEE/CVF international conference on computer vision (ICCV)
26. Kays R, Crofoot MC, Jetz W et al (2015) Terrestrial animal tracking as an eye on life and planet. Science, 348(6240). https://doi.org/10.1126/science.aaa2478
27. Kingma DP, Ba J (2017) Adam: A method for stochastic optimization. arXiv:1412.6980
28. Kipf T, Fetaya E, Wang KC et al (2018) Neural relational inference for Interacting systems. In: International conference on machine learning (ICML), pp 4209–4225
29. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International conference on learning representations (ICLR
30. Klicpera J, Weißenberger S, Günnemann S (2019) Diffusion improves graph learning. In: Advances in neural information processing systems (NeurIPS). https://proceedings.neurips.cc/paper/2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf
31. Knauf K, Memmert D, Brefeld U (2016) Spatio-temporal convolution kernels. Mach Learn 102(2):247–273. https://doi.org/10.1007/s10994-015-5520-1
32. Le HM, Yue Y, Carr P et al (2017) Coordinated multi-agent imitation learning. In: International conference on machine learning (ICML), pp 1995–2003
33. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
34. Linke D, Link D, Lames M (2020) Football-specific validity of TRACAB's optical video tracking systems. PLoS ONE 15(3):e0230,179. https://doi.org/10.1371/journal.pone.0230179
35. Low B, Coutinho D, Gonçalves B et al (2020) A systematic review of collective tactical behaviours in football using positional data. Sports Med 50(2):343–385. https://doi.org/10.1007/s40279-019-01194-7
36. Low B, Rein R, Raabe D et al (2021) The porous high-press? An experimental approach investigating tactical behaviours from two pressing strategies in football. J Sports Sci. https://doi.org/10.1080/02640414.2021.1925424
37. Lucey P, Bialkowski A, Carr P et al (2013) Representing and discovering adversarial team behaviors using player roles. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2706–2713

38. Lutz J, Memmert D, Raabe D et al (2020) Wearables for integrative performance and tactic analyses: Opportunities, challenges, and future directions. Int J Environ Res Public Health 17(1):1–26. https://doi.org/10.3390/ijerph17010059
39. Memmert D, Raabe D (2018) Data analytics in football. Routledge, London. https://doi.org/10.4324/9781351210164
40. Memmert D, Lemmink KA, Sampaio J (2017) Current approaches to tactical performance analyses in soccer using position data. Sports Med 47:1–10. https://doi.org/10.1007/s40279-016-0562-5
41. Miller HJ, Dodge S, Miller J et al (2019) Towards an integrated science of movement: Converging research on animal movement ecology and human mobility science. Int J Geogr Inf Sci 33(5):855–876. https://doi.org/10.1080/13658816.2018.1564317
42. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: International conference on machine learning (ICML), ICML'10, pp 807–814
43. Passos P, Davids K, Araújo D et al (2011) Networks as a novel tool for studying team ball sports as complex social systems. J Sci Med Sport 14(2):170–176. https://doi.org/10.1016/j.jsams.2010.10.459
44. Paszke A, Gross S, Massa F et al (2019) PyTorch: An imperative style high-performance deep learning library. In: Advances in neural information processing systems (NeurIPS)
45. Perl J (2004) A neural network approach to movement pattern analysis. Hum Mov Sci 23(5):605–620. https://doi.org/10.1016/j.humov.2004.10.010
46. Praça GM, Lima BB, Bredt SDGT et al (2019) Influence of match status on players' prominence and teams' network properties during 2018 FIFA World Cup. Front Psychol 10:695. https://doi.org/10.3389/fpsyg.2019.00695
47. Rein R, Memmert D (2016) Big data and tactical analysis in elite soccer: Future challenges and opportunities for sports science. SpringerPlus 5(1):1410–1410. https://doi.org/10.1186/s40064-016-3108-2
48. Rein R, Raabe D, Memmert D (2017) Which pass is better? Novel approaches to assess passing effectiveness in elite soccer. Hum Mov Sci 55:172–181. https://doi.org/10.1016/j.humov.2017.07.010
49. Sampaio J, Macas V (2012) Measuring tactical behaviour in football. Int J Sports Med 33(5):395–401. https://doi.org/10.1055/s-0031-1301320
50. Sarmento H, Marcelino R, Anguera MT et al (2014) Match analysis in football: A systematic review. J Sports Sci 32(20):1831–1843. https://doi.org/10.1080/02640414.2014.898852
51. Scarselli F, Gori M, Tsoi AC et al (2009) The graph neural network model. IEEE Trans Neural Netw Learn Syst 20(1):61–80. https://doi.org/10.1109/TNN.2008.2005605
52. Shuman DI, Narang SK, Frossard P et al (2013) The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process Mag 30(3):83–98. https://doi.org/10.1109/MSP.2012.2235192
53. Silva P, Duarte R, Sampaio J et al (2014) Field dimension and skill level constrain team tactical behaviours in small-sided and conditioned games in football. J Sports Sci 32(20):1888–1896. https://doi.org/10.1080/02640414.2014.961950
54. Szegedy C, Liu W, Jia Y et al (2014) Going deeper with convolutions. arXiv:1409.4842
55. Terroso-Sáenz F, Muñoz A (2021) Nation-wide human mobility prediction based on graph neural networks. Appl Intell. https://doi.org/10.1007/s10489-021-02645-3
56. Von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416. https://doi.org/10.1007/s11222-007-9033-z
57. Wagenaar M, Okafor E, Frencken W et al (2017) Using deep convolutional neural networks to predict goal-scoring opportunities in soccer. In: International conference on pattern recognition applications and methods. https://doi.org/10.5220/0006194804480455
58. Wang S, Cao J, Yu P (2020) Deep learning for spatio-temporal data mining: Survey. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3025580
59. Xu C, Zhang A, Xu C et al (2021) Traffic speed prediction: Spatiotemporal convolution network based on long-term, short-term and spatial features. Appl Intell. https://doi.org/10.1007/s10489-021-02461-9
60. Yeh RA, Schwing AG, Huang J et al (2019) Diverse generation for multi-agent sports games. In: IEEE/CVF Conference on computer vision and pattern recognition, pp 4610–4619

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dominik Raabe received his master's degree in scientific computing from the Technical University Berlin, Germany, in 2020. He is currently a PhD student and research associate at the Institute of Exercise Training and Sport Informatics, German Sport University Cologne. His research interests include match and performance analysis, collective movement behavior, and geometric deep learning.

Daniel Memmert is a Professor and Executive Head of the Institute of Exercise Training and Sport Informatics at the German Sport University Cologne. In 2014, he was a visiting professor at the University of Vienna, Austria. His scientific work focuses on movement science (cognition and motor function), sports psychology (attention and motivation), and computer science in sports (Big Data, pattern recognition and simulation). His institute cooperates with various Bundesliga soccer teams, the German national soccer team and DAX companies and organizes the first international master's degree course in "Match Analysis".