ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/359079429

Detection of tactical patterns using semi-supervised graph neural networks

Conference Paper · March 2022

CITATIONS
3

READS
1,478

4 authors:

Gabriel Anzer
University of Tuebingen
14 PUBLICATIONS **95** CITATIONS
SEE PROFILE

Pascal Bauer
University of Tuebingen
16 PUBLICATIONS **147** CITATIONS
SEE PROFILE

Ulf Brefeld
Leuphana University Lüneburg
122 PUBLICATIONS **2,579** CITATIONS
SEE PROFILE

Dennis Faßmeyer
Leuphana University Lüneburg
4 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Machine Learning for Tactical Match Analysis in Football (Soccer) View project

Project Dependable AI - SafeAI View project

All content following this page was uploaded by Pascal Bauer on 08 March 2022.

The user has requested enhancement of the downloaded file.

MIT SLOAN SPORTS ANALYTICS CONFERENCE presented by ESPN 42 Analytics
MARCH 4-5, 2022

Detection of tactical patterns using semi-supervised graph neural networks

Gabriel Anzer[1,2], Pascal Bauer[2,3], Ulf Brefeld[4], Dennis Fassmeyer[4]
1 Hertha BSC Berlin | 2 Tübingen University | 3 DFB-Akademie | 4 Leuphana University Lüneburg

1. Introduction

In a fluent invasion sport like soccer, tactical analysis typically breaks a match down to distinct game-phases in which teams perform tactical patterns. A tactical pattern is defined as repeatable, coordinated movements of either the whole team or a group of players and conducted in specific situations [1],[2],[3]. To derive meaningful insights in the tactics of a team, these patterns are identified and annotated manually by experts while observing the video footage [4], [5]. Although video analysis departments are a well-established part of professional soccer and specialists in tagging such patterns, this task is often repetitive, time-consuming, and subjective. Thus, with tracking and play-by-play data being available at large scales, the automated detection of tactical patterns has become one of the most important jobs in (team) sports [6],[7]. Successful applications include pattern detection for goal scoring [8], counterpressing [4], counterattacks [9], and defending corner kicks [10],[11]. Their implementation in existing workflows have been shown to save video analysts a great deal of time by (semi-)automating otherwise purely manual processes.

Detecting tactical patterns is not available off-the-shelf but bears three inherent challenges that need to be addressed appropriately to obtain a method that can be used in practice: (I) The detection task is usually considered a supervised learning problem that relies on a great deal of manually annotated situations so that a classifier can be trained to detect patterns of interest. The data used for the training is assembled by forming pairs of situations (the input) and corresponding manual labels (the output) where the classifier learns to act as a function approximator that transforms situations into labels. (II) The second challenge is the permutation problem that is inherent to all multi-agent scenarios like team sports. An appropriate solution needs to be agnostic about the ordering of the players to account for different lineups, in-game role changes and substitutions. Hence, the pattern detector must be in- or equivariant with respect to the ordering of players. Unfortunately, existing approaches often oversimplify the problem by breaking the high-dimensional spatiotemporal input data down to lower dimensional hand-crafted features [4], [8], [11]. (III) Last but not least, the number of players involved in a pattern is often unknown beforehand and may vary from pattern to pattern. This holds particularly for applications in opponent analysis where the goal is to detect novel and so far, unknown patterns. Apart from computing individual scores for players [12], there is, as of now, no convincing solution to compute the subgroup of players that is important for a given situation.[1]

Thus, an appropriate solution to the detection of tactical patterns should (I) require only little manual effort, (II) be in- or equivariant with respect to the ordering of players, and (III) identify relevant subgroups of players automatically. Fortunately, these requirements are not unique to sports analytics and have been studied in other contexts before. For example, semi-supervised

[1] Mathematically, the power set $\pi(X)$ contains all possible subsets of a set X and is of size $2^{|X|}$. While, in basketball computing $2^{10} = 1{,}024$ different subsets per timestep may perhaps be tractable, we have $2^{22} = 4{,}194{,}304$ different subsets of players per frame in soccer which is clearly infeasible.

learning deals with scenarios where labeled instances are scarce but unlabeled ones can be accessed in abundance. Equivariance and subgroups of agents can be captured by graph neural networks (GNNs) [13] that have also been shown to work well in spatiotemporal domains like invasion sports [14], [15].

Putting everything together, we propose a semi-supervised and sequential variational autoencoder (VA) for detecting tactical patterns on team-, group- and player-levels. Internally, a graph neural network remedies permutation problems and allows to focus the algorithms attention on arbitrary subgroups of players, including opponents. Our contribution can thus be seen as a sequential and order equivariant generalization of semi-supervised variational autoencoders [16], [17], [18], or, alternatively, as a semi-supervised extension of existing approaches for detecting patters in trajectory-based data [19].

Our framework allows to address a variety of problems relevant in practice, ranging from the detection of patterns on team-, group- and player-levels like, for example, overlapping runs. Recently, Fassmeyer et. al. [17] proposed a similar approach. Though their approach is tailored to the detection of team-patterns, they showed its usefulness for the detection of counterattacks [9]. In this study, we are the first to explore arbitrary tactical patterns, involving small groups of players. An example is *overlapping runs*[2], a pattern conducted by only two players of a team, as well as patterns involving larger groups of players on the example of *chances without a shot*.[3] The latter constitutes a complex tactical event with varying numbers of players involved and serves to demonstrate the universal applicability of the proposed approach to arbitrary tactical patterns. In both tasks, the proposed method outperforms purely supervised competitors. The superiority of the proposed methodology is further highlighted by another set of experiments focusing on only the movement model of the proposed approach. Though not explicitly designed for movement prediction, the latter certainly constitutes an important part in the detection of tactical patterns with multiple moving players at a time. We report on a side experiment in the Appendix, showing that the movement prediction of our method significantly advances the state-of-the-art on an open-source Basketball dataset.

We conclude the paper by presenting useful practical applications, again using the detection of *overlapping runs* as a showcase. Our method not only saves a great deal of time for video analysts by finding the relevant scenes of a match automatically, but also enables coaching staffs to derive long-term insights by processing data at large scales to support opponent analysis as well as scouting talents.

[2] Two examples of *overlapping runs* can be found here: https://bit.ly/3o2zIQ5. Both the overlapping—the player running behind the player in ball possession—and the overlapped player—the player in possession of the ball—are highlighted in the video.
[3] *Chance without shot* events are defined as situations with a significant chance to score a goal without a shot actually being taken or without an owngoal being scored. A video of an exemplary chance without shot event can be found here: https://bit.ly/3E4LrTD.

2. Section

2.1. Data and Expert-Labeling

For the purpose of this study, we make use of in total 14 matches of tracking and event data of the German National team played in 2021, including their four matches played at the European Championship. Tracking data are acquired using the Chyronhego TRACAB System (Generation 5) evaluated in [20] and are sampled at 25 frames per second. The respective event data, acquired by Sportec Solutions AG (following the official event data catalogue of German Bundesliga Match-Data) are synchronized with the positional data using the methodology of Anzer et. al [21]. We will focus on tracking data in this paper and make use of event data only in the evaluation. More information on both, positional and event data can be found in [4], [8], [17], [21].

In the following, we define an *overlapping run* as a separated two (attackers) versus one (defender) situation with one attacker being in possession of the ball and dribbling toward the defender. The second attacker, performing the overlapping run, runs with high speed past the ball possessing attacker toward the opposing goal line. By doing so, the runner creates a new passing option for his teammate. These situations are difficult to defend since the running player has no direct opponent and the opposing defender must choose between defending the runner or the dribbling player.

According to that definition, a professional match analyst from the German National team hand-labeled all overlapping run situations (in total 32) of four matches of the German national team played in 2021. For all situations both the overlapping (the player running past his teammate) and the overlapped player (the ball possessing player) is annotated by the expert, however, only the former is used as input to the algorithm. Two of those matches are independently labeled by a second analyst from a Bundesliga club to determine the inter-annotator agreement.

2.2. Preprocessing

An architecture based on graph neural networks (GNNs) allows us to directly operate on tracking data as input. We aim to predict target variables that originate either from the available event data (*chance without shot*) or from external annotations (*overlapping run,* expert labels). While the event data uses the same time code as the tracking data, the time format for the expert annotations uses a video footage clock and needs to be synchronized with the tracking data. Thus, the latter case includes a preprocessing step to align the manually provided timestamps to their corresponding frames in the tracking data. Since the coarse timing information and the subsequent transformation process render the label information ambiguous, it is important to include a short context window around the identified frame. Since the expert annotations denote the start of an overlapping situation, we additionally annotate the next 10 frames as well with the expert label.

The tracking data is given as a long consecutive sequence. By contrast, our models operate on game segments of different lengths and we need to split the game at hand into smaller chunks. Also, the model objective consists of a supervised and unsupervised part such that the proposed SequentialM2 (further described in the next section) requires simultaneous sampling of labeled and unlabeled data points. We simply generate unsupervised data by applying a sliding window of length 50 with 50% overlap on the full game data. For the supervised part, we extract “positive” training sequences by adding a contextual window around the timestep where a label of an arbitrary agent is active, and randomly sample from the remainder of the data to construct a "negative" dataset of the same size. The trajectories are centered and normalized to the interval [-1,1] both in x- and y-direction. Finally, the trajectory data is converted to velocity information by subtracting consecutive player coordinates. Note that this does not involve any loss of information with respect to player locations since the movement can be reconstructed from the x/y-coordinate of the first timestep and the subsequent velocity vectors.

2.3. Method

We employ strategies from the work of Kingma et al. [18], who propose a principled probabilistic approach to semi-supervised learning based on the idea of variation autoencoders [26], [27]. VAEs use neural networks to parametrize a generative model (or encoder) and an associated variational distribution (or decoder), and aim to reconstruct the given input data via a low-dimensional bottleneck z.

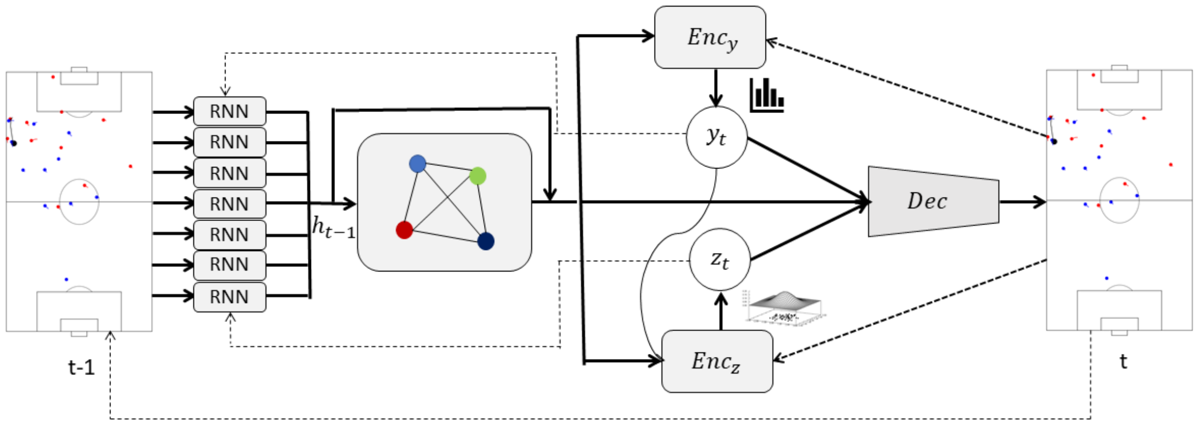

Figure 1: Each players' trajectory is fed through a recurrent neural network (RNN) with shared parameters and subsequently updated by a GNN.

To apply this setting to the semi-supervised domain, Kingma et al. [18] integrate discrete label information y into the data generation process (alongside continuous factors z). Their resulting objective function, referred to as M2 model, can be divided into a supervised and unsupervised part: The supervised part can be seen as a regularized classification objective, where the reconstruction task of the VAE acts as a regularization term. In the case of unlabeled data points, the label information is treated as an unobserved latent variable, so that inference also needs to be performed for the labels. This mechanism provides an effective way to learn from unobserved data as it encourages the model to assign high probability values to label values that achieve a small reconstruction loss. Thus, after training, the variational distribution on y can be used for classifying new data points.

Due to the sequential nature of spatiotemporal tracking data, it is vital to elevate the M2 principle to a sequential definition. A popular realization of generative and inference processes that extend VAEs to model complex sequential distributions is found in the VRNN framework proposed by Chung et al. [18]. Their approach can be seen as a VAE realization at each timestep, with each VAE conditioned on the hidden state of a recurrent neural network (RNN) at the previous timestep. We adopt this idea and combine it with the M2 formulation to derive a general semi-supervised approach for the sequential domain: Our framework, referred to as SequentialM2, can therefore be viewed as a combination of the two previously mentioned methods (VRNN + M2).

The overall layout of the proposed approach is depicted in **Figure 1**. To capture the time dependence of successive observations, we leverage the hidden state of an RNN network [24] that operates on input x and both latent variables (z and y). The RNN state at time t-1 encodes a selective summary of previous time steps and latent variables. The main idea is to use this state (h_{t-1}) as an additional functional input for all components of the encoding and decoding parts governed by the M2 assumption. Thus, the emerging training criterion can be interpreted as the M2 objective summed over each timestep of the input sequence.

2.4. Multi-Agent Setting

To generalize the derived framework to a multi-agent setting for applications in (team) sports, we integrate an attention-based graph neural network (GAT) [28], into the architecture. Such deep approaches are usually the model of choice when dealing with graph-structured data. Specifically, they operate by learning a chain of hidden representations for each node through an iterative process that relies on aggregating messages from interactive nodes. In fact, different proposed architectural variants such as GCN [26], GraphSAGE [27] or GATs [28] mainly differ in their notion of passing messages along the edges of the graph and aggregating neighborhoods. However, since the ordering of nodes in a graph structure is arbitrary, the aggregation operators are required to be permutation invariant (mean or sum are frequent choices).

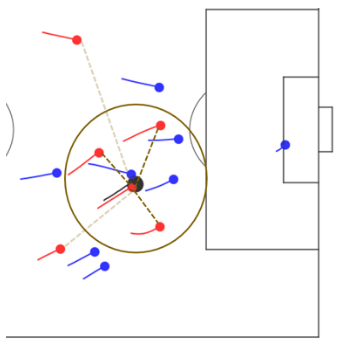

Figure 2: Euclidean distance between nodes (players). Only nearby players influence the computation of the node embeddings.

Graph attentive networks improve previous GNN configurations by computing node representations according to an attention strategy. An attention mechanism implicitly assigns weights to nodes in a neighborhood, thus governing the learning process towards the important pieces of data for the task at-hand. While there are various forms to realize such a mechanism, we use a simple single-layer neural network operating across pairs of nodes. To stabilize the learning process, we replicate this process multiple times with differing weight parametrizations, and subsequently concatenate their separately inferred embeddings, which defines a single GNN layer output (this process is referred to as multi-head attention, see for example [28]).

To embed the previously described concept into our overall architecture, we represent multi-agent trajectory data encoding each player as a node of the graph. This way, the model can utilize a GNN as a tool for information propagation, capable of modeling interaction patterns among different players. For predictions on player-levels, it is crucial to model their interaction with players nearby. Thus, we obviate direct influence of distant players in the computations by using a k-nearest neighbor approach (see Figure 2).[4] However, it is worth noting that eradicating remote players in the graph does not curtail the (implicit) aggregation of remote information. Specifically, by adding depth to the network (stacking GNN layers upon each other), the receptive field, i.e., the number of nodes that affect computations, is effectively increased. Additionally, we introduce a variant of skip connections as presented in Xu et al. [25]: the final feature representation is directly aggregated from all node embeddings of the previous layers. This mechanism adds to the expressiveness of the network and allows the model to accommodate structural information from different levels of granularity at prediction time.

To put everything together, at each timestep, the GNN updates recurrent player states by additionally considering neighborhood information, see **Figure 1**. Since the hidden states serve as input to all encoding and decoding modules, the computed quantities (i.e., latent variables and agent positions) depend indirectly on the respective preceding quantities.[5] This allows the model to properly account for coordination among the different players.

3. Empirical Setup

The experiments conducted in this paper are based on a dataset consisting of four distinct soccer games from the German national team. We use two (concatenated) games for training, one for model selection, and one game for testing. Each soccer game consists of roughly 140,000 timesteps, where each timestep encodes the x/y-coordinates of the players, plus a per-match-average of eight manually provided overlapping situations as well as three *chance without shot* situations. In the training process, parameters of the proposed methods are optimized and saved when the corresponding weight configuration improves the current best AUC value on the validation game. The reported results denote the quantities on the test game with the best parametrization after 100 epochs. All methods are implemented from scratch using PyTorch [29].

3.1. Learning Tasks:

The first task focuses on overlapping runs. An *overlapping run* is an established tactical pattern in soccer that involves a pair of two players: the ball carrier dribbles the ball (typically in the opposing half close to the sideline), a teammate runs past him at high speed, and creates a pass option towards the opposing goal.[6] Although the offensive pattern is rather simple, a well-timed overlapping run can pose a critical problem for defenders. In total 32 overlapping runs from four matches were tagged manually, including the information of the two players involved. Two matches were labeled independently of a second expert agreeing on 22 out of 26 scenes corresponding a pairwise inter-annotator agreement of 84.61 %. As a second task, we evaluate detecting individual

[4] Also, choosing a k-nearest neighbor adjacency matrix alleviates the oversmoothing problem where all node representations collapse to the same quantity.

[5] Note here that we opt for the most general formulation of achieving multi-agent coordination. However, under certain circumstances it may be useful to additionally employ GNN modules for the encoding and decoding building blocks.oppo

[6] A video with exemplary overlapping run situations can be found here: https://bit.ly/3o2zIQ5.

patterns using the *chance without shot* event.[7] The task is straight forward because these situations are already annotated in the event data. Nevertheless, the task allows us to study the effect of the number of provided labels and predictive accuracy.

To assess the detection performance, we mainly use two different performance metrics. Having described the overall goal under the viewpoint of extracting a candidate set of game situations to increase efficiency for game analysis, the area under the ROC curve (AUC) is a suitable metric as it aggregates true positive (TP) and false positive (FP) values for different model thresholds. While it is a reasonable performance metric in the present context, there are still two points to consider when stating AUC values. Firstly, the action labels of interest are given only for individual timestamps and not for the broad game sequence they refer to, a fact that also reflects on the computation of TP and FP values: A potential user of the system contemplates situations holistically and detecting an unlabeled event that is only a couple of frames off the actual annotation is unproblematic. Secondly, a highly unbalanced label distribution for the test set, as given here, may be overly optimistic, since good AUC values are frequently driven by the ease of achieving good FP-rates. To address these two issues, we also report the F_1-score on the test set for contiguous game sections. The components defining the F_1-score are computed in accordance with [17] and ground on probability estimates for frames in the test game. Since the F_1-score is basically the harmonic mean between precision and recall, false positives and false negatives are equally taken into consideration. The model outputs denote agent probabilities, consequently, we define the test game probabilities by choosing the maximum probability value across the agent dimension for each timestep of the game.

3.2. Baselines:

To properly assess the performance of the SequentialM2, we introduce a (fully supervised) sequential baseline model denoted as DetNet. The DetNet essentially adheres to an RNN's logic for classifying sequences. However, to account for spatiotemporal tracking data, the model additionally incorporates a GNN similarly to the SequentialM2 model, which processes RNN states to aggregate player dependencies. In this scenario, a soccer game is essentially treated as a long-supervised sequence where all time frames other than the situations of interest are labeled as negative. Accordingly, the model parameters are learned by randomly sampling game segments of length 50 from the fully annotated training games and optimizing a standard binary cross-entropy loss.

For the detection of overlapping runs, we also incorporate a rule-based detection, designed by professional match-analysts from the German National team. For all individual ball possessions (i.e., distance between player and ball < 2m) in the offensive third (i.e., more than 17.5m in the respective opposing half) and the outside lane (i.e. distance to the vertical midline of the pitch bigger than 17m), the rule-based system classifies situations where a teammate passed by the ball possessing player on the side of the closest sideline (i.e., there are two consecutive frames t and t+1 in which the overlapping player passed behind the ball possessing player in the vertical direction of the pitch with a speed higher than 10 km/h. To ensure comparability to the proposed deep architectures, we add a context of +/- two seconds to the detected timestamps, and compute F1-scores accordingly.

We report the quantitative results of both our model and the two baseline models in **Table 1**. The table shows that for both tasks, the detection of Chances without shot and for overlapping runs, the

[7] A video of an exemplary chance without shot event can be found here: https://bit.ly/3E4LrTD.

SequentialM2 model performs best in AUC as well as in F_1-Scores. The performance disparity to the (highly overfitting) purely supervised baseline highlights the importance of an effective regularization mechanism that operates solely on the surplus of unsupervised positional data. Also, this demonstrates the universal applicability of the proposed concept, which underlines its advantage over task-specific rule-based solutions.

Table 1: Results of the tactical pattern detection.

Task	**Model**	**AUC**	**F_1-Score**
Overlapping run	DetNet	0.62	-
	Rule-based baseline	-	0.31
	SequentialM2	**0.93**	**0.4**
Chance without a shot	DetNet	0.69	-
	SequentialM2	**0.98**	**0.37**

Although overlapping runs can be clearly defined and accordantly labeled by different experts, it cannot be captured as accurately by hand-crafted rules as the proposed method allows us to. Although patterns like chance without shot are identifiable, the involved experts in our experiment were not able to define a rule-based detector of this event. However, using our method, **Figure 3** shows a correctly identified example for the Chance without shot detection task. The lower part of the figure shows the detection probabilities inferred from the inference network of the SequentialM2. The red vertical indicator reflects the ground-truth annotation from the event data. The blue vertical indicator reflects a probability amplitude within the shown game excerpt. The two pictures above display the associated snapshots for the blue (left picture) and red (right picture) vertical lines from the test match, respectively. As can be seen, the model assigns the highest probability value to the location of the ground-truth assignment, highlighting its potential in learning the inherent patterns of agent-level game situations.

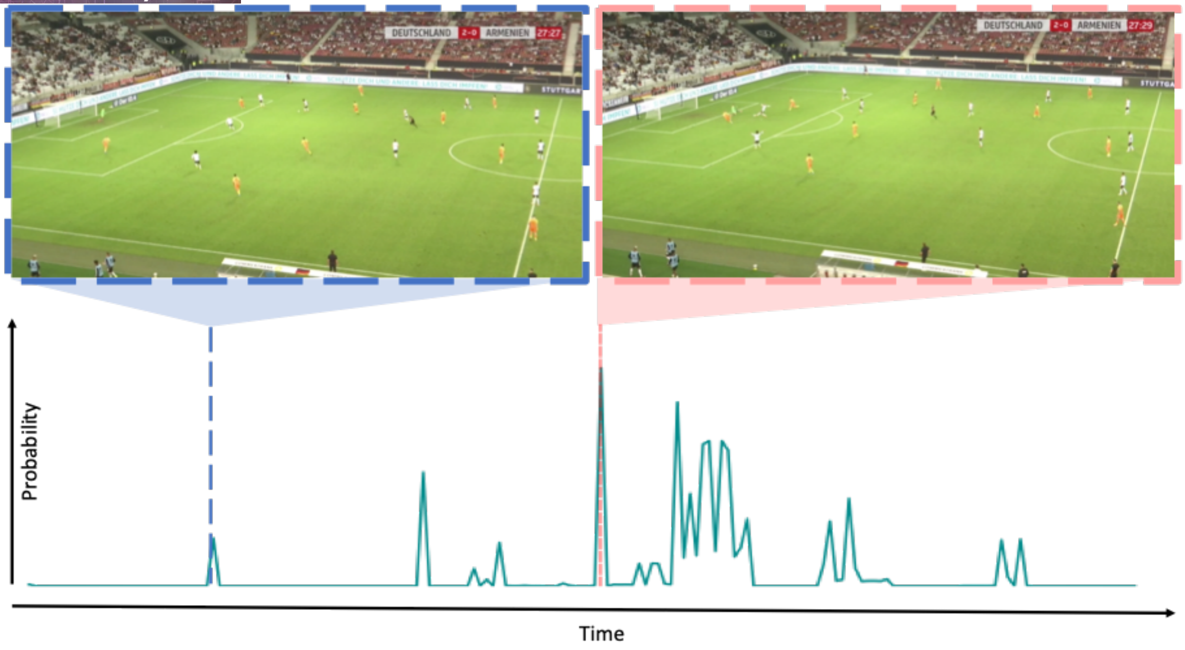

Figure 3: TP example for agent label "chance without shot". The figure shows the predicted probability values for an approx. 7 seconds excerpt from the validation game. The red vertical line indicates the event annotation. The respective video can be found here: https://bit.ly/3E4LrTD.

4. Practical Application

In this section we show how the automatic detection of overlapping runs can support the everyday processes of professional match-analysts and coaching staffs. **Figure 4** shows an excerpt of a tactical match-report that is created immediately after every match. Within this one figure, the coaches get an overview of all overlapping runs that were detected during the match. Each run is plotted at its location (playing direction normalized for each team, i.e., Germany playing from left to right; France vice versa). The involved players are shown by their jersey number and the color of the line also indicates whether the player conducting the overlapping run received the pass (green trajectory) or not (red trajectory). Instead of just providing the absolute number, the percentage close to the center of the pitch compares the number of overlapping runs per side with the historical team average helping coaches to put the numbers into context. Additionally, to that plot, timecodes of the respective overlapping runs are provided to the coaches in form of a Hudl Sportscode XML-File which they can directly import into their video analysis software.[8]

Figure 4 entails *overlapping run* situations detected in the opening match of Group F during the European Championship in 2021 in Munich.[9] Both Robin Gosens (jersey number 20) on the left German side and Benjamin Pavard (jersey number 4) on the right French side overlapped twice but never received the ball. In total the French team conducted their overlapping runs with a higher

[8] A detailed case study of how automated extracts seamlessly fit into the processes of match analysts is presented in [4].

[9] A summary of the match, including various details, can be found here: https://www.youtube.com/watch?v=20xO2Cwvs7U

diversity, with four different players performing overlapping over their left attacking side. In contrast, Germany only performed two overlapping runs over their right side with Thomas Müller (jersey number 25) and Matthias Ginter (jersey number 4) receiving the ball. As mentioned above, another helpful information for the German coaching staff is that their team conducted less overlapping run situations than they typically do (-23%).[10]

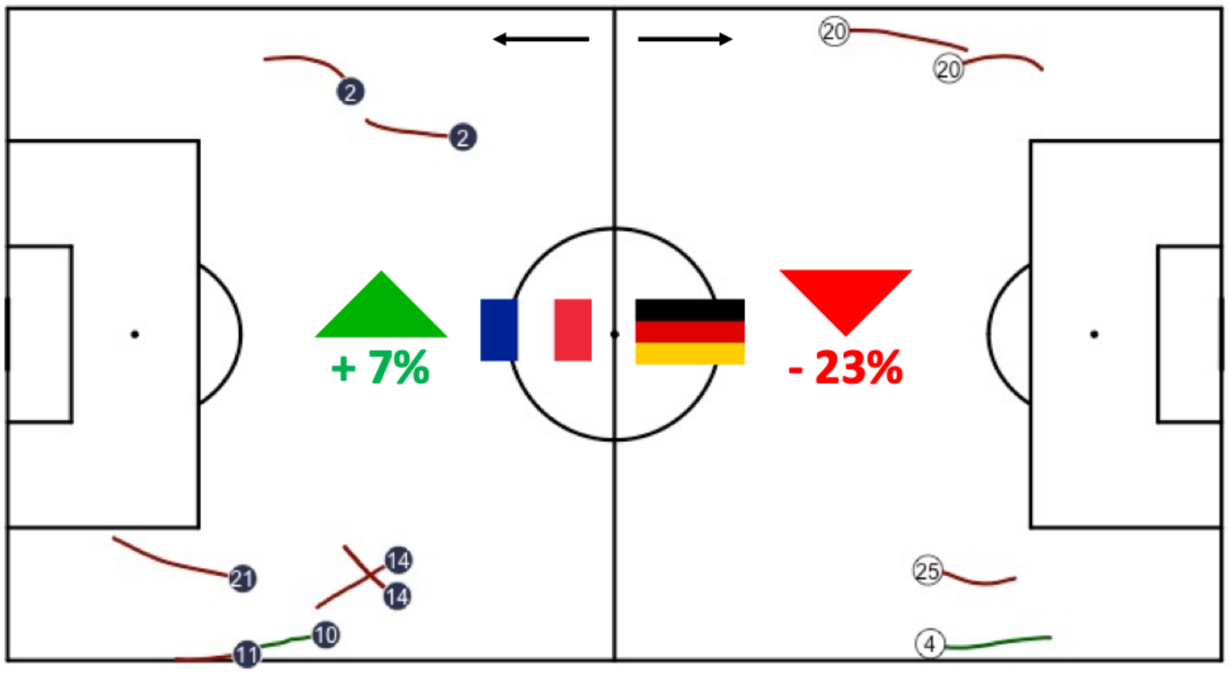

Figure 4: Excerpt from a Match Report showing all overlapping runs conducted by both teams.

Another relevant use-case is opponent analysis. Since overlapping runs are often difficult to defend, knowing which opposing players are typically involved in the pattern, is a strategic edge in tactical match-preparation. This information is usually gathered by video analysts, who manually annotate multiple matches of an opposing team by tagging the respective scenes of interest and bringing them together as a resulting statistic. Using our automated detection, the results presented in **Table 2**. an be produced automatically for each upcoming opponent. It shows which players perform overlapping runs (rows) and who is being overlapped the most (columns). The matrix also reveals preferences of pairings that conduct overlapping run situations together.

The fact that Leroy Sané, Kai Havertz and Thomas Müller—three players with a lot of playing time as offensive wingers—are overlapped frequently is kind of expectable due to their position. Further analysis shows that the pairing of Leroy Sané and Matthias Ginter creates many overlapping run situations as well. For the overlapping runners, it is also no surprise that this is usually performed by defensive wingbacks like Robin Gosens or Ridle Baku. However, the fact that Joshua Kimmich, a player with only three matches as defensive back and the remaining ones as central defender, is involved in so many overlapping run situations (8 times overlapping; 6 times overlapped) is a valuable insight.

[10] Note that the benchmark for the French team purely relies on a small data sample, i.e. all their matches at the European Championship 2021.

Table 2: Frequency of overlapping runs for the German National Team in Season 2021/2022.

		Overlapped																
		All	Can	Gnabry	Goretzka	Gündogan	Günter	Havertz	Hofmann	Kehrer	Kimmich	Müller	Musiala	Nmecha	Baku	Reus	Sane	Werner
	All		↓ 2	↓ 4	↓ 1	→ 10	↓ 1	→ 11	↓ 1	↓ 1	↓ 6	→ 10	↓ 5	↓ 1	↓ 2	↓ 1	↑ 22	↓ 5
Overlapping	Arnold	↓ 1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	Can	↓ 2		0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
	Ginter	↑ 14	0	1	0	1	0	0	0	0	3	2	0	0	0	0	6	1
	Gnabry	↓ 3	0		1	1	0	0	0	0	1	0	0	0	0	0	0	0
	Goretzka	↓ 1	0	0		1	0	0	0	0	0	0	0	0	0	0	0	0
	Gosens	↑ 10	0	0	0	0	0	3	0	0	2	1	1	0	0	0	3	0
	Gündogan	↓ 2	0	0	0		0	1	0	0	0	0	0	0	0	0	1	0
	Günter	↓ 3	0	0	0	0		0	0	0	0	0	0	0	0	0	3	0
	Havertz	↓ 2	0	0	0	0	0		0	0	0	1	0	0	0	0	1	0
	Hofmann	→ 6	0	1	0	0	0	0		0	0	2	0	0	2	1	0	0
	Kimmich	→ 8	1	1	0	1	0	0	0	0		2	0	0	0	0	3	0
	Klostermann	↓ 4	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0
	Kroos	↓ 1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	Max	↓ 5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	3
	Müller	↓ 2	0	0	0	0	0	1	0	0	0		0	0	0	0	1	0
	Musiala	↓ 2	1	0	0	0	0	1	0	0	0	0		0	0	0	0	0
	Baku	↑ 12	0	0	0	0	0	2	1	1	0	1	3	0		0	3	1
	Sane	↓ 4	0	0	0	2	1	0	0	0	0	1	0	0	0	0		0
	Younes	↓ 1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Similarly, this approach could also be used in a player recruitment setting. If a team is searching for a wing back with a strong offensive drive, the frequency with which he is supporting his offensive winger through overlapping runs serves as a good indicator. This way, the identified candidates are evaluated independently of their number of such actions, which is the only information available in traditional scouting databases. Additional insights can be generated by augmenting overlapping runs into key performance indicators like pitch control to understand how much space was created and where, or quantifying the value of an overlapping run with subsequent xG (expected goals, e.g. [21]) values.

5. Conclusion and Future Work

We proposed a semi-supervised graph neural network for pattern detection in team sports. The method worked successfully with only a few labeled data points and was observed to empirically outperform fully supervised competitors. The flexibility of the granularity of the detected patterns as well as the drastically reduced labeling effort, poses a huge benefit for practical applications. Future work will address the detection of a variety of new patterns as well as testing the approach in other team sports such as basketball.

References

[1] M. Kempe, A. Grunz, and D. Memmert, "Detecting tactical patterns in basketball: Comparison of merge self-organising maps and dynamic controlled neural networks," *Eur. J. Sport Sci.*, vol. 15, no. 4, pp. 249–255, 2015.

[2] Q. Wang, H. Zhu, W. Hu, Z. Shen, and Y. Yao, "Discerning tactical patterns for professional soccer teams: An enhanced topic model with applications," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 2015-Augus, no. April 2016, pp. 2197–2206, 2015.

[3] A. Grunz, D. Memmert, and J. Perl, "Tactical pattern recognition in soccer games by means of special self-organizing maps," *Hum. Mov. Sci.*, vol. 31, no. 2, pp. 334–343, 2012.

[4] P. Bauer and G. Anzer, "Data-driven detection of counterpressing in professional football—A supervised machine learning task based on synchronized positional and event data with expert-based feature extraction," *Data Min. Knowl. Discov.*, vol. 35, no. 5, pp. 2009–2049, 2021.

[5] P. Bauer, G. Anzer, and O. Höner, "The Identification of Counterpressing in Football," in *Match Analysis, Routledge (in press)*, 2021, pp. 228–235.

[6] J. Courel-Ibáñez, A. P. McRobert, E. O. Toro, and D. C. Vélez, "Collective behaviour in basketball: A systematic review," *Int. J. Perform. Anal. Sport*, vol. 17, no. 1–2, pp. 44–64, 2017.

[7] R. Montoliu, R. Martín-Félez, J. Torres-Sospedra, and A. Martínez-Usó, "Team activity recognition in Association Football using a Bag-of-Words-based method," *Hum. Mov. Sci.*, vol. 41, pp. 165–178, 2015.

[8] G. Anzer, P. Bauer, and U. Brefeld, "The origins of goals in the German Bundesliga," *J. Sport Sci.*, 2021.

[9] J. Hobbs, P. Power, L. Sha, H. Ruiz, and P. Lucey, "Quantifying the Value of Transitions in Soccer via Spatiotemporal Trajectory Clustering," *MIT Sloan Sport. Anal. Conf. Bost.*, pp. 1–11, 2018.

[10] P. Power, J. Hobbs, H. Ruiz, X. Wei, and P. Lucey, "Mythbusting Set-Pieces in Soccer," *MIT Sloan Sport. Anal. Conf.*, vol. 102, no. 2, pp. 1–12, 2018.

[11] L. Shaw and S. Gopaladesikan, "Routine inspection: A playbook for corner kicks," *MIT Sloan Sport. Anal. Conf. Bost.*, 2021.

[12] J. Lasek, Z. Szlávik, and S. Bhulai, "The predictive power of ranking systems in association football," *Int. J. Appl. Pattern Recognit.*, vol. 1, no. 1, p. 27, 2013.

[13] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," *Preprint (ArXiv)*. 2018.

[14] M. Stöckl, T. Seidl, D. Marley, and P. Power, "Making Offensive Play Predictable - Using a Graph Convolutional Network to Understand Defensive Performance in Soccer," *MIT Sloan Sport. Anal. Conf. Bost.*, 2021.

[15] U. Dick, M. Tavakol, and U. Brefeld, "Rating Player Actions in Soccer," *Front. Sport. Act. Learn. (Special Issue Using Artif. Intell. to Enhanc. Sport Performance)*, vol. 3, p. 174, 2021.

[16] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," *Adv. Neural Inf. Process. Syst.*, vol. 4, no. January, pp. 3581–3589, 2014.

[17] D. Fassmeyer, G. Anzer, P. Bauer, and U. Brefeld, "Toward Automatically Labeling Situations in Soccer," *Front. Sport. Act. Living*, vol. 3, no. November, 2021.

[18] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, “A recurrent latent variable model for sequential data,” in *Advances in Neural Information Processing Systems*, 2015, vol. 2015-Janua, pp. 2980–2988.
[19] J. Haase and U. Brefeld, “Mining positional data streams,” *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.*, vol. 8983, pp. 102–116, 2015.
[20] D. Linke, D. Link, and M. Lames, “Football-specific validity of TRACAB’s optical video tracking systems,” *PLoS One*, vol. 15, no. 3, pp. 1–17, 2020.
[21] G. Anzer and P. Bauer, “A Goal Scoring Probability Model based on Synchronized Positional and Event Data,” *Front. Sport. Act. Learn. (Special Issue Using Artif. Intell. to Enhanc. Sport Performance)*, vol. 3, no. 0, pp. 1–18, 2021.
[22] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, no. Ml, pp. 1–14, 2014.
[23] D. J. Rezende, S. Mohamed, and D. Wierstra, “Stochastic backpropagation and approximate inference in deep generative models,” *31st Int. Conf. Mach. Learn. ICML 2014*, vol. 4, pp. 3057–3070, 2014.
[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” pp. 1–9, 2014.
[25] K. Xu, C. Li, Y. Tian, T. Sonobe, K. I. Kawarabayashi, and S. Jegelka, “Representation learning on graphs with jumping knowledge networks,” *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 12, pp. 8676–8685, 2018.
[26] T. Kipf, E. Fetaya, K. C. Wang, M. Welling, and R. Zemel, “Neural relational inference for Interacting systems,” *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 6, pp. 4209–4225, 2018.
[27] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 1025–1035, 2017.
[28] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, “Graph attention networks,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–12, 2018.
[29] D. Mazza and M. Pagani, “Automatic differentiation in PyTorch,” *Proc. ACM Program. Lang.*, vol. 5, no. POPL, pp. 1–4, 2021.
[30] Huang, Y., Bi, H., Li, Z., Mao, T., & Wang, Z. (2019). STGAT: Modeling spatial-temporal interactions for human trajectory prediction. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October, 6271–6280. https://doi.org/10.1109/ICCV.2019.00637
[31] Amirian, J., Hayet, J. B., & Pettre, J. (2019). Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2019-June, 2964–2972. https://doi.org/10.1109/CVPRW.2019.00359
[32] Monti, A., Bertugli, A., Calderara, S., & Cucchiara, R. (2020). Dag-net: Double attentive graph neural network for trajectory forecasting. *Proceedings - International Conference on Pattern Recognition*, 2551–2558. https://doi.org/10.1109/ICPR48806.2021.9412114
[33] Zhan, Eric, et al. "Generating multi-agent trajectories using programmatic weak supervision." *arXiv preprint arXiv*:1803.07612 (2018).
[34] Felsen, Panna, Patrick Lucey, and Sujoy Ganguly. "Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Appendix

Modeling Movement Patterns

We already showed the empirical success of our approach in detection performance. However, to demonstrate the generality of our approach, we validate its additional functionality by modeling future agent behavior. For detection tasks, the model needs to reason about the whereabouts of players in the near future and we would like to use the opportunity to additionally evaluate this feature. To ensure comparability with existing methods designed for trajectory forecasting, we run experiments on a public basketball dataset provided by STATS SportVU. The dataset comprises tracking positions of offensive plays from the 2016 NBA regular season covering more than 1200 different games. Each game segment consists of 50 timesteps sampled at 5 frames per second, with each timestep encoding x/y-positions for all 11 agents (the 10 players on the court and the ball). The dataset contains already preprocessed multi-agent trajectories and therefore comprises merely the final preprocessing step of **Section 2.2**.

Recent work emphasizes the importance of location-based goals in modeling human motion patterns [32], [33]. These methods are characterized as weakly supervised since they use the given trajectory data and an externally selected speed threshold to identify stationary points for each player. Such (weak) label information can be obtained, for example, by capturing the playing field as a grid of macro-areas, where each cell can represent a potential long-term goal of an agent. At each timestep, the agent's ground-truth objective is then defined as the location where the agent is stationary, i.e., moving below a prespecified speed threshold. Since the proposed framework incorporates discrete supervised signals into the assumption of the data generating process, such information can be naturally integrated into our overall scheme. Specifically, the task can be formalized within the SequentialM2 workflow as maximizing the supervised part of our objective function, where supervision in this case refers to the inferred long-term grid locations.

Table 3 summarizes the results. The Mean L2-Error represents the mean square error between the real observed and the predicted positions over the entire sequence. The Final L2-Error follows the same logic but uses only the last timestep for comparison. Since offensive and defensive players inherently materialize different strategies and thus different trajectory patterns, we train distinct models for both subsets of players. The reported metrics refer to a prediction interval of 40 and 30 timesteps, with an observation/burn-in period of 10 and 20 timesteps, respectively.

The proposed SequentialM2 model framework accommodates not only mutual influences among different players, but also discrete generative factors, which yields a better approximation of the underlying multi-modal data distribution. This aspect is reflected in the impressive quantitative results shown in **Table 3**: though our method is not explicitly designed for trajectory forecasting, we exceed current state-of-the-art in nearly all tested scenarios. Further, the consistently lower performance errors in defensive scenarios highlight their reactive nature and accompanying lower complexity compared to offensive strategies.

A visual representation of the modeling task for an offensive rollout can be found in **Figure 5:** After an initial observation phase, the model predictions are evaluated against the ground-truth trajectories. In addition, we illustrate their corresponding label information with black boxes, where the color intensity corresponds to the frequency of the (weakly obtained) location-based labels. The picture underlines the models' ability for effective sampling from the label space, since it captures highly complex changes in movement directions.

Table 3: The Mean L2-error represents the mean square error between the ground-truth and the predicted positions over the entire sequence. The final L2-error follows the same logic, but only refers to the last timestep. Bold is the highest in the L2 and final L2 column for each task.

Task	Model	L2	Final L2
10-40 (ATK)	STGAT [30]	9.94	15.80
	Social-Ways [31]	9.91	15.19
	Weak-Supervision [33]	9.47	16.98
	DAG-Net [32]	8.98	14.08
	SequentialM2	**8.41**	**12.64**
10-40 (DEF)	STGAT [30]	7.26	11.28
	Social-Ways [31]	7.31	10.21
	Weak-Supervision [33]	7.05	10.56
	DAG-Net [32]	6.87	9.76
	SequentialM2	**6.45**	**8.85**
20-30 (ATK)	C-VAE [34]	7.08	-
	DAG-Net [32]	6.66	-
	SequentialM2	**6.49**	10.62
20-30 (DEF)	**C-VAE** [34]	**4.98**	-
	DAG-Net [32]	5.01	-
	SequentialM2	5.04	7.44

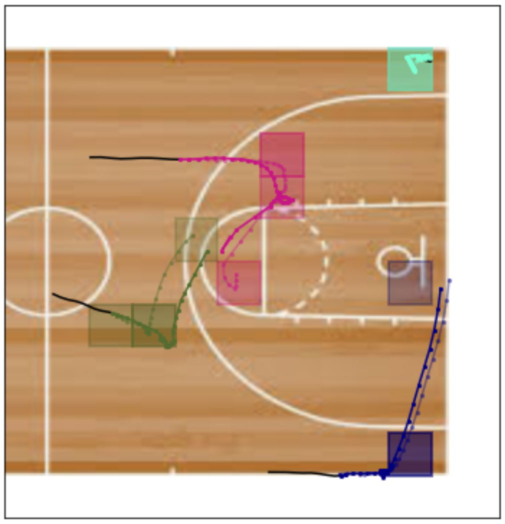

Figure 5: Modeling offensive player trajectories by observing 10 timesteps and predicting 40.

View publication stats