Pass Receiver and Outcome Prediction in Soccer Using Temporal Graph Networks

Pegah Rahimian[1], Hyunsung Kim[2], Marc Schmid[3], and Laszlo Toka[1,4]

[1] Budapest University of Technology and Economics, Budapest, Hungary
[2] Seoul National University, Seoul, South Korea
[3] Technische Universität München, München, Germany
[4] ELKH-BME Cloud Applications Research Group, Budapest, Hungary
pegah.rahimian@edu.bme.hu

Abstract. This paper explores the application of the Temporal Graph Network (TGN) model to predict the receiver and outcome of a pass in soccer. We construct two TGN models that estimate receiver selection probabilities (RSP) and receiver prediction probabilities (RPP) to predict the intended and actual receivers of a given pass attempt, respectively. Then, based on these RSP and RPP, we compute the success probability (CPSP) of each passing option that the pass is successfully sent to the intended receiver as well as the overall pass success probability (OPSP) of a given situation. The proposed framework provides deeper insights into the context around passes in soccer by quantifying the tendency of passers' choice of passing options, difficulties of the options, and the overall difficulty of a given passing situation at once.

Keywords: Soccer Analytics · Multi-Agent Analysis · Temporal Graph Network · Pass Receiver Prediction · Pass Outcome Prediction.

1 Introduction

Passes are the most frequent event in soccer, so analyzing them is essential to evaluate players' performance or match situations [1, 11]. Particularly, focusing on individual passing options in a given passing situation enables domain participants to characterize the general tendency of players' decision-making or assess their decisions. There are two main aspects of analyzing passing options: either in terms of player (i.e., selecting a player to receive the pass) [1,11] or space (i.e., selecting a specific location on the pitch to send the ball to) [6, 16, 18].

In reality, it is difficult for players to pass the ball to a specific point on purpose, so we focus on players rather than the space to concretize the passer's intention more intuitively. We employ Temporal Graph Network (TGN) [17] to predict the intended and actual receivers in a given passing situation. By leveraging the TGN's ability to capture temporal dependencies, we estimate for a given moment the receiver selection probability (RSP) that the passer intends to send the ball to each of the teammates and the receiver prediction probability (RPP) that each player becomes the actual receiver of the pass.

Based on these RSP and RPP, we compute the success probability (named as CPSP in our paper) of each passing option that the pass is successfully sent to the intended receiver as well as the overall pass success probability (OPSP) of a given situation. Especially, we mathematically prove that dividing the RSP of a teammate by the corresponding RPP is equal to the CPSP of the passing option to the teammate. We analyze 358,790 passes from the 330 Belgian Pro League matches to estimate their average success probabilities in 18 zones of the pitch for both the start and end locations of the passes.

The proposed framework provides deeper insights into the context around passes in soccer by quantifying the tendency of passers' choice of passing options, the difficulties of the options, and the overall difficulty of a given passing situation at once. Another contribution is that this study suggests the potential of applying the TGN model to team sports data for handling the interaction between players. Also, we have made the source code available online for reproducibility [5].

2 Related Work

Several studies have tried to quantify the risk of a pass in a given passing situation. Spearman et al. [18] proposed a physics-based framework named Pitch Control to estimate the probability of a pass being successful given that the pass is sent to each location on the pitch. Power et al. [11] employed logistic regression to estimate the risk and reward of a pass based on handcrafted features. Fernández et al. [6] performed a similar task to that of Pitch Control, but by implementing a CNN-based deep learning architecture instead of physics-based modeling. Anzer and Bauer [1] predicted the intended receiver leveraging the approach of Pitch Control, and trained an XGBoost [5] model to estimate the success probability of each passing option. Most recently, Robberechts et al. [16] proposed a framework named un-xPass that measures a passer's creativity.

Meanwhile, analyzing players' movements in soccer is a cumbersome task due to its spatiotemporal and permutation-invariant nature, so several methods have been proposed to deal with this nature. Some studies [6,14–16] treated each moment of the data as an image and apply a convolutional neural network (CNN) to encode it, and others [10,12,13] sorted players by a rule-based ordering scheme starting from the ball possessor. A better approach to model interaction between players and the ball is to employ graph-based [3] or Transformer-based [19] neural networks. To name a few, Anzer et al. [2] and Bauer et al. [4] constructed graph neural networks (GNN) to detect overlapping patterns and to divide a match into multiple phases of play, respectively. Kim et al. [8] deployed Set Transformers [9] to predict the ball locations from player trajectories.

3 Decomposing the Pass Success Probability

In this section, we formulate a relationship between the selection and success probabilities of each passing option in a given passing situation. Based on this

[5] `https://github.com/hsnlab/sports_analitica`

relationship, we calculate the success probability of each passing option as well as the overall pass success probability in the situation.

Strictly speaking, a pass is said to be "successful" only if it is sent to the intended receiver. However, since there is no direct way of knowing the intention of each pass, many studies simply define that a pass is successful if one of the passer's teammates receives the pass. This definition involves an assumption that for each successful pass, the actual receiver (who is a teammate of the passer) is the expected receiver to whom the passer intended to send the ball. In other words, for the random variables E and R indicating the expected and actual receivers of the pass in a given state $S = s$, respectively, we assume

$$P(E = R|S = s, O = o^+) = 1. \tag{1}$$

where $O = o^+$ denotes the event that the pass is successful. Starting from this assumption, we can prove the following proposition.

Proposition 1. For each teammate i of the passer,

$$P(R = i|S = s) = P(E = i, R = i|S = s).$$

Proof. Since $P(E = i, R = i|S = s) = P(R = i|S = s) \cdot P(E = i|S = s, R = i)$, it is enough to show that $P(E = i|S = s, R = i) = 1$ for every $i \in T^+$ where T^+ is the set of the passer's teammates: Suppose $P(E = i|S = s, R = i) < 1$ for some $i \in T^+$, then

$$\begin{aligned}
&P(E = R|S = s, O = o^+) \\
&= \sum_{i \in T^+} P(E = i, R = i|S = s, O = o^+) \\
&= \sum_{i \in T^+} P(E = i|S = s, O = o^+, R = i) \cdot P(R = i|S = s, O = o^+) \\
&= \sum_{i \in T^+} P(E = i|S = s, R = i) \cdot P(R = i|S = s, O = o^+) \\
&< \sum_{i \in T^+} P(R = i|S = s, O = o^+) \\
&= 1 \quad \text{(by the definition of a successful pass).}
\end{aligned}$$

This contradicts Eq. 1 and thus completes the proof. □

Based on Proposition 1, we can decompose the overall pass success probability as follows:

$$\begin{aligned}
P(O = o^+|S = s) &= \sum_{i \in T^+} P(R = i|S = s) && (2) \\
&= \sum_{i \in T^+} P(E = i, R = i|S = s) && (3) \\
&= \sum_{i \in T^+} P(E = i|S = s) \cdot P(R = i|S = s, E = i). && (4)
\end{aligned}$$

Four types of probabilities are related to the above equations:

- **Overall pass success probability** $P(O = o^{+}|S = s)$ (OPSP) that a pass in a given state s is successful.
- **Receiver selection probability** $P(E = i|S = s)$ (RSP) that the passer at s intends to send the ball to a teammate i.
- **Receiver prediction probability** $P(R = i|S = s)$ (RPP) that i is the actual receiver of the pass.
- **Conditional pass success probability** $P(R = i|S = s, E = i)$ (CPSP) given that i is the expected receiver of the pass being successfully sent to i.

In Section 4, we construct separate TGN architectures to estimate RSPs and RPPs, respectively. Then, we can obtain the OPSP for a given state from Eq. 2, i.e., by adding up the RPPs for all the teammates of the passer. Moreover, from Proposition 1, dividing the RPP of each teammate by the corresponding RSP results in the CPSP indicating the success probability of the hypothetical pass from the passer to the teammate, i.e.,

$$P(R = i|S = s, E = i) = \frac{P(E = i, R = i|S = s)}{P(E = i|S = s)} = \frac{P(R = i|S = s)}{P(E = i|S = s)}. \quad (5)$$

4 Constructing Temporal Graph Networks

In this section, we explain the tasks of estimating RSPs and RPPs introduced in Section 3 using separate TGN models. The common goal of our RSP and RPP models is to find the most likely receiver (either expected or actual) in a given passing situation. What differentiates them is that the candidate receivers of the former are the teammates (10 in general) of the passer and those of the latter are all the players (21 in general) other than the passer. In Section 4.1 and 4.2, we elaborate on the common fundamentals of our TGN models. In Section 4.3, we describe how to train the TGN for each type of probability.

4.1 Model Definition

The TGN model for each task takes a sequence of time-stamped events that occurred during each "possession" in soccer matches and produces the probability of the pass being received by each of the players (or the teammates) on the pitch in a given game state. Here a possession is defined as a time interval that a team continues to touch the ball except for fewer than three consecutive actions by the opponents. Namely, we assume that a possession ends when the next three actions are performed by the opposite team.

First, we make a graph with **nodes** corresponding to players, and **interactions** (i.e., temporal edges) indicating pass attempts between players. We label an interaction as a successful pass if it connects the two nodes of the same team, and an unsuccessful pass otherwise. Also, we extract temporal features for each of the nodes and edges on top of event and tracking data collected from the given match. More specifically, **node features** include a player's (x, y) location, velocity, distance, and angle from the ball carrier, and a flag indicating whether the player is the ball carrier for each time-step. Meanwhile, **edge features** include the distance and relationship (i.e., teammates or opponents) between the two interacting nodes.

Then, we model a TGN as a sequence of events $G = \{x(t_1), x(t_2), ...\}$ at times $0 \leq t_1 \leq t_2 \leq \cdots$, where $x(t)$ is either (1) a **node-wise event** $\mathbf{v}_i(t)$ of a player i such as the change of his location or (2) an **interaction event** $\mathbf{e}_{ij}(t)$ represented by a temporal edge between two nodes i and j such as a pass or a change in the distance between the two players.

4.2 TGN Architecture

In this section, we elaborate on the building blocks of the proposed TGN. It consists of an encoder-decoder pair, where an encoder is a function that maps players' interactions to node embeddings and a decoder takes the node embeddings as input and performs link prediction of the future time-steps. Figure 1 depicts the architecture of our network, which consists of the following modules:

- **Memory:** A memory of the model $\{\mathbf{s}_i(t)\}_{i \in P}$ at time t is a representation of the node's history that the model has seen until t. It consists of a state vector $\mathbf{s}_i(t)$ for each player $i \in P$ with the set P of all the players in the game at t, which is updated after an event $x(t)$. (Note that $x(t)$ can be either node-wise or interactive.) When a substitute is sent onto the pitch and a new node is created, the network initializes a zero vector for it, and then updates the memory after each event the player is involved in.
- **Message Function:** For each event $x(t)$ involving player i, the model computes a message $\mathbf{m}_i(t)$ to update i's memory. When a node-wise event $\mathbf{v}_i(t)$ happens, a single message for i is computed as:

$$\mathbf{m}_i(t) = \mathrm{msg_n}(\mathbf{s}_i(t^-), t, \mathbf{v}_i(t)).$$

  Likewise, an interaction event $\mathbf{e}_{ij}(t)$ induces the computation of messages for the passer i and the receiver j as follows:

$$\mathbf{m}_i(t) = \mathrm{msg_s}(\mathbf{s}_i(t^-), \mathbf{s}_j(t^-), \Delta t, \mathbf{e}_{ij}(t))$$
$$\mathbf{m}_j(t) = \mathrm{msg_d}(\mathbf{s}_j(t^-), \mathbf{s}_i(t^-), \Delta t, \mathbf{e}_{ij}(t))$$

  where $\mathbf{s}_i(t^-)$ is the memory of i at the time of the last event before t in which the player is involved and $\mathrm{msg_n}, \mathrm{msg_s}, \mathrm{msg_d}$ are learnable message functions.
- **Message Aggregator:** Since each player i can be involved in multiple events until time t, we aggregate all the memories $\mathbf{m}_i(t_1), \ldots, \mathbf{m}_i(t_b)$ of i generated before t by averaging them, i.e.,

$$\bar{\mathbf{m}}_i(t) = \mathrm{mean}(\mathbf{m}_i(t_1), \ldots, \mathbf{m}_i(t_b)).$$

- **Memory Updater:** For each event $x(t)$, a learnable memory update function updates the memory $\mathbf{s}_i$ of each player i involved in the event:

$$\mathbf{s}_i(t) = \mathrm{mem}(\bar{\mathbf{m}}_i(t), \mathbf{s}_i(t^-)).$$

In this work, we employ the structure of Long Short-Term Memory (LSTM) [7] for this memory update function.

- **Embedding:** Even if $\mathbf{s}_i$ is not updated at time t because the player i is not involved in the event $x(t)$, the context around i can change by interactions of other players. To reflect this, we also deploy the embedding module to generate the temporal embedding $\mathbf{z}_i(t)$ of i at any time t by

$$\mathbf{z}_i(t) = \mathrm{emb}(i,t) = \sum_{j \in P_{-i}} h(\mathbf{s}_i(t), \mathbf{s}_j(t), \mathbf{e}_{ij}, \mathbf{v}_i(t), \mathbf{v}_j(t))$$

  where P_{-i} is the set of all 21 players other than i and h is Temporal Graph Attention (TGA) proposed in Rossi et al. [17].
- **Link Prediction:** To predict the most likely receiver (either expected or actual) of a pass attempt, we put the temporal embeddings $\mathbf{z}_i(t)$ into a fully connected layer that outputs link values between nodes. After applying softmax to these values, we obtain a set of probabilities that add up to 1 and mean which candidate would be the receiver of the pass. Note that any passer does not intend to send the ball to an opponent, so we restrain the candidates to the passer's teammates for the RSP model. Meanwhile, all the players including opponents are the candidates for the RPP model. See Fig. 2 depicting the resulting probabilities in a passing moment as an example.

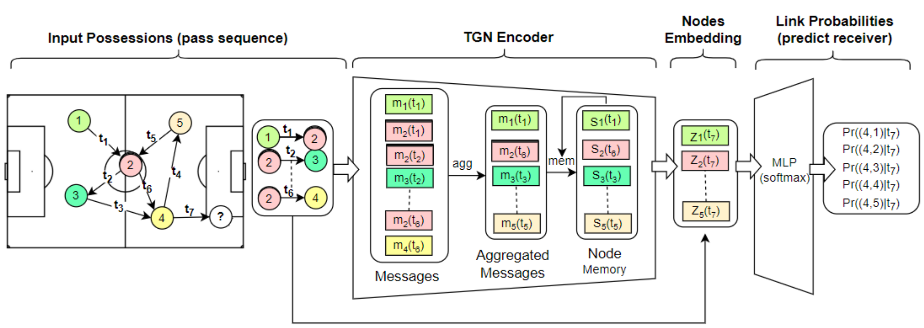
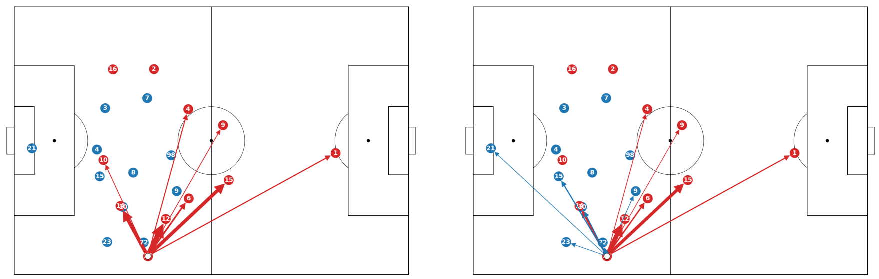


Fig. 1: TGN architecture for outcome prediction.

4.3 Training RSP and RPP Models

We train separate TGN models for estimating RSPs and RPPs, respectively. Both models take features for all the players in the state s. On the other hand, we impose different restrictions on model outputs as described in Section 4.2: While only a teammate can be a candidate receiver for the RSP model, all the players including teammates and opponents are candidates for the RPP model.



(a) Receiver selection probabilities (b) Receiver prediction probabilities

Fig. 2: Visualizations of RSP and RPP for an example match situation. Every passing option from the ball carrier to a player with a probability larger than 0.01 is expressed as an arrow whose width indicates the probability value.

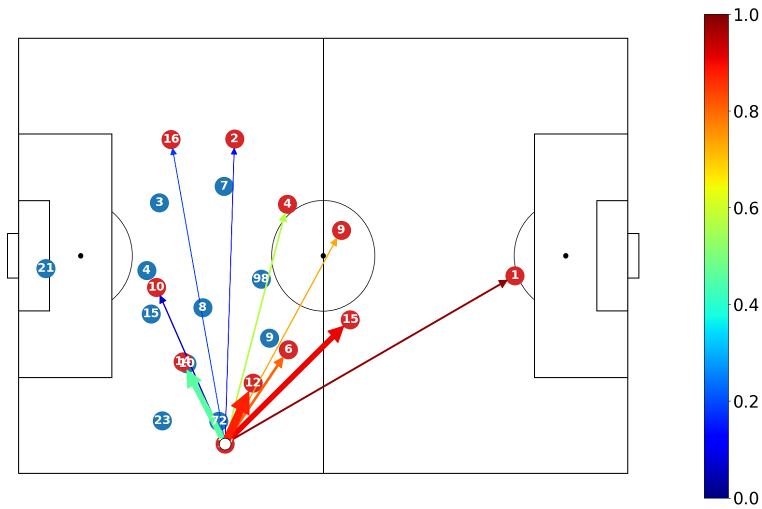


Fig. 3: Combined visualization of RSP and CPSP for the same match situation as Fig. 2. The width of an arrow indicates the selection probability (RSP) of the corresponding passing option (same as Fig. 2a) and the color of it stands for the success probability (CPSP) of such option.

For the RSP model, we aim to estimate $\hat{y}^E_{s,i} = P(E = i|S = s)$ for each of the passer's teammates i, the probability that the passer intends to send the ball to i in a given state s. While we cannot know the expected receiver of an unsuccessful pass, we have assumed that for a successful pass, the actual receiver is the expected receiver in Section 3. Hence, we take successful passes $\mathcal{D}^+$ in the training dataset and use the actual receivers of them as the true labels indicating the expected receivers for training. Namely, the model is trained by minimizing the cross-entropy loss

$$\mathcal{L}^E = \frac{1}{|\mathcal{D}^+|} \sum_{s \in \mathcal{D}^+} \sum_{i \in T^+} y^E_{s,i} \log \hat{y}^E_{s,i}$$

between the output $\hat{y}^E_{s,i}$ and the true label $y^E_{s,i}$. Here $y^E_{s,i} = 1$ if i receives the pass, and $y^E_{s,i} = 0$ otherwise.

For the RPP model, we want to estimate $\hat{y}^R_{s,i} = P(R = i|S = s)$ for each of the players i (either a teammate or an opponent), the probability that i actually receives the pass. Other than the RSP model, we do know the true receiver for every pass (either successful or failure) in the dataset. Thus, we train the model with the entire training dataset $\mathcal{D}$ by minimizing the cross-entropy loss

$$\mathcal{L}^R = \frac{1}{|\mathcal{D}|} \sum_{s \in \mathcal{D}} \sum_{i \in T^+ \cup T^-} y^R_{s,i} \log \hat{y}^R_{s,i}$$

where $y^R_{s,i} = 1$ if i receives the pass, and $y^R_{s,i} = 0$ otherwise.

While the two models are trained on different datasets, they can be applied to any passing situation regardless of its outcome. They produce $\hat{y}^E_{s,i}$ for each teammate i and $\hat{y}^R_{s,j}$ for each player j (either a teammate or an opponent) for the situation. Then, as mentioned in Section 3, we can compute the OPSP from Eq. 2 and the CPSP per teammate from Eq. 5, i.e.,

$$P(O = o^+|S = s) = \sum_{i \in T^+} P(R = i|S = s) = \sum_{i \in T^+} \hat{y}^R_{s,i} \tag{6}$$

$$P(R = i|S = s, E = i) = \frac{P(R = i|S = s)}{P(E = i|S = s)} = \frac{\hat{y}^R_{s,i}}{\hat{y}^E_{s,i}}. \tag{7}$$

For example, the OPSP of Fig. 2 can be obtained by summing the widths of red arrows in Fig. 2b. Also, the CPSP per the passer's teammate is calculated by dividing the width of the corresponding arrow in Fig. 2b by that of its counterpart in Fig. 2a. The results are illustrated as the arrows' colors in Fig. 3.

5 Experiments

5.1 Dataset

The dataset consists of high-resolution spatiotemporal tracking and event data covering all 330 games of the 2020-21 season of Belgian Pro League collected by Stats Perform. The tracking data include the (x, y) coordinates of all 22 players and the ball on the pitch for 25 observations per second. The event data includes on-ball action types such as passes, shots, dribbles, etc. annotated with additional features such as period ID, the ball carrier's ID, start and end locations of the ball. We then merged tracking data with event data. Each record of our merged dataset includes all players and the ball coordinates with their corresponding features for each snapshot, i.e., every 0.04 seconds.

Table 1: Feature sets of different models.

Model	Selected features
XGBoost trained with event data	Normalized start and end locations of the passes, pass length, pass angle, pass direction, a flag indicating whether the pass goes inside the penalty box, angle and distance to goal, pass type.
XGBoost trained with event and tracking data	Velocity of the passer, velocities of the nearest defenders toward the passer and the receiver, Distances from the passer and receiver to their respective nearest defenders, the nearest defenders' angles to the passing line, time from regaining possession.
TGN (trained with event and tracking data)	(1) Node features: player's location, velocity, distance and angle from the ball carrier, and a flag indicating whether the player is the ball carrier for each time-step. (2) Edge Features: distance and relationship (i.e., teammates or opponents) between the two interacting nodes.

Table 2: Model performance on the test dataset.

Model	F_1 score	AUC	Log loss
Naive	-	0.50	0.45
XGBoost (event only)	0.81	0.79	0.38
XGBoost (event + tracking)	0.86	0.82	0.29
TGN	**0.95**	**0.92**	**0.18**

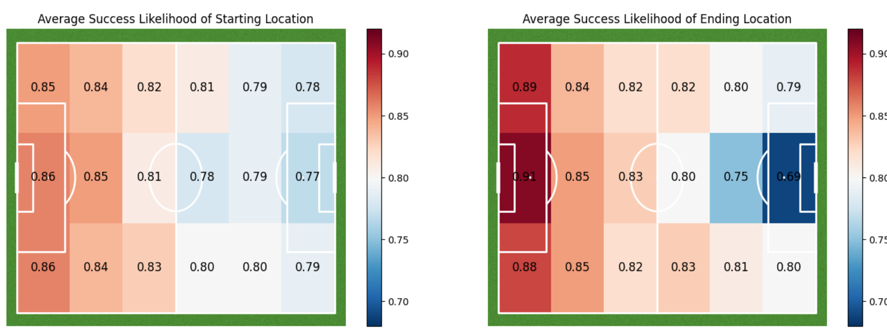
5.2 Evaluating Model Performance

We aim to evaluate the performance of our framework for estimating CPSPs. To this end, we trained our TGN models with 358,790 passes from the 330 Belgian Pro League matches. The data was split by using 80% of games for model training, 10% for validation, and 10% for test according to the chronological order of the matches in the league. Also, we implemented three other baseline models to estimate the success probability of a hypothetical pass for comparison. First, we implemented a naive model that assigns a fixed success probability of 84.5% (the average success rate in our dataset) to every pass. Second, we trained a binary classifier with XGBoost [5] using only the features that can be derived from event data (i.e., the first row in Table 1). Lastly, we trained XGBoost with hand-crafted features derived from event and tracking data [11] to showcase the necessity of positional data in prediction (i.e., the second row in Table 1). As a consequence, Table 2 demonstrate that our TGN model outperforms the other baselines, achieving 0.95 in F_1 score and 0.92 in Area Under the Curve (AUC).

5.3 Pass Difficulty in Different Areas of the Pitch

To show that the resulting success probabilities agree with our intuition, we compared the CPSPs in different areas of the pitch. Specifically, we split the pitch into 18 zones as in Fig. 4 and calculated the average CPSP for the passes starting from each zone. Also, we performed the same aggregation using the end locations of passes. The resulting average probabilities of each zone signify how it is challenging that a player in the area makes a successful pass (aggregation by starting zone) or that a player successfully sends to the ball to that area (aggregation by ending zone). Namely, the lower the average values of the zone, the more difficult to make a successful pass from or to the area.

Figures 4a and 4b depict the average CPSP for all passes in different areas of the pitch. The cooler colors of a pitch zone show lower average values indicating more difficulty. Both figures demonstrate that it is more difficult to make a pass in or towards the attacking area (i.e., zones in the central channel and the final third). Particularly, partitioning by the destination of a pass exhibits more deviation of probabilities than that by the origin. These observations accord with our general intuition that it is generally more challenging to succeed in passing in the scoring zone, and even harder to successfully send the ball to the zone.



(a) Partitioning by starting location (b) Partitioning by ending location

Fig. 4: Average CPSPs in different areas of the pitch.

6 Conclusion

In this study, we propose the application of a Temporal Graph Network (TGN) to pass receiver and outcome prediction in soccer. By leveraging the TGN's predictive capabilities, our framework can analyze a given passing situation with segmentalized components. Specifically, it quantifies the tendency of passers' choice of passing options, the difficulties of the options, and the overall difficulty of a given passing situation at once. A direction for future work is to assess the offensive and defensive performance of players leveraging our metrics.

Acknowledgment

Project no. 2021-1.2.4-TÉT-2021-00053 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the 2021-1.2.4-TÉT funding scheme.

References

1. Anzer, G., Bauer, P.: Expected passes: Determining the difficulty of a pass in football (soccer) using spatio-temporal data. Data Mining and Knowledge Discovery **36**, 295–317 (2022)
2. Anzer, G., Bauer, P., Brefeld, U., Fassmeyer, D.: Detection of tactical patterns using semi-supervised graph neural networks. In: 16th MIT Sloan Sports Analytics Conference (2022)
3. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V.F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, H.F., Ballard, A.J., Gilmer, J., Dahl, G.E., Vaswani, A., Allen, K.R., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M.M., Vinyals, O., Li, Y., Pascanu, R.: Relational inductive biases, deep learning, and graph networks. CoRR **abs/1806.01261** (2018), `http://arxiv.org/abs/1806.01261`
4. Bauer, P., Anzer, G., Shaw, L.: Putting team formations in association football into context. Journal of Sports Analytics **9**(6), 39–59 (2023)
5. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
6. Fernández, J., Bornn, L.: SoccerMap: A deep learning architecture for visually-interpretable analysis in soccer. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (2020)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
8. Kim, H., Choi, H.J., Kim, C.J., Yoon, J., Ko, S.K.: Ball trajectory inference from multi-agent sports contexts using set transformer and hierarchical bi-LSTM. In: Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2023)
9. Lee, J., Lee, Y., Kim, J., Kosiorek, A.R., Choi, S., Teh, Y.W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In: Proceedings of the 36th International Conference on Machine Learning (2019)
10. Mehrasa, N., Zhong, Y., Tung, F., Bornn, L., Mori, G.: Deep learning of player trajectory representations for team activity analysis. In: 12th MIT Sloan Sports Analytics Conference (2018)
11. Power, P., Ruiz, H., Wei, X., Lucey, P.: Not all passes are created equal: Objectively measuring the risk and reward of passes in soccer from tracking data. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017)
12. Rahimian, P., Oroojlooy, A., Toka, L.: Towards optimized actions in critical situations of soccer games with deep reinforcement learning. In: Proceedings of the 8th IEEE International Conference on Data Science and Advanced Analytics (2021)

13. Rahimian, P., da Silva Guerra Gomes, D.G., Berkovics, F., Toka, L.: Let's penetrate the defense: A machine learning model for prediction and valuation of penetrative passes. In: Proceedings of the 9th International Workshop on Machine Learning and Data Mining for Sports Analytics (2022)
14. Rahimian, P., Van Haaren, J., Abzhanova, T., Toka, L.: Beyond action valuation: A deep reinforcement learning framework for optimizing player decisions in soccer. In: 16th MIT Sloan Sports Analytics Conference (2022)
15. Rahimian, P., Van Haaren, J., Toka, L.: Towards maximizing expected possession outcome in soccer. International Journal of Sports Science and Coaching **0** (2023)
16. Robberechts, P., Roy, M.V., Davis, J.: un-xPass: Measuring soccer player's creativity. In: Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2023)
17. Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., Bronstein, M.M.: Temporal graph networks for deep learning on dynamic graphs. CoRR **abs/2006.10637** (2020), `https://arxiv.org/abs/2006.10637`
18. Spearman, W., Basye, A., Dick, G., Hotovy, R., Pop, P.: Physics-based modeling of pass probabilities in soccer. In: 11th MIT Sloan Sports Analytics Conference (2017)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)