Temporal Graph Machine Learning for Live Outcome Prediction in Football Using Passing Networks

Ties Bloemen t.t.j.bloemen@student.tudelft.nl

Marc Delgado m.delgadosanche@student.tudelft.nl

Szymon Gołębiowski sgolebiowski@student.tudelft.nl

Olaf Heijl o.p.heijl@student.tudelft.nl

Oriol Monge o.mongegirones@student.tudelft.nl

Abstract

Football's global popularity has driven a growing interest in data-driven models capable of predicting match outcomes in real time. Prior work has shown that passing networks capture meaningful patterns of team behavior and player interaction, yet most existing models rely on static representations that aggregate all passes up to the prediction time, overlooking the temporal evolution of the game.

In this work, we apply temporal graph-based models to passing network analysis for live outcome prediction. We construct dynamic graphs over fixed time windows and evaluate multiple temporal graph learning architectures. Our experiments reveal that models explicitly encoding temporal dependencies consistently outperform static counterparts, particularly in the early stages of a match when uncertainty is highest. We further investigate the influence of window length, loss function design, and limited historical context on predictive accuracy.

1 Introduction

Football is the most popular sport in the world, with an estimated 3.5 billion followers and the largest market size among all sports. This immense popularity has spurred a growing demand for advanced analytics and predictive insights, both from fans and professional stakeholders. Modern football broadcasts increasingly integrate data-driven features such as the EPL's *Win Probability*, providing viewers with real-time estimations of the likelihood of each team winning. Beyond entertainment, such dynamic predictions are also of interest to analysts, coaching staff, and betting markets, who seek to understand how match events influence outcome probabilities over time.

In recent years, machine and deep learning have gained traction in the domain of sports analytics. However, much of the existing work has predominantly focused on pre-match outcome prediction using static features [1–3]. In contrast, **in-game** or **real-time prediction**, where the estimated outcome evolves as the match unfolds, remains relatively underexplored.

Among the various actions in football, passing is the most frequent and fundamental. Consequently, researchers have devoted significant attention to analyzing passing behaviour to infer team strategies and performance. **Passing networks**, where players are represented as nodes and passes as edges, have emerged as a powerful tool to model and interpret team interactions [4–6]. Such networks have

Preprint. Under review.

been utilized in various works, mainly for tactical analysis [7] and evaluation of passing tendencies [8].

More recently, a notable study applied **Graph Attention Networks** (**GAT**s) for real-time match outcome prediction using passing networks [9]. However, this approach constructs a single aggregated network from all passes up to a given time, thereby discarding the temporal distribution of passing interactions. Similarly, other graph-based frameworks for sports outcome prediction have relied on static game states, and explicitly highlight that modeling sequences of evolving graphs for real-time predictions remains an open research challenge [10].

In this work, we address this gap by applying **temporal graph machine learning models** to sequences of passing networks over successive time windows. We incorporate temporal dependencies between passing patterns to predict match outcomes dynamically as the game progresses. Specifically, we explore different architectures capable of learning from graph-structured time series and evaluate their effectiveness in real-time outcome prediction.

2 Methods

2.1 Data Collection and Graph Construction

We generate our dataset using the `SoccerData` Python package, which provides structured access to WhoScored.com event data [11]. We focus on the English Premier League from 2015-2016 to 2023-2024, including timestamped events such as passes or goals, along with player identities and pass locations.

From these events, we build temporal passing networks by dividing each match into fixed time windows (e.g., 5 minutes). For each window, all successful passes are extracted to construct directed graphs $G_t^H = (V_t^H, E_t^H)$ and $G_t^A = (V_t^A, E_t^A)$ for the home and away teams, where nodes represent players and edges $(i, j) \in E_t$ with weight e_{ij} denote the number of passes from i to j. Node features $\mathbf{x}_v^t$ include mean pass location $(\bar{x}, \bar{y})$ and a validity flag. Each temporal snapshot inherits the match outcome $y \in \{\text{home win}, \text{draw}, \text{away win}\}$. The resulting temporal dataset $\mathcal{G}_t = (\{G_1^H, \ldots, G_t^H\}, \{G_1^A, \ldots, G_t^A\}, \{\mathbf{s}_1, \ldots, \mathbf{s}_t\})$ can be queried at each interval t to simulate live prediction. Here, $\mathbf{s}_t$ denotes the global match features, which include the start and end of the current time window as well as the current score for both teams.

2.2 Problem Formulation

Live outcome prediction is formulated as a temporal multi-graph classification task. Given $\mathcal{G}_t$, the model outputs a probability distribution $\hat{y}_t$ over match outcomes. The training objective combines cross-entropy for the categorical outcome with Poisson losses for goal counts when included:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{CE}}(y, \hat{y}_t) + \frac{\beta}{2}\Big(\mathcal{L}_{\text{Poisson}}(g_H, \hat{g}_H) + \mathcal{L}_{\text{Poisson}}(g_A, \hat{g}_A)\Big). \tag{1}$$

The goal-based component of the loss models the number of goals scored by each team as a Poisson random variable and applies a likelihood-based loss as described in [12, 13], encouraging the model to predict expected goal counts consistent with the observed outcomes.

2.3 Models

To evaluate the effect of incorporating both spatial (graph) and temporal information, we compare several baseline methods with our proposed temporal graph learning architectures.We consider three baseline approaches:

- **Time-only models:** A Vector AutoRegressive Moving Average (VARMA) model and a Recurrent Neural Network (RNN) trained solely on global match statistics $\mathbf{s}_t$, without incorporating graph information.
- **Graph-only model:** A static Graph Attention Network (GAT) that aggregates all passes up to timestamp t into a single graph representation, following the approach proposed in [9].

Our main contribution lies in extending GNNs to account for temporal dependencies in evolving passing networks. We implement and compare three architectures:

- **Disjoint model (GAT + RNN):** Each passing network G_t^H and G_t^A is independently encoded using a GAT. The resulting team-level embeddings are concatenated with the global statistics vector $\mathbf{s}_t$ and processed by a Recurrent Neural Network (RNN) that maintains a hidden state across time. This setup models temporal dynamics after spatial aggregation.
- **Graph-RNN:** Temporal dependencies are integrated directly within message passing. The hidden representation of each node is carried over time, enabling the model to track player roles and interactions as the match progresses, thus coupling spatial and temporal reasoning.
- **Product Graph GCNN:** We construct a **Cartesian** product graph where each node represents a specific player at a particular time step. Running a GCNN on this expanded graph allows the model to jointly capture player interactions within a single time step and their evolution across time. The resulting representation aggregates both spatial and temporal information to predict the match outcome.

3 Numerical Experiments and Results

3.1 Experimental Setup

To prevent temporal leakage, matches are split by their unique identifiers into training and testing sets using an 80/20 ratio. At each interval t, the models generate a prediction for the match outcome, allowing us to track how predictive performance evolves over time.

We evaluate model performance using two complementary metrics: *Accuracy* and the *Ranked Probability Score* (RPS). Accuracy measures the correctness of discrete outcome predictions, while RPS evaluates the quality and calibration of the entire predicted probability distribution.

All models are trained[1] for $n = 50$ epochs using the Adam optimizer with a learning rate of $\lambda = 5 \times 10^{-4}$ and weight decay (L2 regularization) of 1×10^{-5}. To ensure stable and efficient training, we apply gradient clipping and use early stopping with a patience of 7 epochs. A learning rate scheduler is also employed to gradually reduce the learning rate during training. We focus on:

- **Model comparison.** The main experiment compares the proposed temporal graph models with the previously described baselines. Model performance is evaluated throughout the match in terms of Accuracy and RPS. All temporal graph models receive 5-minute windows as input.
- **Effect of time window length.** We analyze the influence of temporal resolution on model performance. *Disjoint* models are trained using different window lengths (5,10, and 15 minutes) to determine whether the way in which a match is segmented into temporal slices affects predictive accuracy.
- **Loss function comparison.** To study the impact of different optimization objectives, we train *Disjoint* models under three loss configurations: (i) cross-entropy loss ($\alpha = 1, \beta = 0$ in Equation 1), (ii) goal-based loss ($\alpha = 0.1, \beta = 1$), and (iii) a combined objective incorporating both terms ($\alpha = 1, \beta = 0.5$).
- **Limited history.** To evaluate the temporal receptive field of the model, we train *Product Graph GCNNs* using only the previous $[2, 3, 4, 5]$ windows for prediction. This experiment quantifies how far into the past the model must look to achieve strong predictive performance.
- **Ablation study.** We perform an ablation experiment on the *Disjoint* model to evaluate the contribution of general match statistics to predictive performance.

3.2 Results

Figure 1 shows that all models incorporating temporal information **outperform** the baseline approaches (*VARMA*, *RNN*, and static *GAT*) in the **early stages** of the match, from kickoff up to approximately the 45-50 minute mark. The *Disjoint* and *Graph-RNN* models perform particularly

[1] GitHub repository accessible at [14]

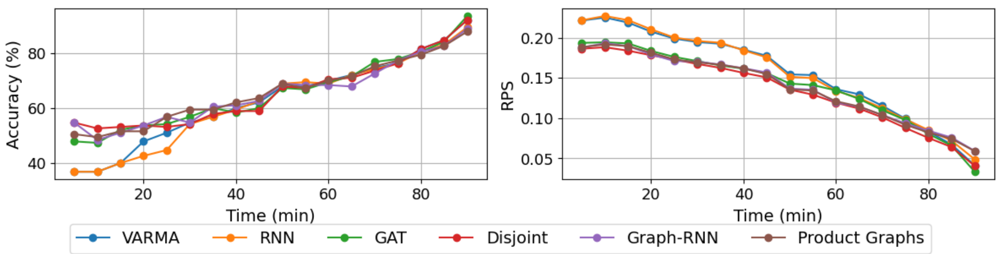

Figure 1: Results of the model comparison experiment.

well at the start, achieving around 55% accuracy from the opening minutes. The *Product Graph GCNN* becomes the strongest performer from the 20-minute mark until the midpoint of the match, after which all models converge to similar performance levels. Toward the end of the match, the static *GAT* and *Disjoint* models yield the highest accuracies. These results indicate that temporal modeling is especially beneficial for **early in-game predictions**, when match dynamics are still developing and uncertainty is high.

Figure S2 shows that model performance remains consistent across different temporal resolutions, with finer slicing (5-minute windows) producing marginally higher accuracy on average. This suggests that increasing temporal granularity offers limited but positive effects on predictive stability.

The comparison of different loss functions yields similar overall performance according to Figure S3, with goal-aware objectives performing slightly better in the early stages of the match. This pattern suggests that including goal-related information helps models interpret early-game conditions more effectively.

Using different numbers of previous windows as input shows minimal variation in overall performance. Figure S4 shows that models with shorter histories tend to produce more volatile accuracy curves, whereas longer histories lead to smoother and more stable predictions. This indicates that temporal dependencies are captured effectively even with limited past context.

Ablation Study

As seen in Figure S5, **removing global match features causes a clear drop in predictive performance**, confirming their importance for accurate outcome prediction. Despite this, the model trained solely on passing networks still achieves around 50% accuracy, well above the 33% random baseline, highlighting that passing structures themselves carry substantial predictive information. Given that real-world applications also have access to such match-level features, their inclusion remains justified.

Limitations and future work

While model selection was generally effective, certain frameworks may not have been optimally suited to the characteristics of our data. In particular, the *Graph-RNN* assumes a fixed node set across time steps, whereas in passing networks the on-field players can change due to substitutions. As a result, some temporal windows may contain more than eleven players, violating this assumption. To address this, we introduced *fake nodes* to ensure that all graphs share a consistent structure across time. Although this workaround allowed training to proceed, it may have introduced artifacts that affect temporal consistency and interpretability.

Another limitation concerns the experimental scope. Due to time constraints, some analyses were only performed on selected models rather than across all architectures. Future work should systematically extend these experiments to all model types.

4 Discussion

The results of this work demonstrate that **applying temporal graph-based models to passing networks improves the prediction of live match outcomes** compared to static baselines. This finding supports our initial hypothesis that temporal dynamics carry crucial information about team performance and match progression that cannot be captured by static representations alone.

Interestingly, the advantage of temporal models is most pronounced in the **first half of the match**, while performances tend to converge across models as the game progresses. This pattern suggests that temporal context is especially informative when little has yet unfolded on the field, but that later in the match, when outcomes are more constrained by the current score and remaining time, simpler models can perform comparably. Moreover, the results indicate that the **optimal model architecture depends on the prediction time**: the Disjoint and Graph-RNN models perform best for early-match predictions, whereas the Product Graph model excels mid-match, and static GNNs recover competitiveness towards the end. This observation implies that real-world applications could benefit from adaptive systems that select different models depending on the current stage of the match.

5 CRediT author statement

T.B Conceptualization, Methodology, Software (Data loaders, Experiments), Validation, Visualization. *78 commits*. **M.D** Conceptualization, Methodology, Software (GAT baseline, Disjoint Model), Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. *5 commits*. **S.G** Methodology, Software (GraphRNN and Product Graph), Data Curation. *13 commits*. **O.H** Methodology, Software (RNN and VARMA baselines), Investigation, Writing - Original Draft. *7 commits*. **O.M** Conceptualization, Methodology, Software (Data loaders, Disjoint Model, Experiments), Visualization. *14 commits*.

References

[1] Anito Joseph, Norman E Fenton, and Martin Neil. "Predicting football results using Bayesian nets and other machine learning techniques". In: *Knowledge-Based Systems* 19.7 (2006), pp. 544–553.

[2] Rahul Baboota and Harleen Kaur. "Predictive analysis and modelling football results using machine learning approach for English Premier League". In: *International Journal of Forecasting* 35.2 (2019), pp. 741–755.

[3] Josip Hucaljuk and Alen Rakipović. "Predicting football scores using machine learning techniques". In: *2011 Proceedings of the 34th International Convention MIPRO*. IEEE. 2011, pp. 1623–1627.

[4] Takahiro Kawasaki et al. "Football pass network based on the measurement of player position by using network theory and clustering". In: *International Journal of Performance Analysis in Sport* 19.3 (2019), pp. 381–392.

[5] Javier M Buldú et al. "Using network science to analyse football passing networks: Dynamics, space, time, and the multilayer nature of the game". In: *Frontiers in psychology* 9 (2018), p. 1900.

[6] Zeyu Wang et al. "Graph Neural Network Recommendation System for Football Formation". In: *Applied Science and Biotechnology Journal for Advanced Research* 3.3 (May 2024). DOI: `10.5281/zenodo.12198843`. URL: `https://doi.org/10.5281/zenodo.12198843`.

[7] Arash Beheshtian-Ardakani, Mostafa Salehi, and Rajesh Sharma. "CMPN: Modeling and analysis of soccer teams using Complex Multiplex Passing Network". In: *Chaos, Solitons & Fractals* 174 (2023), p. 113778.

[8] Pegah Rahimian et al. "Pass receiver and outcome prediction in soccer using temporal graph networks". In: *International Workshop on Machine Learning and Data Mining for Sports Analytics*. Springer. 2023, pp. 52–63.

[9] Jinmo Lee, Eunil Park, and Angel P del Pobil. "We know who wins: graph-oriented approaches of passing networks for predictive football match outcomes". In: *Journal of Big Data* 12.1 (2025), p. 147.

[10] Peter Xenopoulos and Claudio Silva. "Graph neural networks to predict sports outcomes". In: *2021 IEEE international conference on big data (big data)*. IEEE. 2021, pp. 1757–1763.

[11] *Football Statistics | Football Live Scores | WhoScored.com — whoscored.com*. `https://www.whoscored.com/`. [Accessed 26-09-2025].

[12] Pieter Robberechts, Jan Van Haaren, and Jesse Davis. "A bayesian approach to in-game win probability in soccer". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 3512–3521.

[13] Pegah Rahimian, Balazs Mark Mihalyi, and Laszlo Toka. "In-game soccer outcome prediction with offline reinforcement learning". In: *Machine Learning* 113.10 (2024), pp. 7393–7419.

[14] T. Bloemen. *Temporal Graph Machine Learning for Live Outcome Prediction in Football Using Passing Networks - GitHub Repository*. `https://github.com/sgol13/tudelft-graph-ml-football-predictions/tree/main`. 2025.

Supplementary Figures

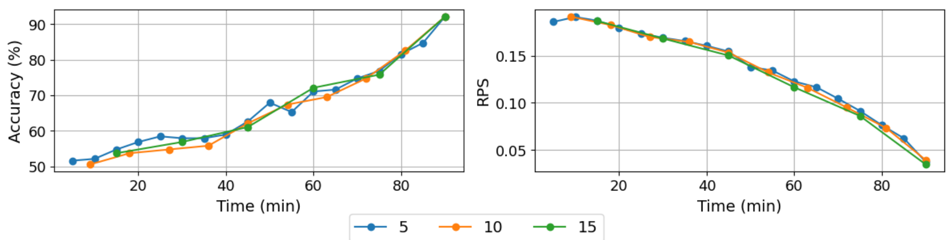

Figure S2: Results of the window size comparison experiment, where we train *Disjoint* models on passing networks resulting from slicing matches into windows of different sizes.

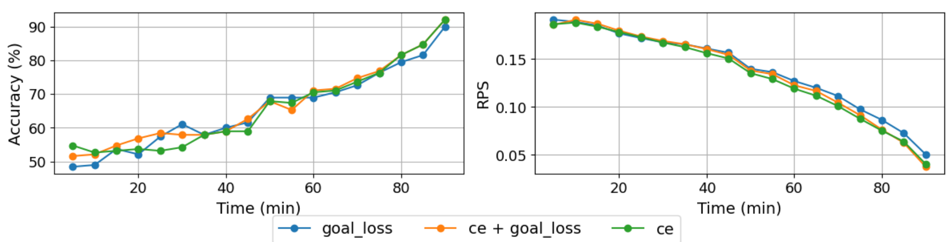

Figure S3: Results of the loss function experiment, where we train *Disjoint* models optimizing different loss functions.

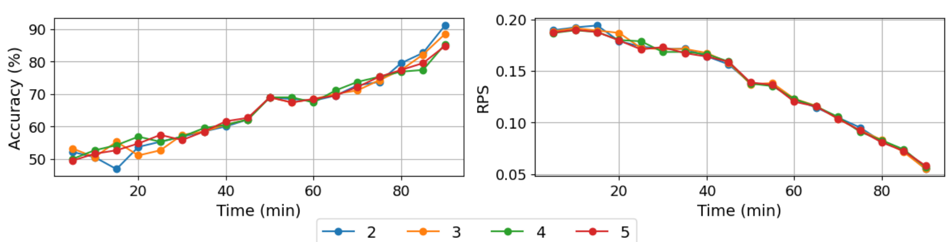

Figure S4: Results of the limited history experiment, where we train *GCNNs* on *product graphs* that use a different number of windows for prediction.

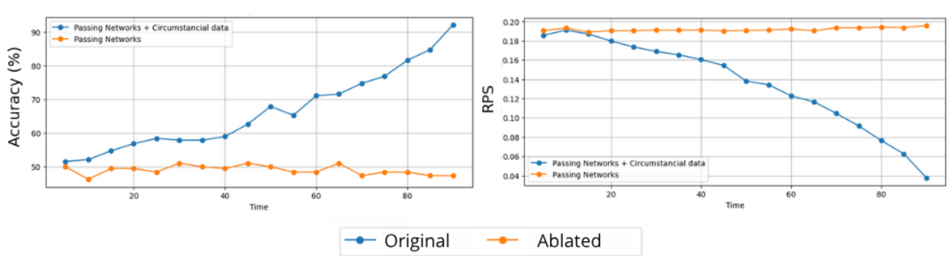

Figure S5: Results of the ablation study of the general match features on the *Disjoint* model.