Modeling Football Player Trajectories During Passes Using Graph-Structured Recurrent Networks

Kiernan McGuigan[1*] Jayden Hsiao[1*] K. Andrea Scott[1]
Sirisha Rambhatla[1] David A. Clausi[1] Lincoln Linlin Xu[2]

[1]University of Waterloo [2]University of Calgary

Abstract

Accurately forecasting player trajectories during the ball-in-flight phase of American football requires models that capture multi-agent interactions while remaining consistent with underlying physical constraints. This work evaluates how input normalization and target parameterization influence the performance of a Spatio-Temporal Graph LSTM model. Three normalization strategies are considered, along with two prediction targets, and the results show that RMS scaling combined with velocity prediction provides the highest accuracy, with an RMSE of 0.684. The findings indicate that normalization choices have a major effect on forecasting quality and that carefully selecting both the input scaling method and the target representation is essential for reliable trajectory prediction.

1. Introduction

Downfield passing plays create high variability in player movement and lead to some of the most consequential events in American football. Once a quarterback releases the ball, all players adjust their routes in ways that depend on spatial configuration, player roles, ball trajectory, and anticipation of potential outcomes. Accurately modeling these multi-agent trajectories requires capturing both temporal evolution and interactions among players.

The 2026 NFL Big Data Bowl Prediction Competition provides pre-pass tracking data and requests predictions of player positions for each frame of the ball-in-flight period. This problem requires:

- Understanding multi-agent dynamics and forecasting movement patterns under uncertainty.
- Producing a sequence of predicted (x, y) coordinates for each relevant player until the ball lands.

2. Related Works

Multi-Agent Trajectory Prediction. Forecasting interacting agents has advanced from social force models with hand-designed potentials [8] to deep recurrent architectures that pool neighbor information [1, 10] and generative models that produce multimodal future distributions [7]. These works establish that accurate prediction requires modeling both individual motion dynamics and interaction-driven variability.

Graph neural networks are now the dominant approach for relational agent modeling. Representing agents as nodes and spatial or semantic connections as edges allows message passing and attention mechanisms to capture proximity- and role-dependent dependencies that evolve over time [2, 9, 11]. This relational inductive bias is central to our Graph LSTM design.

Player Movement Modeling in Sports. Tracking data has enabled detailed analyses across sports, including defensive assignments in basketball [5], formation and pitch-control models in soccer [3], and route and coverage analyses in football [4, 6]. Football trajectory forecasting is particularly challenging because all players adjust the moment the ball is released, requiring models that capture rapidly evolving multi-agent dependencies.

3. Methodology

3.1. Dataset and Constraints

The dataset includes multiframe player tracking information recorded at ten frames per second. Each play provides pre-pass contextual features followed by the ball-in-flight period for which predictions are required. Only frames up to the moment of ball release are provided as inputs during training and evaluation. The forecasting target consists of the spatial trajectories of the targeted receiver and relevant defenders from ball release until pass completion or incompletion. An overview of the prediction task and the scoring protocol is shown in Figure 1.

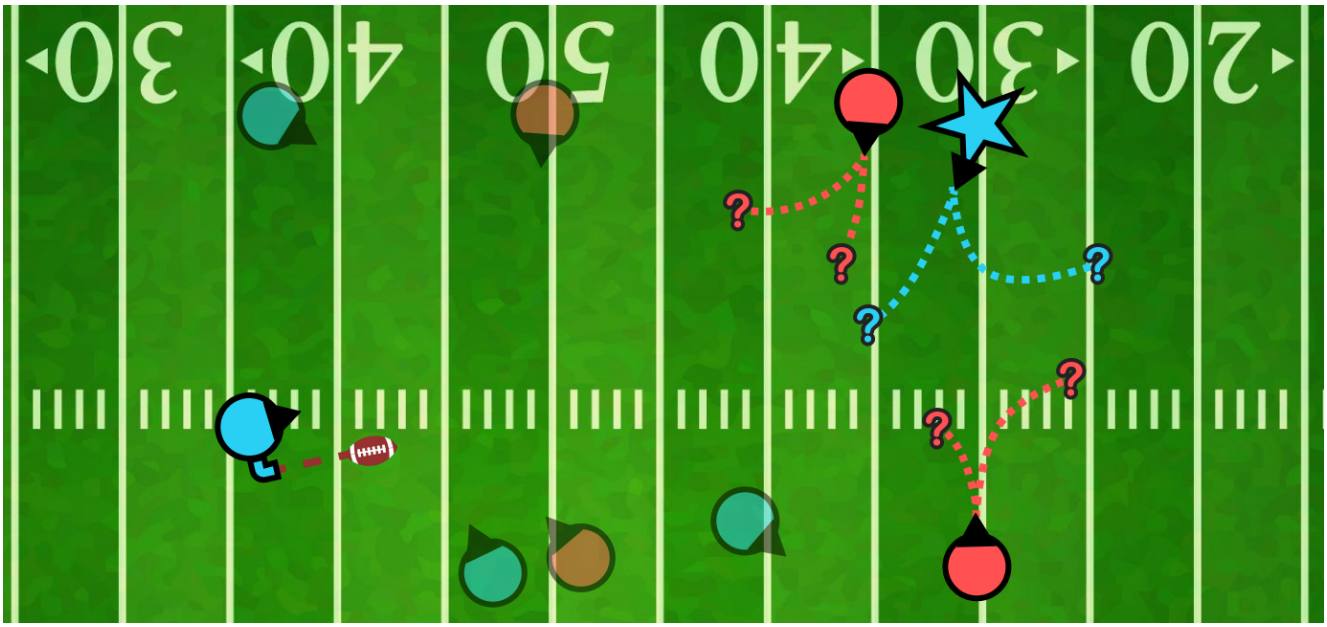

Figure 1. Illustration of the forecasting task. Player positions are observed until ball release, after which a model predicts player trajectories while the ball is in the air. Only the targeted receiver, symbolized with a star, and relevant defenders have their trajectories scored.

The data include the following major categories:

- **Identifiers** including game, play, and player identifiers for grouping sequences.
- **Player Attributes** such as position, physical measurements, team side, and role in the play.
- **Spatial Features** including absolute field coordinates, orientation, and directional movement.
- **Kinematic Features** including instantaneous speed and acceleration at each frame.
- **Contextual Features** summarizing play direction, situational variables, and ball landing location.
- **Targets** consisting of future (x, y) positions over the sequence of post-release frames.

Sequence lengths vary considerably across plays (Figure 2), requiring the model to handle variable-length inputs and outputs while maintaining consistent temporal alignment. The competition enforces a nine-hour compute budget, disables internet access during evaluation, and restricts external data to publicly available sources.

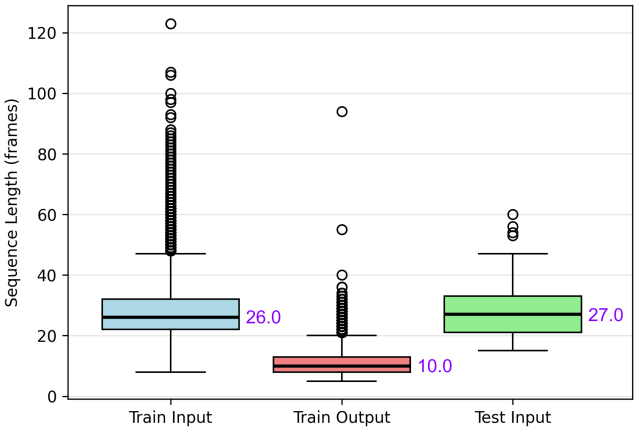

Figure 2. Summary box plots comparing train input, train output, and test input sequence lengths. The input covers frames prior to ball release; the output covers the ball-in-flight period. All plays are recorded at 10 fps. Median values are highlighted in purple.

3.2. Modeling

We formulate trajectory forecasting as a sequence-to-sequence task on a dynamic graph $G = (V, E)$. Given the state of all agents $\mathbf{X}$ for T_{in} frames prior to ball release, we predict the future positions $\mathbf{Y}$ for a target subset $V_{pred} \subseteq V$ (the targeted receiver and defenders) until the play concludes. The graph is fully connected, allowing the network to learn implicit dependencies between any pair of agents regardless of initial spatial proximity.

3.2.1. Spatio-Temporal Architecture

We propose a Spatio-Temporal Graph LSTM comprising an encoder–decoder structure, illustrated in Figure 3. Input tracking data (players $\times$ frames $\times$ features) is normalized before being processed by three parallel encoders whose outputs are fused and passed through stacked Graph LSTM layers. The decoder is initialized from the encoder's hidden states and generates future positions autoregressively.

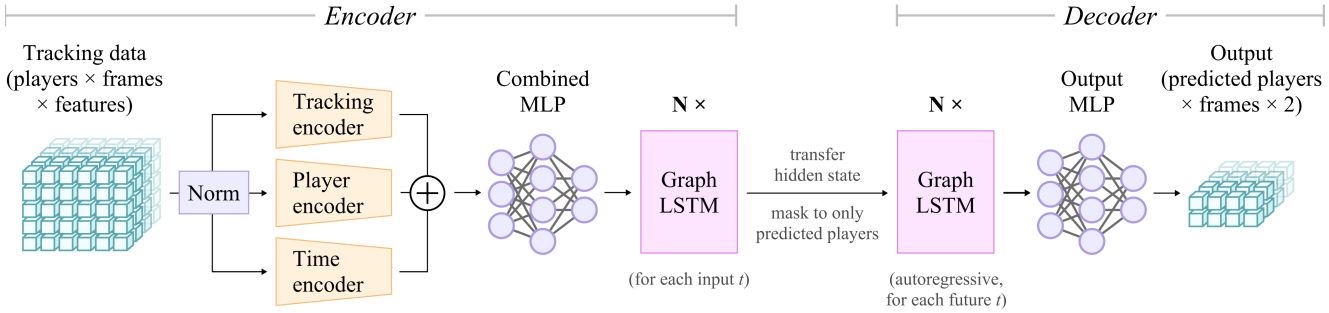

Figure 3. High-level architecture of the Spatio-Temporal Graph LSTM model. Input tracking data is normalized, encoded with tracking, player, and time encoders, fused through a combined MLP, and processed by N stacked Graph LSTM layers. Decoder states are initialized from the encoder and generate future positions autoregressively, operating only on the predicted players.

Encoder. We handle heterogeneous inputs via three separate encoding branches:

- **Tracking encoder:** Dynamic kinematic features (velocity, acceleration, orientation, etc.) and relative coordinates (distance to ball landing) are processed via a Multi-Layer Perceptron (MLP).
- **Player encoder:** Static attributes such as height, weight, and age are linearly projected, and learned embeddings are produced for a player's position, role, and side (offense or defense).
- **Time encoder:** A dedicated MLP encodes the frame index to preserve temporal fidelity across variable-length sequences.

The outputs of all three branches are concatenated and passed through a combined MLP to mix information across sources. The resulting representation is fed into N stacked Graph LSTM layers. Each layer integrates a Graph Attention Network (GATv2) with an LSTM cell: at each time step t, GATv2 aggregates spatial information using learnable attention weights, and the aggregated features update the LSTM hidden states, maintaining a memory of motion momentum. This structure is illustrated in Figure 4.

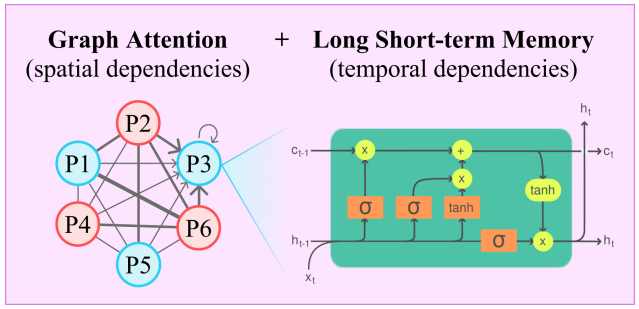

Figure 4. Illustration of the Graph Attention and LSTM components within each Graph LSTM block. The attention module models spatial interactions among players, while the LSTM cell captures temporal dependencies across frames.

Decoder. An autoregressive Graph LSTM decoder is initialized with the hidden states transferred from the encoder's LSTM cells. Unlike the encoder, the decoder operates only on V_{pred}. At each step, it predicts a velocity vector $\mathbf{s}_t$ (shift) rather than absolute coordinates. The position is updated through $\hat{\mathbf{y}}_t = \hat{\mathbf{y}}_{t-1} + \mathbf{s}_t$, stabilizing training by focusing on local dynamics.

3.3. Experiments

Submissions are evaluated using the root mean squared error over predicted and observed player positions. For a set of N prediction targets with true positions $(x_{t,i}, y_{t,i})$ and predicted positions $(x_{p,i}, y_{p,i})$, the metric is

$$\text{RMSE} = \sqrt{\frac{1}{2N}\sum_{i=1}^{N}[(x_{t,i} - x_{p,i})^2 + (y_{t,i} - y_{p,i})^2]}.$$

All models are trained with 10 GB of VRAM and 10 CPU cores, for up to 400 epochs with a patience of 16 epochs for early stopping, using $N = 2$ Graph LSTM layers.

3.3.1. Normalization Strategies

Two important factors to consider when modeling human tracking data are (1) providing the model with similarly distributed inputs, to avoid undue importance being assigned to variables based on their individual scales, and (2) maintaining the physical relationships between variables, to avoid breaking kinematic constraints. This study evaluates three normalization approaches crossed with two target formulations to determine the optimal configuration.

Field Scaling (Baseline) Positions are divided by the field dimensions to place them in a $[0, 1]$ range. All other features such as speed and acceleration are left unscaled. This serves as a baseline to highlight potential gains from more

thorough normalization. Two instances are evaluated predicting either player velocity or acceleration. When velocity is predicted it is multiplied by the inter-frame time interval to obtain displacement. When acceleration is predicted, it is used to update a stored velocity which is then used to compute displacement.

Max Scaling Features are divided by their known maximum values; positions continue to use field scaling. A benefit of this approach is that the resulting bounds can be used to enforce physical plausibility constraints: the known maximum human speed and acceleration prevent the model from predicting physically impossible movements. Two instances are again evaluated, one predicting velocity and the other predicting acceleration.

RMS Scaling Positions are centered at the field midpoint and divided by half the field size, placing them in a $[-1, 1]$ range. All other features (speeds, accelerations, distances) are divided by the root-mean-square (RMS) value of those features. This approach attempts to balance pushing input variables toward similar distributions while preserving the proportional mathematical relationships that inherently exist among the tracking variables. The target variables are again either the velocity or acceleration of a player.

4. Results

Normalization	Target Var.	RMSE ↓
Field	Velocity	0.755
Field	Acceleration	0.936
Max	Velocity	0.702
Max	Acceleration	0.911
RMS	Velocity	**0.684**
RMS	Acceleration	0.699

Table 1. Test root-mean-square error (RMSE) for Graph LSTM trajectory forecasts under six configurations combining three input-normalization methods with two target formulations. Errors represent average positional deviation in yards for all predicted post-release frames.

The results reveal several consistent trends:

- RMS scaling across all features achieves the lowest errors.
- Simple field scaling is less effective than balanced feature normalization.
- Velocity as a prediction target outperforms acceleration under all normalization conditions.

RMS scaling stands out because it reduces feature-scale disparities without distorting the proportional relationships among spatial and kinematic variables. This consistency appears important for graph-recurrent networks, which rely on coherent feature geometry to maintain stable attention weights and smooth temporal dynamics.

Acceleration-based targets perform worse across all normalization conditions. This gap reflects a deeper modeling challenge: acceleration is a second-order quantity affected by factors absent from the dataset, such as cleat–turf friction, elastic energy return, and instantaneous gait adjustments. Asking the model to predict acceleration therefore requires it to infer unobserved forces, and the resulting errors compound when positions are reconstructed through double integration. Velocity targets avoid this amplification because short-horizon motion is sufficiently smooth for a first-order approximation, and the network can leverage local continuity directly. The narrower performance gap under RMS scaling suggests that normalization reduces noise in the acceleration signals, but not enough to overcome the structural difficulty of the task.

Overall, the results indicate that preprocessing choices must respect both the statistical properties of the inputs and the physical constraints governing athlete motion. Misaligned choices in either direction can overwhelm any gain achieved by model complexity.

5. Conclusion

This study evaluates how normalization strategy and target formulation influence trajectory forecasting in American-football tracking data. Three key findings emerge:

- **RMS scaling preserves physical relationships** while reducing feature-scale disparities, making it particularly effective for graph-recurrent networks.
- **Acceleration-based targets introduce additional uncertainty** and integration error, limiting predictive accuracy relative to velocity targets.
- **Graph LSTM effectively leverages multi-agent interactions**, demonstrating the value of relational modeling for trajectory forecasting in sports.

These findings underscore that improvements in feature design and target parameterization can yield substantial accuracy gains without altering the model architecture. Future work should investigate hybrid target representations that encode physically meaningful constraints, as well as regularization strategies that enforce plausible motion dynamics. Approaches that incorporate uncertainty estimation or physics-aware penalties may further improve reliability in complex multi-agent environments.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. pages 961–971, 2016. 1

[2] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and koray kavukcuoglu. Interaction Networks for Learning about Objects, Relations and Physics. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 1

[3] Alina Bialkowski, Patrick Lucey, Peter Carr, Iain Matthews, Sridha Sridharan, and Clinton Fookes. Discovering Team Structures in Soccer from Spatiotemporal Data. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2596–2605, 2016. 1

[4] B. Burke and Espn Analytics. DeepQB: Deep Learning with Player Tracking to Quantify Quarterback Decision-Making & Performance. 2019. 1

[5] Daniel Cervone, Alex D'Amour, Luke Bornn, and Kirk Goldsberry. A Multiresolution Stochastic Process Model for Predicting Basketball Possession Outcomes. *Journal of the American Statistical Association*, 111(514):585–599, 2016. arXiv:1408.0777 [stat]. 1

[6] Joachim Gudmundsson and Michael Horton. Spatio-Temporal Analysis of Team Sports. *ACM Comput. Surv.*, 50 (2):22:1–22:34, 2017. 1

[7] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks, 2018. arXiv:1803.10892 [cs]. 1

[8] Dirk Helbing and Peter Molnar. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51(5):4282–4286, 1995. arXiv:cond-mat/9805244. 1

[9] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, 2017. arXiv:1609.02907 [cs]. 1

[10] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents, 2017. arXiv:1704.04394 [cs]. 1

[11] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks, 2018. arXiv:1710.10903 [stat]. 1