ATTENTION IS ALL YOU NEED, FOR SPORTS TRACKING DATA

PREPRINT

Udit Ranasaria
SumerSports
udit.ranasaria@sumersports.com

Pavel Vabishchevich
SumerSports
pavel.vabishchevich@sumersports.com

August 20, 2024

ABSTRACT

The rapid advancement of spatial tracking technologies in sports has led to an unprecedented surge in high-quality, high-volume data across all levels of play. While this data has catalyzed innovations in various domains of sports analytics, current methodologies often struggle with the inherent challenges of sports tracking data, particularly the player-ordering problem. This paper showcases the application of Transformer architectures to address these challenges in sports analytics. Our approach operates end-to-end on raw player tracking data, processes unordered collections of player vectors, and is inherently designed to learn pairwise spatial interactions between players. We argue that this framework satisfies critical criteria for being widely adopted in sports modeling: minimal feature engineering, adaptability across diverse problems, and accessibility in terms of understandability and reproducibility. Specifically, while our highlighted approach is presented for its task-agnostic nature, we demonstrate its effectiveness over other commonly used approaches at predicting tackling location in the NFL, a recently and prominently explored task in the public domain on Kaggle. This work aims to catalyze a paradigm shift in sports analytics research methodologies, moving from traditional models to Transformer-based architectures, potentially unlocking new insights into player dynamics, team strategies, and game outcomes across various sports domains.

Keywords sports analytics · sports tracking data · latent representation learning · multi-agent spatiotemporal modeling

1 Introduction

The prominence of sports analytics has been accelerating rapidly, propelled by unprecedented advancements in spatial tracking technologies. This surge in high-quality, high-volume data acquisition—facilitated by optical systems, computer vision algorithms, and chip-based tracking solutions—is permeating across all sports and levels of play. The confluence of this rich data with innovative research and sophisticated modeling techniques precipitates disruptions in multiple domains, including sports science, broadcasting, player evaluation, team building optimization, injury prevention, and tactical strategy formulation.

Kovalchik [2023] comprehensively delineates the burgeoning corpus of innovative research contributions leveraging sports tracking data, with the most advanced and modern modeling emphasis in Latent Variable Estimation, Event Prediction, and Value Attribution. Despite the sophistication of these methodologies, their potential is often hampered by their reliance on traditional approaches to circumvent the *player order problem*. This challenge arises from the dynamic nature of team sports, where the roles and formations of players are inconsistent and can vary between games. The absence of a persistent intrinsic order of players in different intervals of play or games conflicts with the input structure requirements of many standard machine learning models, necessitating hand-crafted feature preprocessing steps to transform raw tracking data into structured feature representations or the imposition of heuristic-based orderings [Horton, 2020].

These hand-designed feature extractors or ordering approaches, while functional, rely on domain expertise and lack generalizability. Moreover, with the proliferation of deep learning, feature engineering has become an anti-pattern[1] to give way to end-to-end learning [LeCun et al., 2015, Ng, 2018]. These advocates for end-to-end learning emphasize that, given sufficient data, models should aim to learn latent features directly from raw inputs, optimizing against training objectives. Consequently, modeling advances in deep learning research typically stem from architectural innovations tailored to the data space rather than feature extraction techniques.

Furthermore, we posit that the majority of modeling tasks involving sports player tracking share a fundamental objective: learning pairwise spatial interactions between players. Models adept at capturing these interactions are likely to excel in event prediction and other supervised tasks that necessitate a nuanced understanding of player positioning within the context of sport-specific rules and dynamics. This observation motivates our exploration of novel architectural approaches that can inherently handle unordered sets of player data while capturing complex spatial relationships.

To address these challenges, this paper highlights the application of Transformer architectures as a modeling framework for sports analytics. Transformers, originally developed for natural language processing tasks, have shown remarkable capabilities in handling sequential and unordered data across various domains. Our proposed approach satisfies several critical criteria:

1. End-to-end operation on raw player tracking data with minimal feature engineering, ensuring flexibility and adaptability across diverse sports modeling problems.
2. Capability to process unordered collections of player vectors, directly addressing the player-ordering problem.
3. Inherently designed for learning pairwise player interactions.
4. Accessibility in terms of understandability, explainability, and reproducibility.

Our objective is to catalyze a paradigm shift in sports analytics research methodologies. We anticipate a transition from traditional models like XGBoost and MLPs, commonly employed in competitions such as the NFL Big Data Bowl, towards variants and extensions of Transformer architectures. This shift has the potential to unlock new insights in player dynamics, team strategies, and game outcomes, ultimately advancing the field of sports analytics and its applications across various domains.

2 Methods

2.1 Sports Tracking Data

Table 1: Example of a Multi-Agent Tracking Frame from the NFL

frame_id	**event**	**nfl_player_id**	**team**	**x**	**y**	**s**	**o**	**dir**
15	handoff	34452	OFF	26.87	27.9	3.0	274.91	242.98
15	handoff	40089	OFF	35.19	34.03	0.51	246.43	84.06
15	handoff	42368	DEF	40.37	25.44	6.15	304.19	297.75
⋮								
15	handoff	54948	DEF	35.33	31.36	2.67	272.22	261.64

Table 1 shows a sample of data from the 2024 NFL Big Data Bowl. This spatiotemporal data in team sports is usually characterized by

- A frame number or time column
- A player identifier
- An indicator for which team a player plays on
- An event stream that annotates "actions" happening at a given moment in time in the sport
- Feature columns representing spatial properties of the player such as position or velocity.

[1] Feature engineering or order imposition should be viewed as a form of regularization that is needed in specific scenarios (e.g., data scarcity or overfitting concerns) rather than as a general starting point for large data modeling.

This paper is focused on modeling the static *multi-agent* aspect of tracking data that exists within a single frame (or timestamp), and thus we will assume that each tracking frame at a given timestamp represents a unique, independent training sample disconnected from the frames temporally around it. We also note that while the unordered multi-agent nature of a tracking frame is largely consistent across team sports, the nature of the time dimension differs significantly as some have an inherently stop-start nature like American football whereas others are nearly fully continuous like soccer. A framework to model generally over the time dimension is explicitly beyond the scope of this paper.

2.2 Task Formulation

Let $P = \{p_1, p_2, ..., p_K\}$ represent the set of K players participating in a particular frame. Similarly, let $V = \{v_1, v_2, ..., v_K\}$ be the set of feature vectors such that each $v_k \in \mathbb{R}^d$ captures all relevant spatial (e.g. position and velocity) and characteristic (e.g., height and weight) features for the player p_k. Notably, both P and V are *unordered* as for most team sports there exists no natural ordering for players.

We now describe a generic supervised learning task over V. Let $y \in \mathcal{Y}$ be the objective label we aim to predict, where $\mathcal{Y}$ is the set of possible outcomes. This could represent various tasks such as predicting future events or outcomes.

We define a model $f : (\mathbb{R}^d)^K \to \mathcal{Y}$ as a function that maps the set of feature vectors V to the label space $\mathcal{Y}$. Formally,

$$\hat{y} = f(V) = f(\{v_1, v_2, ..., v_K\}) \tag{1}$$

where $\hat{y}$ is the predicted label.

To train this model, we define a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that measures the discrepancy between the true label y and the predicted label $\hat{y}$. The optimization problem can then be formulated as:

$$f^* = \arg\min_f \mathbb{E}_{(V,y)\sim\mathcal{D}}[\mathcal{L}(y, f(V))] \tag{2}$$

where $\mathcal{D}$ represents the underlying data distribution from which our training samples are drawn.

2.3 Notable Prior Work

As discussed in the Introduction1, often sports researchers identify this player ordering issue and then decompose the modeling as:

$$f(V) = g(\Phi(V)) \tag{3}$$

where $\Phi : (\mathbb{R}^d)^K \to \mathbb{R}^m$ is a feature extraction process that maps the unordered set of player feature vectors to an ordered fixed-dimensional representation, and $g : \mathbb{R}^m \to \mathcal{Y}$ is typically implemented using MLPs or gradient boosted tree models. Note that just applying a fixed ordering over V based on domain heuristics is still considered a special case of Φ where: $\Phi_{\text{fixed}} : (\mathbb{R}^d)^K \to \mathbb{R}^{d \cdot K}$.

Here are some specific notable examples that follow this decomposition paradigm:

- Both Fernández et al. [2019] and Yurko et al. [2020] propose novel frameworks to decompose the complex sports of soccer and American football into a continuous time value based metric. However both papers invest heavily in deriving "a wide set of spatiotemporal features" to feed individual models that comprise the frameworks.
- Amirli and Alemdar [2022] in building a model to infer ball location from tracking data identify that it is "it is impossible to find a correct ordering for the individual players to be represented in the feature matrix" and implement a segment-based role assignment algorithm to fix an order and extract features before feeding into an MLP.
- Both Le et al. [2017] and Schmid et al. [2021] employ deep imitation learning to learn "ghosting" and evaluate team strategy in soccer and American football, respectively. Yet, to featurize a tracking frame as input into the recurrent nets they relied a role-based assignment step to "[provide] additional context to impose ordering on the training input".
- Felsen et al. [2018] built a Conditional Variational AutoEncoder model capable of synthetically generating basketball player trajectories conditioned on identity and context. Yet, they had to separately develop a algorithm to solve for the "significant challenge in encoding multi-agent trajectories is the presence of permutation disorder"
- Mehrasa et al. [2018] innovates with convolutional network filtering over the time dimension but to avoid "implicitly enforce an order among this set of players" they use an anchor based sorting scheme.

This collection is only a subset and serves to justify one of the premises of this paper: advanced and modern sports research is often encountering this player-ordering problem and solving them in imperfect ways. Given the patterns over past work in expect an end-to-end deep learning solution to perform better by discovering latent features from the data.

2.4 General Transformer Model Architecture

The Transformer [Vaswani et al., 2017] revolutionized Natural Language Processing by introducing a self-attention mechanism that enables learning from direct pairwise interactions between all elements in a sequence, regardless of their order. This approach effectively addresses the challenge of modeling long-range dependencies, a limitation of commonly used recurrent models.

Crucially, while the self-attention operation is permutation-invariant, the overall Transformer architecture maintains permutation equivariance: any permutation of the input sequence results in the same permutation of the output embeddings, without affecting the values of the embeddings in any way. This property is *exactly* what is needed to directly learn over the unordered set of player feature vectors end-to-end while capturing player interaction relationships.

We define our Transformer-based model as:

$$f(V) = g(\text{TransformerEncoder}(V)) \tag{4}$$

where the TransformerEncoder function can be expressed as:

$$\text{TransformerEncoder}(V) = \text{LayerNorm}(V + \text{FFN}(\text{LayerNorm}(V + \text{MultiHead}(V, V, V)))) \tag{5}$$

The function g is a problem-specific "decoder" pooling + MLP layer that maps the Transformer Encoder's learned salient latent player embeddings to the desired label space $\mathcal{Y}$. The pooling operation is necessary primarily when we need to aggregate information across all latent player embeddings for a single shared prediction. In Figure 1, we visualize this architecture at a high-level demonstrating its generalizability across problems in Sports Tracking. [2].

2.5 Other Related General Equivariant Work

This paper is not the first example of developing general-purpose Deep Learning frameworks over unordered sets in the sports player tracking domain. We compare them here to the Transformer approach.

2.5.1 DeepSets

Horton [2020] targeted this problem of proposing a canonical end-to-end modeling of raw trajectory data for the purpose of learning latent representations generally across problems and sports. This paper, released before Transformers demonstrated widespread modeling success outside of the NLP domain, used an equivariant architecture *DeepSets* [Zaheer et al., 2018] that relies on global pooling of element-wise transformations.

While DeepSets offers permutation equivariance, it is likely inferior to Transformers in sports modeling scenarios where the set size is small but require deep, long-range pattern recognition. Transformers' self-attention mechanism allows for more nuanced interactions between all players, capturing complex spatial and temporal dependencies crucial in team sports. Moreover, Transformers have become ubiquitous across various domains, leading to extensive research, optimizations, and pre-trained models, which DeepSets lack.

2.5.2 Graph Neural Nets

Yeh et al. [2019] proposed using graph neural networks (GNNs) as a permutation-equivariant method to model over unordered players where each player represents a node in a fully connected graph.

While GNNs offer a natural way to model relationships between players, Transformers present several advantages in the context of sports modeling with fully connected player interactions. Firstly, in a fully connected scenario, the multi-hop message passing of GNNs becomes redundant, as all nodes are directly connected. Transformers, with their self-attention mechanism, can model these direct interactions more efficiently. Secondly, the sparsity benefits typically associated with GNNs are nullified in a fully connected graph, negating one of their key advantages. Alcorn and Nguyen [2021] also demonstrates empirically that Transformers outperform this GNN approach.

[2] While we do not dive into the details of the internal pieces of the Transformer, we note that they have been proven to be highly effective learners in many domains. For a comprehensive understanding of Transformer internals, we recommend many of the excellent public sources on the subject

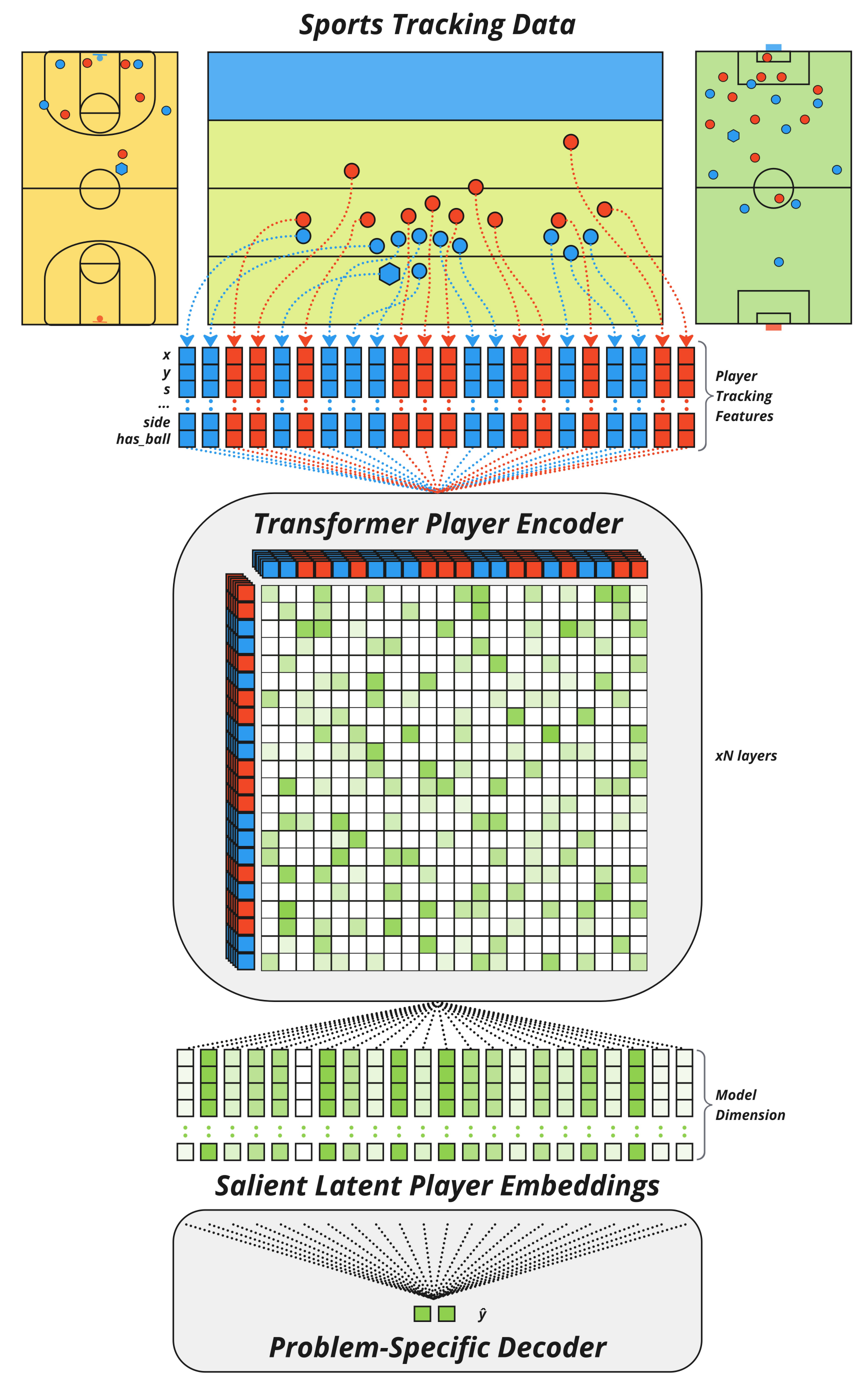

Figure 1: A generalized end-to-end Transformer Encoder modeling solution for sports tracking data analysis. The architecture consists of: (1) an input layer ingesting raw tracking features as unordered player vectors, (2) a Transformer Player Encoder that transforms these into salient player embeddings through repeated Multi-Head Attention Transformer layers, and (3) a problem-specific decoder that pools information (if needed) from the embeddings to learn a final $\hat{y}$.

A single head of self-attention is visualized in the center of the figure. This is the key aspect of why this approach fits the multi-agent problem. Each player vector is updated with information from each other player in a weighted manner as the model learns to identify important patterns for the objectives. The process of players attending to each other creates rich embeddings, which we believe is a common pattern across most tracking sports modeling tasks.

2.5.3 The Zoo

The Zoo [2020] is a research group that developed a model architecture that won the 2020 NFL Big Data Bowl[3] in Kaggle challenge. This victory, along with the Big Data Bowl's growing prominence, led to The Zoo Architecture (TZA) becoming a de-facto equivariant deep learning approach used by entrants in following competitions. Furthermore, this architecture powers models and stats distributed by the NFL's advanced wing *Next Gen Stats*.
In their submission, they identified the importance of designing for the player-order equivariance problem but discovered that their custom equivariant TZA design *outperformed* Transformers with Multi-Head Attention. TZA, as shown in their architecture diagram2, relies on feature engineering vectors pairwise between each offensive and defensive player, and then operating dense layers over each pairwise vector *independently*, with pooling operations to eventually reduce the dimensionality into one final prediction. While not explicitly cited, we find many similarities between TZA and the DeepSet approach. We expect architecturally TZA to be inferior for similar reasons and note that TZA is not an end-to-end approach.
Furthermore, in our Experiments 3, we demonstrate confirm this inferiority empirically in a similar, but slightly more general task than what TZA was originally optimized for. We find that while TZA may have achieved impressive performance for that original dataset and specific task, it does not extend generally to other problems, even those that are just slightly different.

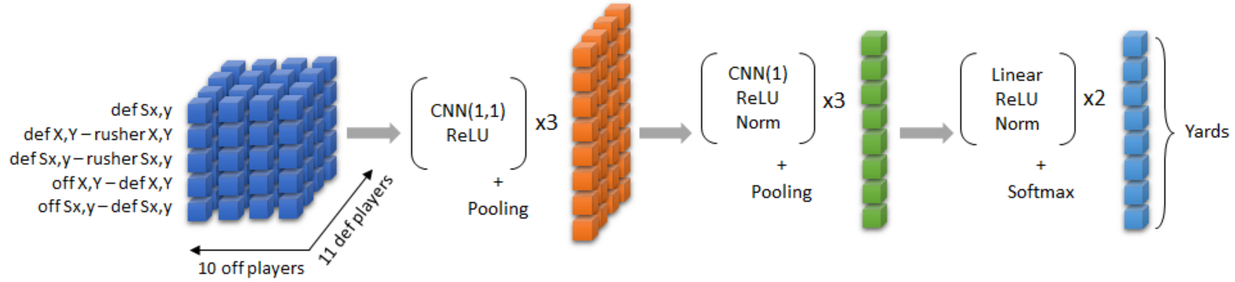

Figure 2: Simplified Structure of The Zoo Architecture that rose to prominence in modeling spatiotemporal NFL tracking data. It was designed to predict a categorical distribution over the number of yards gained on rushing plays using tracking frames at time of handoff. The model relies on manually constructing "interaction" feature vectors pairwise between each offensive and defensive player. The model then essentially treats these "interaction" feature vectors as independent throughout, with no mechanism to learn across player dimension. After applying a few dense layers to each interaction vector, pooling is applied to collect the most salient learned features across the offensive player and then defensive player dimension.

3 Experiments

The model code, data, and results for our experiments will be available shortly.

3.1 Dataset

While we believe the Transformer is useful for its task-agnostic nature for sports, we demonstrate its efficacy on a large set of recently released public sports tracking data in the 2024 NFL Big Data Bowl[4]. This dataset includes the location, speed, and orientation of all 22 players on the field for Weeks 1-9 of the 2022 NFL Season. The theme of this competition was tackling and thus the only tracking frames provided are those where there is a clear ball-carrier and the defense is focused on the task of tackling. This means our total dataset includes 136 games, $\sim 2,000$ unique plays, and $\sim 80,000$ frames.

We choose our modeling objective to be predicting the (x, y) position of the tackle. This was chosen because it aligns with the dataset but is also very similar to what TZA was designed for. The 2 notable differences are:

1. The dataset TZA used in 2020 only contained frames from the *handoff* event. This means that with our experiment we are performing a similar task but across many more varied football situations where the ball

[3] https://www.kaggle.com/c/nfl-big-data-bowl-2020/overview

[4] https://www.kaggle.com/competitions/nfl-big-data-bowl-2024/data

carrier could have just caught a pass on the sideline, be mid-scramble, or already be breaking away for a huge gain.

2. TZA's modeling objective was classifying the tackle location as categorical of yards gained. Instead, we are treating it as a regression problem and asking the models to predict both the horizontal and vertical location of the eventual tackle.

As is often standard practice in NFL modeling, we standardize our data so that the offense is always moving the right and our (x, y) positions based on the *play origin* [5]. We also augment our data by mirroring across the y-axis, effectively doubling our training data frames. We do not use the ball tracking data for our experiments, instead using a column identifying which player is currently the ball carrier.

We note that while each frame is being treated as an independent input, the truth labels for tackle location are unique at a play level. Thus, we used a training/validation/test split of 70/15/15 over unique plays. Our train dataset has around unique 9k plays and 750k frames while our test and validation have around 2k plays and 150k frames.

3.2 Model

We compare our proposed Transformer architecture against a baseline of The Zoo Architecture[6]. We design the model classes such that they share the same AdamW optimizer, learning rate, and loss calculation. We used the Smooth L1 Loss function with $\beta = 1.0$ defined as $\mathcal{L} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ between the true tackle location $y = (x_{tackle}, y_{tackle})$ and the predicted tackle location $\hat{y}$:

$$\mathcal{L}(y, \hat{y}) = \text{SmoothL1Loss}((x_{tackle}, y_{tackle}), (\hat{x}_{tackle}, \hat{y}_{tackle})) \tag{6}$$

3.2.1 Transformer Model Experiment

For the Transformer model in this experiment, we define the set of feature vectors $V = \{v_1, v_2, ..., v_K\}$ over the $K = 22$ players, where each $v_k \in \mathbb{R}^6$ represents the features of player p_k. Specifically, each feature vector v_k is composed of:

$$v_k = [x_k, y_k, vx_k, vy_k, o_k, b_k] \tag{7}$$

where:

- (x_k, y_k) represent the spatial coordinates of player p_k
- (vx_k, vy_k) represent the velocity components of player p_k
- $o_k \in \{0, 1\}$ is a binary indicator for offense (1) or defense (0)
- $b_k \in \{0, 1\}$ is a binary indicator for whether player p_k is the ball carrier (1) or not (0)

The label space $\mathcal{Y}$ for our task is the predicted tackle location, defined as $\mathcal{Y} = \mathbb{R}^2$, representing the (x, y) coordinates of the predicted tackle.

The TransformerEncoder model $f : (\mathbb{R}^6)^{22} \to (\mathbb{R}^d)^{22}$ maps the set of player feature vectors V equivariantly into a set of player embeddings of size model dimension d. Then we apply the task-specific decoder $g : (\mathbb{R}^d)^{22} \to \mathbb{R}^2$ as an average pooling over the players followed by an MLP to get the predicted tackle location:

$$\hat{y} = (\hat{x}_{tackle}, \hat{y}_{tackle}) = g(f(V)) = g(f(\{v_1, v_2, ..., v_{22}\})) \tag{8}$$

3.2.2 The Zoo Architecture Model Details

For our TZA, we have to perform a complex pairwise feature interaction process $\Phi : V \to (\mathbb{R}^{10})^{10 \cdot 11}$ that converts $V = \{v_1, v_2, ..., v_K\}$ over the $K = 22$ players into 110 interaction vectors. Specifically, each interaction vector u_{ij} between offensive player p_i and defensive player p_j with ball carrier p_b is composed of:

$$u_{ij} = [vx_j, vy_j, x_j - x_b, y_j - y_b, vx_j - vx_b, vy_j - vy_b, x_i - x_j, y_i - y_j, vx_i - vx_j, vy_i - vy_j] \tag{9}$$

where:

- (x_k, y_k) represent the spatial coordinates of player p_k

[5] The play origin is the location of the ball at the time of snap. This standardizes tracking data around 0 more and serves as a preprocessing normalization that helps training efficiency

[6] We couldn't find code from the original authors but referred to a publically reproduced implementation here: `https://github.com/juancamilocampos/nfl-big-data-bowl-2020/blob/master/1st_place_zoo_solution_v2.ipynb`

- (vx_k, vy_k) represent the velocity components of player p_k
- p_b is the ball carrier

Then TZA model $g : (\mathbb{R}^{10})^{10 \cdot 11} \rightarrow \mathbb{R}^2$ applies successive MLPs to interaction vectors independently, only pooling across to reduce dimensionality.

4 Results

Table 2: Test Set Event-Frame Performance Comparison (Mean Squared Error)

event	n	Transformer	Zoo	% diff
ball snap	1916	66.3	67.1	1.2
handoff	1776	39.7	40.7	2.5
pass caught	1684	16.0	18.2	12.1
first contact	3156	9.0	12.5	28.0
out of bounds	544	3.7	13.0	71.5
tackle	2994	1.5	9.2	83.7

Over our held-out test-set we calculate Mean Squared Error (MSE) to find that the Transformer performs 16.1% better overall. In Table 2 we break down the MSE by frames which had a tagged event. We observe that the model performance difference is actually near negligible for the events *ball snap* and *handoff*. Notably, *handoff* was the specific modeling problem that TZA was optimized for and this provides credence to the idea that TZA was a very compelling architecture for that specific problem. When we look at other events, the performance gap between the models expands considerably. Notably, the events of *out of bounds* and *tackle* are frames where we'd expect the model to have a relatively easy task since the target is just the current ball carriers location, as long as they can identify that the play is likely near over. The Transformer model expectedly reduced its error as frames arrived closer to the tackle time whereas TZA model seems like it was unable to generalize to these various situations. Figure 3 shows a visual example of the Transformer generalizing to unseen frames in the test set significantly better than TZA is able to.

5 Conclusion

In summary, the adoption of transformers in sports data modeling promises to address the limitations of existing methodologies by providing a simple, generalized, and scalable framework. Our methodology demonstrates the superior performance of transformer-based models compared to traditional approaches like The Zoo Architecture, highlighting their potential in capturing complex spatial interactions with minimal feature engineering. This paradigm shift could facilitate more robust and flexible analyses, ultimately advancing the field of sports data science.

5.1 Further Work

It was with intention that we kept the experiments and scope of this paper narrow. We wanted to stoke interest in Transformers as a powerful tool to be used in sports modeling, but leave plenty of room for innovation and further work to extend this. For example, while we have made claims about the generalizability of this approach we only had the resources to explore one application in this paper in one sport. We welcome additional effort to rigorously compare it to solutions that are not end-to-end and in other data spaces.

References

Stephanie A. Kovalchik. Player tracking data in sports. *Annual Review of Statistics and Its Application*, 10(Volume 10, 2023):677–697, 2023. ISSN 2326-831X. doi:https://doi.org/10.1146/annurev-statistics-033021-110117. URL https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-033021-110117.

Michael Horton. Learning feature representations from football tracking. In *MIT Sloan Sports Analytics Conference*, Boston, March 2020. MIT sloan sports analytics conference Boston, MA, USA. URL https://www.sloansportsconference.com/research-papers/learning-feature-representations-from-football-tracking. Presented on March 6–7, 2020.

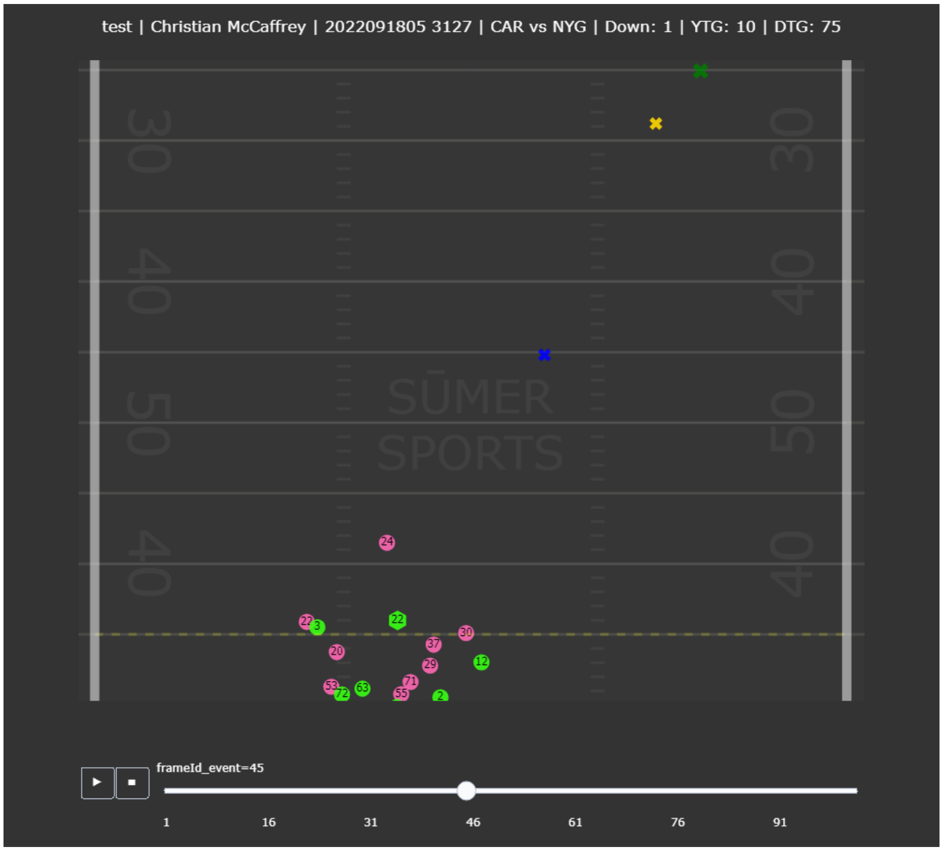
test | Christian McCaffrey | 2022091805 3127 | CAR vs NYG | Down: 1 | YTG: 10 | DTG: 75
30
40
50
40
30
40
50
40
SŪMER
SPORTS
24
22
3
22
30
37
20
29
12
53
72
63
71
55
2
frameId_event=45
1
16
31
46
61
76
91

Figure 3: Visualization of a frame from the test set showing improved generalization from the Transformer. The green hexagon (#22) is the ball-carrier, green cross is the true tackle location. Yellow and blue crosses represent predictions from Transformer and Zoo models respectively

Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015. doi:10.1038/nature14539.

Andrew Ng. *Machine Learning Yearning*. deeplearning.ai, Mountain View, CA, 2018. URL `https://info.deeplearning.ai/machine-learning-yearning-book`. Section 47: "The rise of end-to-end learning", pages 91–96.

Javier Fernández, Luke Bornn, and Dan Cervone. Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer. In *13th MIT Sloan Sports Analytics Conference*, volume 2, 2019.

Ronald Yurko, Francesca Matano, Lee F Richardson, Nicholas Granered, Taylor Pospisil, Konstantinos Pelechrinis, and Samuel L Ventura. Going deep: models for continuous-time within-play valuation of game outcomes in american football with tracking data. *Journal of Quantitative Analysis in Sports*, 16(2):163–182, 2020.

Anar Amirli and Hande Alemdar. Prediction of the ball location on the 2d plane in football using optical tracking data. *Academic Platform Journal of Engineering and Smart Systems*, 10(1):1–8, 2022.

Hoang M Le, Peter Carr, Yisong Yue, and Patrick Lucey. Data-driven ghosting using deep imitation learning. In *MIT Sloan Sports Analytics Conference*. MIT sloan sports analytics conference Boston, MA, USA, 2017.

Marc Schmid, Patrick Blauberger, and Martin Lames. Simulating defensive trajectories in american football for predicting league average defensive movements. *Frontiers in Sports and Active Living*, 3, 2021. ISSN 2624-9367. doi:10.3389/fspor.2021.669845. URL `https://www.frontiersin.org/journals/sports-and-active-living/articles/10.3389/fspor.2021.669845`.

Panna Felsen, Patrick Lucey, and Sujoy Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Nazanin Mehrasa, Yatao Zhong, Frederick Tung, Luke Bornn, and Greg Mori. Deep learning of player trajectory representations for team activity analysis. In *11th mit sloan sports analytics conference*, volume 2, page 3, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL `https://arxiv.org/abs/1706.03762`.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018. URL `https://arxiv.org/abs/1703.06114`.

Raymond A. Yeh, Alexander G. Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4605–4614, 2019. doi:10.1109/CVPR.2019.00474.

Michael A. Alcorn and Anh Nguyen. baller2vec: A multi-entity transformer for multi-agent spatiotemporal modeling, 2021. URL `https://arxiv.org/abs/2102.03291`.

The Zoo. 1st place solution - nfl big data bowl 2020. Kaggle, 2020. URL `https://www.kaggle.com/c/nfl-big-data-bowl-2020/discussion/119400`. Winning submission for the NFL Big Data Bowl 2020.