Approaching In-Venue Quality Tracking from Broadcast Video using Generative AI

Soccer Track

Paper ID 193972

1. Introduction

Soccer tracking data has now been available for 25 years, with Derby County the first to utilize the outputs for analysis [1]. This system relied on a set of cameras installed in the stadium and on humans to manually annotate player locations at 10 frames-per-second. The data generated was used for measuring fitness outputs (i.e., how far a player ran, number of sprints etc.), and subsequently was used for tactical analysis, a key factor in enabling Sam Allardyce's Bolton Wanderers to overachieve for several seasons.

By 2008, advances in computer vision allowed for the automatic tracking of players and the ball, enabling real-time data analysis. However, the full potential of tracking data was initially constrained by limited availability, often restricted to a team's own games. This hindered the use of tracking data for broader applications like scouting and recruitment. Subsequently, league-wide deals were arranged, but this still limited analysis to within-leagues, precluding cross-league analysis and comparison.

The advent of broadcast tracking systems promised to overcome these limitations by providing data for any game where broadcast video was available. However, the data obtained from broadcast footage is inherently incomplete due to several factors, such as players being out of the main camera's view, close-up shots, picture quality issues, and scenes where players obscure each other from view. In essence, even though broadcast tracking data can be generated – these *occlusions* mean that it still only captures a portion of what one could expect from tracking data from an in-venue setup.

1.1 Limitations of Existing Data Streams for Scalable Tactical Analysis

To understand the limitations of incomplete data when conducting nuanced tactical analysis, we focus on the motivating task of predicting each player's likelihood of receiving a given pass from a teammate. This task is ostensibly a simple one within the context of sport analytics, but can reveal deep tactical insights into the ways professional teams press, build-up, and create goal-scoring opportunities. While expert coaches can perform this task for an

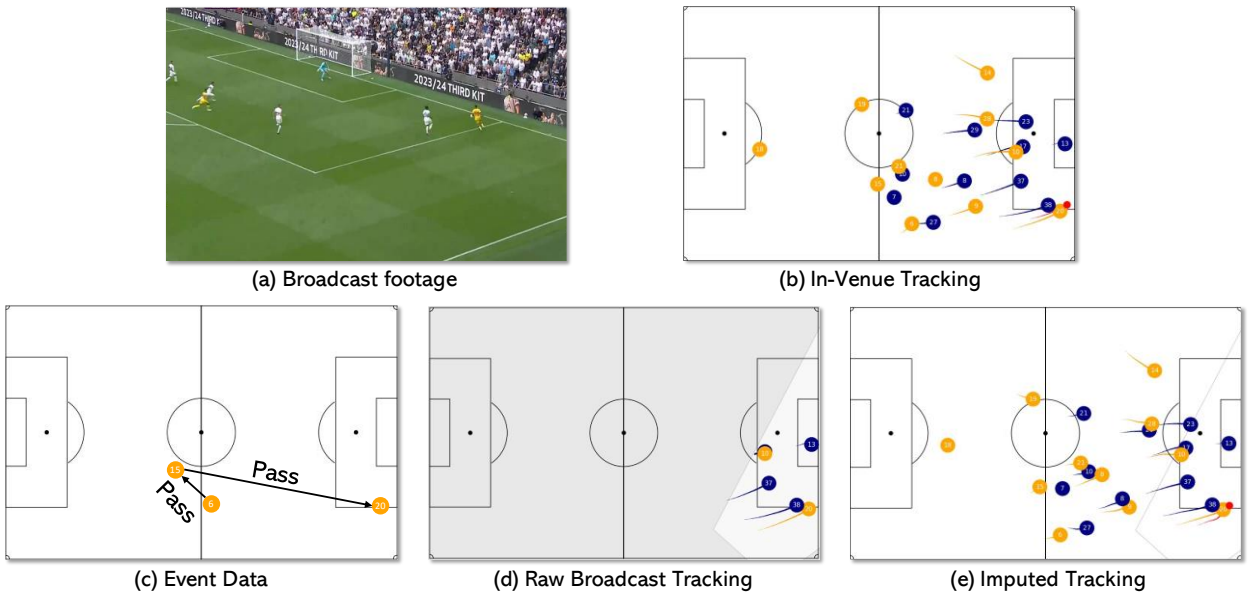


Figure 1: (a) Shows the moment where a player approaching the byline (Yellow #20) is preparing to pass the ball. Which players are the most likely receivers? The only way to answer this question in a data-driven way is with In-Venue Tracking data (b), as Event Data (c) lacks the sufficient granular information to answer this question, and Raw Broadcast Tracking (d) suffers from heavy occlusions where only 1 of the passer's 10 teammates are visible. In (e) we show how our Imputed Tracking data generates realistic behaviors for players that are occluded in broadcast tracking, thus enabling more nuanced downstream analysis.

individual pass, how can this task be performed for every pass in a match? Or for every pass in a season? Or for every pass through the hundreds of thousands of men's and women's professional matches that are played across the globe every year? The only way to perform these analyses is with *complete* (with all agent locations being provided continuously throughout the game) and *scalable* (across all games) data digitization.

Although in-venue tracking systems generate highly accurate and complete tracking data, licensing agreements and the operational costs have meant that these systems have not scaled across the world game. One data stream that is widely available is event data, which logs the sequential stream of semantic events within games. While event data covers the majority of professional games, it only captures the player events that are on-ball, missing off-ball actions (e.g., how a player positions themselves to receive a pass). In this sense, event data is incomplete, and cannot be used to perform tasks that require perception of all player behaviors.

While broadcast tracking addresses the limitations of in-venue tracking systems by being able to scale globally, like event data it is also not a complete data stream. We have discussed the importance of having complete tracking data for understanding a team's passing patterns. In one example game, in the frames where passes occur, only an average of 43% of players can be visually perceived in broadcast tracking, meaning that over half the players on average are occluded. These occlusions impact the visibility of the most important agents during passes: passers, receivers, and the ball. In 21% of pass frames the passer is occluded,

and in 39% of pass frames the receiver is occluded. Furthermore, the ball's small size and fast movement mean that its trajectory is also heavily occluded and/or noisy. The difficulties that occluded data poses in capturing the context around pass events is perhaps even more acute in the case of goal-scoring opportunities. In the evaluation game, of the passes that were made to receivers who subsequently attempted a shot, the receiver was occluded 17% of the time when the pass was made. Soccer is a low-scoring game where scoring opportunities are sparse. The absence of this important context in broadcast tracking impairs the ability to perform complete and nuanced analysis.

In this paper, we demonstrate that generative AI (specifically diffusion models) can be utilized to impute highly realistic behaviors for agents (players and the ball) when they are occluded in broadcast tracking. We showcase how this approach makes a considerable step from incomplete raw broadcast tracking, towards the generation of in-venue quality tracking without in-venue cameras. This is a landmark contribution towards the democratization of tracking data across the global game. A comparison of our imputed tracking data and broadcast footage, in-venue tracking, event data, and broadcast tracking can be viewed in Figure 1.

2. Generating Complete Tracking Data via Diffusion

In this section we give a high-level overview of our imputation method, including how we encode broadcast tracking data, fuse broadcast tracking with event data, and utilize generative AI models to produce highly photorealistic trajectories.

2.1 Encoding Broadcast Tracking Data

The first task of imputation is to encode broadcast tracking data, which forms the strongest signal for inferring the locations of occluded agents. The two key challenges of encoding tracking data are: (1) modelling each agent's past behaviors, and (2) representing inter-agent spatial dynamics. In our setting, the first challenge is especially difficult, because players often remain occluded for long periods of time (up to a minute). As a result, it is vital to be able to encode multiple minutes of broadcast tracking at a time.

In previous works, tracking data has been visualized as a 2D top-down image and processed through computer vision models [2, 3] i.e., convolutional neural networks [4]). However, while agent spatial inter-relationships can be perceived from a single image, the agents' long-term temporal histories cannot. Furthermore, the high dimensionality of images makes it intractable to jointly process more than a few consecutive image frames at a time. In our setting, where multiple **minutes** of tracking context is required, image-based approaches are not appropriate.

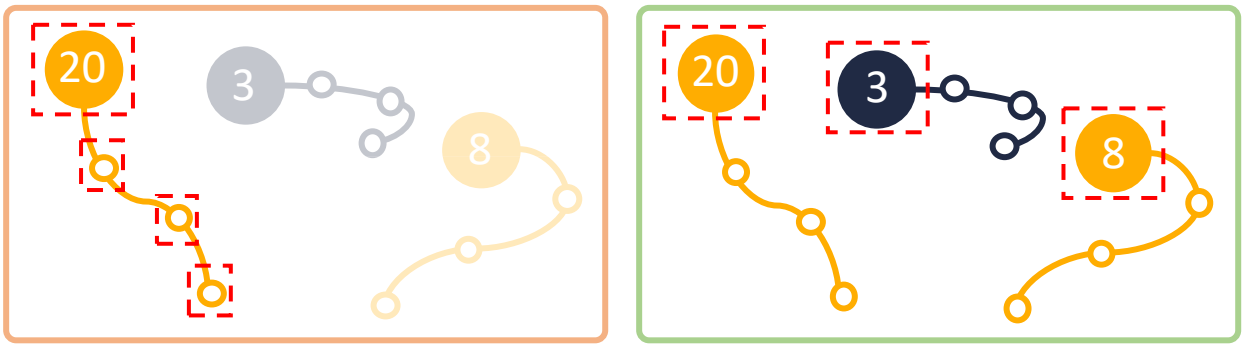
Tracking data is an inherently compressed data representation, and therefore it makes more sense to impute behaviors by using this stream directly. One important challenge of using tracking data directly is the permutation problem. AI models generally assume that their inputs are consistently ordered e.g., words passed to a large language model are always

entered sequentially. However, there is no natural ordering of players that persists from frame to frame and from game to game, which means that most standard deep learning models are forced to learn the same relationships for each of the $(10!)^2$ possible permutations of agent orderings i.e., the number of ways in which the two teams of 10 outfield players can be ordered. One approach that addresses the permutation problem is to consistently order players by inferring their instantaneous spatial role within a formation template [5, 6]. This method is limited by its use of a single static template, failing to represent how player roles change depending on the current phase of play (e.g., corners, dead-balls, counterattacks).

Another way of tackling the permutation problem is by using *permutation invariant* models i.e., models where changing the order of the players has no impact on the model's output. One such family of models that have this property are Graph Neural Networks (GNNs) [7], which encode information that has an underlying graph structure. These models have been applied to sports tracking data by representing each agent as a node in a fully connected graph, i.e., where there is an edge between every pair of nodes [8, 9, 10]. While formulating tracking data as a graph solves the spatial modelling challenge, existing applications have only endowed GNNs with short-term temporal context (i.e., <10 seconds).

The backbone of nearly all modern state-of-the-art AI models (e.g., OpenAI's ChatGPT [11]) are Transformers [12], which are neural networks that are closely related to GNNs. Transformers primarily rely on a single simple operation: self-attention. For a given collection of tokens (e.g., a sequence of words) the attention mechanism will infer each token's (e.g., word's) dependence on every other token from large amounts of training data. After this operation, each token is updated with the context with respect to all other tokens. The success of the attention mechanism on language modelling problems has shown its ability to learn complex long-term interdependencies within sequential data. This is an appealing property for encoding tracking data, which contains long-term spatial and temporal dependencies.

We argue that the optimal way to adapt Transformers to sports tracking data is through spatiotemporal axial attention (SAA) [13, 14], which consists of two interleaved attention modules: temporal attention and axial attention. In temporal attention (Figure 2a), each agent's temporal context is encoded by completing self-attention between each of an agent's past locations. In spatial attention (Figure 2b), the spatial relationships within a single frame are modelled by completing self-attention between each agent's locations at that instant. By interleaving these operations, both the temporal and spatial dependencies within the sporting scene are jointly modelled. SAA has two key advantages: First, SAA avoids the permutation problem as no ordering is imposed on agents. Secondly, temporal attention is an extremely computationally efficient method for modelling agent's long-term histories. This is particularly crucial when accurately predicting the behaviors of agents that are occluded for long periods of time.



(a) Temporal Attention (b) Spatial Attention

Figure 2: Illustration of Spatiotemporal Axial Attention from the perspective of player #20 at the current timestep. In Temporal Attention (a) each agent's past behaviors are encoded by constructing a fully connected graph comprised of a player's past and present locations. Spatial attention (b) is the mechanism where an agent's spatial dependencies are modelled. This is done by constructing a fully connected graph comprised of each agent's present locations. Interleaving these operations collectively results in the modelling of both temporal and spatial dependencies.

2.1 Enhancing Broadcast Tracking with Event Data

Although broadcast tracking provides an essential signal for the accurate synthesis of complete tracking data, it has several limitations. First, broadcast tracking struggles to track the ball continuously and accurately, due to its small size and fast movement. Secondly, there are many continuous periods of the game where broadcast tracking does not provide any coverage. Although these periods are typically relatively short (<10 seconds), synthesizing accurate agent behaviors for these segments is extremely difficult without additional contextual information. We address these challenges by integrating event data with broadcast tracking data to estimate occluded agent behaviors. This is a paradigm shift away from prior works that treat sport as a unimodal domain i.e., only using tracking data. Our insight is that sport is *multi-modal*, consisting of multiple spatiotemporal input modes – namely tracking data and event data.

We use the insight that, like tracking data, event data can also be framed as a spatiotemporal modality, consisting of a temporal dimension (i.e., the chronological ordering of each player's events), and a spatial dimension (i.e., representing each specific player) and thus can be encoded using SAA. We utilize the flexibility of the Transformer architecture by jointly processing these modalities together to produce an encoding that contains both tracking and event context, as is shown in Figure 3. Collectively, this architecture enables the first fusion of event and tracking data in a deep learning model, which is a landmark moment for the ways in which sports data is understood and processed by AI models.

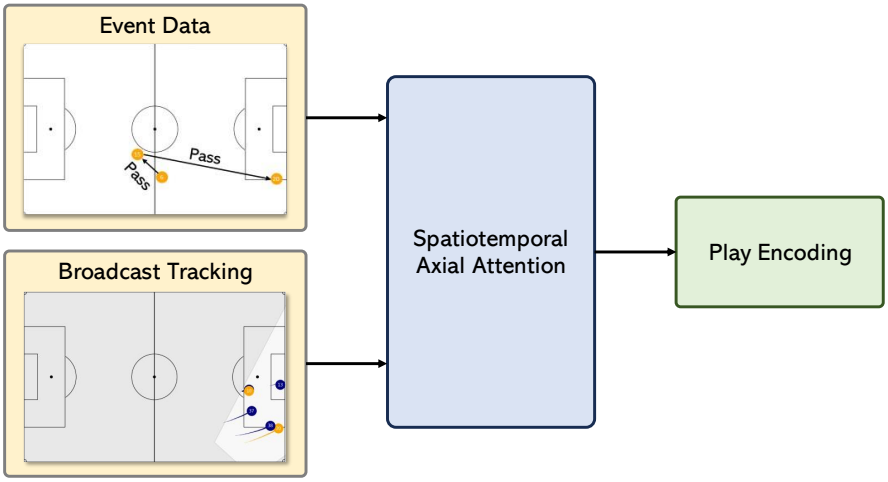


Figure 3: Illustration of how we fuse where Event Data and Broadcast Tracking data to produce a play-level encoding. The key module within this architecture is Spatiotemporal Axial Attention, which processes spatiotemporal data via interleaved temporal and spatial attention.

2.3 Generating Photorealistic Tracking Data via Diffusion

To this point, the approach we have outlined for fusing event data with broadcast tracking data can accurately predict agent locations, however these locations collectively do not necessarily form realistic human motion. This is caused by the high level of uncertainty in agent locations, particularly due to the noise and heavy occlusions in the broadcast tracking input. In practice, this means that behaviors generated in this way often exhibit jitter (i.e., unsmooth trajectories) and occasionally teleport between locations.

To alleviate these issues in generating agent behaviors, we utilize diffusion [15], which is a family state-of-the-art generative AI models that have most notoriously been used in the generation of highly realistic images from captions [16, 17]. At a simple level, diffusion models generate data via iteratively denoising from a random initial state. Starting with pure noise, diffusion models progressively refine the sample, gradually creating a higher and higher fidelity generation. The process of iterative denoising is largely what makes the diffusion approach well-suited to the generation of images. Iterative denoising means that models both learn to construct the coarse features (e.g., the main subject of an image) and granular features (e.g., visual textures) that comprise an image, resulting in highly photorealistic generations. Diffusion has similar advantages in the generation of tracking data, which also contains both rich coarse features (e.g., agents' rough locations) and granular features (e.g., the smoothness of agent motion). Moreover, just as images can be generated by diffusion models by conditioning on textual captions, we generate complete tracking data through conditioning on our broadcast and event data encodings. We visualize the diffusion generation process in Figure 4.

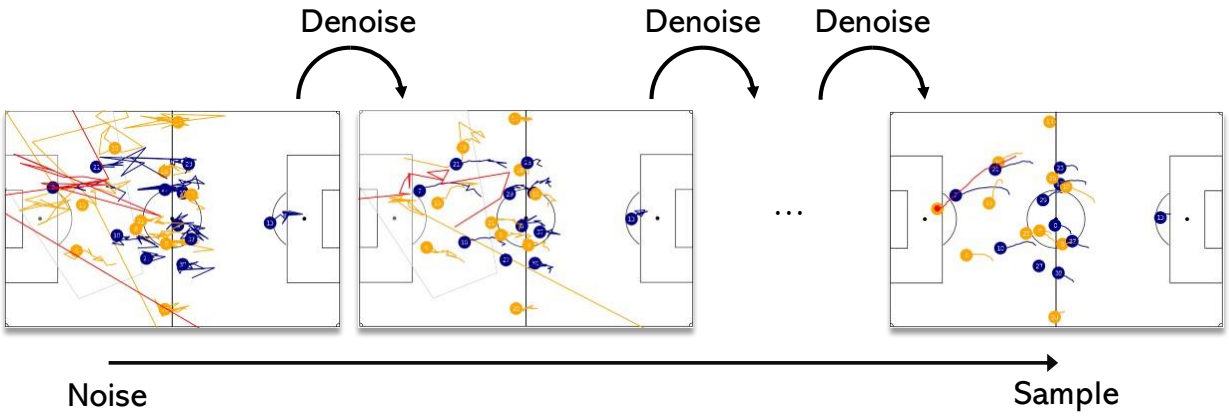


Figure 4: Visualizes how tracking data is generated with diffusion via iteratively denoising an initial pure noise sample. Gradually, this noise is refined to form a highly realistic tracking data.

3. Unlocking Downstream Analysis with Imputed Data

To evaluate the accuracy of our imputation, we separately extract downstream metrics from in-venue tracking, imputed tracking, and broadcast tracking on a test game. This allows us to determine how similar the remote tracking datasets (broadcast tracking and imputed tracking) are to in-venue tracking data. While there are many valid discriminative tasks that could be used for this downstream sporting analysis, here we use the task discussed in Section 1.1 - for a given pass, what is the probability that each attacking player will be the pass receiver. This is often called the xReceiver metric [10] and is dependent both on agents' coarse locations and on fine-grained details such as agent velocities, accelerations, and body orientations. For the xReceiver outputs to match the outputs of in-venue tracking, our imputed data must correctly generate these complex features in trajectory space. In this section, we outline our method for implementing the xReceiver model, along with a discussion of the xReceiver outputs for in-venue tracking, raw broadcast tracking, and our imputed tracking.

3.1 xReceiver Dataset

We train and validate our xReceiver model on 130 games of tracking data. For each of these games, we have access to both the in-venue tracking and broadcast tracking data. We focus on all successful passes, using 5 seconds of tracking context leading up to 0.2 seconds before the pass. For the purposes of this analysis, we only inspect passes where the pass' temporal context exists entirely in in-play segments of the game. Using tracking data directly rather than extracting handcrafted temporal features (e.g., velocity and acceleration) increases the amount of information available to models and is less sensitive to small amounts of noise. We use a 90:10 training and validation split, with features including each agent's (x, y) locations, the agent's type (i.e., goalkeeper, ball, or outfield player), and an indicator as to whether the agent is on the attacking team.

3.2 **xReceiver Model**

We use SAA as the underlying xReceiver model architecture, demonstrating the flexibility of having a powerful model which extracts spatiotemporal dependencies from tracking data. In essence, we argue that SAA should be the de facto AI model for processing tracking data, rather than having intermediate representations (e.g., images) or handcrafting temporal features. All agents' trajectories are processed by a SAA module followed by a linear projection layer. Then, each attacking agent's outputs are fed through a softmax activation function, which ensures that the xReceiver model maintains the Law of Total Probability (i.e., all attacking player xReceiver values sum to 1). Models are trained using cross entropy loss. Two instances of this model are trained, one using in-venue tracking to comprise agent locations, and another that uses broadcast tracking.

For testing, the xReceiver model that is trained on in-venue tracking is applied to in-venue tracking. Likewise, the xReceiver model that is trained on raw broadcast tracking is applied to the raw broadcast tracking data. In the case of imputed tracking, the model trained on in-venue tracking is used. This enables an analysis of imputed tracking data's ability to be substituted for in-venue tracking.

3.3 **xReceiver Quantitative Results**

We use two metrics to compare the quality of raw broadcast and imputed tracking's xReceiver outputs with the in-venue outputs. First, we report how frequently the true receiver is among the top-k most likely predicted receivers from each dataset. The second metric directly compares how similar the in-venue xReceiver outputs are with the remote tracking dataset (broadcast tracking / imputed tracking) xReceiver outputs. For every pass in each dataset, we extract the set of high likelihood receivers (i.e., players with an xReceiver value over 0.1). Then, we compare these sets of players using the Intersection over Union (IoU) metric. IoU computes the ratio between the intersection (i.e., the players that both data sources agree are high probability receivers) and the union (i.e., the total number of unique players in the two sets of high probability receivers). The IoU metric is computed separately between (1) in-venue and raw broadcast tracking, and (2) in-venue and imputed tracking. These results are shown in Table 1.

The first notable result is that broadcast tracking exhibits the weakest performance in each of the extracted metrics. This clearly shows the difficulties that arise in conducting downstream analyses on incomplete data which is heavily impaired by occlusions. Comparatively, our imputed data has much stronger performance. In terms of the top-k metrics, our imputed data's xReceiver outputs closely approach the accuracy shown with in-venue tracking data. In terms of the IoU metric, our imputed tracking also considerably outperforms the raw broadcast tracking data. Despite this, our imputed data still does not reach the theoretical limit of 1 (the point at which the high probability receivers in the two sets are identical for every pass), showing that our imputation model can still be improved in future work. While we only focus on the xReceiver task in this paper, this evaluation will

be expanded to encompass the rich suite of downstream analytics tasks that have been developed in the sporting community over the last 15 years.

Table 1: Quantitative Results showing each dataset's (Raw Broadcast, Imputed, In-Venue) results in terms of the top-k metrics and the Intersection over Union between the Raw Broadcast and In-Venue xReceiver outputs, as well as between the Imputed and In-Venue xReceiver outputs.

Metric	**Raw Broadcast**	**Imputed**	**In-Venue**
Top-1	42.03	55.43	59.06
Top-2	59.42	76.81	79.35
Top-3	71.01	88.04	90.94
Top-4	80.43	93.84	93.48
Intersection over Union	0.5105	0.7121	-

3.3 **xReceiver Qualitative Results**

Now, we qualitatively analyze the xReceiver outputs. Returning to the play shown in Figure 1, all the event data can tell us is that this play ends with Yellow #20 crossing the ball to Yellow #28, who registers a shot-on-target from near the penalty spot. But which other players were available? Was this the most threatening pass? How likely was this pass to succeed? We only focus on the first of these questions for this paper, but with complete tracking data each of these questions can be answered in a data-driven way. We visualize each dataset's xReceiver outputs for this example in Figure 5. For this example, in-venue tracking data's xReceiver outputs tell us that there are three likely receivers of the pass, one of which is the actual receiver.

In contrast, the broadcast tracking's xReceiver model predicts three likely receivers, none of which are the actual receiver. These three players appear to be deemed high probability receivers because they are the players most visible in the xReceiver model's 5 seconds of tracking context. This highlights the heavy bias against occluded players that emerges when conducting data-driven analysis on incomplete tracking data. It is also notable that the xReceiver outputs of broadcast tracking also cannot be easily interpreted, due to the uncertainty concerning agents' real locations.

Visually, our imputed tracking data is complete, with player locations closely resembling in-venue tracking. Additionally, player trajectories are smooth, representing realistic human motion. However, one discrepancy is that unlike the in-venue outputs, the imputed data deems player #28 to be a high likelihood receiver. This highlights the sensitivity of the xReceiver prediction to minor differences in player velocities and accelerations. However, the fact that this is the only discrepancy between the in-venue and imputed outputs for this example is a strong result, especially considering the large occlusions present in broadcast tracking. This result strongly motivates the importance of imputation for enabling more nuanced data-driven analysis when in-venue tracking is not available. More examples can be found in Appendix A (images) and B (video).

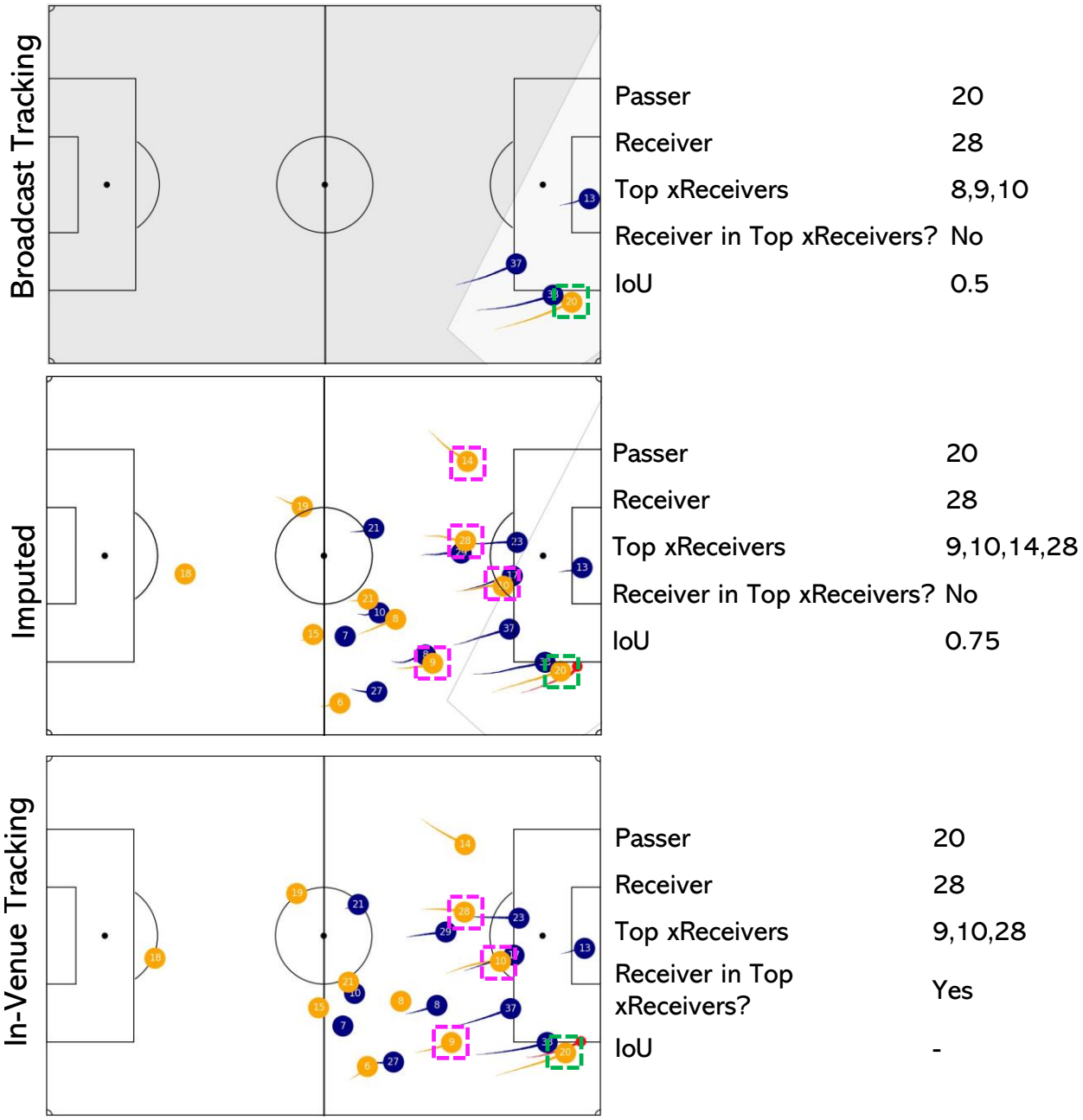


Figure 5: Illustrates the xReceiver outputs for a pass using raw broadcast tracking (1st row), our imputed tracking (2nd row), and in-venue tracking (3rd row). The player who passes the ball (Yellow #20) has a **green border**, while the predicted high likelihood receivers have a **pink border**.

The three high likelihood receivers predicted in the broadcast tracking are heavily biased towards the most recently players, which is problematic as the actual receiver (Yellow #28), who is making an attacking run into the box and has been occluded for a long period of time, is therefore not deemed a high likelihood receiver. With our model, we impute highly realistic motion for occluded agents, which results in xReceiver outputs that much more closely match the in-venue outputs. Notably, in both the in-venue and imputed results, the actual receiver is deemed a high probability receiver.

4. Summary

In this paper we have presented the importance of imputation for analyzing sports broadcast tracking data. This is a landmark step on the road towards making complete player tracking available across the global game. In future work we will broaden this evaluation to include the rich suite of data-driven sporting analyses that have been developed in the sports community over the last 15 years. Another contribution in the paper is a review of the previous methods in which AI models process sports tracking data. We argue that spatiotemporal axial attention (a Transformer adapted specifically to process spatiotemporal data) should be the de facto approach to all machine learning tasks that use sports tracking data as an input. Finally, we show that spatiotemporal axial attention can be extended in a simple and principled manner to jointly process both event and tracking data. We see this as a paradigm shift away from a unimodal representation (tracking data only) towards seeing sports data as multimodal, including semantic (event data) and fine-grained (tracking) streams.

References

[1] P. Hay, "How Prozone sparked a football analytics boom," The Athletic, 16 November 2020. [Online]. Available: https://theathletic.com/2193722/2020/11/16/prozone-analytics-ramm-mylvaganam-analysis-premier-league/. [Accessed 27 November 2023].

[2] U. Brefeld, J. Lasek and S. Mair, "Probabilistic movement models and zones of control," *Machine Learning,* vol. 108, no. 1, pp. 127-147, 2019.

[3] J. Fernández and L. Bornn, "Soccermap: A deep learning architecture for visually-interpretable analysis in soccer," *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14--18, 2020, Proceedings, Part V, 2021,* pp. 491-506, 2021.

[4] G. E. Hinton, I. Sutskever and A. Krizhevsky, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, 2012.

[5] P. Lucey, A. B. P. Carr, S. Morgan, I. Matthews and Y. Sheikh, "Representing and Discovering Adversarial Team Behaviors Using Player Roles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[6] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan and I. Matthews, "Identifying Team Style in Soccer Using Formations Learned from Spatiotemporal Tracking Data," 2014.

[7] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks,* vol. 20, no. 1, pp. 61-80, 2008.

[8] M. Horton, "Learning feature representations from football tracking," in *MIT Sloan Sports Analytics Conference*, 2020.

[9] G. Anzer, P. B. U. B. and D. Faßmeyer, "Detection of tactical patterns using semi-supervised graph neural networks," in *MIT Sloan Sports Analytics Conference*, 2022.

[10] M. Stöckl, T. Seidl, D. Marley and P. Power, "Making offensive play predictable-using a graph convolutional network to understand defensive performance in soccer," in *Proceedings of the 15th MIT sloan sports analytics conference*, 2021.

[11] OpenAI, "Introducing ChatGPT," OpenAI, 30 November 2022. [Online]. Available: https://openai.com/blog/chatgpt. [Accessed 30 November 2023].

[12] A. Vaswani, N. S. N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[13] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat and B. Sapp, "2023 IEEE International Conference on Robotics and Automation (ICRA)," 2023.

[14] A. Monti, A. Porrello, S. Calderara, P. Coscia, L. Ballan and R. Cucchiara, "How many observations are enough? knowledge distillation for trajectory forecasting," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[15] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, 2015.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022.

[17] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125,* vol. 1, p. 3, 2022.

Appendix A

In this appendix there are three additional examples of the tracking and xReceiver outputs of broadcast tracking, imputed tracking, and in-venue tracking. These examples further reinforce the importance of quality imputation to deliver highly accurate nuanced analyses where in-venue tracking is not available.

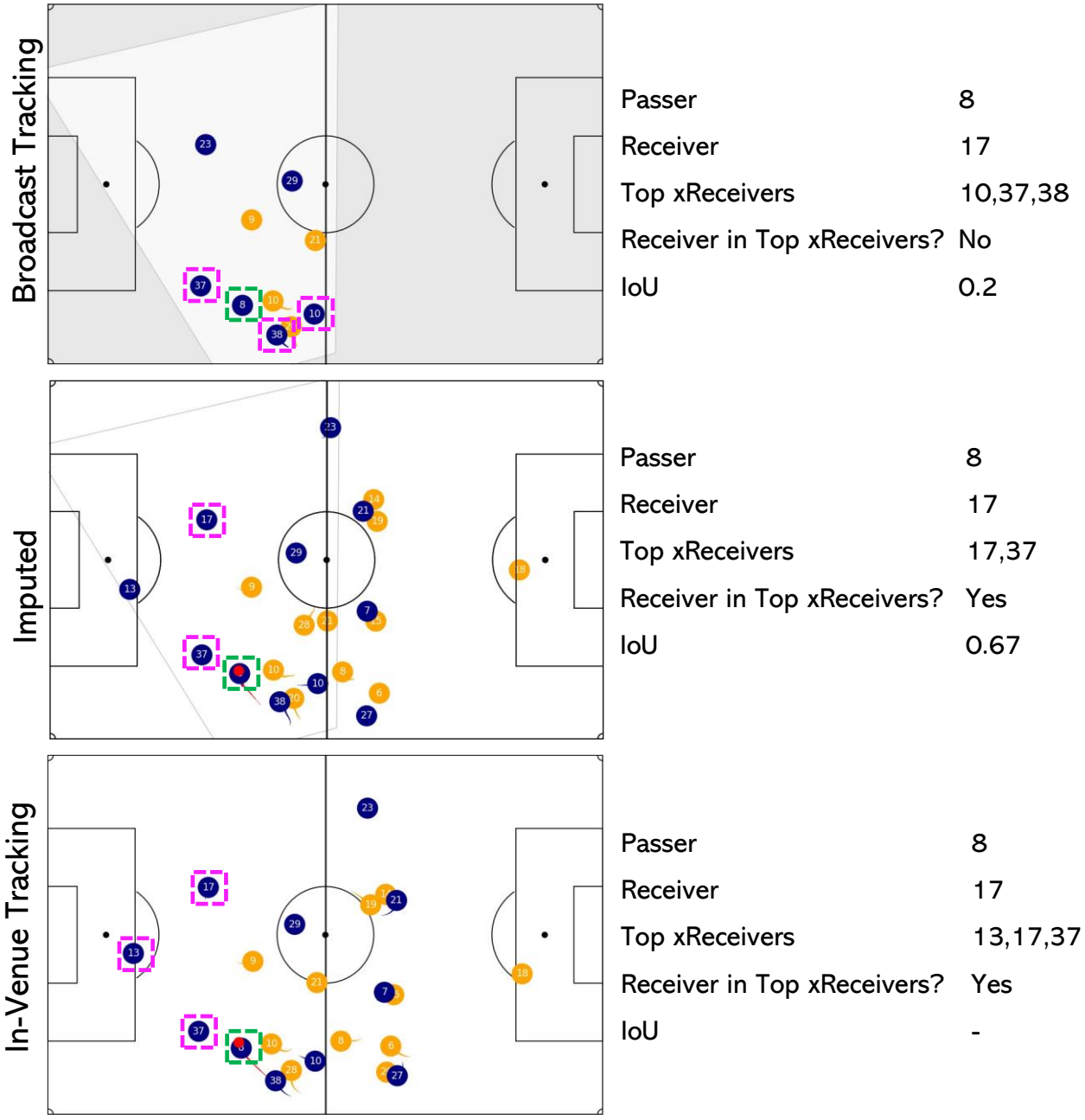


Figure A-1: Illustrates the xReceiver outputs for a pass using raw broadcast tracking (1st row), our imputed tracking (2nd row), and in-venue tracking (3rd row). The player who passes the ball (Yellow #20) has a **green border**, while the predicted high likelihood receivers have a **pink border**.

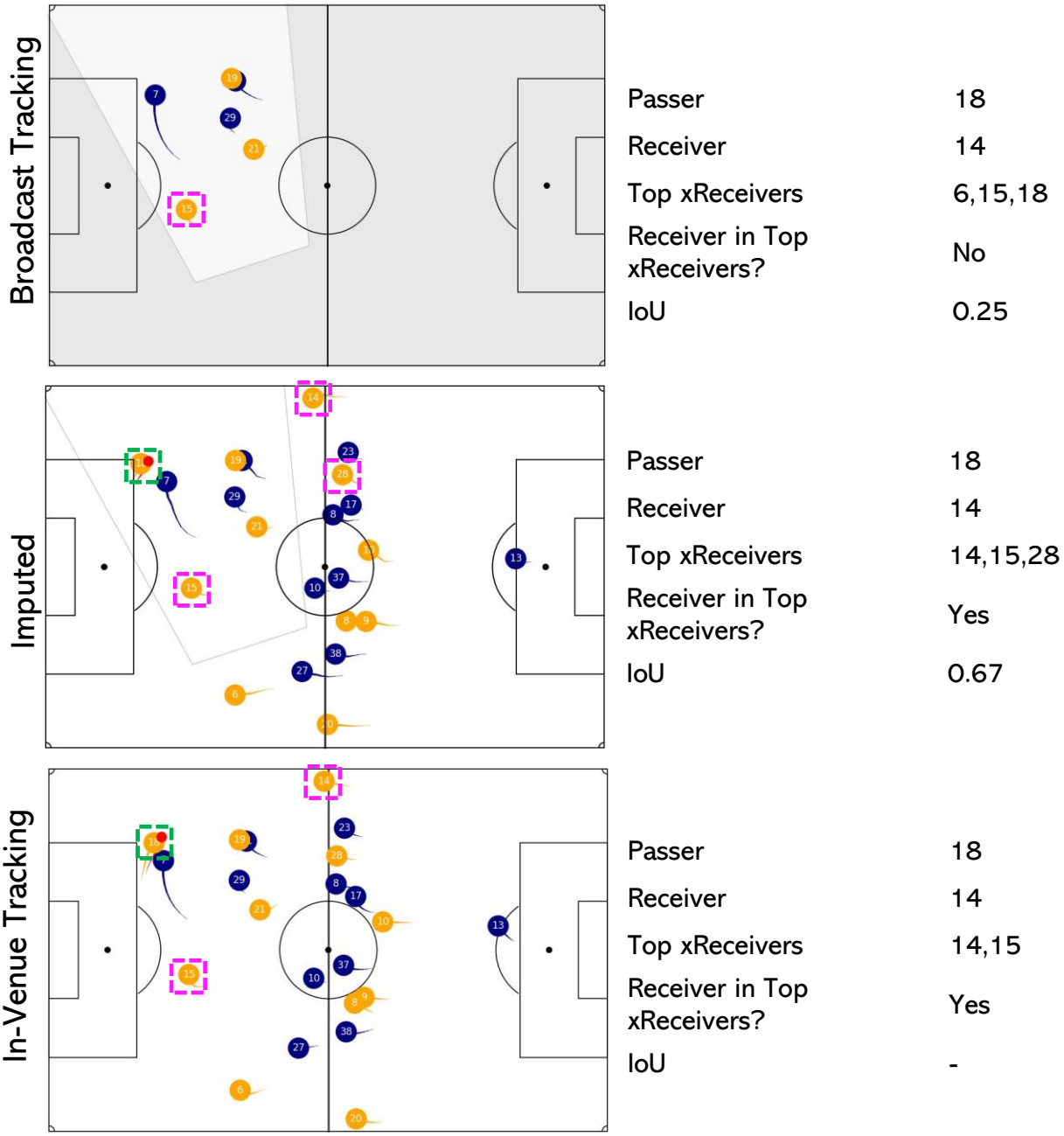


Figure A-2: Illustrates the xReceiver outputs for a pass using raw broadcast tracking (1st row), our imputed tracking (2nd row), and in-venue tracking (3rd row). The player who passes the ball (Yellow #20) has a **green border**, while the predicted high likelihood receivers have a **pink border**.

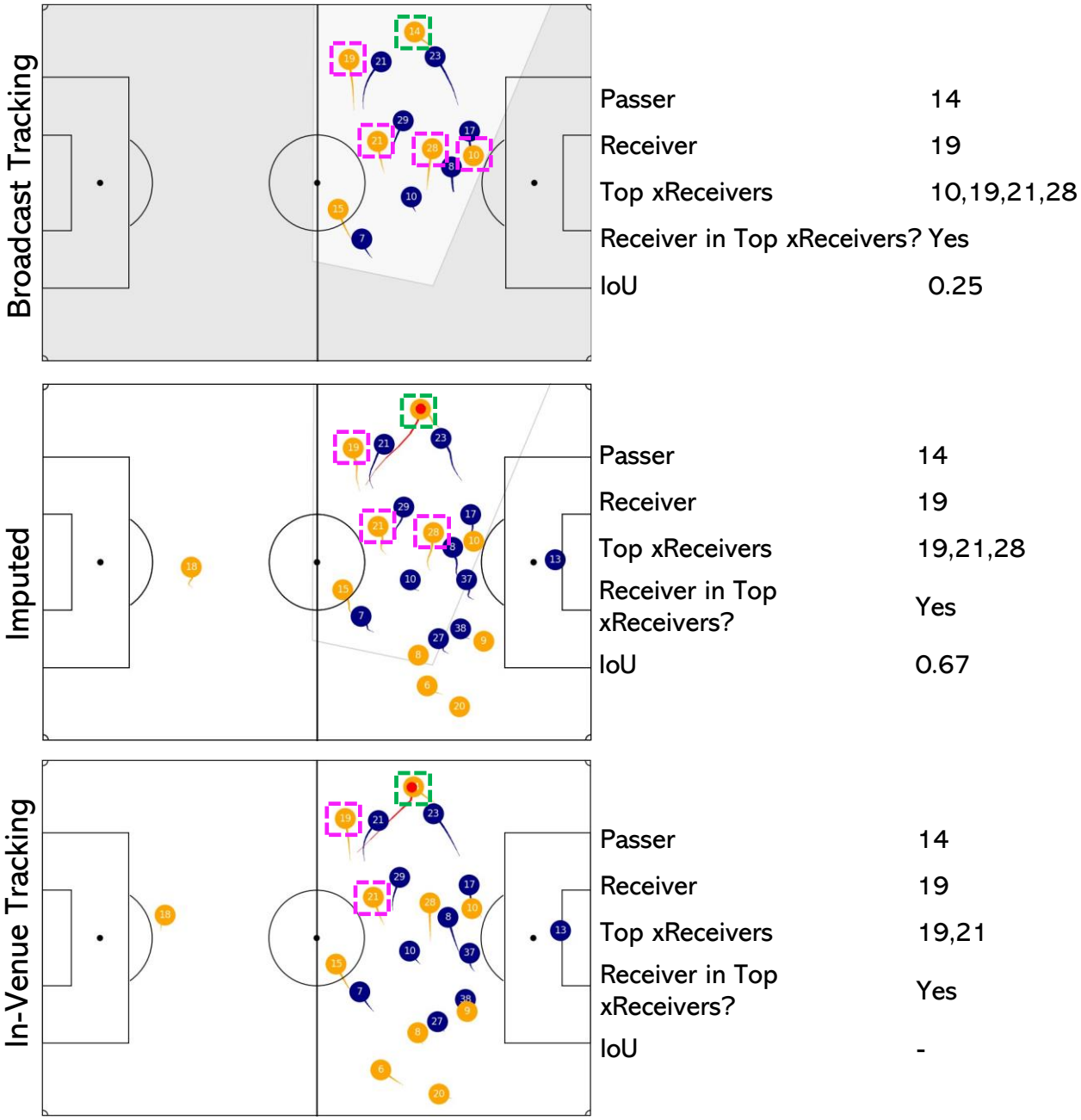


Figure A-3: Illustrates the xReceiver outputs for a pass using raw broadcast tracking (1st row), our imputed tracking (2nd row), and in-venue tracking (3rd row). The player who passes the ball (Yellow #20) has a **green border**, while the predicted high likelihood receivers have a **pink border**.

Appendix B

Supplementary Video Link:

- https://www.dropbox.com/scl/fi/q7du2g2j1rncxxxqad8zr/Sloan-Video-1.mp4?rlkey=nav8pjv3pktc3218nmvu6vmd4&dl=0
- Password: sloan2024

GitHub evaluation link:

- https://www.dropbox.com/scl/fi/6rzat1wmski5zkq7wfeap/merged_outputs.parquet?rlkey=byaqpcf92f0wfjcieuwavg8vy&dl=0
- Password: sloan2024