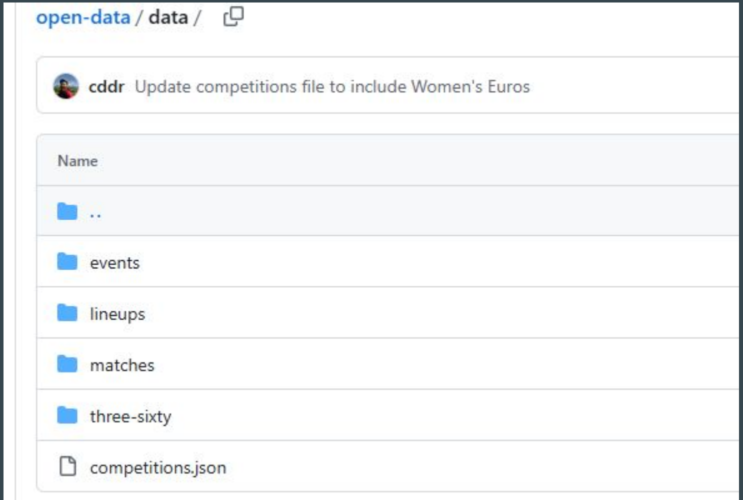
Player Similarity via Embedding Models in Soccer Analytics

[John Zhang, Jaehyeon Park, Cris Gao, Eason Huang, Ted Tao]

Introduction

- ***Problem Statement***
  - Traditional summary statistics in soccer are used to compare players, but these metrics cannot capture how players actually make decisions.
- ***Research Goal***
  - In order to quantitatively evaluate how similar two players are, we aim to develop an embedding model to encode how players think and act in specific contexts.

Data-Data Source & Collection

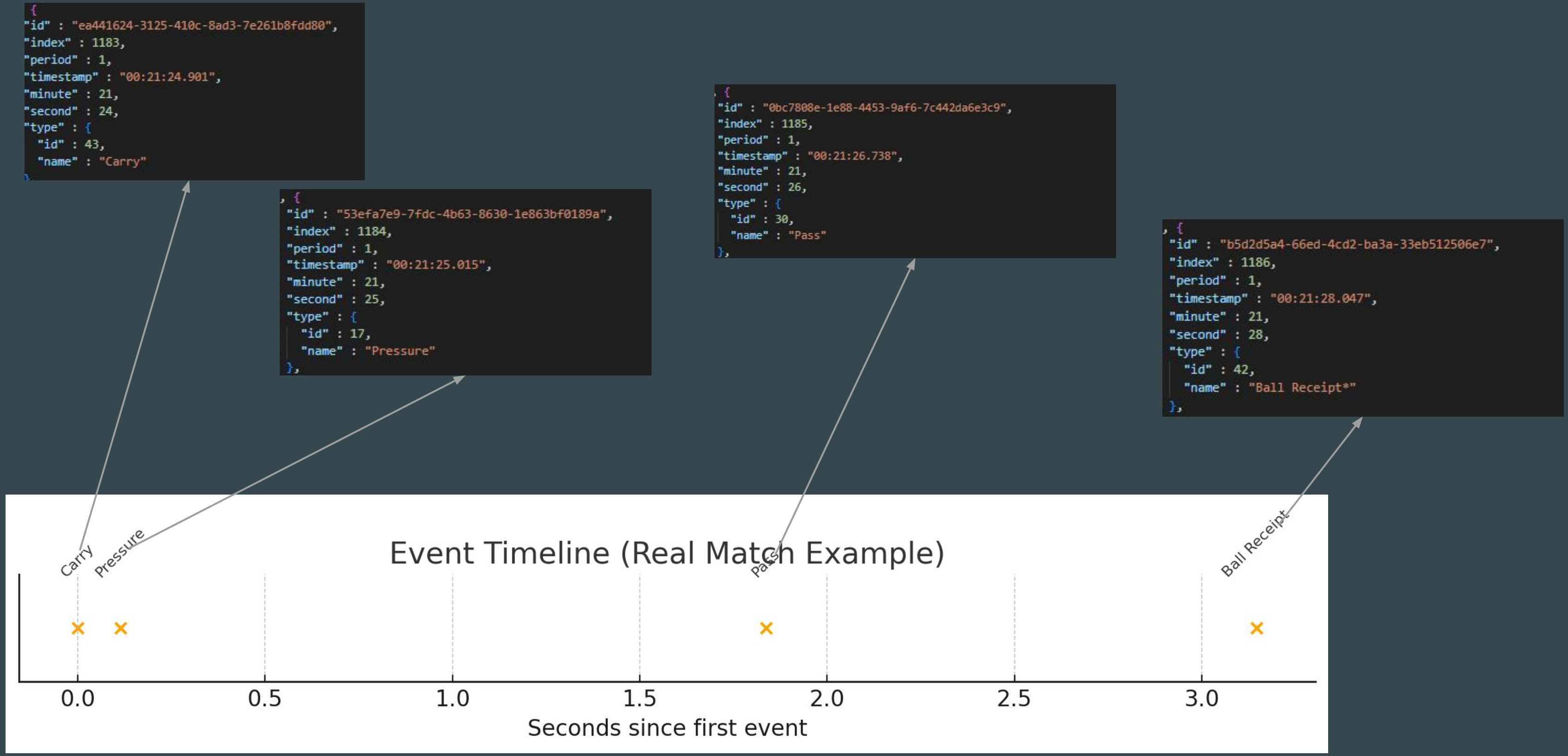


StatsBomb open-data → /data folder structure screenshot

https://github.com/statsbomb/open-data/tree/master/data

- 34 Bundesliga matches from Leverkusen (2023–24)
- Events (actions), Lineups (players), 360 (positions), Match metadata
- JSON format with nested fields
- Events = manually annotated
- 360 freeze-frames extracted from broadcast video

Data- Temporal scale

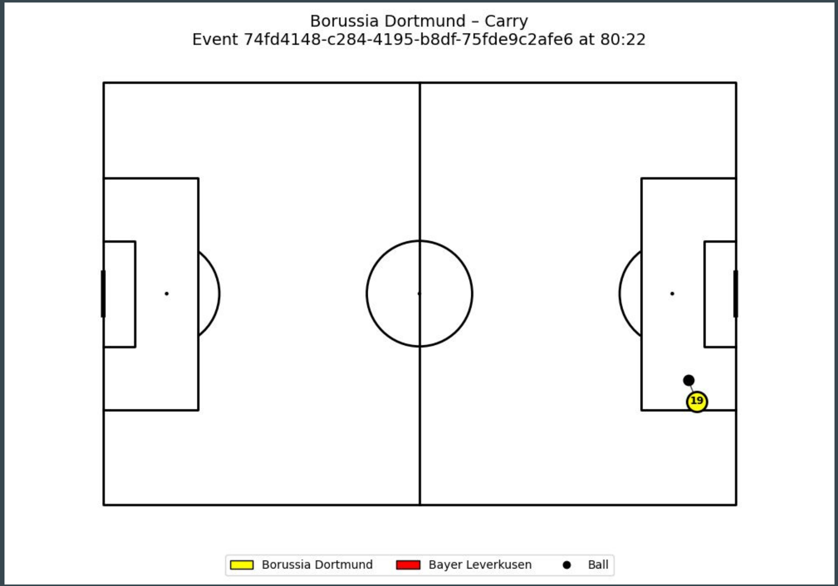


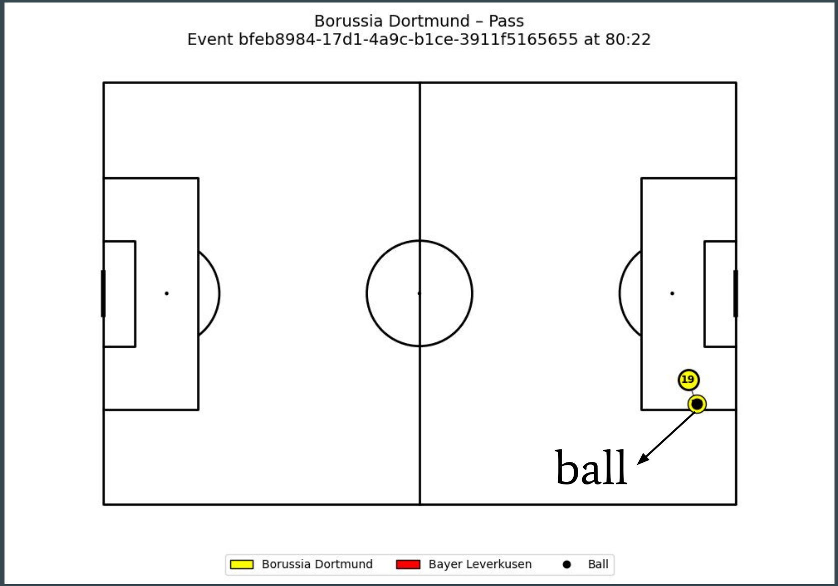
The temporal structure is event-based, with irregular intervals between actions.

Spatial scale

Source: StatsBomb Open Data – Events Spec Appendix 2.

Frequency scale





One spatial snapshot per event, not continuous tracking.

Data - IDs & Timing

`id, index, period, timestamp, minute, second, possession, possession_team`

- **id / index** – unique labels for each event
- **period** – which chunk of the game: 1st half, 2nd half, extra time, etc.
- **timestamp, minute, second** – how far into the match this happened.
- **possession, possession_team** – ID for a **possession sequence**, all events with the same possession ID belong to the same attacking move, and which team is in control for that possession.

Data - Who & Where

`team, player, position, jersey_number, location, end_location`

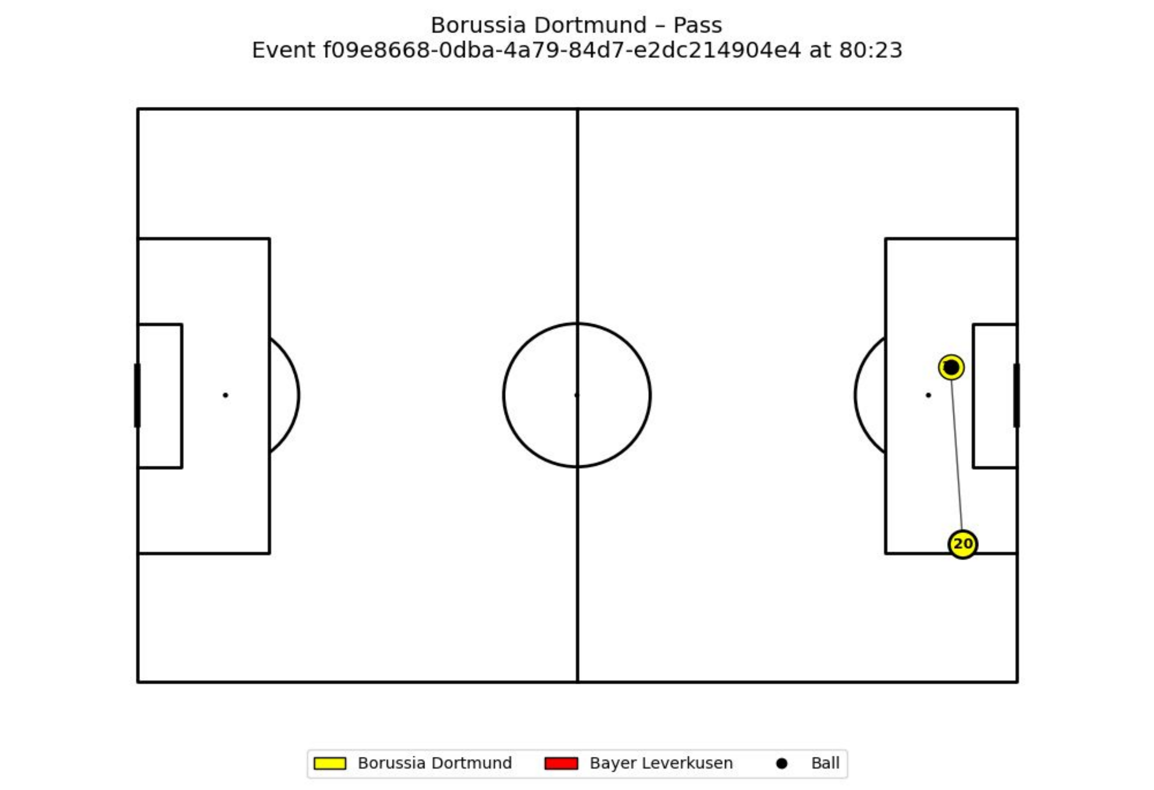
- **team** – which team the event belongs to
- **player** – the on-ball player
- **position** – their tactical role at that moment
- **jersey_number** – shirt number from the lineup
- **location** – the starting point of the event, as (x, y) on a 120×80 pitch.
- **end_location** – where the ball ends up after the event.

Data - Action-Specific: Passes

`recipient, length, angle, height, cross, goal_assist`

- **recipient** – who receives the ball.
- **length** – distance the ball travels in metres.
- **angle** – the direction of the pass (forwards, backwards, sideways).
- **height** – ground, low, or high pass.
- **cross** – whether it's a cross into the box.
- **goal_assist** – whether this pass directly sets up a goal.

Data - Action-Specific: Passes



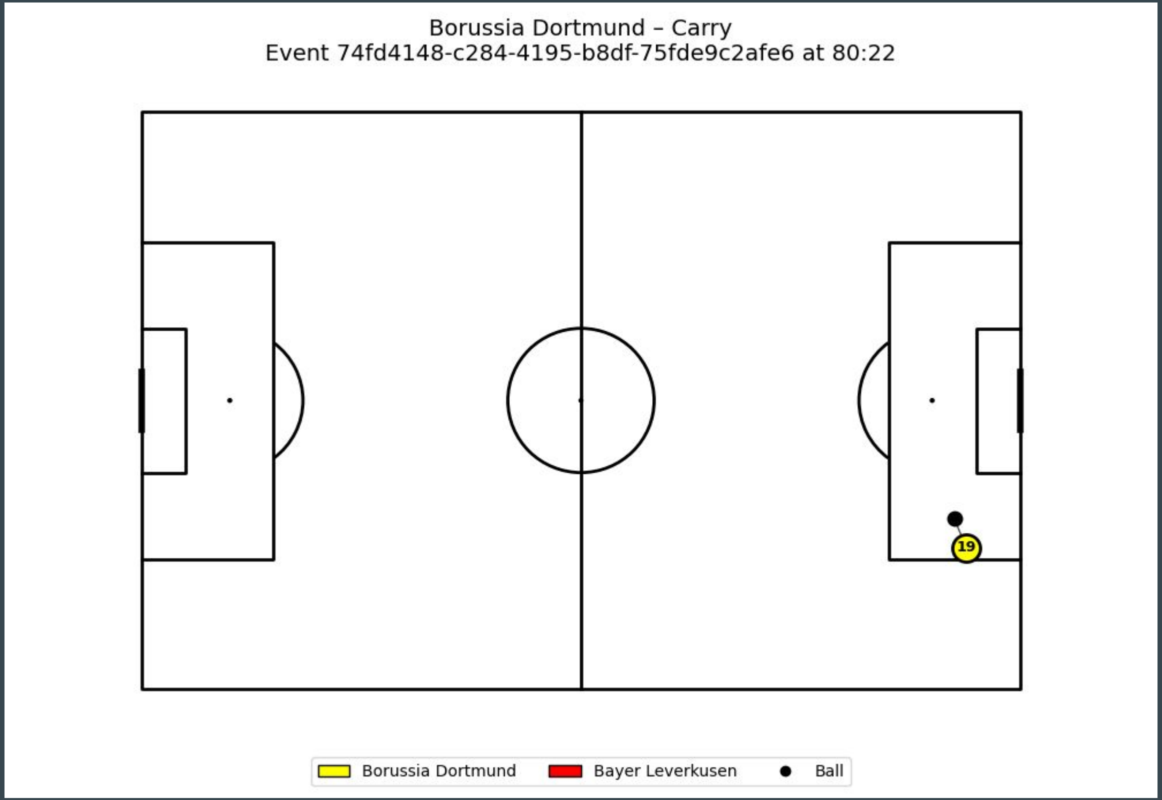
Marcel Sabitzer (No.20) makes a cross to the middle of the box

Data - Action-Specific: Carries

```
end_location, duration
```

- **end_location** – where the carry finishes.
- **duration** – how long the player is moving with the ball.

Data - Action-Specific: Carries



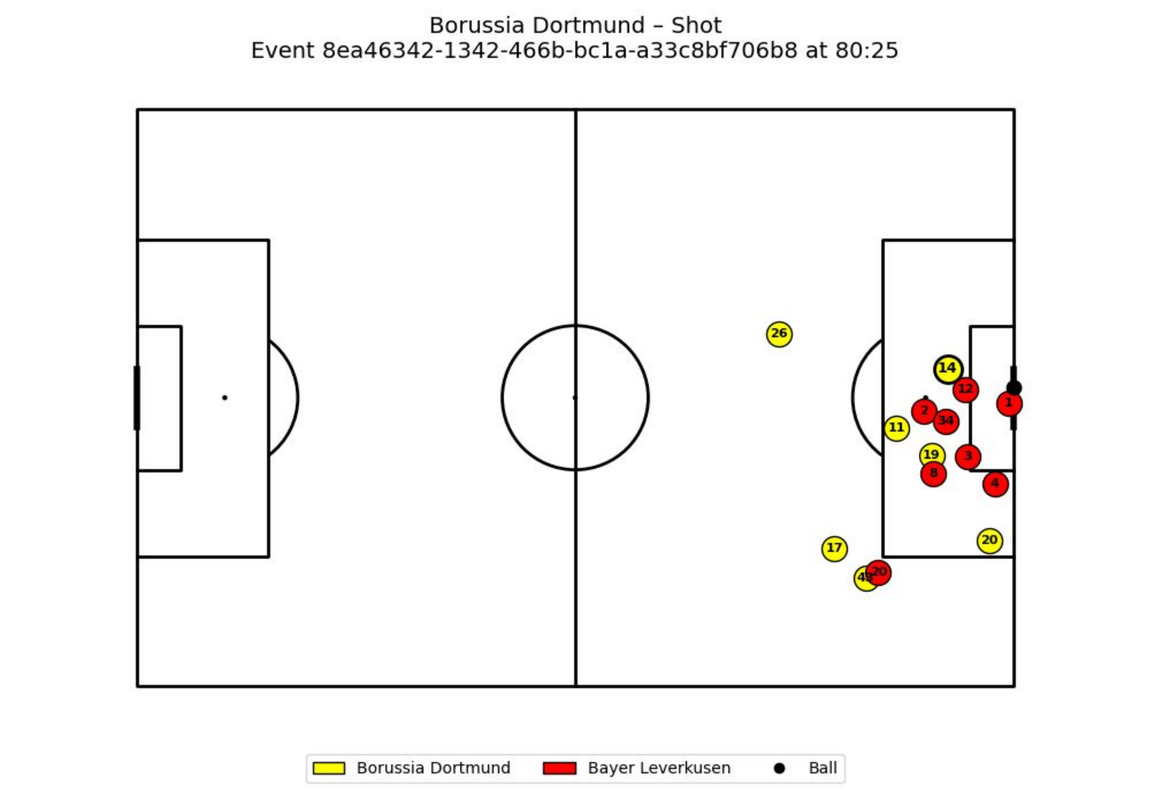
Julian Brandt (No.19) carries the ball deeper inside the box

Data - Action-Specific: Shots

`end_location, outcome, body_part, type, technique`

- **end_location** – coordinates in/around the goal
- **outcome** – Goal, Saved, Off Target, etc.
- **body_part** – right foot, left foot, head, etc.
- **type** – open play, set piece, penalty...
- **technique** – volley, header, half-volley, etc.

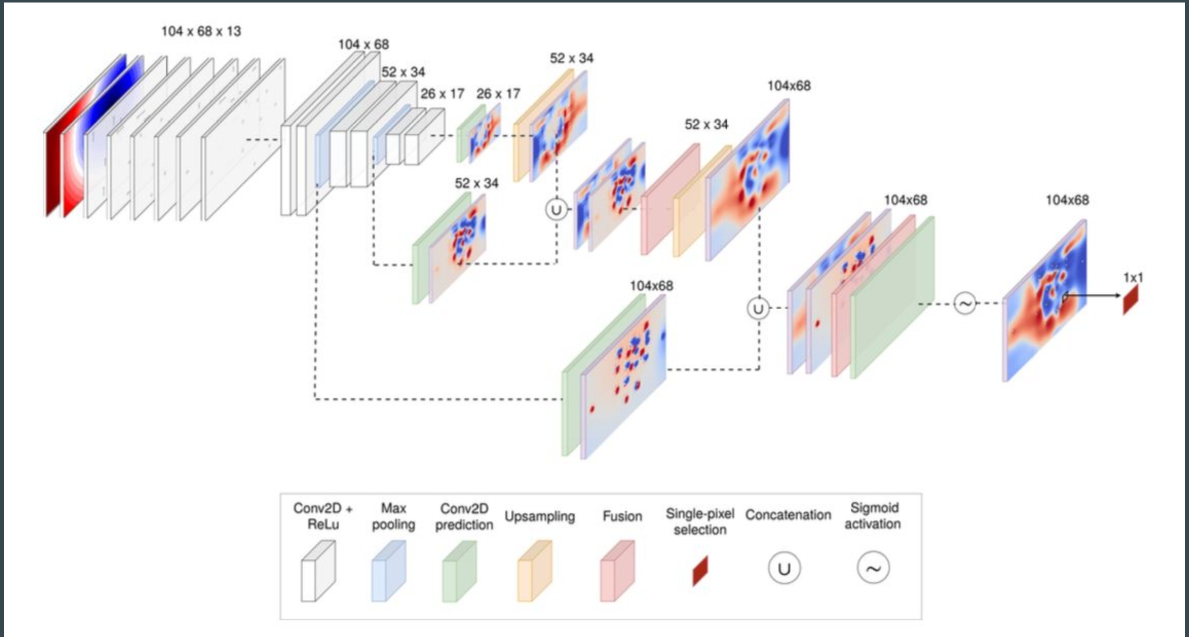
Data - Action-Specific: Shots



Niklas Fullkrug (No.14) takes a shot inside the box


KONAMI
QATAR
Allianz
Allianz
SAP
tipico
SIEMENS
KONAMI

Methodology: SoccerMap CNN

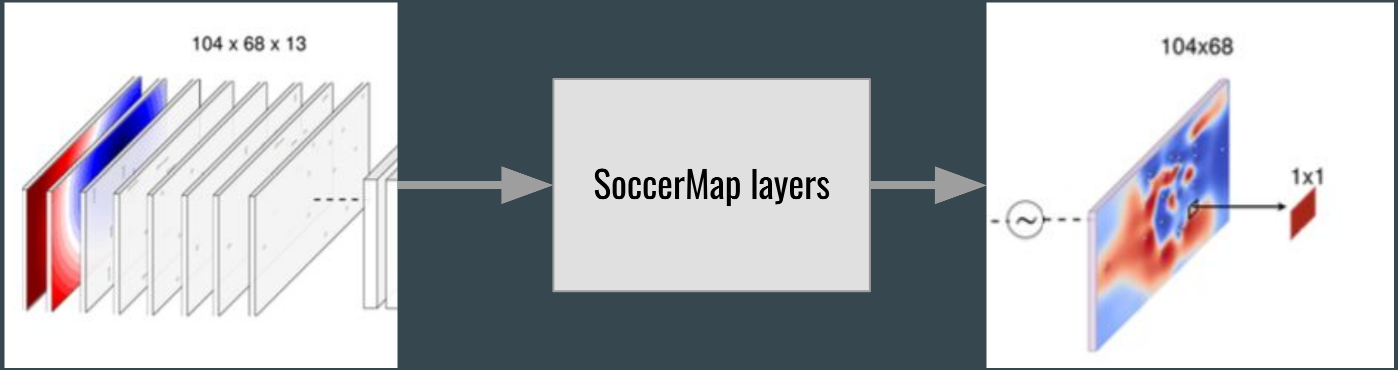


- Multi-scale convolutional neural network for spatial decision mapping (Fernandez & Bornn, 2020)
- Serves as our starting point for spatial understanding of game states

Input and Output Representation

Input

Output



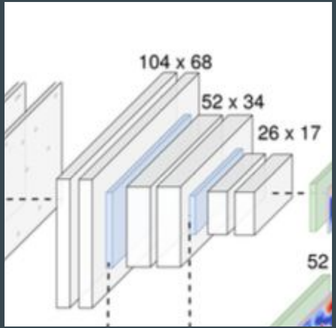
104 x 68 x 13 tensor of player, ball, and game contextual features

104 x 68 probability map (sigmoid)

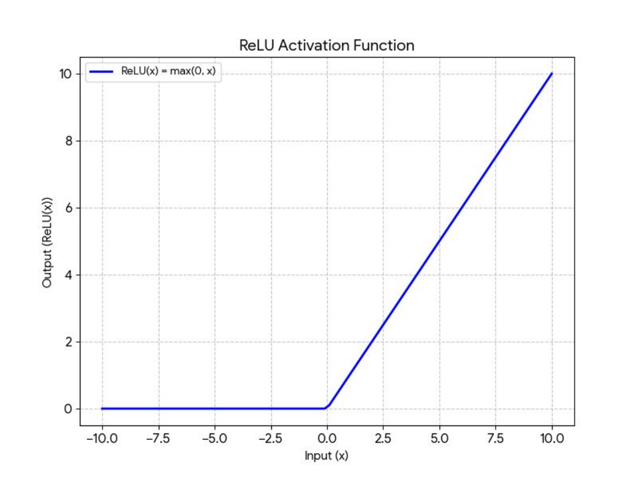
example provided in [Appendix A]

Convolution Layers

- The model uses 3 convolutional blocks, each operating at different spatial scale
- Each block scan the field (input channels) and learn patterns



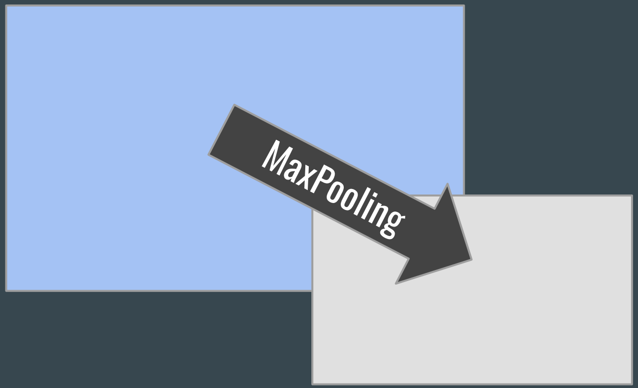
- After each convolution, a ReLU activation function is applied to filter out noise and keep only the important positive signals.



example provided in [Appendix B & C]

Maxpooling & Upsampling

MaxPooling



104 x 68

52 x 34

⋮

Upsampling

Reduces the size of feature maps by selecting the strongest signals

Restores the feature map to the original field dimensions

example provided in [Appendix D]

Loss Function

$$L(f(x_k;\theta), y_k, d_k) = \text{logloss}\left(f(x_k;\theta)_{d_k}, y_k\right)$$

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{N}\sum_i \left[y_i \cdot \log(\hat{y}_i) + (1 - y_i)\cdot \log(1 - \hat{y}_i)\right]$$

- $f(x_k;\theta)_{d_k}$ is the predicted probability at the pass destination d_k.
- $y_k \in \{0, 1\}$ is the actual outcome of the pass.

Next Steps

Dataset for Initial Evaluation

a. We will evaluate the baseline SoccerMap model using **Bundesliga 2023/24** StatsBomb dataset

Planned Model Improvements

a. Improved Feature Engineering
   i. Use input features that better match the **StatsBomb 360 dataset**
   ii. Add context-aware features (e.g. pressure, spacing, etc.)
b. Expansion of the Model Architecture
   i. **Attention Layer** to capture long-range player interactions
   ii. **Temporal Component** to model short sequences and recent actions

Questions?

Appendix A: Input Representation

Each frame is encoded into 13 spatial channels capturing player locations, velocities, distances, and directional geometry relative to the ball and goal

Player Location & Velocities (6 sparse channels)

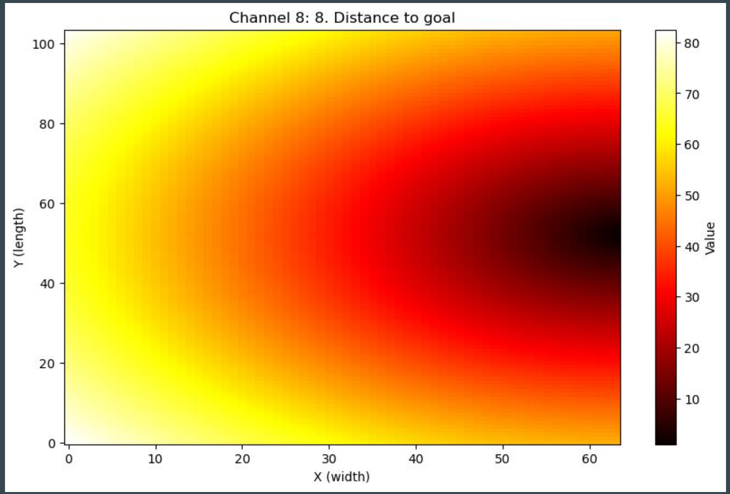
- Offense: location, X-vel, Y-vel
- Defense: location, X-vel, Y-vel

Distances & Angles (5 dense channels)

- Distance to ball and goal
- sin(angle to goal), cos(angle to goal), angle (radians)

Ball – Carrier Direction (2 sparse channels)

- sine & cosine of angle between ball-carrier velocity and teammate direction



Appendix B: Example Convolution Layers

$$\begin{bmatrix} 1 & -1 & 2 & 0 \\ 0 & 3 & -2 & 1 \\ 1 & 1 & 0 & -1 \\ 2 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & -6 & 5 \\ 0 & 2 & -3 \\ -2 & 3 & 0 \end{bmatrix}$$

$$\to \mathrm{ReLU}\left(\begin{bmatrix} 4 & -6 & 5 \\ 0 & 2 & -3 \\ -2 & 3 & 0 \end{bmatrix}\right) = \begin{bmatrix} 4 & 0 & 5 \\ 0 & 2 & 0 \\ 0 & 3 & 0 \end{bmatrix}$$

Convolution: the kernel slides over the input grid and computes a weighted sum at each location

Kernel: small matrix of weights that scans the input grid. It learns to detect patterns such as open space, pressure, or important player configurations

ReLU activation: Negative values are removed (set to zero) so the model only keeps useful positive signals

Optimization: kernel weights are adjusted during training to minimize the loss function

Appendix C: The “Convolution” & Prediction

$$\begin{bmatrix} 1 & -1 & 2 & 0 \\ 0 & 3 & -2 & 1 \\ 1 & 1 & 0 & -1 \\ 2 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & -6 & 5 \\ 0 & 2 & -3 \\ -2 & 3 & 0 \end{bmatrix}$$

*The first output row comes from “**convoluting** ” the 2×2 kernel over the top of the input*

Prediction

$$F = \begin{bmatrix} 4 & 0 \\ 2 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 0.3 & -0.1 \\ 0.5 & 0.2 \end{bmatrix}$$

$$z = F \cdot W = 2.4$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} \approx 0.9168$$

The final 1×1 convolution combines all values in the last feature map into a single score z, which is passed through a sigmoid to produce the predicted pass probability

Appendix D: Example Maxpooling & Upsampling

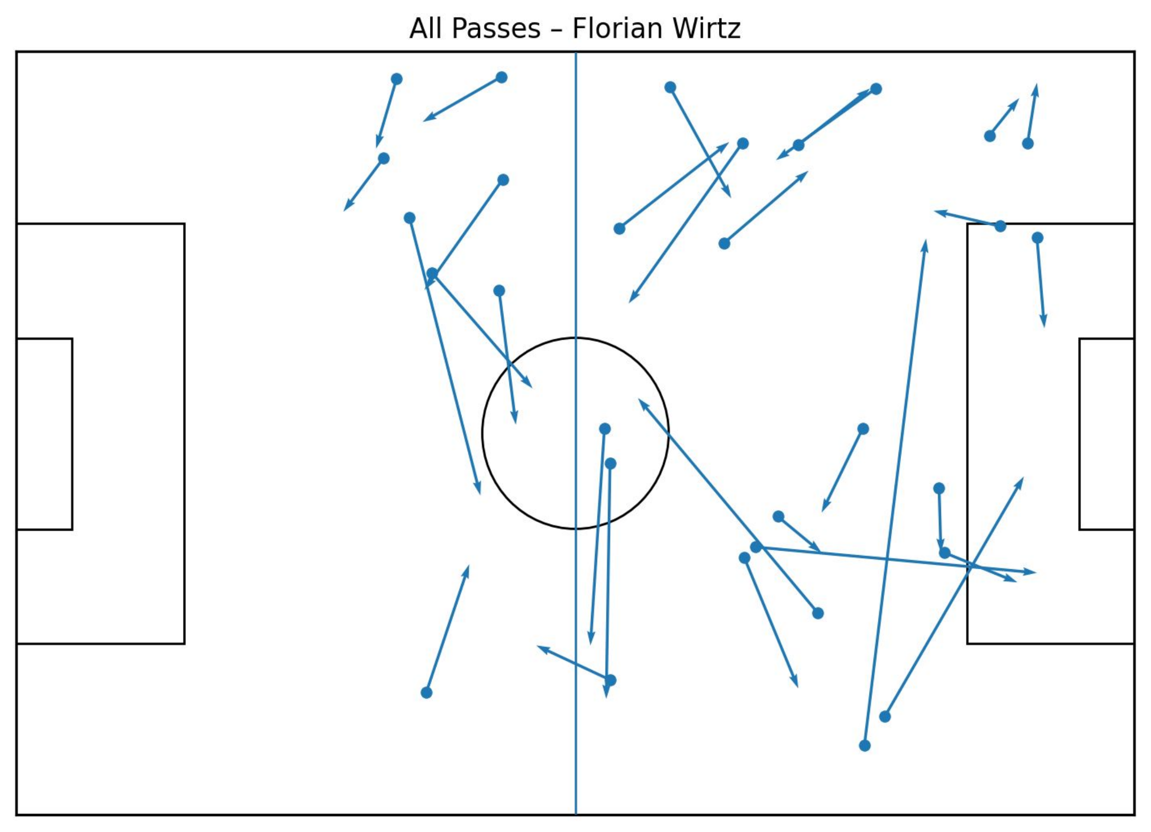
$$\mathrm{MaxPool}\left(\begin{bmatrix} \boxed{\begin{matrix}4 & 0\\ 0 & 2\end{matrix}} & \boxed{\begin{matrix}5 & 1\\ 0 & 3\end{matrix}} \\ \boxed{\begin{matrix}1 & 3\\ 0 & 1\end{matrix}} & \boxed{\begin{matrix}2 & 0\\ 4 & 2\end{matrix}} \end{bmatrix}\right) = \begin{bmatrix} \max(4,0,0,2) & \max(5,1,0,3) \\ \max(1,3,0,1) & \max(2,0,4,2) \end{bmatrix} = \begin{bmatrix} 4 & 5 \\ 3 & 4 \end{bmatrix}$$

$$\mathrm{Upsample}\left(\begin{bmatrix} 4 & 5 \\ 3 & 4 \end{bmatrix}\right) = \begin{bmatrix} 4 & 4 & 5 & 5 \\ 4 & 4 & 5 & 5 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix}$$

Maxpooling: After ReLU, select the largest value inside each 2x2 region. This reduces feature map, while keeping the strongest signals

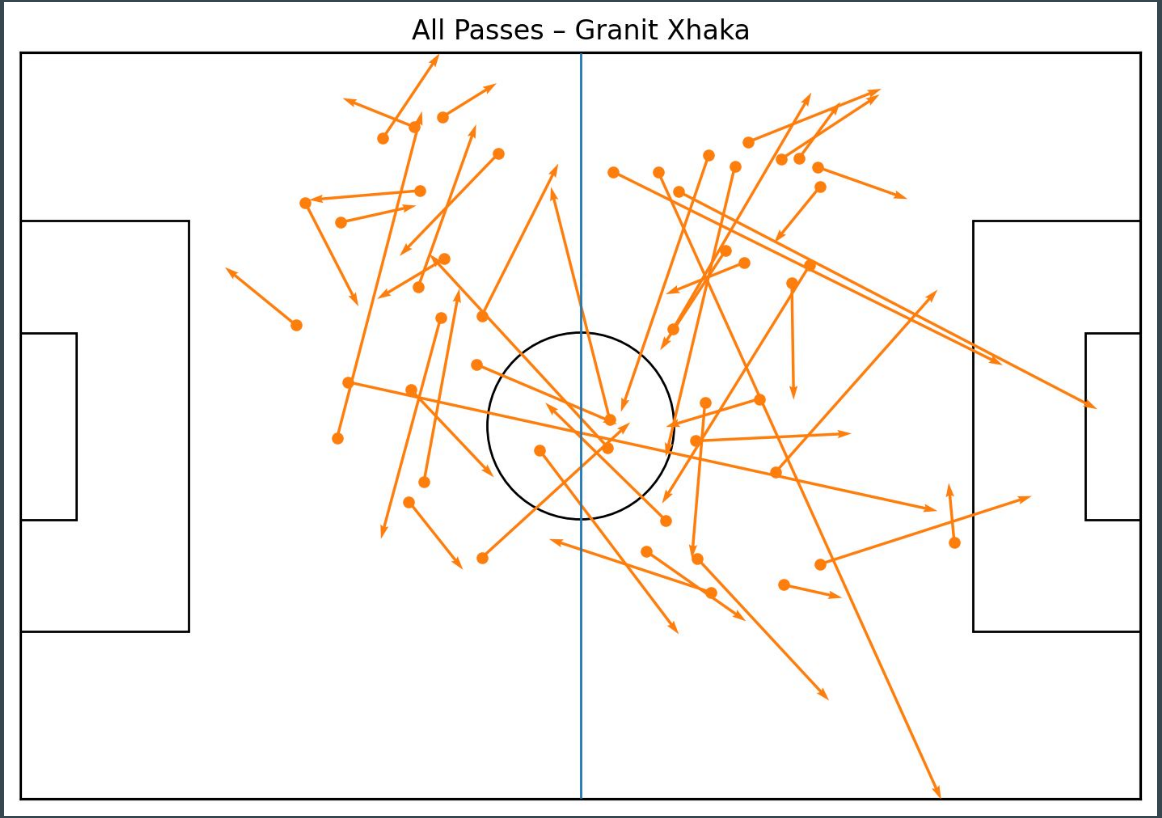
Upsampling : increases spatial resolution by repeating each value into a larger block

Appendix E: Exploratory Data Analysis



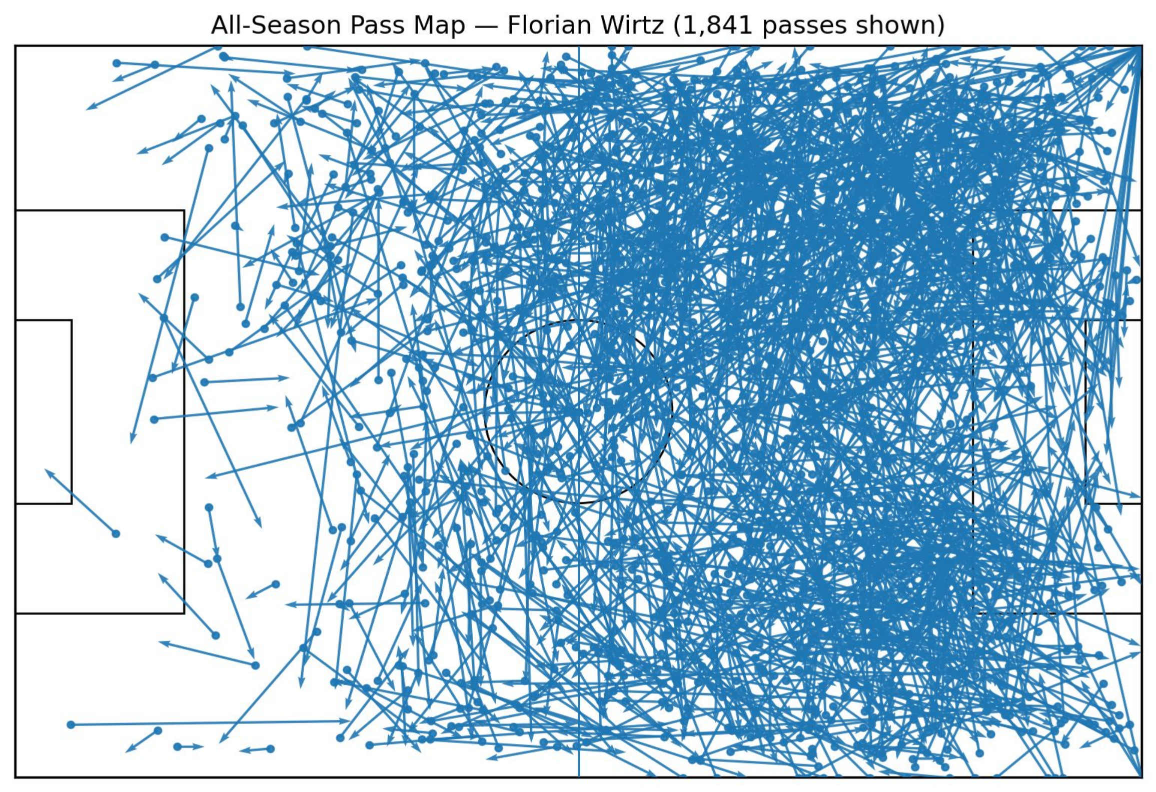
All passes of Florian Wirtz in a single game

Appendix E: EDA



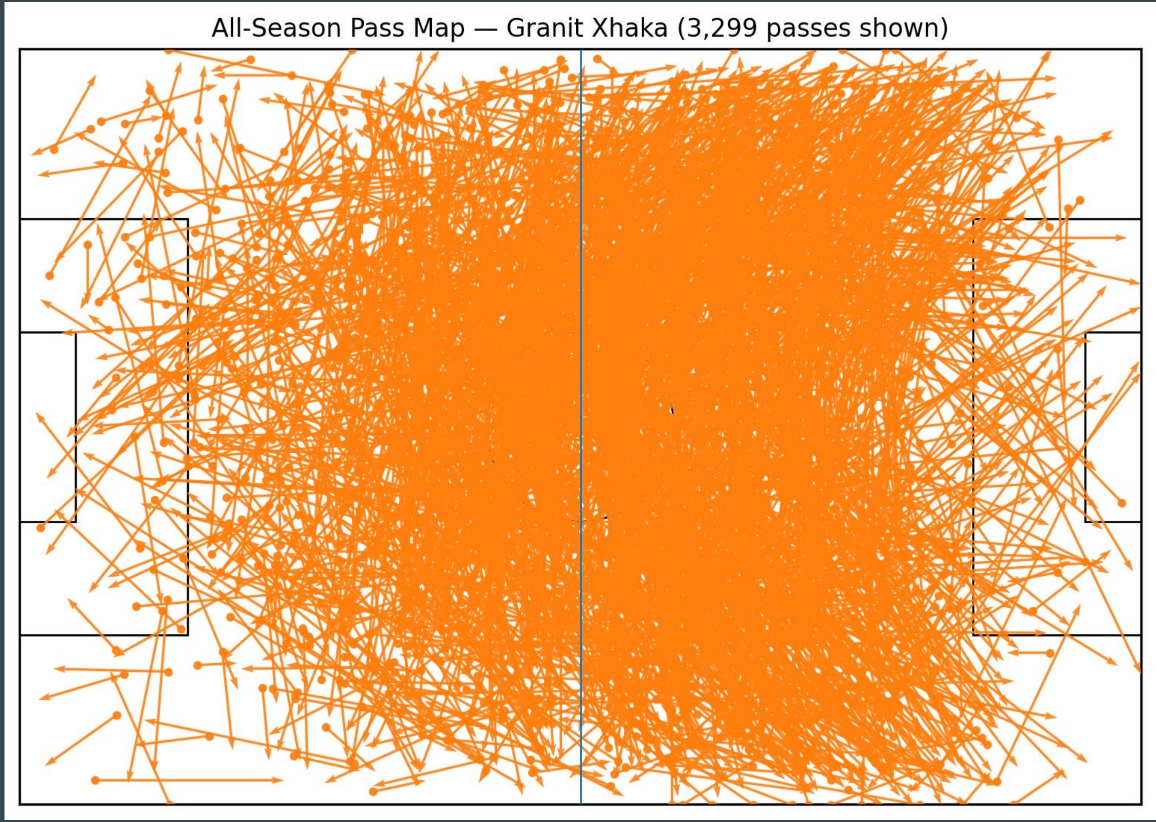
All passes of Granit Xhaka in a single game

Appendix E: EDA



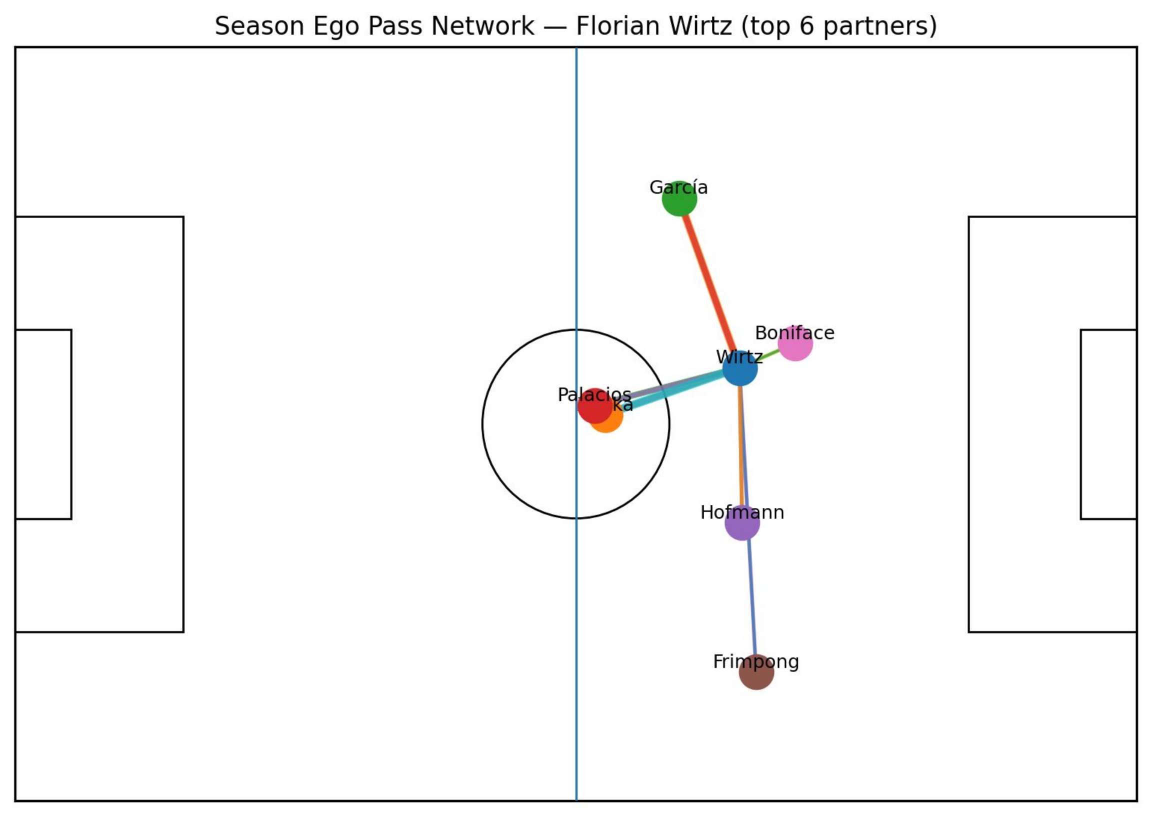
All passes of Florian Wirtz for the whole season

Appendix E: EDA



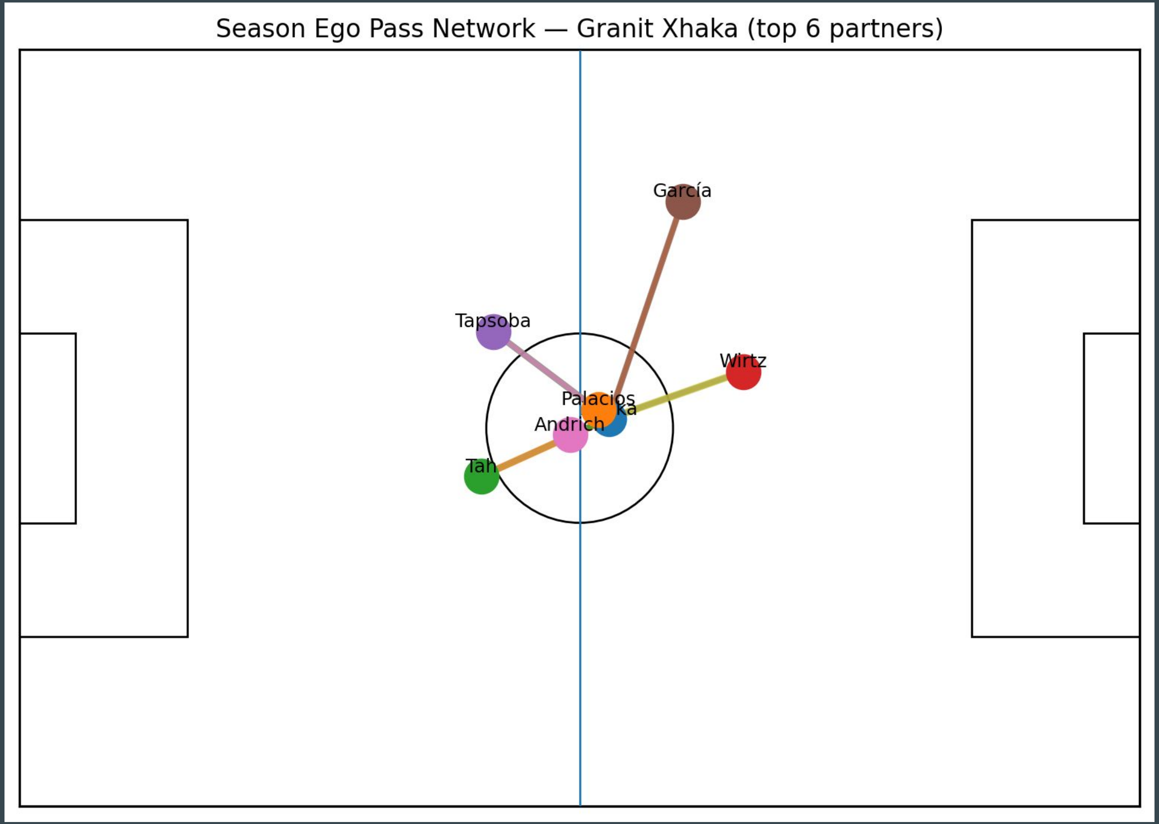
All passes of Granit Xhaka for the whole season

Appendix E: EDA



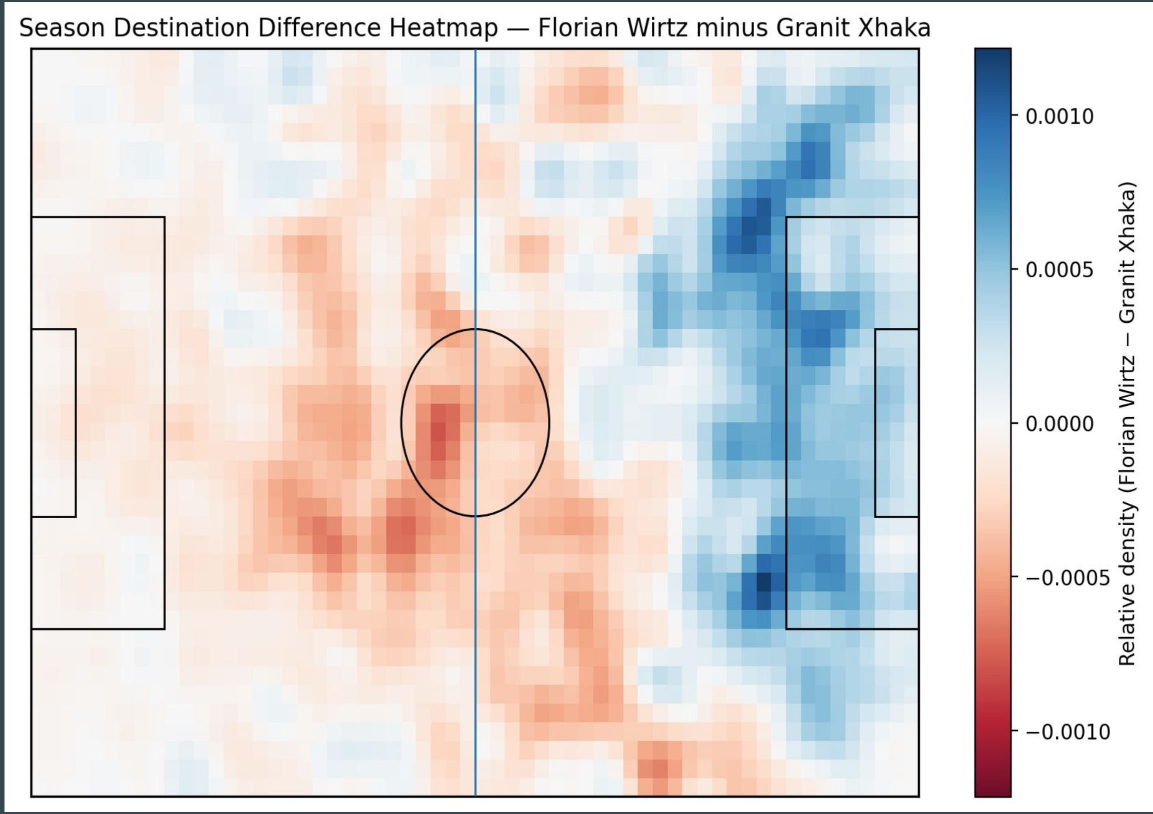
Ego Pass Network of TOP-6 partners of Florian Wirtz for the whole season

Appendix E: EDA



Ego Pass Network of TOP-6 partners of Granit Xhaka for the whole season

Appendix E: EDA



Pass Destination Difference Heatmap of Wirtz and Xhaka for the whole season