Supplementary Information:

Automatic event detection in football using tracking data

July 27, 2022

S1 Set piece events in the presence of tracking data errors

In this section, we complement the dead ball - set piece detection framework introduced in Section 2.4.1 to account for errors and inaccuracies in the tracking data. The most prevalent issues we observed were (1) the in-play/dead boolean being off-sync (namely the game has resumed but the boolean still signals the ball is dead); (2) missing ball data during in-play intervals; and (3) absence of tracking data for players that were beyond the pitch lines. To that end, we also gather information on incomplete triggers for corners and throw-ins, that is where the set piece trigger has been activated from d_1 until a given frame $d_i < d_c$. To ensure the algorithm can better detect DBEs in the presence of tracking data errors, we propose the strategy depicted on the right side of Fig. S1, that applies only if the algorithm has not detected any complete trigger, highlighted with a dashed vertical line in S1. In this case, instead of automatically adjudicating the set piece to be a free kick, we perform additional examinations on the triggers and patterns to help mitigate the three recurrent issues identified above.

If the free kick pattern (a player having the ball within their possession zone (PZ) on the first in-play frame) is activated by a player that does not move more than ϵ_{f} before the ball is passed, we conclude that this player was the set piece executor. However, to prevent errors where the dead ball boolean wrongly indicates the ball is in-play when it is dead, we perform an extra check on all the triggers and patterns for the set piece executor to ensure the event is indeed a free kick, see issue 1 on Fig. S1.

Conversely, if the free kick pattern is not activated, we first check whether a corner or a throw-in had been incompletely triggered for a player that was then not detected within $[d_{i+1}, d_c]$. Since this corresponds with high certainty to the triggering player not being tracked for being out of bounds, we impute the corresponding set piece event to that player, see issue 2 on Fig. S1.

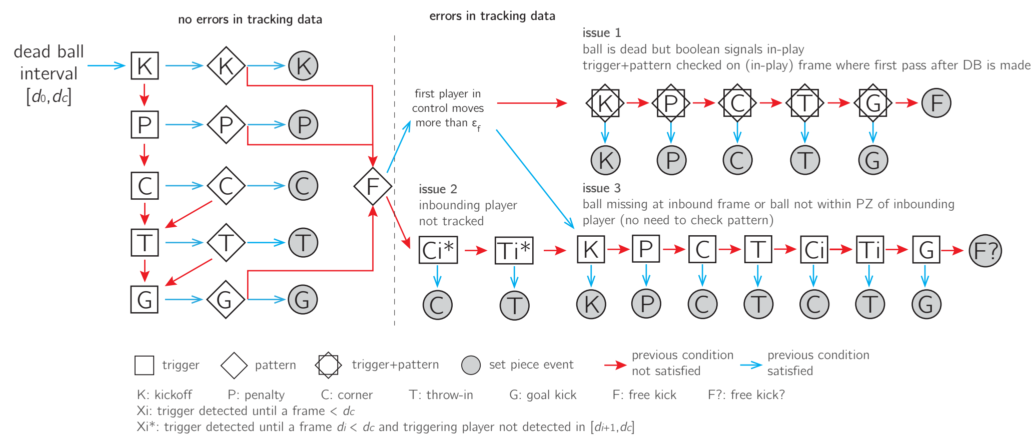

Figure S1: Flowchart to detect set pieces following a dead ball interval, separating between absence and presence of tracking data errors. For the latter, three different common tracking data issues are identified, along with the mitigating strategies to maximize set piece detection accuracy.

Finally, if the free kick pattern is activated by a player that moves more than ϵ_f (hence not being the set piece executor) or is not activated and no incomplete corner/throw-in with a missing player was detected, we have a situation of either missing ball at inbound time or the set piece executor being far from the ball due to data inaccuracies. We then propose the dead ball hierarchy shown in issue 3 of Fig. S1, only focusing on triggers since due to the nature of the issue we cannot investigate patterns (the ball is not close to any player when the game resumes). Kickoffs and penalties are checked first due to their distinct signature. Corners and throw-ins are then checked for a complete trigger, followed by an incomplete trigger. The reason for this order is that we have observed higher detection rate when favoring complete throw-in over incomplete corner. If no trigger is activated, the algorithm labels the set piece as a `free kick?`, thus signaling that the algorithm is confused on this instance and may require additional input. Incorporating these three strategies to accommodate tracking data errors increased the mean F1 score on detecting all set pieces from 85% to 95% for the datasets investigated.

S2 Event benchmarking

To assess the performance of the automatic event detection introduced in Sections 2.3 and 2.4 of the manuscript, we use the annotated events by Sportec Solutions (STS) as the "gold standard" and compare them with the discrete events predicted by the algorithm.

A discrete event is correctly detected if it is executed by the **annotated player and is within a $\pm 10\,\mathrm{s}$ window of the annotated time**.

We explore four categories of events: open play passing events, set piece events, goals and saving events, and present the detection results using confusion matrices. Open play (not from set piece) passing events include `shot`, `pass`, `cross`, `fair play`, `nutmeg`, `goal disallowed`, `backheel`, `possession loss before goal` and `bicyclekick` as defined by STS. Besides passes, shots and crosses, the proposed event detection algorithm lacks the context to discern among them, hence we relabel `fair play`, `nutmeg`, `backheel`, `possession loss before goal` and `bicyclekick` as `pass`, and also `goal disallowed` as `shot` for better comparisons. Set piece events are passing events that resume the game after a dead ball interval, namely `kickoff`, `penalty kick`, `corner kick`, `throw-in` and `free kick`. We include the additional category `free kick?` to depict the instances where the algorithm was confused, see Fig. S1. Goal events are signaled in STS by either `own goal` or as the outcome `successful shot` of a `shot` event. We included the category `goal?` to reflect the uncertainty on shooting events where the ball surpasses one of the goals, the game is interrupted and ends before resuming, and hence we lack context as to whether a goal has been scored or not (ball may have gone to corner or goal kick for instance).

Finally, savings events by goalkeepers include `saved shot` as an outcome of a `shot` event and the event `ball claimed`, but there was no STS annotation for loose ball receptions. To that end, we assume that these occurred as a result of passing events where the recipient was the goalkeeper and the passer was an opponent. Since one should draw limited conclusions from this benchmarking, the saving events' results are included below and not in the manuscript.

S2.1 Saving events

The results for saving events are shown in Fig. S2. The performance of the algorithm degrades when predicting saves, claims and loose ball receptions. Upon detailed inspection, the errors committed by the algorithm stemmed from a myriad of reasons. The main source pertains to annotated shot-save events where the shooter was closely guarded by a defender and the algorithm wrongly assumes they were the defender, due to both tracking data inaccuracies and model limitations, hence no save was recorded. There were also instances when the shot originated from an offside position that was later called by the referee (hence the save was not annotated by STS), but the tracking data still signals the ball being in-play at the time of the save. Moreover, since loose ball receptions are not annotated by STS, there are instances in which these receptions occur but there was no annotated record of the goalkeeper receiving the ball, thus leading to benchmarking errors. Finally, the event detection algorithm is based off the logic outlined in Fig. 6 in the manuscript, which may not fully capture the context on certain saving events, e.g.

the importance of ball z-coordinate for claims or the speed of shots to discern between a shot-save sequence and a pass-loose ball reception sequence.

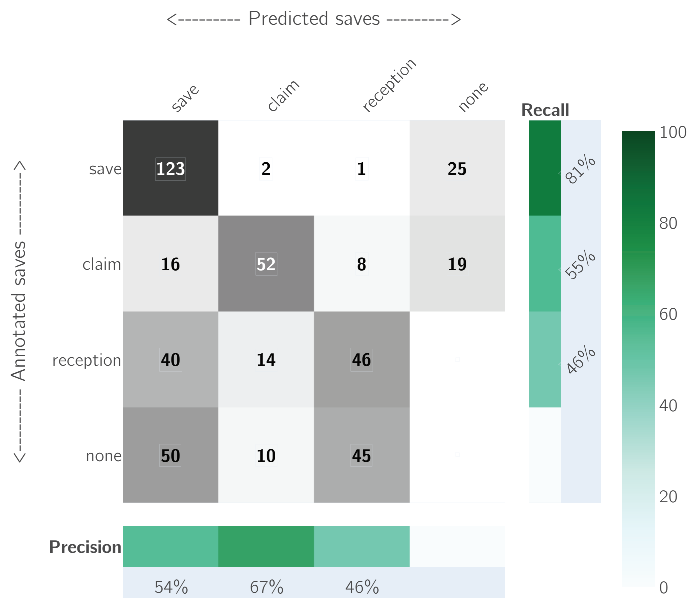

Figure S2: Confusion matrix comparing the saving events predicted by the event detection algorithm with the annotated events from STS, together with precision and recall for each category. Matrix cells are colored according to the relative number of instances per row.

S3 Tracking and event data accuracy for passing events

In this section, we perform a data quality analysis for both tracking and event data, with the objective of understanding what are the potential limitations of that will impact the accuracy of the proposed algorithm for automatic event detection.

To that end, for each passing event annotated by STS, we analyze the minimum distance D between the ball and the player that executed the event on a $\pm 10\,\mathrm{s}$ window centered on the annotated time, which should ideally be a small number. In addition, we also collect the time lag τ between the annotated time t_{event} and the instant t_D in the tracking data where such minimum distance is attained, namely $\tau = t_{\text{event}} - t_D$ which evaluates the time accuracy of the manual annotation, since the tracking data was synced with the start of period. The ball-player distance is the backbone of the event detection algorithm, therefore this analysis allows us to gauge how large does the PZ need to be to effectively capture in-game events. Furthermore, it also sheds light on the accuracy of the time annotations that are provided with the manually annotated events, which can later inform the event benchmarking process.

For this analysis, we group the games by different tracking data provider, namely nine games of Track160 (hereafter referred to as provider A), 19 games of Tracab (hereafter referred to as provider B) and three games of Hawk-Eye (hereafter referred to as provider C), for a total of 9K, 18K, and 3K passing events respectively. The results are shown as hexagonal scatterplots on Fig. S3, together with the marginal distributions of time lag τ and minimum distance D. For visualization purposes, we only show the minimum distances below 1.5 m, since larger minimum distance values correspond to errors in either the tracking data (swapped players) or the event data (wrong player annotation or time lag larger than ± 10 s). For reference, the percentage of events in which minimum distance was above 1.5 m was 1.8%, 1.1% and 2.2% respectively.

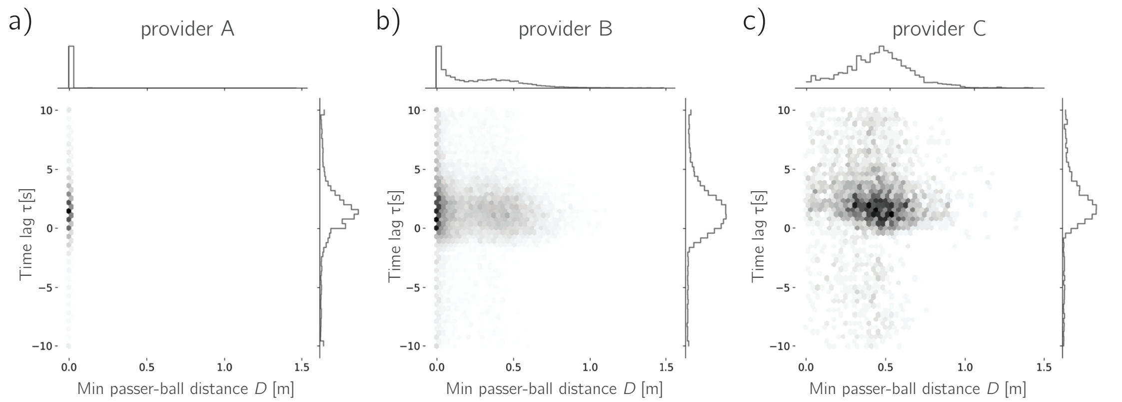

Figure S3: Hexagonal bin scatterplot and marginal distributions of minimum passer-ball distance (in meters) and time lag (in seconds) for all passes/crosses/shots, grouped by tracking data provider: a) provider A with nine games and 9K events; b) provider B with 19 games and 18K events and c) provider C with three games and 3K events. Minimum distance between the ball and the player who is annotated as passer, using tracking data within the ± 10 s window centered around the manually annotated time. Positive time lags indicate minimum distance in tracking data occurred before the annotated event time.

The main conclusion regarding event data is that the manually annotated events exhibit a mean positive lag of 1.5 s with the minimum passer-ball distance attained with the tracking data, which signals that the events were annotated later than when they occurred. Moreover, there are 16% (resp. 8%) of the passing events where $|\tau|$ is greater than 5 s (resp. 7.5 s), which should be taken into account when benchmarking and using the event data for further analysis.

For tracking data, the most important takeaway is that the distribution of minimum distance passer-ball differs significantly across providers. For the provider A dataset, 99% of the 9K passing events exhibited a minimum passer-ball distance of exactly 0. For the provider B dataset 80% (resp. 99%) of the 18K passing events exhibited a minimum distance of less than 50 cm (resp. 1 m), whereas for the provider C[1] dataset 67% (resp. 99%) of the 3K passing events exhibited a minimum distance of less than 50 cm (resp. 1 m). These numbers suggest that choosing a PZ radius $R_{pz} < 1$m for provider B and provider C would negatively impact the performance of the algorithm: the minimum distance between the passer and the ball needs to be smaller than the R_{pz} for the possession (and therefore the subsequent event) to be detected by the proposed methodology. Conversely, for provider A the choice of R_{pz} will not greatly affect the capacity of the algorithm to capture possession and events.

S4 Optimal choice of parameters

S4.1 Possession zone

The analysis on player-ball distance at passing time for different providers in Section S3 can be directly used to determine the choice of the possession zone radius R_{pz}, which has a notable impact on the performance of the possession and event detection algorithms. An optimal choice of the PZ radius depends on the quality of the tracking data with respect to the distances between player and ball, as shown in Fig. S3, and should be a balance between minimizing false positives (instances where the player does not come in contact with the ball but a large R_{pz} incorrectly assigns possession) and false negatives (instances where a player does come in contact with the ball but not within R_{pz} due to tracking data inaccuracies).

The value of the required PZ radius can vary between tracking data providers and across competitions, since it depends on the accuracy of the tracking data. After discussions with experts at FIFA, we argue the R_{pz} should be within 50 and 150 cm. The optimal PZ radius for the different datasets presented here was found by running the algorithm with $R_{pz} = \{50, 75, 100, 125, 150\}$cm and then evaluating the F1 score in detecting open-play passing events compared to the manual annotations, using the event benchmarking approach described above. The results are shown in Fig. S4, grouped by tracking data provider, where for each one we highlight the R_{pz} that renders the highest F1 score. These values are 50 cm for provider A and 1 m for both provider B and provider C. Unsurprisingly, these optimal values are in close relationship with the minimum distance player-ball results in Section S3 and Fig. S3, since the R_{pz} should reflect the minimum player-ball distance whereby most passing events can be captured. The radius of the duel zone was taken to be $R_{dz} = 1$ m for all providers.

[1] Player tracking data was given as a 3D pose consisting of 18 joints, but for this study we have use the 2D center of mass data computed by provider C from the 18 joints.

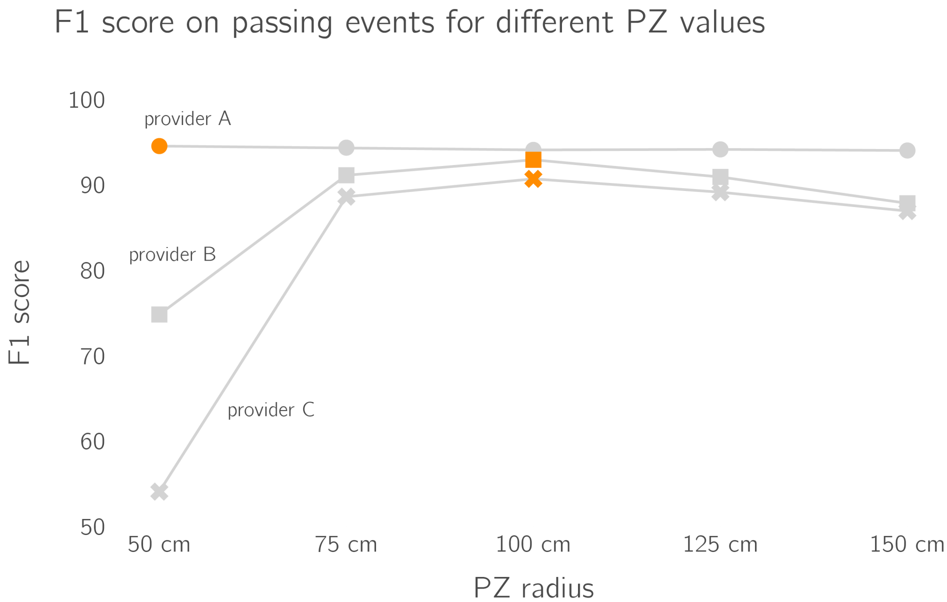

Figure S4: F1 score on detecting open-play passing events (passes/shots/crosses) for different tracking data providers and values of the possession zone radius. The optimal PZ radius and F1 score for each provider is highlighted in orange.

S4.2 Set piece tolerances and hyperparameter selection

The proposed event detection framework relies various tolerances to more effectively capture the set piece triggers and patterns. The need for tolerances arises not only from tracking data inaccuracies, but also to accommodate situations where the football rules regarding spatial location of players are not strictly enforced (*e.g.* non-penalty takers entering the penalty area before the shot, or a goal kick pass from beyond the goal area). To find the tolerance values that maximize the accuracy in set piece detection, for each type of set piece event we perform a grid search on tolerance configurations, and for each combination we benchmark the results of automatic set piece detection. This process has been undertaken using all available games together, hence the tolerances are the same for all providers. More extensive datasets can be helpful to further tailor the tolerances for each separate provider.

First, we focus on kickoffs, which are determined by $\epsilon_{\mathrm{k1}}, \epsilon_{\mathrm{k2}}$. We find that $\epsilon_{\mathrm{k1}} \in 3-5\,\mathrm{m}$ and $\epsilon_{\mathrm{k2}} \geq 2\,\mathrm{m}$ render perfect detection, while lower values of ϵ_{k1} degrade the performance of the algorithm. For instance, the kickoff F1-score drops to 9%, 82% and 95% if we reduce ϵ_{k1} to 0, 0.5 and 1 m respectively, while keeping $\epsilon_{\mathrm{k2}} = 2\,\mathrm{m}$.

Secondly, we investigate the tolerances $\epsilon_{\mathrm{p1}}, \epsilon_{\mathrm{p2}}, \epsilon_{\mathrm{p3}}$ on penalty kicks. Interestingly, setting the tolerance that controls the location of the remaining players $\epsilon_{\mathrm{p3}} = 0\,\mathrm{m}$ results in no detected penalty kicks, whereas $\epsilon_{\mathrm{p3}} = 2\,\mathrm{m}$ only achieves a maximum F1-score of 82%. The optimum results were attained with $\epsilon_{\mathrm{p1}} = 2\,\mathrm{m}$, $\epsilon_{\mathrm{p2}} = 2\,\mathrm{m}$ and $\epsilon_{\mathrm{p3}} = 4\,\mathrm{m}$. However, the games analyzed only comprise 13 penalty kicks, hence a more extensive dataset is required before drawing further conclusions.

Finally, we study the remainder of set pieces together since the detection framework shown in Fig. 5 follows a hierarchical order from corner kicks to free kicks. By exploring the detection results on multiple events, the optimal tolerance configuration will inevitably depend on the objective function that we choose to optimize. Choosing a homogeneous weighted F1-score, that is errors are penalized equally regardless of the set piece event, we found that $\epsilon_{\mathrm{c}} = 3\,\mathrm{m}$, $\epsilon_{\mathrm{t}} = \epsilon_{\mathrm{g}} = 0.5\,\mathrm{m}$ and $\epsilon_{\mathrm{f}} = 1\,\mathrm{m}$ render the optimal weighted F1-score (93.4%). Note that other choices of objective functions (weighing set piece mistakes differently) may lead to different values for the tolerances.

In terms of hyperparameter selection, we performed a cross-validation study on a subset of games and found that $\{\epsilon_\theta = \cos 10°, \epsilon_v = 5\,\mathrm{m\,s^{-1}}, \epsilon_s = 10\,\mathrm{cm}\}$ were the optimal hyperparameters for provider B and provider C and $\{\epsilon_\theta = \cos 10°, \epsilon_v = 1\,\mathrm{m\,s^{-1}}, \epsilon_s = 5\,\mathrm{cm}\}$ for provider A.

S5 Match contextualization: proposed attributes

The computational framework presented here gives rise to four in-game categories: passing events, receiving events, control sequences and no-possession sequences. Passing events are discrete and comprise the labels `pass`, `shot on target`, `shot off target`, `cross`, `own goal`. Receiving events are discrete and include saving events (`save and deflect/retain`, `claim and deflect/retain`, `reception from loose ball`, `unsuccessful save`), receiving events (`reception` and `interception`), instances where a player gains and loses possession on the same frame (which we label as a passing event but is also a receiving event) and instances within a longer control sequence involving multiple players where no pass is made but possession changes. A passing event is typically followed by a reception event, but this logic does not always hold: a passing event may lead to a dead ball interval; a receiving event may stem from a duel in which one player gains possession without a prior passing event.

Control sequences are continuous events that start with a receiving event and end when: (1) there is a change in possession with or without a passing event there can be a duel where a player turns the ball over to an opponent or a player can gain control of the ball from a teammate; (2) the game stops; (3) or the game ends. Finally, no-possession sequences are time intervals where no player has the ball within their PZ and there is no duel.

Combining the predicted events table with these four categories enables the partition of in-game intervals into non-overlapping sequences. This concept is illustrated in Fig. S5a for the following sample scenario: the game is resumed by a blue player who passes the ball (1) to a teammate that receives it (2) progresses and passes it (3); the ball arrives to a blue teammate who passes the ball with one touch, hence the reception and the pass occur on the same frame (4); the ball is received by a blue player (5), who maintains possession until a red player comes nearby and a duel starts (6); the blue player then turns the ball over (7) to the red player, who passes the ball (8) to a red teammate; this player receives the ball (9) and after a short possession passes the ball (10). The ball goes out of bounds and the game is therefore stopped.

Both the discrete events and the control sequences can be complemented with several contextual attributes that can be computed or modeled from the tracking data, in the form of extra columns to the final events table. In this section, we illustrate how to evaluate some important attributes identified by FIFA, but it should be noted that the list presented below is by no means exhaustive.

For passing events (discrete), we evaluate the **outcome**: `completed` if it reaches a teammate, `intercepted` if it reaches an opponent, `duel` if there is a duel at reception time, `unclear` if there is a duel at passing time, `goal` if a goal is scored, `dead ball` if a dead ball interval follows, and `end of half` if the game ends before the pass reaches its destination. We also collect the **distance traveled by the ball** until the next receiving event (if applicable), the **location** of the passer relative to the opponents' formation, the **outgoing angle** of the ball, the **opponents overtaken** and the **span** of both teams, see Fig. S5b-d.

For receiving events (discrete), we evaluate the **distance traveled by recipient** since previous passing event (if applicable), the **origin** of the pass (directly linked to the pass **outcome** described above) the **location** of the receiver relative to the opponents' formation, the **incoming angle** of the ball and the **span** of both teams, see Fig. S5c.

For control sequences, we evaluate the **outcome**: `pass` if it ends in a passing event, `goal` if it ends in a goal, `lost` if possession changes to an opponent without a passing event, `maintained` if possession changes to a teammate without a passing event, `deflect` if the goalkeeper deflects it, `dead ball` if a dead ball interval follows without a passing event, and `end of half` if the game ends before there is a passing event, a dead ball or a change in possession. We also monitor the **distance traveled** by the player in control as well as the **opponents overtaken** during this continuous sequence.

Finally, for the dead ball event `goal` we update the **score** attribute according to the team that scored, while evaluating if there was an assist that lead to the goal. The scoring player, scoring team and assisting player (if any) metadata is included in the **description** attribute for additional clarity. Similarly, for the set piece event `penalty kick` we annotate whether the penalty was scored or missed by a given player and team, and provide this

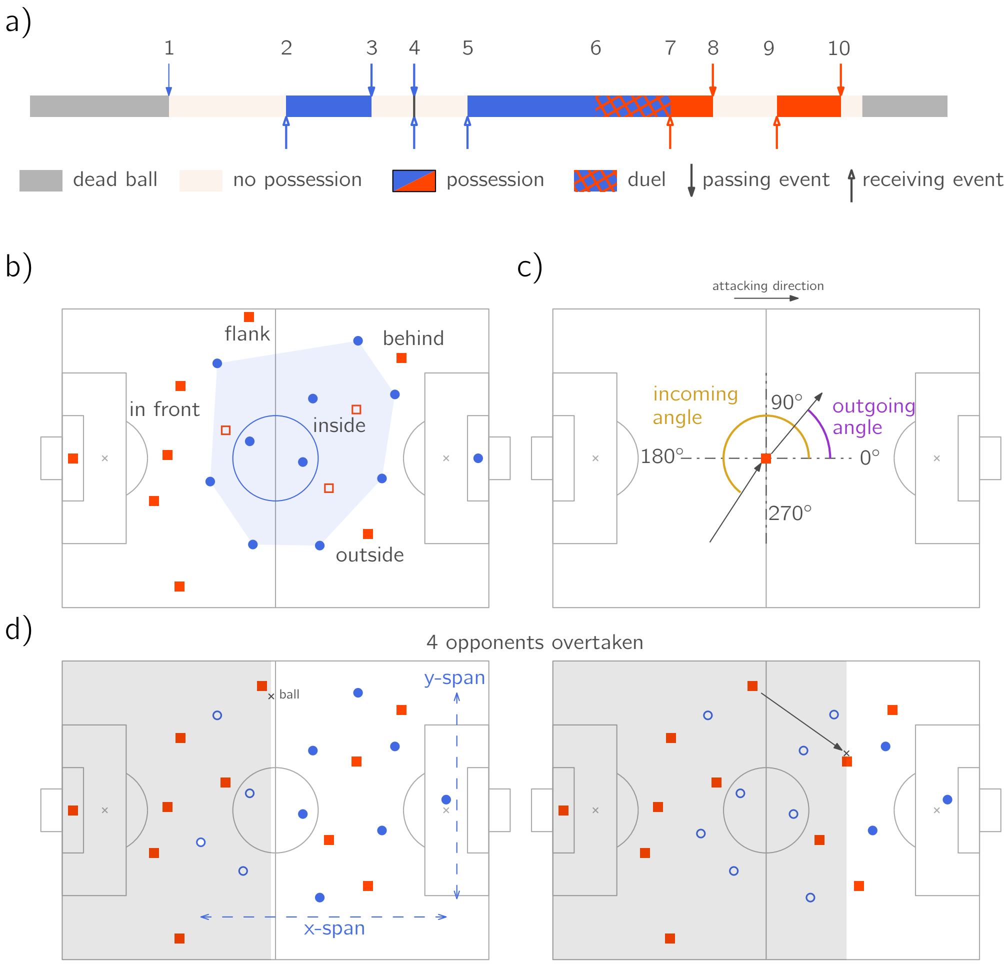

Figure S5: a) Sample scenario of how the in-game sequences of a football game may be partitioned using passing/receiving events and control/no possession sequences. b) Different passing/receiving locations of a red player relative to the blue team. c) Computation of incoming (yellow) and outgoing (purple) ball trajectory angles for an arbitrary player. d) The number of opponents overtaken by a passing event or control sequence for the red team is measured by the difference in blue players behind the ball before and after the event. x and y-span of blue team is also shown, and may be analogously computed for the red team.

information on the **description**.

These attributes therefore lead to a more granular definition of events, and enable the use of automatically generated event data for more advanced statistical analyses of football games, as shown in Section 3.2 of the manuscript and S6.

S6 Additional applications

In this section, we present three additional applications to the ones introduced in Section 3.2 of the manuscript.

S6.1 Match-aggregated possession information

First, since the proposed methodology detects possession on each frame, we can visualize the aggregated possession time of both teams, as well as a player breakdown. The purpose of the aggregated possession visualization is to provide a high-level description of the game from the possession standpoint, which also incorporates the non-negligible time where the ball is not within the PZ of any player (*e.g.* ball is traveling), see Fig. S6a. We can also visualize the player possession time on both periods of the game, see Fig. S6b, where we focus on the 10 players with the most possession throughout the entire game while distinguishing between home and away.

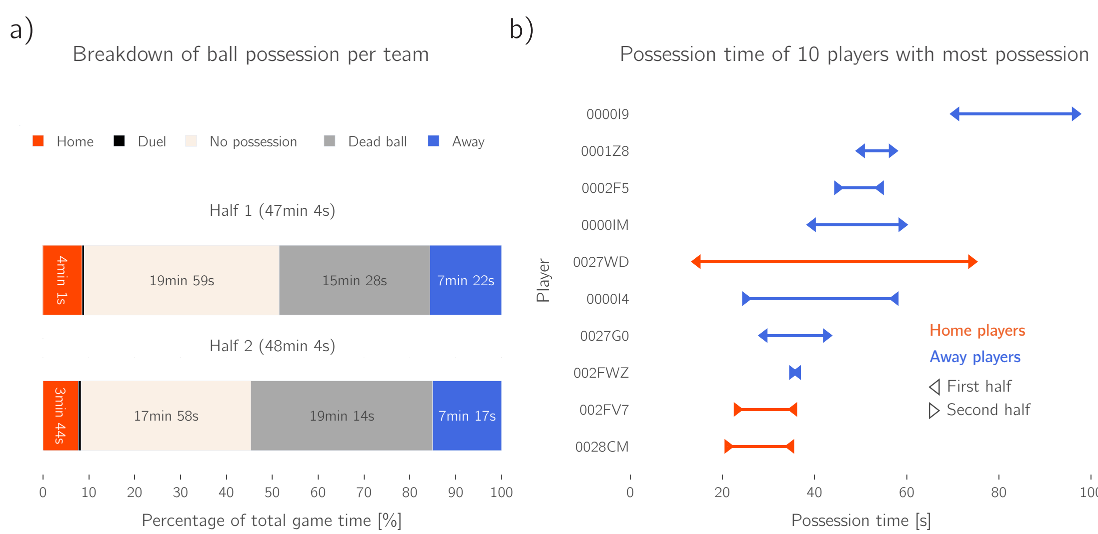

Figure S6: Aggregated possession information for a single match. a) Possession time by teams and periods, including dead ball and ball not in control of any player (no possession). b) Ranking of total possession time per player and period, showing only the top 10 players with the most combined possession across periods.

S6.2 Timeline of events

Second, we can describe the game using a timeline of the most relevant in-game and set piece events, see Fig. S7. This visualization offers a high-level description of the game from the events standpoint, and is designed to be interactive, such that different events can be toggled on/off and additional information on each event, for instance player, team, game time and a brief description, is available. Such an interactive timeline may be enhanced with a video tool to display properly synced footage for each event.

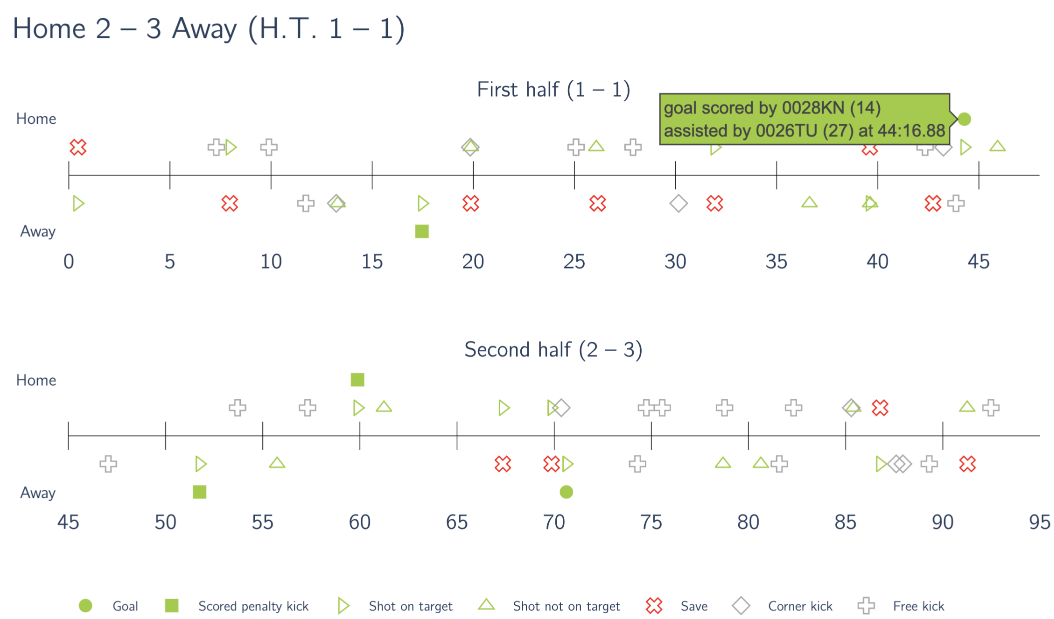

Figure S7: Timeline of events for a match, showing goals, penalties scored, shots (on/off target), corner kicks, free kicks and saves. This timeline is designed to be interactive, offering metadata (player, team, time and brief description) on each event in the web version, such as the highlighted message for the first-half goal scored the home team.

S6.3 Multiple match-aggregated possession information

For this application, we use again data from a team for which we have five games and show the player possession time per period for multiple games. The possession time can be shown both in absolute seconds and as a percentage of the total playing time (excluding dead ball) of each player, to avoid penalizing for players getting substituted, see Fig. S8.

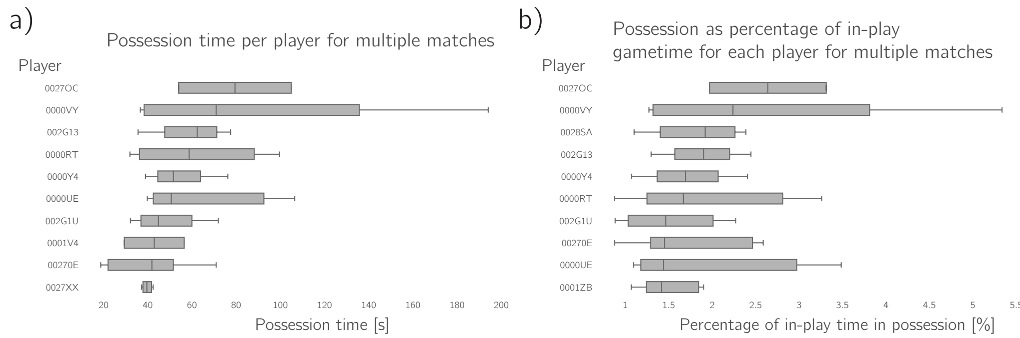

Figure S8: Possession time per player and match shown in boxplots. a) Total seconds. b) Percentage of in-play gametime, excluding dead ball time.

S6.4 Dataset-aggregated possession and event information

For the last application, we can aggregate data from many the players and teams according to any metric of interest to investigate how they compare to each other. For this example, we consider the seven games on the FCWC20 with tracking data from provider B, and focus on two different metrics. The chosen representation is a scatterplot, where the players are depicted by individual dots. The metric count is shown in the horizontal axis, multiplied by a playing time factor ≤ 1, defined for each player as the ratio between their in-play time over the total in-play playing time, to account for the reduced playing time from substitutes. The vertical axis corresponds to the mean in-possession percentage time for that player, computed as the sum of their in-possession seconds over the sum of their in-play playing time. Hence, players to the right of the graphic exhibit a larger metric count, whereas players to the top of the graphic exhibit more possession time. The symbols reflect the different number of games the player participated in, from one to three.

The first metric is the number of forward-going successful passes (outgoing angle lesser than 90° or greater than 270°, see Fig. S5c) executed on the attacking half aggregated by player, see Fig. S9a, where the players are colored by their mean relative x-coordinate at passtime (0 for own endline, 1 for opposing endline, hence larger x-coordinate value means a more advanced passing position and $x \geq 0.5$ signals the attacking half). The second metric accounts for passes that attempt to break closed defenses, and we tentatively define it as the number of completed passes executed on the final third, surpassing at least two opponents and that at the time of pass the opponent's team x-span was less than 20 m, see Fig. S9b. For this example, the players are colored by the total number of minutes (both in-play and dead) played across the tournament.

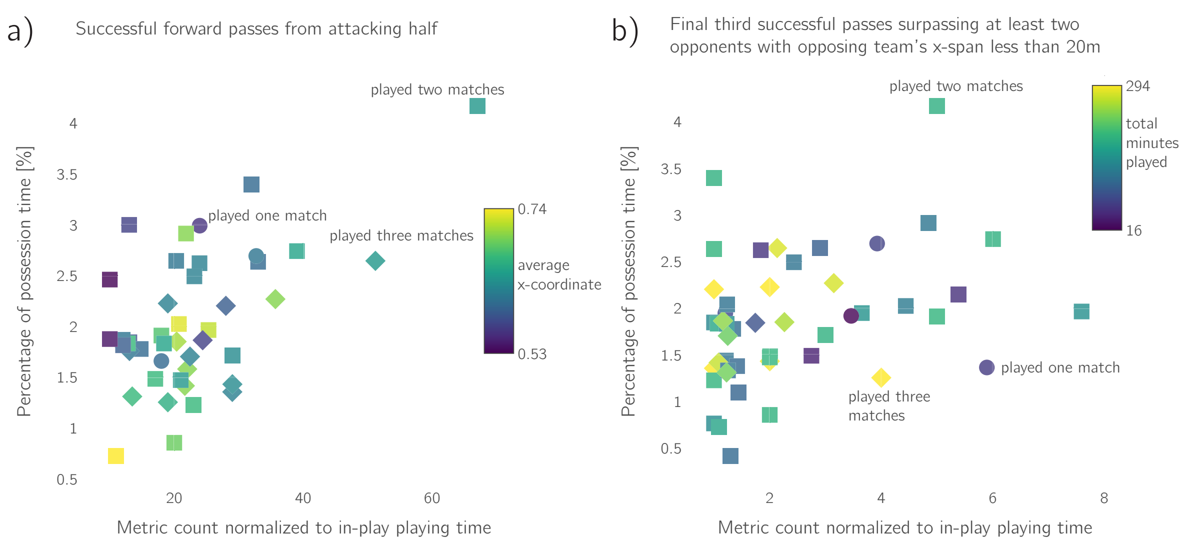

Figure S9: Scatterplots of two passing metrics for all players in the FCWC20. Each player is a dot, the symbol indicates the amount of matches played, the x-axes corresponds to the metric count multiplied by the ratio between the player's in-play time over the total in-play time, whereas the y-axis corresponds to the in-possession percentage time for that player. a) The metric is the number of successful passes going forward executed from attacking half, colored by mean relative x-coordinate at time of pass (0 is own endline, 1 is opposing endline). b) The metric is the number of final third successful passes that surpassed at least two opponents, in situations where the opposing team's x-span at time of pass was less than 20 m, colored by the total amount of minutes on the pitch (both in-play and dead).