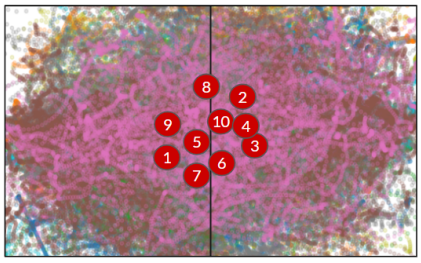
FootballCPD: Formation and Role Change-Point Detection of Football Players from Spatiotemporal Tracking Data — Supplementary Materials

1 FURTHER DESCRIPTION OF THE APPLIED ALGORITHMS

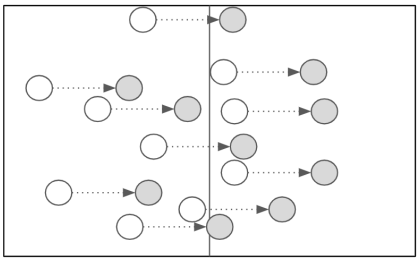
Here we give further description of the two algorithms that we employ in the process of FootballCPD: role representation (Bialkowski et al., 2014) for the frame-by-frame role assignment and discrete g-segmentation (Song and Chen, 2020) for the formation/role change-point detection. We also explain how to implement these algorithms when building our code in Python.

1.1 Role Representation

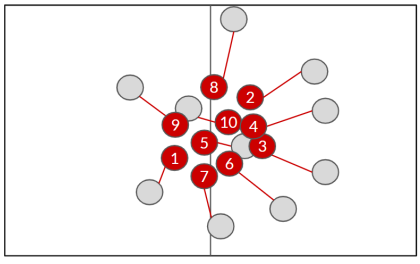
The core of the role representation is an EM algorithm that assigns a unique role to each player at every frame. This algorithm is analogous to that of K-means, except using the Hungarian algorithm in E-step to avoid for multiple players to take the same role at the same time. Fig. 1 depicts the specific learning procedure.



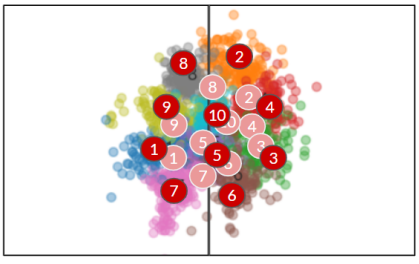
(a) Assign a unique role label (from 1 to 10) to each outfield player and estimate probability densities of each role. The numbered circles indicate the mean locations of the role distributions.



(b) For each frame, translate the player locations by the team average to reduce the situational variation.



(c) **E-step:** Apply the Hungarian algorithm per frame to reassign a roles to players using the cost matrix based on the log probability that each role generates the corresponding player location.



(d) **M-step:** Update the role probability densities using the labels assigned in (c).

Figure 1: The learning procedure of role representation.

The algorithm terminates after 3–4 iterations if there is no significant improvement in the assignment cost. The result satisfies that: (1) each player takes varying roles throughout the given period, (2) no two players in a team have the same role at the same time, and (3) the role-by-role probability distributions exhibit minimal overlap.

1.2 Discrete g-segmentation

Discrete g-segmentation is an extended version of g-segmentation first proposed by Chen and Zhang (2015) and reformed by Chu and Chen (2019). g-segmentation first constructs a similarity graph G such as a minimum spanning tree on the given observations. Then, the edges can be classified into three types for each time stamp t: type-0 edges connecting observations before and after t, type-1 edges connecting observations before t, and type-2 edges connecting observations after t. It gives the counts $R_0(t)$, $R_1(t)$, and $R_2(t)$ of the corresponding types of edges. The generalized edge-count scan statistic is defined as

$$S(t) = \begin{pmatrix} R_1(t) - \mathbb{E}[R_1(t)] \\ R_2(t) - \mathbb{E}[R_2(t)] \end{pmatrix}^T \mathbf{\Sigma}^{-1}(t) \begin{pmatrix} R_1(t) - \mathbb{E}[R_1(t)] \\ R_2(t) - \mathbb{E}[R_2(t)] \end{pmatrix} \tag{1}$$

where $\mathbf{\Sigma}(t)$ is the covariance matrix of $(R_1(t), R_2(t))^T$ under the null distribution obtained by shuffling the time indices. Since similar observations are more likely to be connected by edges than distinct ones, large $S(t)$ implies the small within-group distances and large between-group distances (where "groups" indicate the time intervals divided by t). As such, the algorithm concludes that $\tau = \arg\max_t S(t)$ is a change-point if $S(\tau)$ exceeds the pre-defined threshold corresponding to a certain p-value (0.01 in our case).

However, g-segmentation has a limitation that it cannot handle data with repeated observations, as the optimal similarity graph is not uniquely defined. Discrete g-segmentation overcomes this drawback by using the *averaging* statistic $(R_{1,(a)}(t), R_{2,(a)}(t))$ or the *union* statistic $(R_{1,(u)}(t), R_{2,(u)}(t))$ introduced in Zhang and Chen (2017) as alternatives for $(R_1(t), R_2(t))$ in Eq. 1. That is, they obtain a unique scan statistic either by "averaging" the test statistic over all optimal graphs, or by calculating the test statistic on the "union" of all optimal graphs. According to Zhang and Chen (2017), one can obtain these statistics using the similarity graph G_0 constructed on the distinct values of the observations and the counts $n_{1k}(t)$ and $n_{2k}(t)$ of each observation k before and after t. To be concrete, $(R_{1,(a)}(t), R_{2,(a)}(t))$ is obtained as

$$\begin{aligned} R_{1,(a)}(t) &= \sum_{k=1}^{K} \frac{(n_{1k}(t))(n_{1k}(t)-1)}{n_{1k}(t)+n_{2k}(t)} + \sum_{(u,v)\in G_0} \frac{n_{1u}(t)n_{1v}(t)}{(n_{1u}(t)+n_{2u}(t))(n_{1v}(t)+n_{2v}(t))}, \\ R_{2,(a)}(t) &= \sum_{k=1}^{K} \frac{(n_{2k}(t))(n_{2k}(t)-1)}{n_{1k}(t)+n_{2k}(t)} + \sum_{(u,v)\in G_0} \frac{n_{2u}(t)n_{2v}(t)}{(n_{1u}(t)+n_{2u}(t))(n_{1v}(t)+n_{2v}(t))}, \end{aligned} \tag{2}$$

and $(R_{1,(u)}(t), R_{2,(u)}(t))$ is obtained as

$$\begin{aligned} R_{1,(u)}(t) &= \sum_{k=1}^{K} \frac{(n_{1k}(t))(n_{1k}(t)-1)}{2} + \sum_{(u,v)\in G_0} n_{1u}(t)n_{1v}(t), \\ R_{2,(u)}(t) &= \sum_{k=1}^{K} \frac{(n_{2k}(t))(n_{2k}(t)-1)}{2} + \sum_{(u,v)\in G_0} n_{2u}(t)n_{2v}(t). \end{aligned} \tag{3}$$

In our case, the choice between the averaging statistic and the union statistic hardly affects the results. That is, corresponding change-points in both options have less than three minutes of differences from each other in most cases. Thus, we use the averaging statistic without lose of generality for the experiments.

1.3 Implementation Details

We have implemented the role representation algorithm using Python 3.8 on our own, while adopted the R package `gSeg`[1] developed by Chen and Zhang (2015), Chu and Chen (2019), and Song and Chen (2020) for discrete g-segmentation. To perform the whole process at once, we utilize the Python package `rpy2` to run the R script with `gSeg` inside our Python implementation. The process was executed on an Intel Core i7-8550U CPU with 16.0GB of memory. (Note that the M1 chip by Apple does not support binaries for `rpy2`'s API mode in Python, raising a memory error.) Our implementation is publicly accessible in a GitHub repository named `footballcpd`[2] with sample GPS data from an anonymized match.

[1] `https://rdrr.io/cran/gSeg/`

[2] The link is currently unavailable due to the double-blind review policy. Instead, we have attached the code files for the reviewers.

2 CASE STUDY

In this section, we take a closer look into match-by-match formation labels. That is, we showcase visualized examples of both success and failure cases, interpreting them based on domain knowledge. We abbreviate formation/role periods to FPs/RPs and the player with uniform number k as Pk.

2.1 Typical Examples of Formation Changes

Example 2.1.1 (Changing the number of defenders) A primary criterion for classifying team formations is the number of defenders. In modern football, almost every team plays with either three defenders (called "back three" where we regard wingbacks as midfielders) or four defenders (called "back four") at the back of their side.

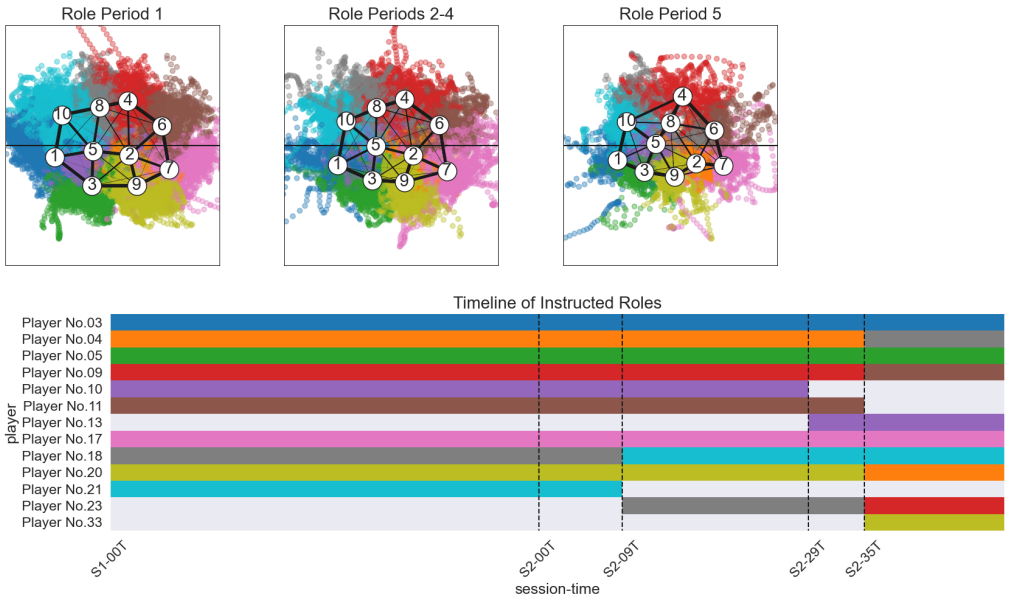


Figure 2: Delaunay graphs and the role change timeline for Match 289.

Fig. 2 and Table 1 illustrate an example of the formation change from back four to back three by substituting the right midfielder P11 with the central defender P33. (Note that there is no human-annotated label for this match since only 28 matches are annotated for model evaluation.)

Table 1: Predicted formation labels for Match 289.

FP	Session	Time	Formation
1	1	0′–46′	4-4-2
2	2	0′–35′	4-4-2
3	2	35′–50′	3-4-3

Example 2.1.2 (Changing the composition of forwards and midfielders) Teams also change the composition of forwards and midfielders during a match. Generally, the number of center forwards (one or two) and the number of defensive midfielders (one or two) are considered significant criteria for distinguishing formations when there are the same number of defenders.

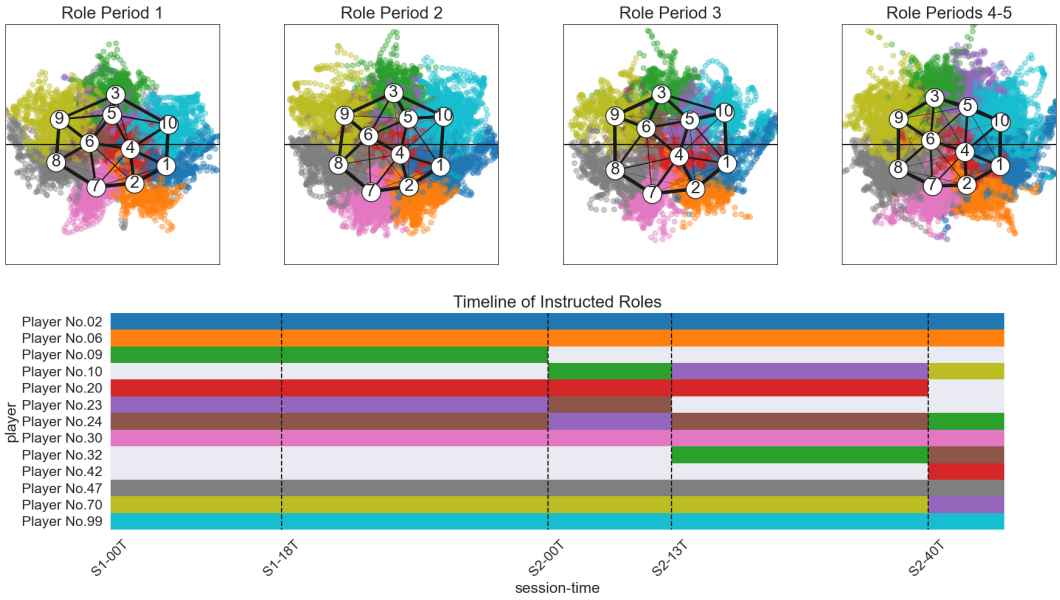


Figure 3: Delaunay graphs and the role change timeline for Match 245.

Fig. 3 and Table 2 show an example where our method correctly detects all the formation changes in both sessions. In this example, the team changes the number of defensive midfielders in the first half (from two to one) and the number of center forwards in the second half (from one to two).

Table 2: Formation labels for Match 245.

(a) Expert-annotated labels

FP	Session	Time	Formation
1	1	0′–15′	4-2-3-1
2	1	15′–46′	4-3-3
3	2	0′–13′	4-3-3
4	2	13′–48′	4-4-2

(b) Predicted labels

FP	Session	Time	Formation
1	1	0′–18′	4-2-3-1
2	1	18′–46′	4-3-3
3	2	0′–13′	4-3-3
4	2	13′–48′	4-4-2

2.2 Typical Examples of Role Changes

Example 2.2.1 (Role change with a player substitution) In many cases, coaches explicitly change the player-role assignment by a player substitution. They instruct substitutes how to play, and the players deliver the instruction after they come inside the pitches. In Fig. 4, the team has made several role changes in the second half (17′, 23′, and 37′) by consecutive player substitutions.

Example 2.2.2 (Role change without a player substitution) Coaches often order role interchanges between players on the pitch without changing the player composition. Players dynamically perform temporary switches throughout a match, but they sometimes intrinsically exchange their dominant roles as from a certain point of time. Therefore, if a set of players maintains the switched positions for enough time (five minutes in our study), we consider this transposition as a consistent and tactically intended change of player-role assignment. Fig. 4 illustrates an example that the two players (P07 and P11) on the pitch exchange their roles in the first half (30′) without a player substitution.

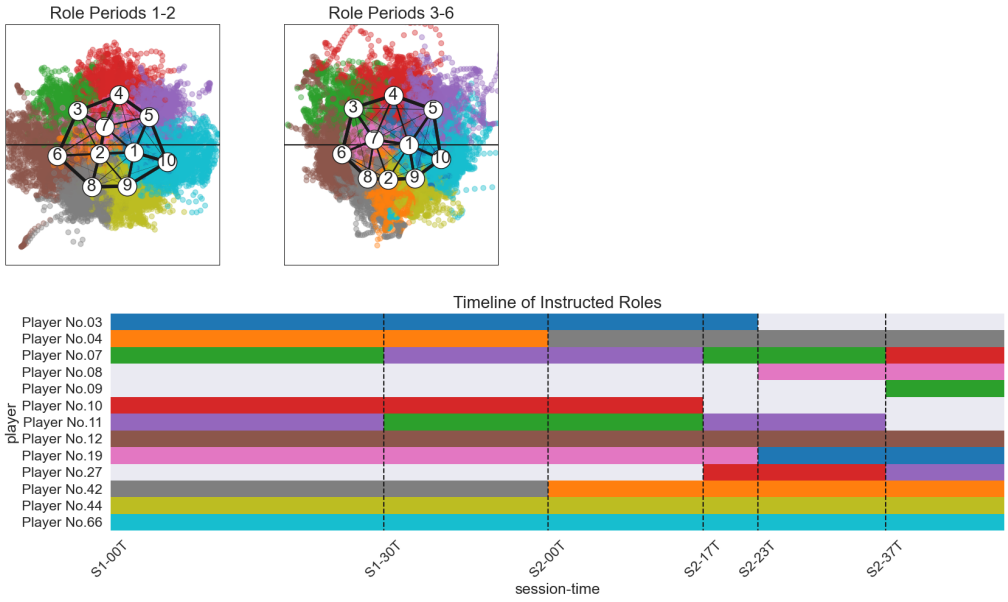


Figure 4: Delaunay graphs and the role change timeline for Match 285.

2.3 Failure Cases

Example 2.3.1 (Detecting unnecessary change-points) Sometimes FootballCPD decides that the detected change-point is significant, even if the formations from both segments eventually get the same label as a result of clustering. This situation often occurs when one of the segments has not enough duration to have a balanced structure as in FP1 and FP4 of Fig. 5. In this example, our method detects significant change-points in both sessions (i.e., 8′ in session 1 and 41′ in session 2) but the actual formations before and after each change-point are the same.

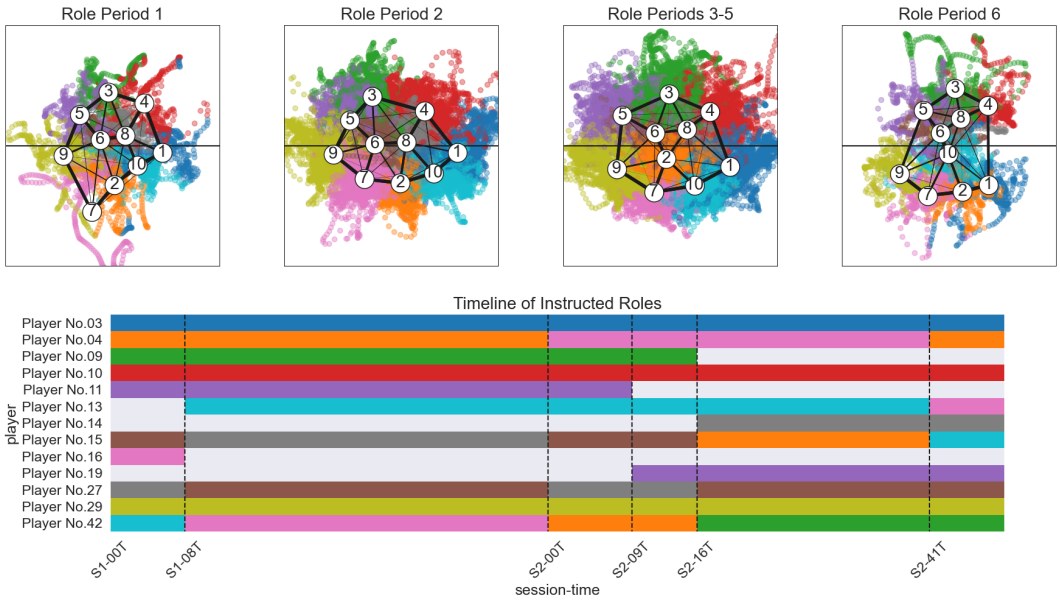


Figure 5: Delaunay graphs and the role change timeline for Match 55.

On the other hand, we also notice from this example that our method assigns the correct formation label to every segment after all (Table 3). Considering that the adjacency relationships are more robust to the formation distortion than the individual role locations, it validates the use of role-adjacency matrices as a representation of formation structure especially in time intervals of varying lengths.

Table 3: Predicted formation labels for Match 55.

FP	Session	Time	Formation
1	1	0′–8′	3-4-3
2	1	8′–47′	3-4-3
3	2	0′–41′	4-4-2
4	2	41′–49′	4-4-2

Example 2.3.2 (Missing a change-point) FootballCPD sometimes fails to detect significant formation changes as shown in Table 4. There is a change from 4-2-3-1 to 3-4-3 in session 2, but it is not detected by our setting. It seems that as a formation lasts for a shorter period, it is more difficult for our method to recognize the change.

Table 4: Formation labels for Match 151.

(a) Expert-annotated labels

FP	Session	Time	Formation
1	1	0′–32′	3-5-2
2	1	32′–46′	4-4-2
3	2	0′–36′	4-2-3-1
4	2	36′–46′	3-4-3

(b) Predicted labels

FP	Session	Time	Formation
1	1	0′–29′	3-5-2
2	1	29′–46′	4-2-3-1
3	2	0′–46′	4-2-3-1

Example 2.3.3 (Confusing between 4-4-2 and 4-2-3-1) FootballCPD sometimes confuses the formations 4-4-2 and 4-2-3-1 because of their similarity. Since the only difference between these two formations is the arrangement of two forward players (horizontal for 4-4-2 and vertical for 4-2-3-1), they have small distance from each other. We have solved this problem up to a point by performing a further clustering only to split 4-4-2 and 4-2-3-1, but there still exist some misclassifications.

Match 151 also corresponds to this kind of mislabeling. As in Table 4 and Fig. 6, our method successfully detects the formation change-point in session 1 from back 3 to back 4 described in Section 2.1. However, it misclassifies the formation in FP2 into 4-2-3-1 even if the two forwards (P09 and P10) are arranged horizontally.

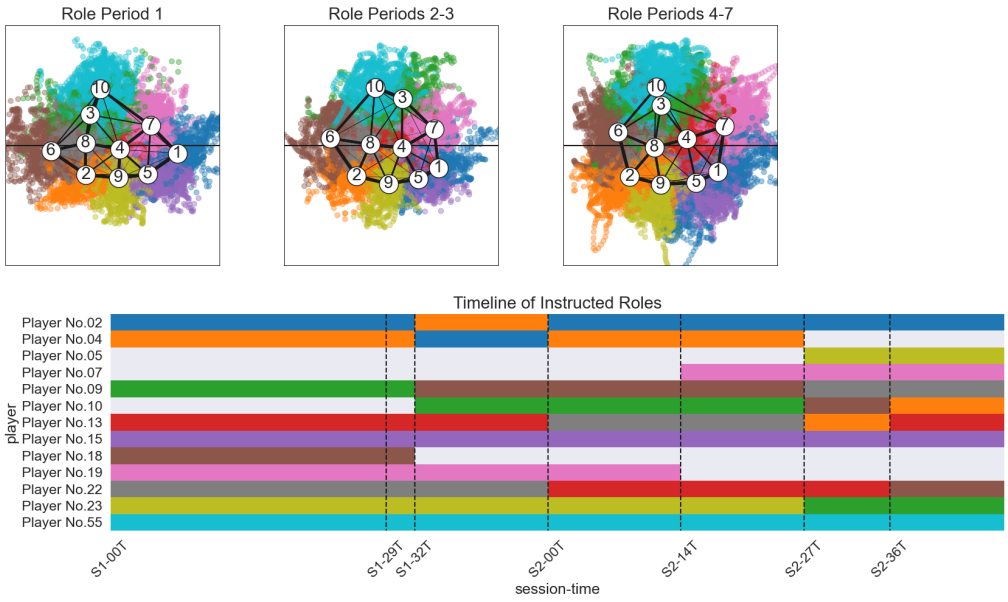


Figure 6: Delaunay graphs and the role change timeline for Match 151.

Example 2.3.4 (Asymmetrical formations) Recently, adventurous football coaches such as Pep Guardiola have began to use atypical strategies by breaking the symmetry of formations. As identifying a team formation based on the knowledge of typical priors, both domain experts and FootballCPD find it difficult to classify these asymmetrical formations into one of the existing formation categories.

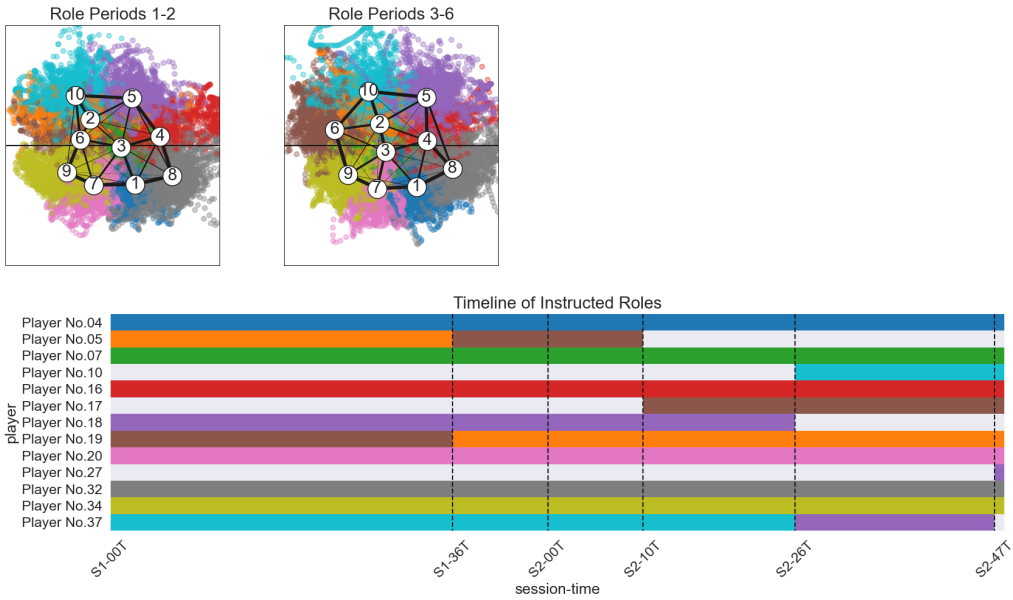


Figure 7: Delaunay graphs and the role change timeline for Match 385.

Fig. 7 illustrates an example of a team using atypical formations. The team's official formation was 3-5-2 with P19 assigned to the left midfielder (i.e., wingback) according to the broadcast. However, we have figured out from the match video that both of P05 and P19 actually had played as left-skewed central midfielders with frequent mutual switches. They also used another asymmetrical formation in the second half by placing a winger (R6) only on the left side. (Note that P16 assigned to R4 played as a central midfielder.) This skewness make both the expert and our method assign the formation into the "others" label (session 1) or leads to a mismatch between experts' annotations and the predicted labels (session 2).

Table 5: Formation labels for Match 385.

(a) Expert-annotated labels

FP	Session	Time	Formation
1	1	0′–46′	others
2	2	0′–48′	4-4-2

(b) Predicted labels

FP	Session	Time	Formation
1	1	0′–46′	others
2	2	0′–48′	3-5-2

Example 2.3.5 (Incomplete player tracking) The last case is the situation that the coach forgets to attach a GPS tracker on the substitute, which results in the decrease of the number of measured players per frame. Since this causes incomplete player-role assignment in the role representation step, the resulting Delaunay graphs become less reliable as a representation of team formation. Moreover, if this partial absence of tracking data occurs from the beginning of the session, our method cannot compute 10×10 role-adjacency matrix for any frame and thus fails to estimate formations in the session.

In Fig 8, the player substituted with P30 was not measured by the GPS tracker. As a result, since the number of player coordinates obtained at each time is at most nine for the second half, our method cannot be performed on the session. Likewise, 105 sessions among 914 in total have been excluded because of the insufficient numbers of measured players.

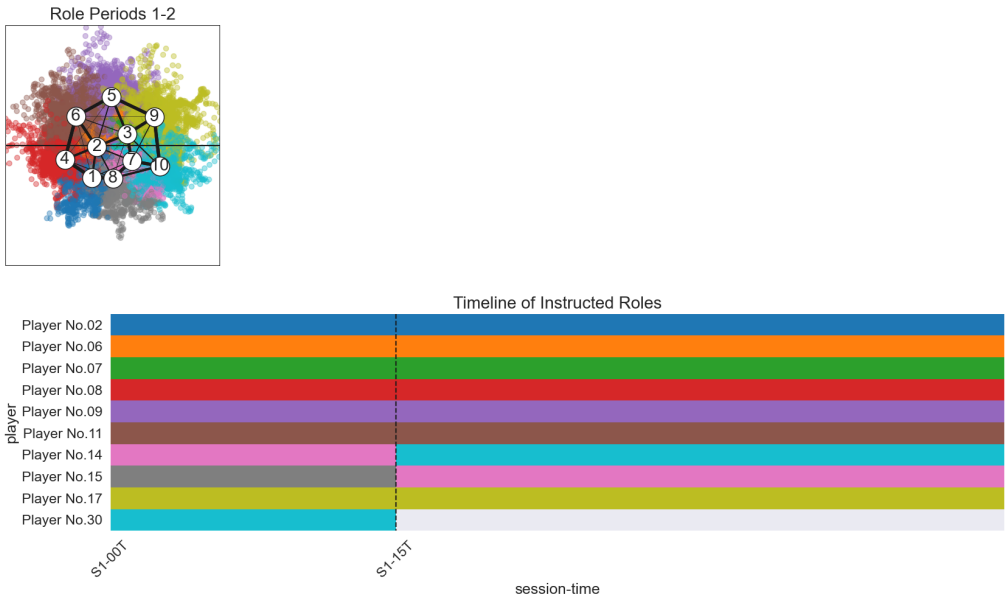


Figure 8: Delaunay graphs and the role change timeline for Match 339.

References

Bialkowski, A., Lucey, P., Carr, P., Yue, Y., Sridharan, S., and Matthews, I. (2014). Large-scale analysis of soccer matches using spatiotemporal tracking data. In *IEEE International Conference on Data Mining*.

Chen, H. and Zhang, N. (2015). Graph-based change-point detection. *Annals of Statistics*, 43(1):139–176.

Chu, L. and Chen, H. (2019). Asymptotic distribution-free change-point detection for multivariate and non-euclidean data. *Annals of Statistics*, 47(1):382–414.

Song, H. and Chen, H. (2020). Asymptotic distribution-free change-point detection for data with repeated observations.

Zhang, J. and Chen, H. (2017). Graph-Based Two-Sample Tests for Data with Repeated Observations.