SoccerCPD: Formation and Role Change-Point Detection in Soccer Matches Using Spatiotemporal Tracking Data

Hyunsung Kim
hyunsung.kim@fitogether.com
Fitogether Inc.
Seoul, South Korea

Bit Kim
bit.kim@fitogether.com
Fitogether Inc.
Seoul, South Korea

Dongwook Chung
dongwook.chung@fitogether.com
Fitogether Inc.
Seoul, South Korea

Jinsung Yoon
jinsung.yoon@fitogether.com
Fitogether Inc.
Seoul, South Korea

Sang-Ki Ko
sangkiko@kangwon.ac.kr
Kangwon National University
Chuncheon, South Korea

ABSTRACT

In fluid team sports such as soccer and basketball, analyzing team formation is one of the most intuitive ways to understand tactics from the domain participants' point of view. However, existing approaches either assume that team formation is consistent throughout a match or assign formations frame-by-frame, which disagree with real situations. To tackle this issue, we propose a change-point detection framework named SoccerCPD that distinguishes tactically intended formation and role changes from temporary changes in soccer matches. We first assign roles to players frame-by-frame and perform two-step change-point detections: (1) formation change-point detection (FormCPD) based on the sequence of role-adjacency matrices and (2) role change-point detection (RoleCPD) based on the sequence of role permutations. The evaluation of the proposed method using the ground truth data annotated by domain experts demonstrates that our method accurately detects the points of tactical changes and estimates the formation and player-role assignment in each segment. Lastly, we introduce practical use-cases that domain participants can easily interpret and utilize.

CCS CONCEPTS

• **Information systems → Data mining**; *Spatial-temporal systems*; • **Mathematics of computing → Time series analysis**.

KEYWORDS

sports analytics; spatio-temporal data analysis; formation analysis; GPS tracking data; change-point detection

1 INTRODUCTION

In fluid team sports such as soccer and basketball, players interact with each other while maintaining a role in a certain formation [24]. Since team formations highly affect the players' movement patterns, analyzing them is one of the most intuitive ways to understand tactics from the domain participants' point of view.

However, even with the recent acquisition of a vast amount of sports data, there remain several technical challenges for formation and role estimation as follows:

- Coaches initially assign a unique role to each player, but they can change their instruction throughout the match.
- Players temporarily switch roles with their teammates.
- Abnormal situations such as set-pieces sometimes occur, in which all the players ignore the team formation.

Several studies [6, 19, 20, 26] have tried to estimate the team formation during a match using spatiotemporal tracking data, but they did not overcome all the above challenges. Bialkowski et al. [6] successfully captured the temporary role switches between players, but they have assumed that the team formation is consistent throughout half of a match. Narizuka et al. [19, 20] found formations for each instant, resulting in too frequent formation changes that are not intended by coaches. Shaw et al. [26] estimated a team formation per two-minute window, but their bottom-up approach has a trade-off between the reliability of detected change-points and the robustness against the abnormal situations. That is, since the endpoints of windows are the only candidate change-points, the discrepancy between predicted and actual change-points increases as the window size increases. On the other hand, a smaller window size becomes more sensitive to the situations such as set-pieces.

To fill this gap, we propose a change-point detection framework named *SoccerCPD* that distinguishes tactically intended formation and role changes from temporary switches in soccer matches. First, we assign roles to players per frame by adopting the role representation proposed by Bialkowski et al. [6]. Next, we perform two-step change-point detections: (1) formation change-point detection (FormCPD) in each half of a match and (2) role change-point detection (RoleCPD) in each segment resulting from FormCPD.

In FormCPD, we formulate the players' spatial configuration—which we call the *role topology* in this paper—at each frame as a role-adjacency matrix calculated from the Delaunay triangulation [19]. Then we deploy a nonparametric change-point detection algorithm named discrete g-segmentation [28] to find change-points in this sequence of matrices. The detected change-points split a match into several *formation periods* during which the team formation is assumed to be consistent. We represent the team's formation in each segment as the mean role-adjacency matrix and translate it into the language of domain participants (such as "3-4-3" or "4-4-2") by clustering all the obtained matrices.

In RoleCPD, we formulate the players' transposition by a sequence of role permutations in each formation period. As in the previous step, applying discrete g-segmentation to the permutation sequence with pairwise Hamming distances gives the role change-points. They split each formation period into several *role periods* where the player-role assignment remains constant. Lastly, we give domain-specific “position” labels (such as “left wing-back” or “central defensive midfielder”) to the roles so that each player takes a single position per role period.

With these results, we make the following four contributions:

- applying a state-of-the-art change-point detection method to the sports tracking data, which discovers the underlying tactics and their changes in an unsupervised manner.
- effectively representing the spatial configuration and role switches of players in a time interval as a mean role-adjacency matrix and a sequence of permutations, respectively.
- suggesting practical applications such as switching pattern mining and automatic set-piece detection, which can be easily interpreted and utilized by domain participants.
- providing a Python implementation of our framework with sample GPS data[1].

2 RELATED WORKS

2.1 Change-Point Detection

Change-point detection (CPD) is the problem of finding points when the governing parameters of the process change. CPD methods are classified into parametric and nonparametric ones depending on whether they specifically estimate the parameters or directly find change-points [30].

Parametric CPD models assume a certain form of distribution for given observations and find change-points of the parameters. They include maximum likelihood estimations [13, 16, 21], piecewise linear models [2, 3, 23], Mahalanobis metric-based models [15], and so on. However, they have limitations in that they are sensitive to the violation of the assumptions of distributions [1].

Nonparametric CPD methods directly detect breakpoints without estimating distribution parameters. In general, they are based on nonparametric statistics such as rank statistics [17], graph-based scan statistics [7, 8, 28], and kernel estimation [10, 11, 27]. Free from the above assumptions of distributions, they have shown greater success for various datasets [1]. Particularly, graph-based methods [7, 8, 28] are considered more flexible in real applications since they only use similarity information and thus are applicable to high-dimensional data or even non-Euclidean data.

2.2 Formation Estimation in Team Sports

Several studies have analyzed spatiotemporal tracking data to estimate team formations and find the roles of players in team sports. Bialkowski et al. [6] proposed the “role representation” that dynamically assigns a unique role to each player frame-by-frame. Narizuka et al. [19] expressed a temporary formation as an undirected graph from Delaunay triangulation [9] and clustered the graphs obtained from a match to characterize the formation. Later, Narizuka et al. [20] combined their method with the role representation to cluster formations over multiple matches. Also, Shaw et al. [26] used relative positions between players to estimate team formation per two-minute window.

Moreover, there have been other approaches relying on the annotated or estimated formation for deeper analyses. Bialkowski et al. [4] and Tamura et al. [29] studied the correlation between team formation and other context information such as the existence of home advantage or the result of the previous match. Machado et al. [18] and Wu et al. [32] suggested visual interfaces showing the dynamic changes of the team formation. Especially, many studies have utilized the result of role representation above for various purposes such as highlight detection [31], similar play retrieval [25], team style identification [5], and player style identification [12].

3 PROBLEM DEFINITION

Soccer coaches change the team formation up to several times during a match, or they command role switches between players without formation change. Thus, our method is divided into two steps to track their decisions: first finds formation change-points and then finds role change-points. In this section, we formalize each step as a separate CPD problem.

3.1 Formation Change-Point Detection

One can consider team formation as a probability distribution over several role topologies. Given the sequence of observations in the form of players’ (x, y) locations in 10 Hz, the goal of FormCPD is to find change-points of this distribution. Thus, we first need to find effective features representing role topology from raw observations and then perform CPD using the sequence of the features.

Formally speaking, given a sequence of features $\{A(t)\}_{t\in T}$ indicating the role topology, we aim to find a partition $T_1 < \cdots < T_m$ of T such that $t \in T_i$ implies $A(t) \sim \mathcal{F}_i$ for some distinct distributions $\mathcal{F}_1, \ldots, \mathcal{F}_m$. Here we call each interval T_i a *formation period* during which the team maintain a certain formation represented as the distribution $\mathcal{F}_i$.

3.2 Role Change-Point Detection

Bialkowski et al. [6] proposed the role representation that gives the frame-by-frame role assignment to the outfield players, but they did not distinguish long-term tactical changes from these temporary role swaps. Given that a player generally possesses a constant role instructed by the coach for a certain period, the goal of RoleCPD is to find change-points where long-term tactical changes occur. That is, we further partition a formation period T_i into several time intervals $T_{i,1} < \cdots < T_{i,n_i}$ named *role periods* where each player possesses a constant role during a single role period. Then, we can deem the frame-by-frame role interchanges (the result of the role representation) inside role periods as temporary swaps.

Formally speaking, for the set P of players in a match time T, the role representation finds a set $\mathcal{X} = \{X_1, \ldots, X_N\}$ of roles and *player-to-temporary-role* (P-TR) maps $\{\beta_t : P \to \mathcal{X}\}_{t\in T}$ such that

$$\beta_t(p) \neq \beta_t(q) \quad \forall p \neq q \in P,\ t \in T \tag{1}$$

Here we aim to express the given P-TR maps $\{\beta_t\}_{t\in T}$ as the composition of *player-to-instructed-role* (P-IR) maps $\{\alpha_t : P \to \mathcal{X}\}_{t\in T}$ and *temporary role permutations* (RolePerm) $\{\sigma_t : \mathcal{X} \to \mathcal{X}\}_{t\in T}$ with $\sigma_t \in \mathrm{S}(\mathcal{X})$ (symmetric group on $\mathcal{X}$), i.e., $\beta_t = \sigma_t \circ \alpha_t$. In other words, given P-TR maps $\{\beta_t\}_{t\in T}$ in T_i, we find a partition $T_{i,1} < \cdots < T_{i,n_i}$ of T_i and P-IR maps $\{\alpha_t\}_{t\in T}$ satisfying the followings:

- (*Period-wise consistency*) The instructed role of every player is constant during each role period $T_{i,j}$. i.e.,

$$\alpha_t(p) = X_p^{(i,j)} \quad \forall t \in T_{i,j} \tag{2}$$

[1] https://github.com/pientist/soccercpd

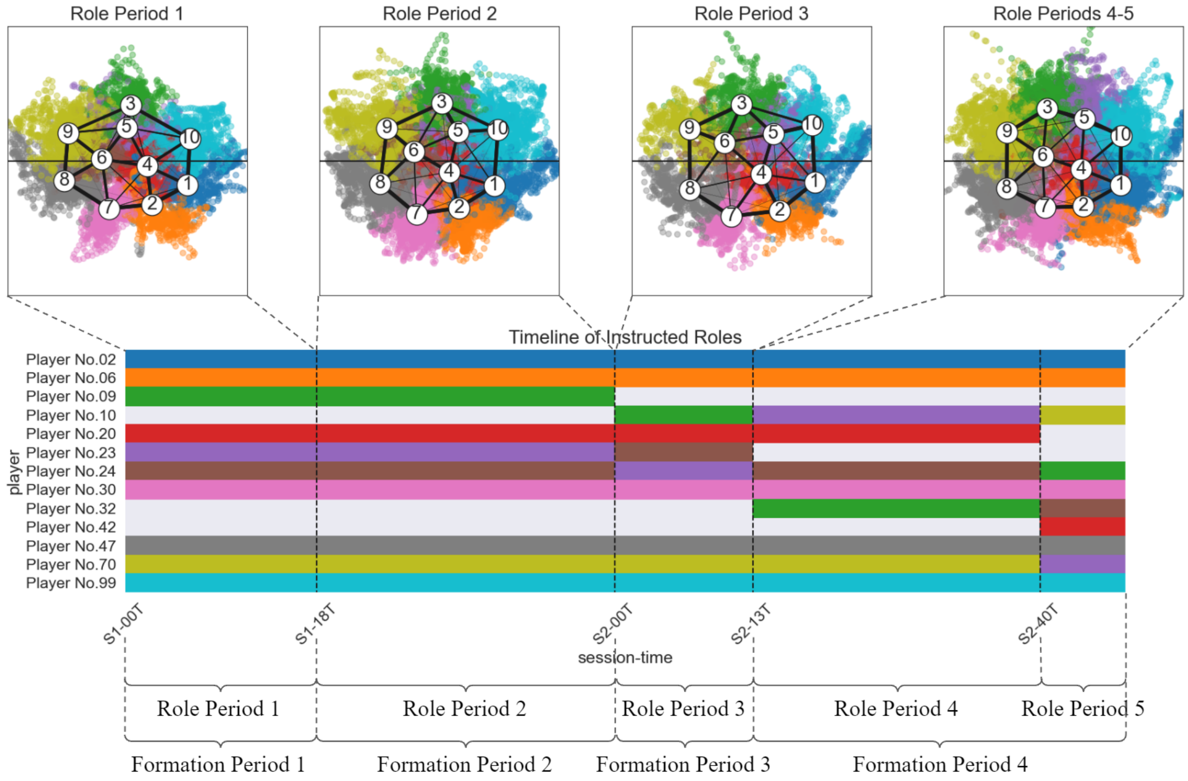


Figure 1: An example of formation and role change timeline as a result of SoccerCPD.

for some $X_p^{(i,j)} \in \mathcal{X}$.

- (*Uniqueness*) No two players are instructed to take the same role in a role period. i.e.,

$$X_p^{(i,j)} \neq X_q^{(i,j)} \quad \forall p \neq q \in P \tag{3}$$

- (*Existence of a role change*) A change of role period implies a change of instruction. i.e., for each $j \in \{1, \ldots, n_i - 1\}$, there exists $p \in P$ such that

$$X_p^{(i,j)} \neq X_p^{(i,j+1)} \tag{4}$$

Note that the last condition is valid only for the change *inside* a single formation period. Namely, adjoining role periods from distinct formation periods can have the same player-role assignment.

4 FORMATION CHANGE-POINT DETECTION

At first, we express the role topology at each frame as a role-adjacency matrix calculated from the Delaunay triangulation (Section 4.1). Then, we detect formation change-points by applying discrete g-segmentation on the sequence of role-adjacency matrices (Section 4.2). Finally, we represent the formation in each segment as the mean role-adjacency matrix and cluster the matrices from the entire dataset to give domain-specific labels such as “3-4-3” or “4-4-2” to individual formation periods (Section 4.3). Please refer to the block diagram in Appendix A.1 for better understanding of the whole pipeline.

4.1 Calculating the Sequence of Role-Adjacency Matrices

First, we assign roles to the players per frame by the role representation proposed by Bialkowski et al. [6]. It groups the player locations recorded throughout a session into role clusters, under the constraint that no two players take the same role at the same time. This is achieved by the EM algorithm below:

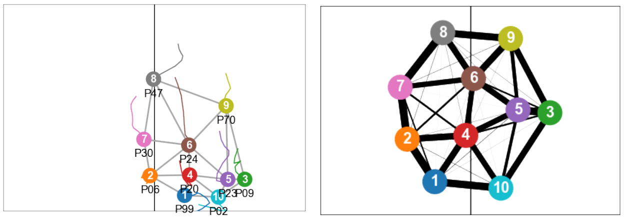
- Initialization: Assign a unique role label from 1 to N to each outfield player and estimate the density of each role.
- E-step: Apply the Hungarian algorithm [14] per frame to reassign roles to players using the cost matrix based on the log probability of a role generating a player location.
- M-step: Update the density parameters based on the role labels reassigned in E-step.

Next, we perform Delaunay triangulation [9] to get the adjacency matrices $\{A(t)\}_{t \in T} \subset \mathbb{R}^{N \times N}$ between the role labels. That is, the components $a_{kl}(t)$ are defined as

$$a_{kl}(t) = \begin{cases} 1 & \text{if the } \textit{roles } X_k \text{ and } X_l \text{ are adjacent at } t, \\ 0 & \text{otherwise} \end{cases}$$

for the set $\mathcal{X} = \{X_1, \ldots, X_N\}$ of roles. Fig. 2a shows an example of Delaunay graph on role locations.

The idea of applying Delaunay triangulation to player tracking data was first proposed by Narizuka et al. [19]. However, they used uniform numbers as indices of the adjacency matrices and applied the role representation only for aligning uniform numbers from multiple matches to a single set of indices [20]. While this method can discover position-exchange patterns well, the resulting player-adjacency matrices are highly affected by temporary switches or irregular situations such as set-pieces which are irrelevant to the original team formation. Hence, we use role labels instead of player labels as the indices of adjacency matrices, which are a more robust representation of the team's spatial configuration.



(a) (b)

Figure 2: (a) A temporary Delaunay graph with player trajectories at a certain frame and (b) the weighted graph drawn from the mean role locations and the mean role-adjacency matrix for a formation period. The number inside each circle indicates a role label from 1 to 10, while the number below each circle indicates the player's uniform number.

4.2 Applying Discrete g-Segmentation to the Sequence of Matrices

In order to find formation change-points, we separately apply CPD to the sequence of role-adjacency matrices obtained from each half of a match.

Since our sequences are high-dimensional ($N \times N$ matrices for FormCPD with $N = 10$ in general) or even non-Euclidean (permutations for RoleCPD in Section 5), we focus on graph-based CPD [7, 8, 28]. They draw a similarity graph such as the minimum spanning tree on the observations and calculate a scan statistic $R(t)$ for each time t by checking how many edges connect observations before and after t. Since a large difference between the observations before and after t leads to large $R(t)$, one can regard $\tau = \arg\max_t R(t)$ as a possible change-point if $R(\tau)$ exceeds a certain threshold after normalization. As they only use similarity information between observations, they are applicable to our sequences with proper distance functions.

Another issue is that our sequences have repeated observations (binary matrices for FormCPD here and permutations for RoleCPD in Section 5), which makes the optimal similarity graph not uniquely defined. Among the series of graph-based CPD methods, discrete g-segmentation [28] can only cover this case by using the average statistic $R_{(a)}(t)$ or the union statistic $R_{(u)}(t)$ introduced by Zhang and Chen [33] as alternatives for $R(t)$. That is, they obtain a unique scan statistic either by "averaging" the scan statistics over all optimal graphs or by calculating the scan statistic on the "union" of all optimal graphs. Therefore, we employ the discrete g-segmentation to find change-points in our discrete high-dimensional/non-Euclidean sequences. (Please refer to Appendix B that empirically justifies the choice of CPD method.)

To elaborate, we use the $L_{1,1}$ matrix norm

$$d_M(A(t), A(t')) = \|A(t) - A(t')\|_{1,1} = \sum_{k=1}^{N} \sum_{l=1}^{N} |A_{kl}(t) - A_{kl}(t')| \tag{5}$$

(which we call *Manhattan distance* hereafter) as the distance measure between role-adjacency matrices. Also, we compute the *switch rate* at t by counting the number of players whose temporary roles differ from their most frequent roles in the session. Then, we exclude frames with high switch rates (> 0.7) to ignore abnormal situations such as set-pieces. Consequently, discrete g-segmentation returns an estimated change-point among the sequence of the remaining *valid* adjacency matrices using the pairwise Manhattan distances.

After finding a change-point during the given period, the algorithm decides its significance based on three conditions. To be specific, we recognize the estimated change-point τ as significant only if (1) the p-value of the scan statistic is less than 0.01, (2) both of the segments last for at least 5 minutes, and (3) the mean role-adjacency matrices calculated from the respective segments are far enough (i.e., have large Manhattan distance) from each other. The threshold duration for (2) comes from the fact that abnormal periods such as injury breaks or VAR checks generally last for 2–3 minutes, and therefore 5 minutes are robust against them. Also, the threshold distance for (3) is empirically set to 7.0.

Since there can be more than one formation change-point during a session, we construct a recursive framework to find multiple change-points. First, if there is a significant change-point in the given period, we respectively apply the CPD algorithm to the sequences before and after the change-point again. Each branch-CPD terminates when there is no significant change-point in the segment of interest. In turn, the given session T is partitioned into several formation periods $T_1 < \cdots < T_m$.

4.3 Formation Clustering Based on Mean Role-Adjacency Matrices

We now represent the formation in a formation period T_i as a weighted graph $F(T_i) = (V(T_i), A(T_i))$ with the *mean role locations*

$$V(T_i) = \frac{1}{|T_i^*|} \sum_{t \in T_i^*} V(t) \tag{6}$$

as vertices and the *mean role-adjacency matrix*

$$A(T_i) = \frac{1}{|T_i^*|} \sum_{t \in T_i^*} A(t) \tag{7}$$

as edge matrix, where T_i^* denotes the set of valid time stamps in T_i with small switch rates and $V(t) = (v_1(t), \ldots, v_N(t))^T \in \mathbb{R}^{N \times 2}$ is the 2D locations of N roles normalized to have zero mean at each frame t (i.e., $\sum_{k=1}^{N} v_k(t) = (0, 0)$). See Fig. 2b as an example.

Unlike Section 4.2, we first need to align the roles when calculating the distance between a pair of formation graphs since each role label from 1 to N has nothing to do with the same label in another graph (especially that from another match). Thus, for a pair of formation graphs, we find the "optimal" permutation of the role labels of one graph relative to the other and calculate the Manhattan distance (Eq. 5) between the adjacency matrices.

To be specific, let $F = (V, A)$ and $F' = (V', A')$ be graphs from a pair of formation periods. Then, we rearrange the rows and columns of A based on the Hungarian algorithm [14] using the pairwise Euclidean distances between V and V'. That is, we find the optimal permutation matrix Q minimizing the assignment cost

$$c(V, V'; Q) = \sum_{k=1}^{N} \|(QV)_k - v'_k\|_2$$

and use QAQ^T and A' when calculating the distance between the formations F and F', i.e.,

$$d(F, F') = d_M(QAQ^T, A') \tag{8}$$

Along with this distance metric, we cluster the formation obtained from the entire dataset as in Bialkowski et al. [6] to decide which formations are the same or different. For this work, we use the GPS data collected from the two seasons (2019 and 2020) of K League 1 and 2, the first two divisions of the South Korean professional soccer league. The data from 809 sessions (i.e., match halves) with at least one moment when ten outfield players were simultaneously measured are split into 864 formation periods and 2,152 role periods. The two-step agglomerative clustering task applied on the pairwise formation distances divides the 864 formations into six main formation groups and "others" group consisting of several clusters with less than 15 observations. (Appendix A.2 describes the detailed process.) Lastly, we give each of the six groups a label commonly used by the domain participants, as in Fig. 3.

The difference of our clustering task from that in Bialkowski et al. [6] is the use of the distance between mean role-adjacency matrices instead of the earth mover's distance (EMD) between role locations. While formations in the previous work are from halves of matches with almost the same duration, ours are from formation periods with shorter and varying durations causing distortion of formation graphs. Using mean role-adjacency matrices is more robust to this distortion since they are affected by the adjacency relationships rather than absolute locations.

5 ROLE CHANGE-POINT DETECTION

The process of RoleCPD is analogous to those of FormCPD, except that it leverages role permutations (calculated in Section 5.1) instead of role-adjacency matrices. Again, discrete g-segmentation recursively finds role change-points in each formation period (Section 5.2). Lastly, we give domain-specific position labels to individual players per role period (Section 5.3).

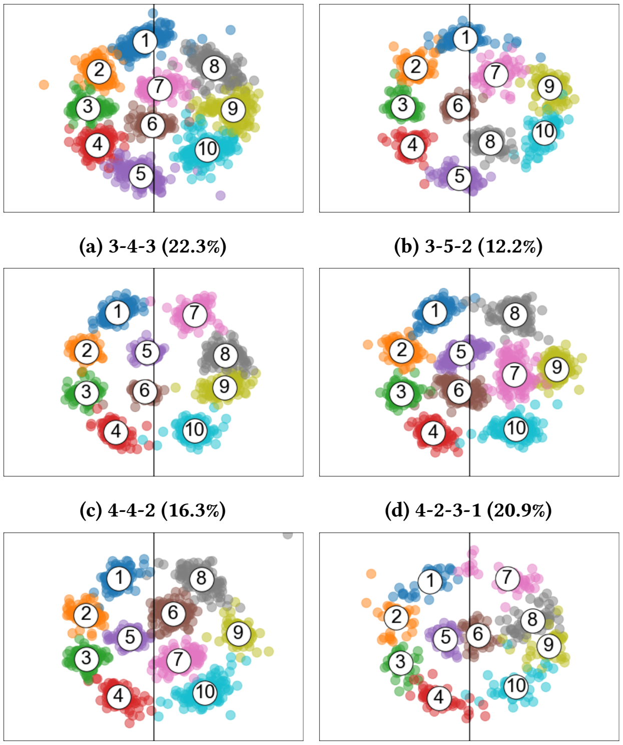
5.1 Calculating the Sequence of Role Permutations

During the initialization step of role representation in Section 4.1, each player p is assigned to an *initial role* X_p according to a canonical order such as uniform number. Then, one can express the temporary role assignment to the players $p_1, \ldots, p_N$ at time t as a permutation $\pi_t = (\pi_t(X_{p_1}) \cdots \pi_t(X_{p_N})) \in S(\mathcal{X})$ of the initial role assignment $(X_{p_1} \cdots X_{p_N})$, i.e.,

$$\beta_t(p) = \pi_t(X_p). \tag{9}$$

Table 1: Assignment rules of position labels to roles.

Role	3-4-3	3-5-2	4-4-2	4-2-3-1	4-3-3	4-1-3-2
1	LWB	LWB	LB	LB	LB	LB
2	LCB	LCB	LCB	LCB	LCB	LCB
3	CB	CB	RCB	RCB	RCB	RCB
4	RCB	RCB	RB	RB	RB	RB
5	RWB	RWB	LCM	LDM	CDM	CDM
6	RCM	CDM	RCM	RDM	LCM	CAM
7	LCM	LCM	LM	CAM	RCM	LM
8	LM	RCM	LCF	LM	LM	LCF
9	CF	LCF	RCF	CF	CF	RCF
10	RM	RCF	RM	RM	RM	RM



(a) 3-4-3 (22.3%) **(b) 3-5-2 (12.2%)**

(c) 4-4-2 (16.3%) **(d) 4-2-3-1 (20.9%)**

(e) 4-3-3 (17.4%) **(f) 4-1-3-2 (5.8%)**

Figure 3: Mean role locations of each formation group with the proportion (%) in terms of playing time.

5.2 Applying Discrete g-Segmentation to the Sequence of Permutations

Similar to Section 4.2, we apply the discrete g-segmentation to the sequence of role permutations $\{\pi_t\}_{t \in T_i}$ obtained from each formation period T_i. Here the algorithm finds an estimated change-point among the sequence of valid permutations (i.e., with switch rates ≤ 0.7) using the Hamming distance

$$d_H(\pi_t, \pi_{t'}) = |\{X : \pi_t(X) \neq \pi_{t'}(X),\ X \in \mathcal{X}\}| \tag{10}$$

as the distance measure between role permutations.

Testing the significance of the change-point also emulates that in FormCPD except for the last condition. Since the goal of RoleCPD is to find the point when the dominant role assignment changes, we recognize the change-point τ as significant only if the most frequent permutations differ between before and after τ.

Finally, the recursive CPD applied to the sequence from each formation period T_i results in a partition $T_{i,1} < \cdots < T_{i,n_i}$ of T_i. We set the instructed role per player in each role period $T_{i,j}$ be the most frequent permutation, and express every temporary roles resulting from the role representation as a permutation of the instructed roles. Formally speaking, we define the P-IR maps $\{\alpha_t : P \to \mathcal{X}\}_{t \in T_{i,j}}$ as

$$\alpha_t(p) = \pi_{(i,j)}(X_p) \quad \text{(constant along } t \in T_{i,j}) \qquad (11)$$

where $\pi_{(i,j)} \in S(\mathcal{X})$ is the most frequent permutation among $\{\pi_t\}_{t \in T_{i,j}}$. The RolePerm $\sigma_t(X_p) = \beta_t \circ \alpha_t^{-1}$ at $t \in T_{i,j}$ can then be obtained as

$$\sigma_t(X_p) = \beta_t(\alpha_t^{-1}(X_p)) = \pi_t(\pi_{(i,j)}^{-1}(X_p)) \qquad (12)$$

from Eq. 9 and Eq. 11.

The resulting P-IR maps satisfy the three conditions presented in Section 3.2. They satisfy the period-wise consistency and uniqueness by Eq. 11 (i.e., $\alpha_t(p)$ is constant along $t \in T_{i,j}$ and distinct across $p \in P$ since it is a permutation of distinct initial roles). Also, the aforementioned significance test guarantees the existence of a role change between adjacent role periods.

5.3 Assigning Domain-Specific Position Labels to Instructed Roles of Players

Since roles are independently initialized based on the players' uniform numbers per session in Section 5.1, each role label from 1 to 10 in a formation graph has nothing to do with the same label in another graph. Thus, we first align role labels between formation graphs so that the locations of the same label from different graphs are close to one another. Then, we give one of the 18 position labels such as "left wing-back (LWB)" or "central defensive midfielder (CDM)" to each of the aligned roles 1–10 per formation group based on domain knowledge.

The role alignment is achieved by clustering the roles in each formation group based on the EM algorithm similar to the role representation [6] as follows:

- Initialization: Partition the role locations in each formation group into ten clusters by agglomerative clustering.
- E-step: Apply the Hungarian algorithm [14] between the ten role locations and the cluster centroids (mean locations) per formation graph to reassign cluster labels to roles.
- M-step: Update the centroid per cluster.

As a result, roles are relabeled as the annotations in Fig. 3 so that the same label in the same formation graph corresponds to a certain "position" in the pitch.

As the last step, we give ten position labels to the instructed roles 1–10 per formation group by the rules in Table 1. This results in each player having one position label among the total of 18 in Fig. 4 per role period. For the "others" formation group, we assign position labels separately for each formation graph based on the Hungarian algorithm [14] between the ten role locations and the mean locations of the 18 position labels obtained from the above. Fig. 5 shows the resultant position-labeled graphs as examples.

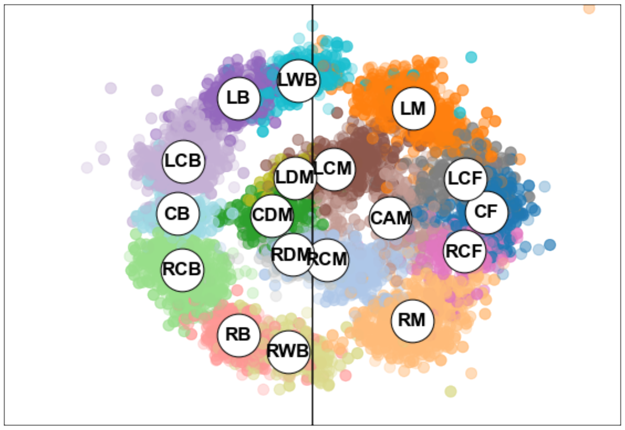
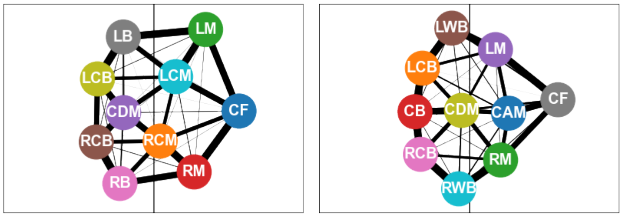


Figure 4: Role locations in the ordinary formation groups colored by position label.



(a) Formation "4-3-3" **(b) Formation "others"**

Figure 5: Examples of position-labeled formation graph.

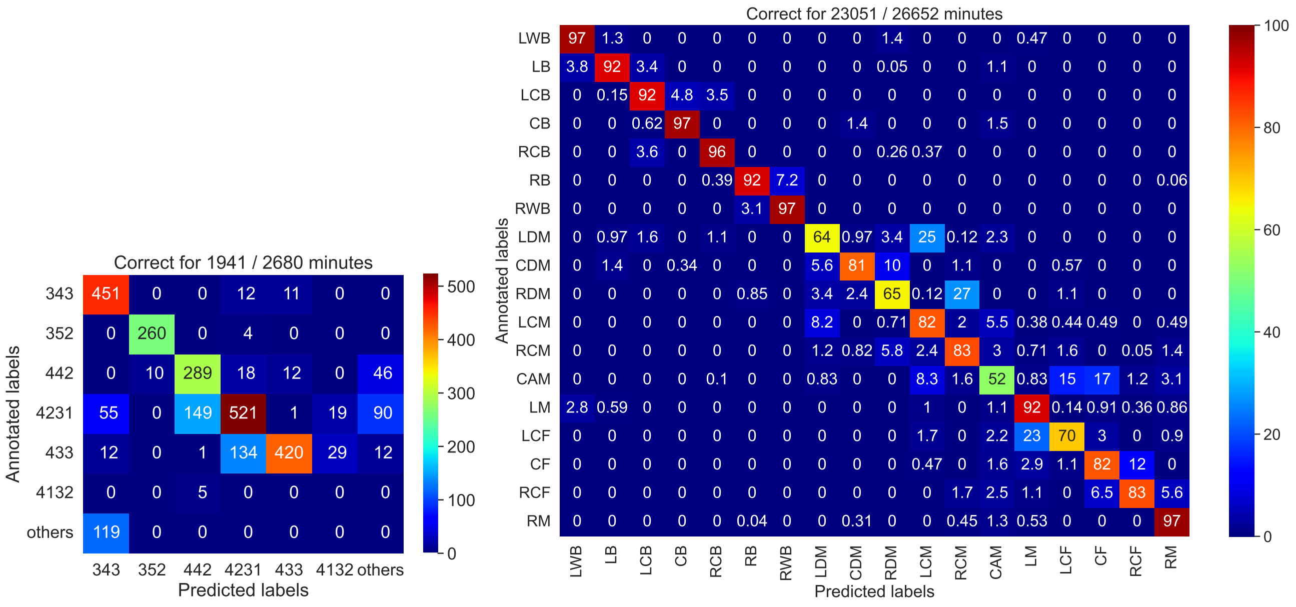
6 EXPERIMENTS

In this section, we first describe the evaluation process of SoccerCPD by comparing our results with annotations by domain experts (Section 6.1). Next, we introduce switching pattern discovery (Section 6.2) and set-piece detection (Section 6.3), which are useful and interpretable applications of our model.

6.1 Model Evaluation

We evaluated the performance of SoccerCPD by measuring the prediction accuracies of (1) team formation and (2) player position. Namely, we calculated the ratios of correctly detected one-minute segments to the total number of segments (total minutes played).

More specifically, we used the ground truth labels annotated by domain experts who worked as coaches or video analysts in elite soccer teams. We divided the whole playing time into one-minute segments and compared the predicted and annotated formation/position labels per segment. (For instance, there are 90 formation annotations and 900 position annotations for a single team's 90-minute match without a red card or missing measurement.) As the minute-wise annotation of formation and role is very costly, we sampled 28 matches, one for each of 28 season-team pairs



(a) Formation accuracy in minutes

(b) Position accuracy in percentage

Figure 6: Confusion matrices for formation and position predictions. Note that we calculate the matrix in minutes and in percentage for Fig 6a and Fig 6b, respectively, since the former has high class imbalance while the latter does not.

(i.e., 15 teams in season 2019 and 13 in 2020), and let the experts perform the annotation only for those matches.

Fig. 6 demonstrates the detailed results in confusion matrices. For the formation prediction, our result accords with the annotation for 1,941 minutes (72.4%) among 2,680 minutes of the total playing time. By analyzing the failure cases, we have found that our method can make a misprediction in the following patterns:

- It sometimes confuses the formations 4-4-2 and 4-2-3-1 because of their similarity. (See the (4,3)-element of the confusion matrix in Fig. 6a.) Particularly, CAM and CF sometimes form a horizontal line when the team is defending, making the prediction more difficult.
- It cannot distinguish 3-4-1-2 (included in the formation "others" in our experiment) from 3-4-3, which leads to the (6,1)-element in Fig. 6a. 3-4-1-2 deploys LCF-CAM-RCF in their forward line, but their arrangement is too similar to the LM-CF-RM line to be separated from 3-4-3.

For the position prediction, since labels are compared player-by-player in each segment, the accuracy is calculated for the 26,652 minutes of player-segment pairs in total (i.e., about ten times the total playing time). Overall, the predicted labels accord with the annotated ones for 23,051 minutes (86.5%). Most positions achieve recalls higher than 80%, except for LDM, RDM, CAM, and LCF whose low recalls are caused by the aforementioned mispredictions of formations. This result is far greater than the formation prediction, implying that our method finds most of the positions correctly even when it fails to predict a formation.

6.2 Switching Pattern Discovery

The RolePerms obtained from Eq. 12 indicate the temporary role switches between players. If every player maintains the originally instructed role at time t, then the RolePerm σ_t becomes the identity permutation. In other words, a non-identity RolePerm implies that there is a switching play at that time. Thus, we can identify teams' playing patterns by analyzing the non-identity RolePerms.

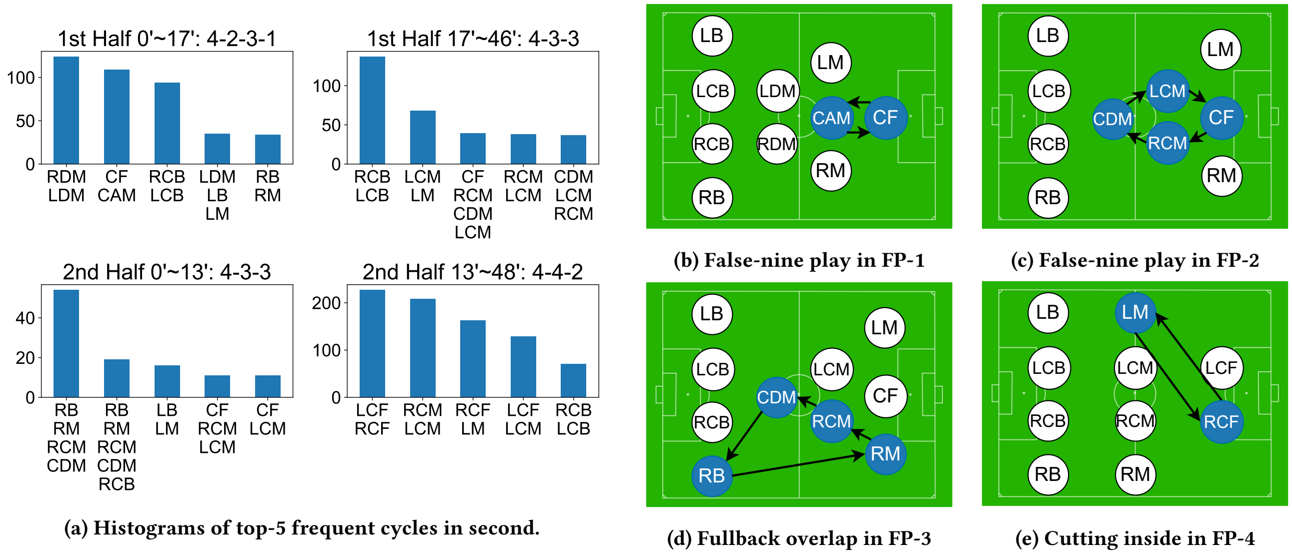
A point to note is that independent switches can occur in the different parts of the pitch at the same time. Thus, we decompose each RolePerm into one or multiple cyclic permutations (which we call *cyclic RolePerms* or simply *cycles*). For example, the RolePerm

$$\sigma = \begin{pmatrix} \mathrm{LB} & \cdots & \mathrm{LM} & \cdots & \mathrm{RM} & \cdots & \mathrm{RCF} \\ \mathrm{LM} & \cdots & \mathrm{LB} & \cdots & \mathrm{RCF} & \cdots & \mathrm{RM} \end{pmatrix}$$

with switch rate 0.4 is the concurrence of the two 2-cycles (LB LM) and (RM RCF). Here, LB and RCF raise a non-identity RolePerm σ together, but they did not switch roles with each other. Considering this kind of cases, we regard concurrent cycles as distinct role switches to solely focus on the "interchanging" patterns of roles.

As a concrete case study, we look into the cycles during the match introduced in Fig. 1 and derive some domain insights. Fig. 7a shows the top-5 frequent cycles with position labels per formation period. Notable observations are as follows, where we abbreviate formation/role periods as FPs/RPs.

- The players dynamically switched their roles in FP-1 and FP-4 (top-5 cycles lasted for about 400 and 750 seconds, respectively), while there were few interchanges in FP-2 and FP-3 (only with 280 and 100 seconds, respectively).
- The center forward performed the "false-nine play" by dropping deep into the midfield positions, generating (CF CAM)



(a) Histograms of top-5 frequent cycles in second.

(b) False-nine play in FP-1

(c) False-nine play in FP-2

(d) Fullback overlap in FP-3

(e) Cutting inside in FP-4

Figure 7: (a) Frequent cycles per formation period, and (b)–(e) some tactically notable cycles from the match in Fig. 1.

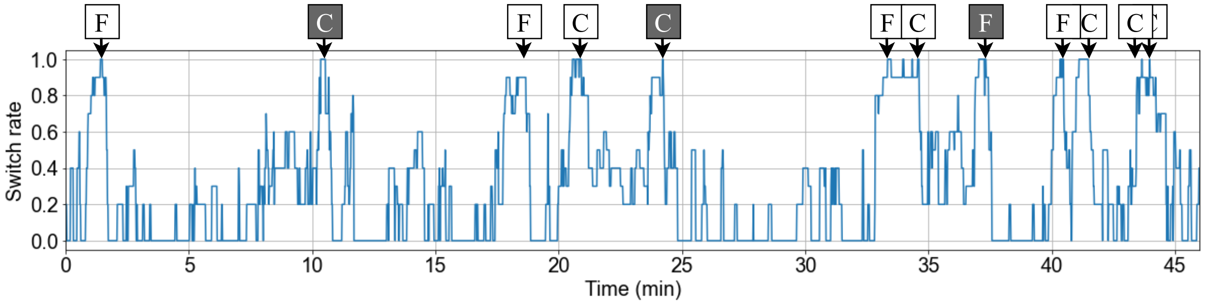


Figure 8: A time-series of switch rates and set-piece occurrence in the first half of the match in Fig. 1. Squares with the labels "C" and "F" mean a corner kick and a close free kick (that the ball was kicked into the box), respectively. The color of a square indicates the attacking team during the situation, where the white corresponds to the measured team of interest.

in FP-1, (CF RCM CDM LCM) in FP-2, (CF RCM LCM) and (CF LCM) in FP-3.

- There were fewer switches in FP-3 by and large, but the overlapping of the right back stood out with the cycles (RB RM RCM CDM) and (RB RM RCM CDM RCB).
- After changing to 4-4-2 in FP-4, the team performed frequent switches between forwards. In particular, the cycle (RCF LM) indicates that the left midfielder kept trying to "cut inside" the box.

6.3 Set-Piece Detection

The "set-piece" in soccer refers to situations such as corner kicks and free kicks that resume the match from a stoppage by kicking the ball into the scoring zone. Since set-pieces are a great opportunity to score a goal [22], teams carefully review set-piece situations from previous matches and set aside special tactics for set plays.

As a byproduct of SoccerCPD, we can also help soccer teams easily extract and manage the set play data by automatically detecting set-pieces based on the statistic *switch rate*. The term switch rate has been defined in Section 4.2, but we slightly alter its definition to the Hamming distance (Eq. 10) between the RolePerm and the identity permutation divided by the number of roles. (Note that we cannot use this definition in Section 4.2, as it is before obtaining the RolePerms.) Since the players are totally mixed up during set-pieces, switch rates in this situation soar close to 1.0. Thus, we can construct a simple, totally unsupervised, but fairly accurate set-piece detection model that picks situations with high switch rates (such as ≥ 0.9). Figure 8 shows the strong co-occurrence pattern of set-pieces and high switch rates.

6.4 Discussion on Limitations

While SoccerCPD offers benefits such as scalability and usability by only relying on player locations, it leaves some limitations as below. For instance, a team can take separate formations for attacking and defending situations, but our framework does not take these contexts into account. Also, ours tends to confuse the formations with similar spatial configurations, since it does not consider on-the-ball actions. The mispredictions of 3-4-1-2 to 3-4-3 and 4-2-3-1 to 4-4-2 in Section 6.1 are cases in point.

Another limitation is that our framework only predicts popular formations well. In other words, since our framework gives formation labels only for the clusters with enough sizes, it just classifies irregular formations into “others” group. This brings about time delay to reflect the tactics of some adventurous managers, such as asymmetrical formations by Pep Guardiola or José Mourinho.

7 CONCLUSION

This study proposes a change-point detection (CPD) framework named SoccerCPD that distinguishes tactically intended formation and role changes from temporary changes in soccer matches. First, we express the temporary role topology and transposition as sequences of binary matrices and permutations, respectively. Considering that these representations are high-dimensional or non-Euclidean data with frequent repetitions, we then apply a graph-based CPD algorithm named discrete g-segmentation to find formation and role assignment change-points.

Since the concept of formation and role is the most fundamental and intuitive way to express soccer tactics, tracking and summarizing the team’s formation/role changes has its own worth for domain participants. In addition, one can also use additional information such as temporary permutations to discover the switching patterns or to detect set-pieces as introduced in our study. Hence, we expect our method to be widespread in the sports industry.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1F1A1073451 and No. 2020R1A4A3079947). Also, the authors thank Taein Lee, Minwoo Seo, and Donghwa Shin at Fitogether Inc. for supporting the model evaluation task by annotating formation and position labels as domain experts.

REFERENCES

[1] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51 (2017), 339–367.

[2] Jushan Bai. 1994. Least squares estimation of a shift in linear processes. *Journal of Time Series Analysis* 15, 5 (1994), 453–472.

[3] Jushan Bai. 1997. Estimation of a change point in multiple regression models. *Review of Economics and Statistics* 79, 4 (1997), 551–563.

[4] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, and Iain Matthews. 2014. “Win at home and draw away”: Automatic formation analysis highlighting the differences in home and away team behaviors. In *MIT Sloan Sports Analytics Conference*.

[5] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. 2014. Identifying team style in soccer using formations learned from spatiotemporal tracking data. In *IEEE International Conference on Data Mining Workshops*.

[6] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. 2014. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *IEEE International Conference on Data Mining*.

[7] Hao Chen and Nancy Zhang. 2015. Graph-based change-point detection. *Annals of Statistics* 43, 1 (2015), 139–176.

[8] Lynna Chu and Hao Chen. 2019. Asymptotic distribution-free change-point detection for multivariate and non-euclidean data. *Annals of Statistics* 47, 1 (2019), 382–414.

[9] Boris Delaunay. 1934. Sur la sphère vide. *Bulletin de l’Académie des Sciences de l’URSS, Classe des Sciences Mathématiques et Naturelles* 6 (1934), 793–800.

[10] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13 (2012), 723–773.

[11] Zaïd Harchaoui, Francis Bach, and Éric Moulines. 2009. Kernel change-point analysis. *Advances in Neural Information Processing Systems* (2009), 609–616.

[12] Hyunsung Kim, Jihun Kim, Dongwook Chung, Jonghyun Lee, Jinsung Yoon, and Sang-Ki Ko. 2021. 6MapNet : Representing soccer players from tracking data by a triplet network. In *ECML-PKDD Workshop on Machine Learning and Data Mining for Sports Analytics*.

[13] Stanley I.M. Ko, Terence T.L. Chong, and Pulak Ghosh. 2015. Dirichlet process hidden Markov multiple change-point model. *Bayesian Analysis* 10, 2 (2015), 275–296.

[14] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97.

[15] Rémi Lajugie, Sylvain Arlot, and Francis Bach. 2014. Large-margin metric learning for constrained partitioning problems. In *International Conference on Machine Learning*. 502–510.

[16] Marc Lavielle and Eric Moulines. 2000. Least-squares estimation of an unknown number of shifts in a time series. *Journal of Time Series Analysis* 21, 1 (2000), 33–59.

[17] Alexandre Lung-Yut-Fong, Céline Lévy-Leduc, and Olivier Cappé. 2015. Homogeneity and change-point detection tests for multivariate data using rank statistics. *Journal de la Société Française de Statistique* 156, 4 (2015), 133–162.

[18] Vinicius Machado, Roger Leite, Felipe Moura, Sergio Cunha, Filip Sadlo, and João L.D. Comba. 2017. Visual soccer match analysis using spatiotemporal positions of players. *Computers and Graphics* 68 (2017), 84–95.

[19] Takuma Narizuka and Yoshihiro Yamazaki. 2017. Characterization of the formation structure in team sports. In *Proceedings of the Institute of Statistical Mathematics*, Vol. 65. 299–307.

[20] Takuma Narizuka and Yoshihiro Yamazaki. 2019. Clustering algorithm for formations in football games. *Scientific Reports* 9 (2019).

[21] Florian Pein, Hannes Sieling, and Axel Munk. 2017. Heterogeneous change point inference. *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 79, 4 (2017), 1207–1227.

[22] Paul Power, Jennifer Hobbs, Hector Ruiz, Xinyu Wei, and Patrick Lucey. 2018. Mythbusting set-pieces in soccer. In *MIT Sloan Sports Analytics Conference*.

[23] Zhongjun Qu and Pierre Perron. 2007. Estimating and testing structural changes in multivariate regressions. *Econometrica* 75, 2 (2007), 459–502.

[24] Robert Rein and Daniel Memmert. 2016. Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus* (2016).

[25] Long Sha, Patrick Lucey, Yisong Yue, Peter Carr, Charlie Rohlf, and Iain Matthews. 2016. Chalkboarding: A new spatiotemporal query paradigm for sports play retrieval. In *International Conference on Intelligent User Interfaces*. 336–347.

[26] Laurie Shaw and Mark Glickman. 2019. Dynamic analysis of team strategy in professional football. In *Barça Sport Analytics Summit*. 1–13.

[27] John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

[28] Hoseung Song and Hao Chen. 2020. Asymptotic distribution-free change-point detection for data with repeated observations. (2020). arXiv:2006.10305

[29] Kohei Tamura and Naoki Masuda. 2015. Win-stay lose-shift strategy in formation changes in football. *EPJ Data Science* 4, 1 (2015), 1–19.

[30] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167 (2020).

[31] Xinyu Wei, Long Sha, Patrick Lucey, Stuart Morgan, and Sridha Sridharan. 2013. Large-scale analysis of formations in soccer. In *International Conference on Digital Image Computing: Techniques and Applications*.

[32] Yingcai Wu, Xiao Xie, Jiachen Wang, Dazhen Deng, Hongye Liang, Hui Zhang, Shoubin Cheng, and Wei Chen. 2019. ForVizor: Visualizing spatio-temporal team formations in soccer. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 65–75.

[33] Jingru Zhang and Hao Chen. 2017. Graph-based two-sample tests for data with repeated observations. (2017). arXiv:1711.04349

A ADDITIONAL EXPLANATION

A.1 Pipeline of the Framework

For better understanding of the whole pipeline, we attach a block diagram for Sections 4 and 5 in Fig. 9. The upper part illustrates the process of applying FormCPD to a session (i.e., half of a match), and the lower part depicts the application of RoleCPD to a formation period resulting from the previous FormCPD. Note that formation clustering (Section 4.3) is the only step that uses data from multiple sessions. Also, formation and position labels on the lower right side of the diagram can vary depending on the dataset and the interpretation of the formation clusters.

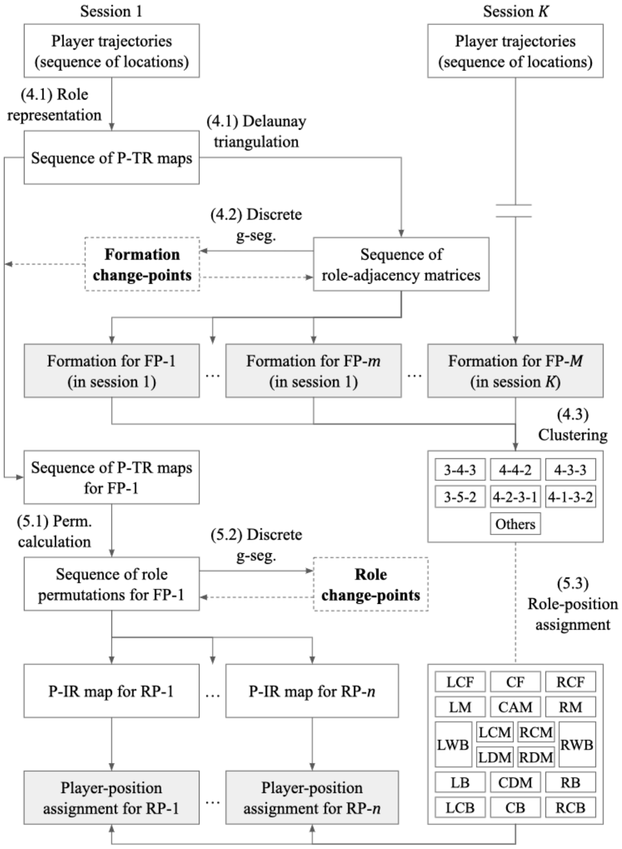

Session 1
Session K
Player trajectories (sequence of locations)
Player trajectories (sequence of locations)
(4.1) Role representation
(4.1) Delaunay triangulation
Sequence of P-TR maps
(4.2) Discrete g-seg.
Formation change-points
Sequence of role-adjacency matrices
Formation for FP-1 (in session 1)
Formation for FP-m (in session 1)
Formation for FP-M (in session K)
(4.3) Clustering
Sequence of P-TR maps for FP-1
3-4-3
4-4-2
4-3-3
3-5-2
4-2-3-1
4-1-3-2
Others
(5.1) Perm. calculation
(5.2) Discrete g-seg.
Sequence of role permutations for FP-1
Role change-points
(5.3) Role-position assignment
P-IR map for RP-1
P-IR map for RP-n
LCF
CF
RCF
LM
CAM
RM
LWB
LCM
RCM
LDM
RDM
RWB
LB
CDM
RB
LCB
CB
RCB
Player-position assignment for RP-1
Player-position assignment for RP-n

Figure 9: A block diagram for Sections 4 and 5. FP and RP are acronyms for formation and role periods, respectively.

A.2 Details of Formation Clustering

In Section 4.3, we proceed the formation clustering in two steps, since it is hard to distinguish 4-4-2 from 4-2-3-1 by a single step of clustering. First, we group 864 formations into 18 clusters where 18 is the minimum number of clusters that main formations except 4-4-2 and 4-2-3-1 are separated from one another. At the next step, the cluster in which 4-4-2 and 4-2-3-1 coexist (i.e., the mixture of Fig. 3c and Fig. 3d) is further split into 18 sub-clusters, with one indicating 4-4-2 and all the others indicating 4-2-3-1. After merging clusters with less than 15 observations into "others" group and those indicating 4-2-3-1 into one group, we get seven formation groups as illustrated in Fig. 3.

A.3 Python Implementation

We have implemented the role representation algorithm using Python 3.8 on our own, while adopted the R package `gSeg`[2] developed by Chen et al. [7], Chu et al. [8], and Song et al. [28] for discrete g-segmentation. To perform the whole process at once, we utilize the Python package `rpy2`[3] 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 `soccercpd`[4] with sample GPS data from an anonymized match.

B COMPARISON AGAINST BASELINES

To empirically justify our model selection, we have performed an additional experiment by comparing the results in Section 6.1 against those using other baseline CPD methods.

Since FormCPD and RoleCPD handle different types of sequences, the sets of compatible baselines also differ from one another. FormCPD takes discrete high-dimensional data (10×10 binary matrices) where parametric methods assuming a certain distribution on a Euclidean space show poor performances. Thus, we adopted non-parametric methods including kernel methods [27] and rank-based methods [17]. We also estimated a formation per session without finding a change-point as a naive baseline. For RoleCPD, the observations are permutations in a symmetric group where we cannot even define an inner product or an order. This means that kernel methods and rank-based methods are not applicable as well as the other off-the-shelf CPD methods for $\mathbb{R}^d$. Hence, we have only used the 5-minute smoothing to find instructed roles per time window as in Bialkowski et al. [6].

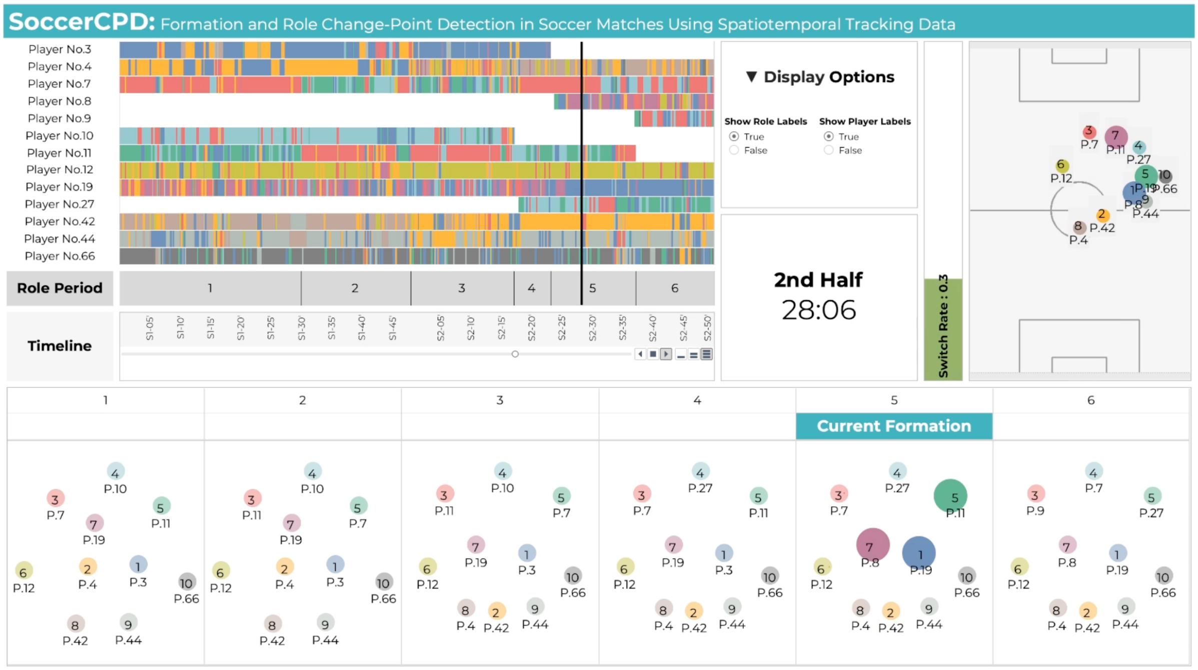
To implement the CPD baselines, we utilize the Python package `ruptures`[5] developed by Truong et al. [30]. The applied methods are listed below.

- **Avg. g-seg.** (for both): Discrete g-segmentation using the average scan statistic. (`gseg1_discrete` in `gSeg`)
- **Union g-seg.** (for both): Discrete g-segmentation using the union scan statistic (`gseg1_discrete` in `gSeg`).
- **RBF kernel** (only for FormCPD): Kernel CPD with the radial basis function kernel (`CostRbf` in `ruptures`).
- **Cosine kernel** (only for FormCPD): Kernel CPD with the cosine similarity kernel (`CostCosine` in `ruptures`).
- **Rank** (only for FormCPD): CPD using a rank transformation (`CostRank` in `ruptures`).
- **None** (only for FormCPD): Estimating a formation from the entire session without applying a CPD method.
- **5-min seg.** (only for RoleCPD): Segmenting a player period (obtained by splitting a session by player substitutions) into

[2] https://rdrr.io/cran/gSeg/
[3] https://rpy2.github.io
[4] https://github.com/pientist/soccercpd
[5] https://centre-borelli.github.io/ruptures-docs/

Table 2: Results of SoccerCPD with different CPD methods.

CPD methods		#. periods		Proportion in playing time							Accuracy	
FormCPD	RoleCPD	FP	RP	3-4-3	3-5-2	4-4-2	4-2-3-1	4-3-3	4-1-3-2	others	Formation	Position
Avg. g-seg.	Avg. g-seg.	864	2152	22.3%	12.2%	16.3%	20.9%	17.4%	5.8%	5.1%	**72.4%**	**86.5%**
Union g-seg.	Avg. g-seg.	862	2263	20.5%	11.1%	13.2%	19.4%	20.2%	4.8%	10.9%	67.4%	85.4%
RBF kernel	Avg. g-seg.	879	2278	21.6%	12.0%	16.6%	23.1%	15.9%	4.4%	6.4%	67.1%	83.8%
Cosine kernel	Avg. g-seg.	883	2269	21.2%	12.0%	14.7%	19.7%	17.9%	8.0%	6.4%	67.1%	84.0%
Rank	Avg. g-seg.	864	2223	17.8%	12.4%	15.3%	19.1%	22.5%	4.1%	8.7%	65.7%	85.1%
None	5-min seg.	807	8536	21.6%	12.4%	15.3%	23.4%	15.7%	6.3%	5.1%	59.1%	80.3%



Show Role Labels: ● True ○ False

Show Player Labels: ● True ○ False

Figure 10: A snapshot of the Tableau animation of SoccerCPD.

5-minute windows and find the most frequent roles for each player per window.

Table 2 shows the results of SoccerCPD for each method. All the baselines except for the non-CPD one split the total of 807 sessions into about 870 formation periods and 2,200 role periods. It clearly demonstrates that higher performances can be obtained when employing CPD methods instead of the naive segmentation. Moreover, discrete g-segmentation with the average scan statistic outperforms the other CPD methods.

C TABLEAU VISUALIZATION

Lastly, we visualize our results as an animation using Tableau 2020.4. Fig. 10 is a snapshot of the generated animation. It dynamically shows all the following tactical information in one dashboard:

- **Player trajectories** colored by temporary roles in the right.
- **Player-to-temporary-role assignments** as colored bar codes in the left timeline, divided into detected role periods.
- **Formation** per role period by the mean role locations in the bottom with the emphasis on the current formation.
- **Player-to-instructed-role assignments** per role period by annotating uniform numbers below the role circles.
- **Switching players** indicated by inflating the corresponding circles both in the formations and the pitch image.

We have published the full-version video on YouTube[6]. We expect that this visual interface effectively supports domain participants to review and analyze the match by tracking tactical changes and discovering switching patterns.

[6] https://www.youtube.com/watch?v=F-tvG1-MLps