IMPROVED STRUCTURAL DISCOVERY AND REPRESENTATION LEARNING OF MULTI-AGENT DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Central to all machine learning algorithms is data representation. For multi-agent systems, selecting a representation which adequately captures the interactions among agents is challenging due to the latent group structure which tends to vary depending on various contexts. However, in multi-agent systems with strong group structure, we can simultaneously learn this structure and map a set of agents to a consistently ordered representation for further learning. In this paper, we present a dynamic alignment method which provides a robust ordering of structured multi-agent data which allows for representation learning to occur in a fraction of the time of previous methods. We demonstrate the value of this approach using a large amount of soccer tracking data from a professional league.

1 INTRODUCTION

The natural representation for many sources of unstructured data is intuitive to us as humans: for images, a 2D pixel representation; for speech, a spectrogram or linear filter-bank features; and for text, letters and characters. All of these possess fixed, rigid structure in space, time, or sequential ordering which are immediately amenable for further learning. For other unstructured data sources such as point clouds, semantic graphs, and multi-agent trajectories, such an initial ordered structure does not naturally exist. These data sources are set or graph-like in nature and therefore the natural representation is unordered, posing a significant challenge for many machine-learning techniques.

A domain where this is particularly pronounced is in the fine-grained multi-agent player motions of team sport. Access to player tracking data changed how we understand and analyze sport (Miller et al., 2014; Franks et al., 2015; Wei et al., 2013; Cervone et al., 2014; Power et al., 2017; Sha et al., 2016; 2018; Yue et al., 2014). More relevantly, sport has risen to an increasingly key space within the machine learning community as an application to expand our understanding of adversarial multi-agent motion, interaction, and representation (Lucey et al., 2013; Le et al., 2017; Felsen et al., 2018; Zheng et al., 2016; Zhan et al., 2018; Yeh et al., 2019; Kurach et al., 2019).

In sport there exists strong, complex group-structure which is less prevalent in other multi-agent systems such as pedestrian tracking. Specifically, the *formation* of a team captures not only the global shape and structure the group, but also enables the ordering of each agent according to a ”role” within the group structure. In this regard, sport possesses relational structure similar to that of faces and bodies, which can be represented as a graph of key-points. In those domains, representation based on a fixed key-point ordering has allowed for cutting edge work across numerous tasks with a variety of approaches and architectures (Antonakos et al., 2015; Cootes et al., 2001; Akhter et al., 2012; Joo et al., 2015; Simon et al., 2017).

Unlike for faces and bodies, the representation graph in sport is dynamic as players constantly move and switch positions. Thus dynamically discovering the appropriate representation of individual players according to their role in a formation affords us both structural information while learning a useful representation for subsequent tasks. This challenge was addressed by the original role-based alignment of Lucey et al. (2013) and subsequently by Bialkowski et al. (2016) and Sha et al. (2017). Role-based alignment allows us to take unstructured multi-agent data, and reformat it into a consistent vector format that enables subsequent machine learning to occur (Fig. 1).

Here we formulate the role-based alignment as consisting of two phases: formation discovery and role assignment. Formation discovery uses unaligned data to learn an optimal *formation template*

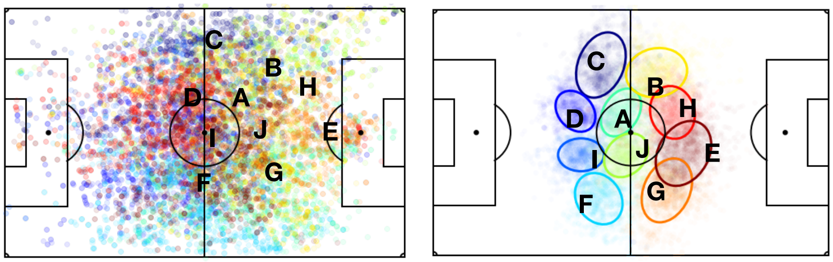

Figure 1: A structure representation enables machine learning of multi-agent data. (Left) Datapoints are colored according to the agent identity (letters denote agents in a given frame). (Right) By learning and aligning data to a formational template, we represent agents in a consistent vector form conducive to learning. Agents are now ordered by the role to which they are assigned.

while the second phase applies a bipartite mapping between agents and roles in each frame to produce ”aligned data”. A major limitation in past approaches was the speed of the template discovery process. In this work we propose an improved approach to the above alignment methods which provides faster and more stable template discovery and representation learning. This enables on-the-fly discovery of the formation templates which reduces computational load and enables new context-specific analysis.

Our main contributions are: the formulation of this problem as a three-step approach (formation discovery, role assignment, template-clustering) with different methods for each, the use of soft-assignment in the template discovery phase thereby eliminating the costly hard-assignment step of the Hungarian Algorithm, a resetting training procedure based on the formation eigenvalues to prevent spurious optima, quantification of the impact of initialization convergence and stability, a restriction of the training data to key-frames for faster training with minimal impact on identical, and a multi-agent clustering framework which captures the covariances across agents during the template-clustering phase.

2 Background

2.1 Representing Structured Multi-Agent Data

A collection of agents is by nature a set and therefore no defined ordering exists *a priori*. Thus to impose an arbitrary ordering introduces significant entropy into the system through the possible permutations of agents in the imposed representation.

To circumvent this, some approaches in representing multi-agent tracking data in sport have included sorting the players based on an ”anchor” agent (Mehrasa et al.). This is limited in that the optimal anchor is task-specific, making the representation less generalizable. An ”image-based” representation (Yue et al., 2014; Zheng et al., 2016; Miller et al., 2014) eliminates the need for an ordering, however, this representation is lossy, sparse, and high-dimensional.

The role-based alignment protocol for sport of Lucey et al. (2013) used a codebook of hand-crafted formation templates against which frame-level[1] samples were aligned. This work was extended by Bialkowski et al. (2016) which learned the template directly from the data. Sha et al. (2017) further employed a hierarchical template learning framework, useful in both retrieval (Sha et al., 2018) and trajectory prediction (Felsen et al., 2018; Yeh et al., 2019).

Le et al. (2017) similarly learned an agent-ordering directly from the data by learning separate role-assignment and motion-prediction policies in an iterative and alternating fashion.

[1] Throughout we use the term ”frame” to indicate a single moment in time in reference to the data being obtained via optical tracking from video.

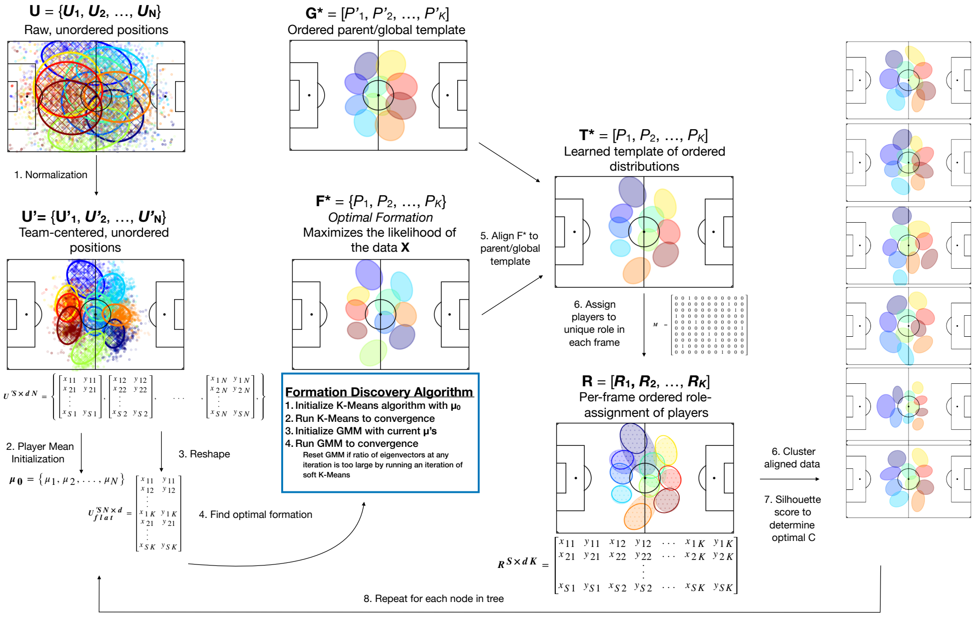
U = {U1, U2, …, UN}
Raw, unordered positions
1. Normalization
U'= {U'1, U'2, …, U'N}
Team-centered, unordered positions
$U'^{S \times dN} = \left\{ \begin{bmatrix} x_{11} & y_{11} \\ x_{21} & y_{21} \\ \vdots \\ x_{S1} & y_{S1} \end{bmatrix}, \begin{bmatrix} x_{12} & y_{12} \\ x_{22} & y_{22} \\ \vdots \\ x_{S2} & y_{S2} \end{bmatrix}, \dots, \begin{bmatrix} x_{1N} & y_{1N} \\ x_{2N} & y_{2N} \\ \vdots \\ x_{SN} & y_{SN} \end{bmatrix}, \right\}$
2. Player Mean Initialization
$\mu_0 = \{\mu_1, \mu_2, \dots, \mu_N\}$
3. Reshape
$U'^{SN \times d}_{flat} = \begin{bmatrix} x_{11} & y_{11} \\ x_{12} & y_{12} \\ \vdots \\ x_{1K} & y_{1K} \\ x_{21} & y_{21} \\ \vdots \\ x_{SK} & y_{SK} \end{bmatrix}$
4. Find optimal formation
G* = [P'1, P'2, …, P'K]
Ordered parent/global template
F* = {P1, P2, …, PK}
Optimal Formation
Maximizes the likelihood of the data X
Formation Discovery Algorithm
1. Initialize K-Means algorithm with μ0
2. Run K-Means to convergence
3. Initialize GMM with current μ's
4. Run GMM to convergence
Reset GMM if ratio of eigenvectors at any iteration is too large by running an iteration of soft K-Means
5. Align F* to parent/global template
T* = [P1, P2, …, PK]
Learned template of ordered distributions
6. Assign players to unique role in each frame
M =
R = [R1, R2, …, RK]
Per-frame ordered role-assignment of players
$R^{S \times dK} = \begin{bmatrix} x_{11} & y_{11} & x_{12} & y_{12} & \cdots & x_{1K} & y_{1K} \\ x_{21} & y_{21} & x_{22} & y_{22} & \cdots & x_{2K} & y_{2K} \\ & & & \vdots & & & \\ x_{S1} & y_{S1} & x_{S2} & y_{S2} & \cdots & x_{SK} & y_{SK} \end{bmatrix}$
6. Cluster aligned data
7. Silhouette score to determine optimal C
8. Repeat for each node in tree

Figure 2: An overview of the proposed method. The procedure consists of (1) Normalization, (2) Initialization, (3) Resetting, (4) Formation Discovery, (5) Template Alignment, (6) Role Assignment, (7) Template Clustering. In the role assignment step, the template distributions are shown as unfilled thick ellipses and the observed distributions of the role-aligned data are shown as the textured ellipses.

2.2 Permutation-Equivariant Approaches

Permutation-equivariant approaches seek to leverage network architectures which are insensitive to the ordering of the input data. Approaches using graph neural networks (GNN) (Kipf & Welling, 2016; Gilmer et al., 2017; Battaglia et al., 2018; 2016; Hoshen, 2017) have become very popular and shown tremendous promise. These approaches are particularly valuable for tasks (e.g. pedestrian tracking) which lack the strong coherent group structure of sport and therefore cannot leverage methods such as role-alignment. Within sport, Kipf et al. (2018) used a GNN to predict the trajectories of players while simultaneously learning the edge-weights of the graph. Yeh et al. (2019) demonstrated the advantages in using GNNs to forecast the future motion of players in sports, surpassing both the role (Bialkowski et al., 2016) and tree-based approaches (Sha et al., 2017) on most metrics.

The success of these approaches, however, does not negate the value of role-based alignment. The learned formation structure provides valuable insight into high-level organization of the group. Furthermore, many traditional machine learning techniques and common deep architectures require an ordered-agent representation. This is again similar to what is seen in the modeling of faces and bodies: great success has been achieved using geometric deep learning (Monti et al., 2017; Kipf et al., 2018), but approaches based on a fixed representation remain popular and effective (Taylor et al., 2017; Kanazawa et al., 2019; Arnab et al., 2019; Walker et al., 2017; Rayat Imtiaz Hossain & Little, 2018).

Interesting work has also been done to learn permutations for self-supervised feature learning or ranking tasks (Adams & Zemel, 2011; Mena et al., 2018; Cruz et al., 2017). Central to these approaches is the process of Sinkhorn normalization (Sinkhorn & Knopp, 1967), which allows for soft-assignment during the training process and therefore a flow of gradients. Exploring the application of Sinkhorn normalization to this task is beyond the scope of this current work, however, we provide additional context on this method in Section A.2.

3 Approach

3.1 Problem Formulation

Mathematically, the goal of the role-alignment procedure is to find the transformation $A : \{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_N\} \times \boldsymbol{M} \mapsto [\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_K]$ which maps the unstructured set $\boldsymbol{U}$ of N player trajectories to an ordered set (i.e. vector) of K role-trajectories $\boldsymbol{R}$. [2] Each player trajectory is itself an ordered set of positions $\boldsymbol{U}_n = [x_{s,n}]_{s=1}^{S}$ for an agent $n \in [1, N]$ and a frame $s \in [1, S]$. We recognize $\boldsymbol{M}$ as the optimal permutation matrix which enables such an ordering.

Thus our goal is to find the most probable set $\boldsymbol{\mathcal{F}}^*$ of 2D probability density functions

$$\boldsymbol{\mathcal{F}}^* = \arg\max_{\boldsymbol{\mathcal{F}}} P(\boldsymbol{\mathcal{F}}|\boldsymbol{R}) \tag{1}$$

$$P(\boldsymbol{x}) = \sum_{n=1}^{N} P(\boldsymbol{x}|n)P(n) = \frac{1}{N}\sum_{n=1}^{N} P_n(\boldsymbol{x}) \tag{2}$$

Bialkowski et al. (2016) transforms this equation into one of entropy minimization where the goal is to minimize the amount of overlap (i.e. the KL-Divergence) between each role. The final optimization equation in terms of the total entropy H then becomes

$$\boldsymbol{\mathcal{F}}^* = \arg\min_{\boldsymbol{\mathcal{F}}} \sum_{n=1}^{N} H(\boldsymbol{x}|n). \tag{3}$$

See A.1 for additional details.

The authors then use expectation maximization (EM) to approximate this solution and note similarity to k-means clustering. However, as they represent that data non-parametrically in terms of per-role heat maps, hard assignment must be applied at each iteration so the distributions may be updated. Instead, we note that equation 2 describes the occupancy of space by any agent in any point in time as a mixture of conditional distributions across each of the N-roles. This is further equivalent to the sum over n-generating distributions. Thus if we model these generating distributions as d-dimensional Gaussian distributions, this reduces the template-discovery process to that of a Gaussian Mixture Model.

3.2 Toy Problem Formulation

Understanding the notion of independence under the different formulations of this problem is key. This may be better understood by considering a toy problem: imagine we have three independent 1D Gaussian distributions we wish to sample S times from each. It is known that we sample from each distribution in rounds, effectively generating the samples in ”triplets”, although the order within the triplets is random. We then seek to reassign the points back to their original distributions.

Following the approach of (Bialkowski et al., 2016), the ”structure” imposed by the triplet sampling is enforced through the hard-assignment at each iteration. Recall, however, that the original distributions were statistically independent; the additional triplet structure we wish to respect is in the assignment step, not the underlying distributions.

Contrastingly, in our method the samples are treated as fully independent; had all samples been taken from the first distribution, followed by the second, followed by the third, the outcome at the ”distribution-discovery” phase would be identical to that having sampled the data in rounds. Only after the three distributions are estimated would the assignment of each point in every triplet be assigned to the distribution which maximized the overall likelihood in that triplet.

Besides being more computationally effective (see Section 4.2), this allows us to find the true MLE of the distributions. Our method will always discover a more optimal estimate of Eq. 2. This can be understood in considering how the assignment is performed during optimization process. For each triplet, the likelihood of assigning each point to each distribution is computed in both approaches.

[2] Generically N need not equal K as a player may be sent off during a game, but for simplicity it is safe to assume $N = K$ in this work.

In our approach, this gives *the* likelihood under each mixture component. In the hard-assignment approach, however, if two (or more) points in a triplet have their highest likelihood under the same component, the exclusionary assignment *must* result in a lower likelihood than assigning each point to its preferred Gaussian.

Furthermore, in our approach, each sample contributes to every component of the mixture, thus the data under the mixture remains ”fixed” during the optimization process. In contrast, as the hard-assignments are made, the samples contributing to each distribution changes each iteration. This, in combination with the sub-optimal likelihood above, effectively ”breaks” the expectation maximization step and can cause solutions to diverge or oscillate, which is inconsistent with a maximum likelihood solution which must monotonically increase.

Thus our approach is computationally efficient, more intuitively captures the independence of the generating distributions versus the structure of the sampling, and ensures a likelihood function that will converge under expectation maximization.

3.3 Formation-Discovery

Our procedure is presented visually in Figure 2 and algorithmically in Algorithm A.3.

Data is normalized so all teams are attacking from left to right and have zero mean in each frame, thereby removing translational effects.

Following the approach of Bialkowski et al. (2016), we initialize the cluster centers for formation-discovery with the average player positions. The impact of this choice of initialization is explored in Section 4.3.

We now structure all the data as a single $(SN) \times d$ vector where S is the total number of frames, N is the total number of agents (10 outfielders in the case of soccer), and d is the dimensionality of the data (2 here). The K-Means algorithm is initialized with the player means calculated above and run to convergence; we find that running K-Means to convergence produces better results than running a fixed number of iterations as is commonly done for initialization. The cluster-centers of the last iteration are then used to initialize the subsequent mixture of Gaussians.

Mixture of Gaussians are known to suffer from component collapse and becoming trapped in pathological solutions. To combat this, we monitor the eigenvalues (λ_i) of each of the components throughout the EM process. If the eigenvalue ratio of any component becomes too large or too small, the next iteration runs a Soft K-Means (i.e. a mixture of Gaussians with spherical covariance) update instead of the full-covariance update. We find that the range $\frac{1}{2} < \frac{\lambda_1}{\lambda_2} < 2$ works well. We refer to this set of K distributions which maximizes the likelihood of the data the *Formation*, which we denote $\boldsymbol{\mathcal{F}}^*$.

Note that the formation is a set of distributions. To enforce an ordering, we must align to a parent template, $\boldsymbol{G}^*$, which is an ordered set of distributions. The specific ordering of this template is unimportant so long as it is established and fixed. We align $\boldsymbol{\mathcal{F}}^*$ to $\boldsymbol{G}^*$ by finding the Bhattacharyya distance between each distribution in $\boldsymbol{\mathcal{F}}^*$ and $\boldsymbol{G}^*$ given by

$D_B = \frac{1}{8}(\mu_{\boldsymbol{\mathcal{F}}^*_i} - \mu_{\boldsymbol{G}^*_j})^T \sigma^{-1}(\mu_{\boldsymbol{\mathcal{F}}^*_i} - \mu_{\boldsymbol{G}^*_j}) + \frac{1}{2}\ln(\frac{\det \sigma}{\sqrt{\det \sigma_{\boldsymbol{\mathcal{F}}^*_i} \det \sigma_{\boldsymbol{G}^*_j}}})$ where $\sigma = \frac{\sigma_{\boldsymbol{\mathcal{F}}^*_i} + \sigma_{\boldsymbol{G}^*_j}}{2}$ to create a $K \times K$ cost matrix and then use the Hungarian algorithm to find the best assignment. We have now produced our *Template*, $\boldsymbol{\mathcal{T}}^*$ an ordered set of distributions with an established ordering that maximizes the likelihood of the data.

3.4 Role-Assignment

The process of role-assignment maps each player in each frame to a specific role with the restriction that only one player may occupy a role in a given frame. To do this, we find the likelihood that each agent belongs to each of the discovered distributions in each frame which was already calculated during the formation-discovery step. This produces a $N \times K$ cost matrix in each frame; the Hungarian algorithm is again used to make the optimal assignment. Thus we have achieved the tasks of formation-discovery and role-assignment having had to apply the Hungarian algorithm on only a single pass of the data. Note that our final role ordering is *identical* to that obtained via the previous method, up to normal convergence differences. We now represent the aligned data as a $S \times (dK)$ matrix $\boldsymbol{R}$.

3.5 Clustering Multi-Agent Data

With an established well-ordered representation, we are now able to cluster the multi-agent data to discover sub-templates and perform other analysis.

Sub-templates may be found either through flat or hierarchical clustering. Generically, we seek to find a set of clusters $\boldsymbol{C}$ which partitions the data into distinct states according to:

$$\arg\min_{\boldsymbol{C}} \sum_{C_k \in \boldsymbol{C}} \sum_{\boldsymbol{R}_i, \boldsymbol{R}_j \in C_k} \| P(\boldsymbol{R}_i) - |P(\boldsymbol{R}_j) \|_2 \tag{4}$$

For flat clustering, a dN-dimensional K-Means model is fit to the data. To help initialize this clustering, we seed the model with the template means plus a small amount of noise. To determine the optimal number of clusters we use a measure similar to Silhouette score (Rousseeuw, 1987):

$$\mathbb{E}(\boldsymbol{R}) = \frac{1}{|\boldsymbol{R}|} \sum_{C_k \in \boldsymbol{C}} \sum_{\boldsymbol{R}_i \in C_k} \frac{\| P(\boldsymbol{R}_i) - \mu_{kn} \|_2 - \| P(\boldsymbol{R}_i) - \mu_k \|_2}{\| P(\boldsymbol{R}_i) - \mu_{kn} \|_2} \tag{5}$$

where μ_k is the mean of the cluster that example $\boldsymbol{R}_i$ belongs to and μ_{kn} is the mean of the closest neighbor cluster of example $\boldsymbol{R}_i$. Equation 5 measures the dissimilarity between neighboring clusters and the compactness of the data within each cluster. By maximizing E we seek to capture the most discriminative clusters.

To learn a tree of templates through hierarchical clustering, we follow the method of (Sha et al., 2017) with minor modification on how the clusters and templates are initialized. Algorithm A.4 outlines this procedure.

4 Results

4.1 Dataset

For this work, an entire season of player tracking data for a professional European soccer league was used; this consisted of 380 games, 6 of which were omitted due to missing data. The data is collected from an in-venue optical tracking system which records the (x, y) positions of the players at 10Hz. The data also contains single-frame event-labels (e.g. pass, shot, cross) in associated frames; these events were used only to identify which frames contained the onset of an event which we call *event-frames*. Unless explicitly noted, the analysis used only event-frames for training, providing over 1.8million samples across the season.

4.2 Run Complexity

Finding the optimal solution to K-Means is NP-hard, even for 2 clusters. However, through standard methods K-means clustering can achieve an average per-iteration complexity of $(samples \cdot clusters \cdot dimensions)$ while gaussian mixture models have a complexity of $(samples \cdot clusters \cdot dimensions^2)$ per iteration due to the additional calculation of the precision matrix (Lloyd, 1982; Verbeek et al., 2003). Note that for all algorithms, $samples$ becomes SN since each agent in each frame contributes to the distributions. The Hungarian Algorithm has a complexity of $elements^3$ per application.

In the original algorithm of Bialkowski et al. (2016), the cost matrix per frame is calculated in a manner resembling that of the GMM, requiring the full distribution (i.e. mean and precision matrix) to be computed so the likelihoods may be calculated. However, the Hungarian Algorithm is then applied across the N-agents in each of the S-frames. This produces a per-iteration complexity of $(SN)Kd^2N^3$. With $K = N$, this simplifies to SN^5d^2 (see Table 1 of Section A.5). Therefore, for a sport like soccer, $N = 10$, causing the original hard-assignment based algorithm to be ~1000 times slower than the proposed approach.

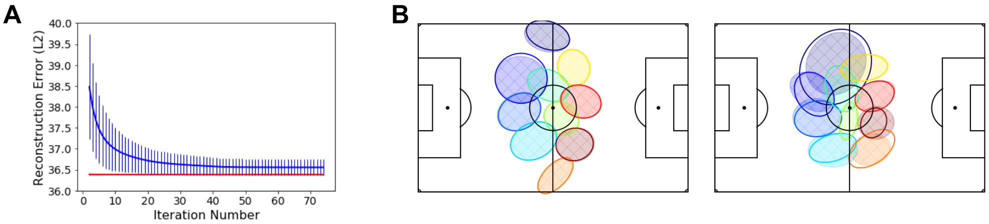

Figure 3: Impact of initialization and key-frame selection. (A) Player-mean initialization (red) enables the K-Means initialization to run to convergence in fewer iterations than random (blue) initialization. (B) Learning the formation on event-only ”key-frames” (thick line, no hashing) results in formations which are very similar to the formations learned on all data (thin line, hashing), but runs significantly faster due the reduced data size and is less prone to find spurious optima. Left: an average example showing the formations learned on the two sets of data are very similar. Right: an unusual ”bad” example showing more disagreement between the two data sets.

4.3 Impact of Initialization and Key-Frames

The original template-learning procedure proposed initializing the algorithm with the distributions of each player, as players tend to spend much of their time in a specific role. In the subsequent work of Sha et al. (2017), a random initialization at each layer was proposed.

To assess the impact of the player-mean initialization, we ran Alg. A.3 20 times per team-period-game, each of which contains about 1500 rows. We recorded the reconstruction error during the K-Means initialization phase of the algorithm. While the exact reconstruction error is sample specific, all samples showed the same trend as Figure 3A: player-mean initialization begins with a much lower reconstruction error and converges significantly more quickly, often within 10 steps. In contrast, the random initialization is much more variable, takes many more iterations to converge, and often does not converge to as good a solution.

The use of event-only ”key frames” is also a key performance and stabilization measure. Limiting the data to event-only frames reduces the data by a factor of ~ 10 , producing a speed-up of the same factor. This has minimal impact on the learned template as seen in Figure 3B. In most instances, the templates learned are almost identical: the average L2-distance between the center of two role distributions is 0.24m and the average Bhattacharyya distance between two role distributions is 0.078.

4.4 Compression Evaluation

Template-based aligned has been shown to produce a compressed representation of multi-agent spatiotemporal data in sport (Lucey et al., 2013). We repeat this analysis here in Figure 4. Similar to the approach in (Sha et al., 2017), we evaluate the compressibility of the approach using clustering and principle component analysis (PCA). We randomly selected 500,000 frames from the larger data set. For the role-based analysis, these frames were aligned according to Algorithm A.3. K-means clustering was applied to both the original unaligned data and aligned data for varying values of K. The average within-cluster-error (WCE) was calculated according to WCE $= \frac{1}{|\boldsymbol{R}|}\sum_{C_k}\sum_{\boldsymbol{R}_i \in C_k} \|\boldsymbol{R}_i - \mu\|_2$ where we again abuse C_k to indicate the k^{th} cluster after K-means clustering. Similarly we run PCA on both the unaligned and aligned data and compute the variance explained by the eigenvectors: Variance Explained $= \frac{\lambda_k}{\sum_{i=1}^{D}\lambda_i}$ where λ_i is the i^{th} eigenvalue indicating the significance of the i^{th} eigenvector.

4.5 Context-Specific Formations

Previously, due to the slowness of the original algorithm, templates had to be learned as a part of a preprocessing step before storage/analysis/consumption. Usually this would be done at the team-game-period level, generating a total of 4 ”specialist” templates per game.

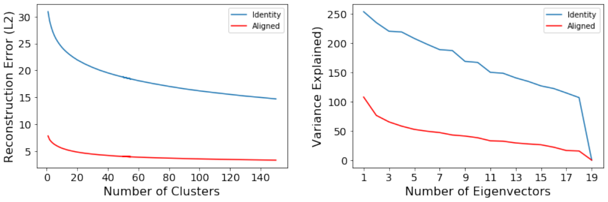

Figure 4: Template-based alignment produces a more compressed representation of the data. Left: reconstruction error as a function of the number of clusters. Right: variance accounted for as a function of the number of eigenvectors used.

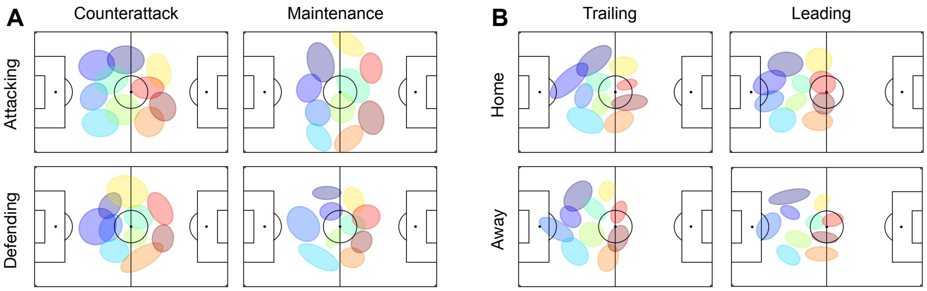

Figure 5: Context-specific templates. (A) We trained distinct templates of a given team while attacking and defending against certain modes (aka. ”styles”) of play. Data is aggregated over multiple games across the season. (B) We trained distinct templates of a given team while defending during the last 10 minutes of the games while trailing and leading, both home and away. Here we have added back in the average team (i.e. group) position to show the overall positioning on the pitch.

In contrast, the proposed method allows templates to be computed ”on the fly”. For several thousand rows of data, the formation can be discovered and aligned in only a few seconds. This allows us to select data under interesting contexts and learn the template that best describes those scenarios across many games.

Figure 5 shows two such analyses this method unlocks. On the left (A) we examine the formation of a team across an entire season when the are attacking in and defending against two very different ”styles” of play (the very offensively aggressive counterattack, and a conservative ”hold the ball” maintenance style) (Ruiz et al., 2017). Similarly, we can examine how a team positions itself when leading or trailing late in a game both at home or away (B). In addition to learning the formation, we can add back in the overall group positioning to see where on the pitch the team attempts to position itself. Additionally, we can learn and align the unique formations of every team across an entire season in a matter of minutes (see Figure 6 in A.6).

5 Summary

For multi-agent systems with a high degree of structure such as that seen in team sport, we are able to learn a mapping which takes the set of agents to an ordered vector of agents without introducing undue entropy from permutation. In this work we have shown an improved method for learning the group representation of structured multi-agent data which is significantly faster and more stable than previous approaches. Our approach exploits the independence of the role-generating distributions during the template-learning phase and enforces the hard assignment of a single agent to a single role only during the final alignment step. This new approach, in combination with a smart choice of key-frame selection and initialization, allows for this representation to be learned over n^3 times faster- a factor of more than 1000 for a sport like soccer. By learning this representation, we are able to perform season-wide contextual and on-the-fly representation learning which were previously computationally prohibitive.

REFERENCES

Ryan Prescott Adams and Richard S Zemel. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011.

Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics*, 31(2):17:1–17:12, April 2012. doi: 10.1145/2159516.2159523.

Epameinondas Antonakos, Joan Alabort-i Medina, and Stefanos Zafeiriou. Active pictorial structures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3395–3404, 2019.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *CoRR*, abs/1612.00222, 2016. URL `http://arxiv.org/abs/1612.00222`.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL `http://arxiv.org/abs/1806.01261`.

Alina Bialkowski, Patrick Lucey, Peter Carr, Iain Matthews, Sridha Sridharan, and Clinton Fookes. Discovering team structures in soccer from spatiotemporal data. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2596–2605, 2016.

Dan Cervone, Alexander DAmour, Luke Bornn, and Kirk Goldsberry. Pointwise: Predicting points and valuing decisions in real time with nba optical tracking data. In *Proceedings of the 8th MIT Sloan Sports Analytics Conference, Boston, MA, USA*, volume 28, pp. 3, 2014.

Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):681–685, 2001.

Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Deeppermnet: Visual permutation learning. *CoRR*, abs/1704.02729, 2017. URL `http://arxiv.org/abs/1704.02729`.

Panna Felsen, Patrick Lucey, and Sujoy Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 732–747, 2018.

Alexander Franks, Andrew Miller, Luke Bornn, Kirk Goldsberry, et al. Characterizing the spatial structure of defensive skill in professional basketball. *The Annals of Applied Statistics*, 9(1): 94–121, 2015.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL `http://arxiv.org/abs/1704.01212`.

Yedid Hoshen. VAIN: attentional multi-agent predictive modeling. *CoRR*, abs/1706.06122, 2017. URL `http://arxiv.org/abs/1706.06122`.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL `http://arxiv.org/abs/1609.02907`.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Karol Kurach, Anton Raichuk, Piotr Staczyk, Micha Zajc, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment, 2019.

Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1995–2003. JMLR. org, 2017.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Patrick Lucey, Alina Bialkowski, Peter Carr, Stuart Morgan, Iain Matthews, and Yaser Sheikh. Representing and discovering adversarial team behaviors using player roles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2706–2713, 2013.

Nazanin Mehrasa, Yatao Zhong, Frederick Tung, Luke Bornn, and Greg Mori. Deep learning of player trajectory representations for team activity analysis.

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. Factorized point process intensities: A spatial analysis of professional basketball. In *International conference on machine learning*, pp. 235–243, 2014.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5115–5124, 2017.

Paul Power, Hector Ruiz, Xinyu Wei, and Patrick Lucey. Not all passes are created equal: Objectively measuring the risk and reward of passes in soccer from tracking data. pp. 1605–1613, 08 2017. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098051.

Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 68–84, 2018.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Hector Ruiz, Paul Power, Xinyu Wei, and Patrick Lucey. The leicester city fairytale?: Utilizing new soccer analytics tools to compare performance in the 15/16 & 16/17 epl seasons. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1991–2000. ACM, 2017.

Long Sha, Patrick Lucey, Yisong Yue, Peter Carr, Charlie Rohlf, and Iain Matthews. Chalkboarding: A new spatiotemporal query paradigm for sports play retrieval. 03 2016. doi: 10.1145/2856767.2856772.

Long Sha, Patrick Lucey, Stephan Zheng, Taehwan Kim, Yisong Yue, and Sridha Sridharan. Fine-grained retrieval of sports plays using tree-based alignment of trajectories. *arXiv preprint arXiv:1710.02255*, 2017.

Long Sha, Patrick Lucey, Yisong Yue, Xinyu Wei, Jennifer Hobbs, Charlie Rohlf, and Sridha Sridharan. Interactive sports analytics: An intelligent interface for utilizing trajectories for interactive sports play retrieval and analytics. *ACM Transactions on Computer-Human Interaction*, 25:1–32, 04 2018. doi: 10.1145/3185596.

Tomas Simon, Hanbyul Joo, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. *CVPR*, 2017.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):93, 2017.

Jakob J Verbeek, Nikos Vlassis, and Ben Kröse. Efficient greedy learning of gaussian mixture models. *Neural computation*, 15(2):469–485, 2003.

Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3332–3341, 2017.

Xinyu Wei, Long Sha, Patrick Lucey, Stuart Morgan, and Sridha Sridharan. Large-scale analysis of formations in soccer. *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2013.

Raymond A. Yeh, Alexander G. Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Yisong Yue, Patrick Lucey, Peter Carr, Alina Bialkowski, and Iain Matthews. Learning fine-grained spatial models for dynamic sports play prediction. In *2014 IEEE International Conference on Data Mining*, pp. 670–679. IEEE, 2014.

Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generative multi-agent behavioral cloning. *arXiv*, 2018.

Stephan Zheng, Yisong Yue, and Patrick Lucey. Generating long-term trajectories using deep hierarchical networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 1543–1551. Curran Associates, Inc., 2016.

A Appendix

A.1 Summary of the Formation Discovery Algorithm of Bialkowski et al. (2016)

To learn the per-role player distributions, the authors cast the problem as one of data paritioning where the goal is to minimize the overlap of an individual role $P_n(\boldsymbol{x})$ and that of the team $P(\boldsymbol{x})$ by placing a penalty V_n on that overlap as defined by

$$V_n = -KL(P_n(\boldsymbol{x})\|P(\boldsymbol{x})) \tag{6}$$

where

$$KL(P(x)\|Q(x)) = \int P(x)\log\left(\frac{P(x)}{Q(x)}\right)dx. \tag{7}$$

Eq. 1 can then be written as

$$\boldsymbol{\mathcal{F}}^* = \arg\max_{\boldsymbol{\mathcal{F}}} V \tag{8}$$

Substituting Eq. 6 into Eq. 8 yields

$$
\begin{aligned}
V = &- \sum_{n=1}^{N} P(n) \int P(\boldsymbol{x}|n) \log P(\boldsymbol{x}|n) dx \\
&- \sum_{n=1}^{N} P(n) \int P(\boldsymbol{x}|n) \log P(\boldsymbol{x}) dx.
\end{aligned} \tag{9}
$$

Written in terms of the entropy $H(x) = -\int_{-\infty}^{+\infty} P(x) \log(P(x)) dx$ this simplifies to

$$
V = -H(x) + \frac{1}{N} \sum_{n=1}^{N} H(\boldsymbol{x}|n) \tag{10}
$$

yielding the final equation of $\boldsymbol{\mathcal{F}}^*$ as given in Eq. 3.

They approximate the solution by using expectation maximization (EM) as summarized here:

Initialization: The data is normalized such that the average team position in each frame is placed at the origin. The formation is initialized by assigning a player to a single role for the entire game. This allows the construction of n independent distributions describing the n different generating roles.

E-Step: An $n \times n$ cost matrix is computed for each frame which is based on the log-probability of each player being assigned to a particular role distribution.

M-Step: The Hungarian Algorithm (Kuhn, 1955) is used to (hard) assign each player to a specific role in each frame. Once all roles have been assigned for that iteration, the role distributions are recomputed.

Termination: The process repeats until convergence.

A.2 Relation to Gumbel-Sinkhorn

In the original template discovery formulation, application of the Hungarian algorithm at each iteration made the strong requirement that a single player be assigned to a single cluster in each frame, that is, $\pi_{s,i} = \pi_{s,j} \forall i, j \in K, \forall s \in S$.

Our method treats each agent in each frame as fully independent; if we examined at the assignment probabilities for a given frame during training, the likelihood would exceed 1. Furthermore, during training, there are no restrictions on the values of each π beyond the standard requirement that $\sum_{k=1}^{K} \pi_k = 1$. Thus at the end of the formation-learning step, there is no requirement that each $\pi_k = \frac{1}{K}$. The requirement that each role be occupied by only a single agent in each frame and that weights are uniform is imposed only at the assignment step.

An approach based on Sinkhorn normalization can be seen as an intermediate between these two paradigms, allowing for soft-assignment of every agent in a frame to every cluster during the learning process.

Thus during training, a player could still contribute to multiple clusters, but the weighting of that contribution is restricted by Sinkhorn normalization which requires the assignment matrix in each frame to be doubly stochastic. Following the notation of Cruz et al. (2017), rows R and columns C are normalized according to

$$
R_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{i,k}}; \qquad C_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=1}^{l} Q_{i,k}} \tag{11}
$$

with the n^{th} iteration defined recursively as

$$
S^n(Q) = \begin{cases} Q & \text{if } n = 0 \\ C(R(S^{n-1}(Q))), & \text{otherwise.} \end{cases} \tag{12}
$$

Similar to our approach, Sinkhorn-based methods employ hard assignment via the Hungarian algorithm only after training completes. The additional benefit of this approach is that the Sinkhorn normalization function is differentiable and thus amenable to gradient-based methods.

However as the our approach is based on expectation maximization, the drawback to Sinkhorn-normalization is that the iterative normalization step is required in each frame at each EM iteration. Thus within the EM framework, it remains cost-prohibitive although gives final assignment results equivalent to both the original and current methods.

A.3 Optimal Template Learning Algorithm

Algorithm 1 Optimal Template Learning

```
Input:
    $\boldsymbol{U} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_N\}$ unordered player positions
    $\boldsymbol{G}^*$ a parent/global template
Output:
    $\boldsymbol{R} = [\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_K]$ player positions ordered by role
    $\boldsymbol{\mathcal{F}}^*$ the learned formation
    $\boldsymbol{\mathcal{T}}^*$ the alignment template

Normalization
normalize the positions in each frame so that the attacking team is going left to right
center-normalize the positional data according to $\boldsymbol{M}_{sn} = \boldsymbol{U}_{sn} - \sum_{n=1}^{N} \boldsymbol{u}_{sn} \quad \forall n \in N$
format $\boldsymbol{M}$ according to $f : \mathbb{R}^{S \times dN} \to \mathbb{R}^{SN \times d}$

Formation Discovery
conduct K-Means clustering for initialization: K-Means$(\boldsymbol{M}, \mu_{init} = [\bar{\boldsymbol{M}}_1, \bar{\boldsymbol{M}}_2, \ldots, \bar{\boldsymbol{M}}_N])$
function EigenvalueResettingGMM
    while lower bound average gain $<$ threshold do
        if $\frac{1}{r} < \frac{\lambda_{n1}}{\lambda_{n2}} < r \quad \forall n \in N$ then
            $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} \leftarrow$ GMM Update
        else
            $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} \leftarrow$ Soft K-Means Update
    return $\boldsymbol{\mathcal{F}}^*$

Template Alignment
function AlignTemplates($\boldsymbol{\mathcal{F}}^*$, $\boldsymbol{G}^*$)
    create cost matrix $C$ s.t. $C_{i,j}$ is the Mahalanobis distance between the $i^{th}$ distribution in $\boldsymbol{\mathcal{F}}^*$
and the $j^{th}$ distribution in $\boldsymbol{G}^*$
    apply Hungarian algorithm to find optimal assignment of $\mathcal{F}_i$ to $\mathcal{G}_j$
    return $\boldsymbol{\mathcal{T}}^*$

Role Assignment
function ApplyAlignment($\boldsymbol{R}$, $\boldsymbol{\mathcal{T}}^*$)
    for $s$ in $S$ do
        create cost matrix $C$ s.t. $C_{i,j}$ is the likelihood the $R_{s,\imath}$ ($i^{th}$-agent in frame $s$), belongs to
the $j^{th}$ distribution of $\boldsymbol{\mathcal{T}}^*$
        apply Hungarian algorithm to find optimal assignment of $R_{s,\imath}$ to $\boldsymbol{\mathcal{T}}^*$
    return $\boldsymbol{R}$
```

A.4 Tree-based Alignment

This algorithm is adapted from Sha et al. (2017). The overall structure is the same with changes to the initializations captured in A.3.

Algorithm 2 Learning process of tree-based alignment

```
Input: $\boldsymbol{R} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_N\}$ unordered player positions
    $\boldsymbol{G}^*$ a parent/global template
    $T = \emptyset, \mathcal{C} = \emptyset$
Output: $T, \mathcal{C}$
function LearnTree($\boldsymbol{R}$)
    for each layer $l$ do
        for each node $n$ do
            learn $\boldsymbol{\mathcal{F}}^{*l}_{n}$, $\boldsymbol{\mathcal{T}}^{*l}_{n}$, $\boldsymbol{R}^{l}_{n}$ from Algorithm A.3 with $\boldsymbol{G}^*$ = $\boldsymbol{\mathcal{T}}^{*l-1}_{n}$ and $\boldsymbol{R} = \boldsymbol{R}^{l-1}_{n}$, the data contained in the parent node
        store $[\boldsymbol{\mathcal{T}}^{*l}_{1}, ..., \boldsymbol{\mathcal{T}}^{*l}_{Z}]$ in $T$
        compute reconstruction loss with Eq. 4
        terminate when stropping criterion is met
        for each node $n$ do
            create $K$ initialization vectors by adding small amounts of noise $\epsilon$ to the cluster means of $\boldsymbol{\mathcal{T}}^{*l}_{n}$
            conduct K-Means on $\boldsymbol{R}^{l}_{n}$ with different $K$
            select cluster set $\boldsymbol{C}^{l}_{n}$ that maximize $E$ partition $\boldsymbol{C}^{l}_{n}$ to child nodes according to $\boldsymbol{C}^{l}_{n}$
        Store $[\boldsymbol{C}^{l}_{\mathbf{1}}, ..., \boldsymbol{C}^{l}_{Z}]$ in $\mathcal{C}$
    return $T, \mathcal{C}$
```

A.5 Run Complexity

Average per-iteration run-complexity for various methods.

Table 1: Algorithm complexity per iteration

ALGORITHM	**COMPLEXITY**
Hungarian EM	$\mathrm{SN}^5\mathrm{d}^2$
K-means	$\mathrm{SN}^2\mathrm{d}$
GMM	$\mathrm{SN}^2\mathrm{d}^2$

A.6 League-Wide Analysis

We ran our formation discovery and alignment algorithm on every team-game-period combination of an entire season. This analysis previously took upwards of 20minutes to process a game under the original approach, but now can be run in less than 10seconds a game.

Some teams consistently operate out of the same formation and therefore the templates and distribution centers are well isolated across the season (e.g. second row, far right). Others play different formations in different matches and therefore the game-to-game templates can vary dramatically (e.g. third row, second from the left; bottom row, second from the left). Our algorithm is able to learn these various templates and align them so that a common, structured representation can be used across matches.

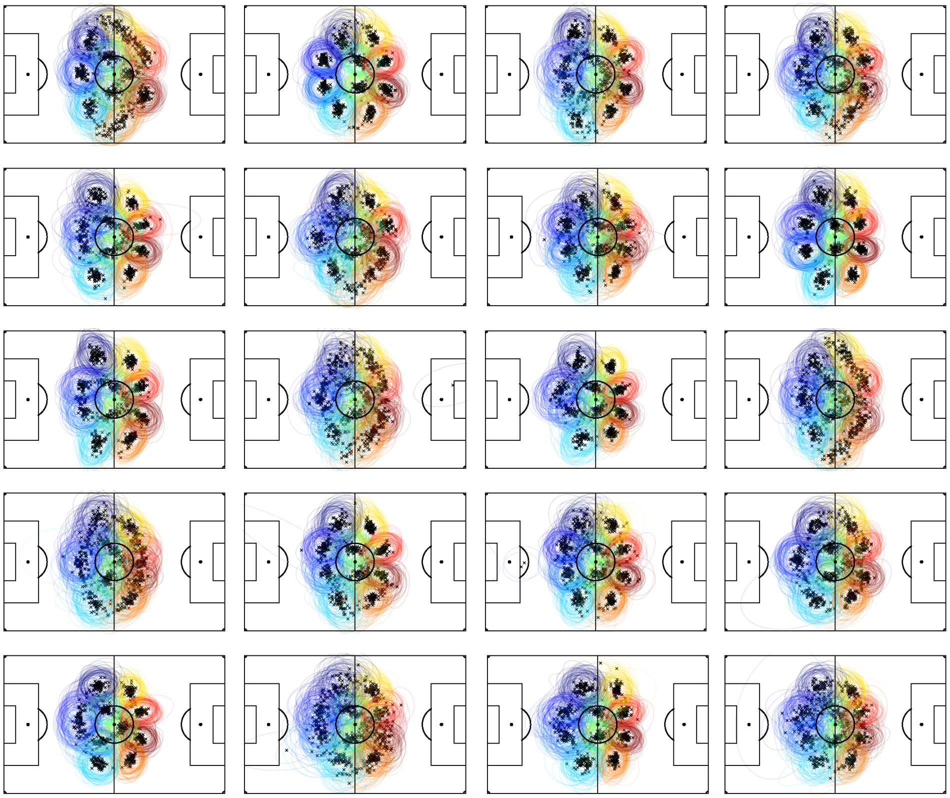

Figure 6: Discovered templates for each team across a season of professional soccer. The global template is learned by selecting data randomly across the season (all teams, all games). Each plot corresponds to a team and a template is learned for each half of every game and aligned to the global template. The centroids of each role-distribution are plotted in black.