Role Discovery in Observed Multi-Agent Systems Over Time through Matrix Factorization

Brian Reily[1], Michael Don[2], John G. Rogers[2], and Christopher Reardon[3]

***Abstract*— Understanding unknown multi-agent systems solely from observations without prior knowledge of the systems' composition or structure is critical for effectively responding to and interacting with them. Whether the unknown system consists of humans, robots, or other entities, the capability to discover the roles that various agents play in the multi-agent system is necessary to fully understand it. While existing work often focuses on predicting future trajectories or behaviors, there has been little research on identifying agents that share roles within a multi-agent system. Discovering shared roles enables a fuller understanding of the system and its future behavior, i.e., agents that share a role could be expected to behave similarly. In this paper, we propose a novel approach for role discovery in an observed multi-agent system. We first present a method to learn a unified temporal representation of the multi-agent system through a temporally weighted approximation of graphs describing relationships between agents at each time step. We then present our main contribution, where we formulate role discovery as a regularized optimization problem with the goal of learning the optimal role assignment based on the unified temporal representation. Our approach learns probabilities that agents play different roles while also discovering the number of distinct roles that exist in the multi-agent system, and is proven to converge to the optimal solution. We also introduce a new role recognition dataset and evaluate on an existing dataset, showing that our approach outperforms existing methods in discovering roles in an observed multi-agent system.**

I. Introduction

Multi-agent systems are ubiquitous in modern life [1]. Whether these agents are robots, software programs, or humans, these systems have complicated structures and internal relationships [2]. Autonomous systems are increasingly tasked with interacting with unknown multi-agent systems for a variety of homeland security, military, and domestic applications, such as automated surveillance systems that monitor crowds of pedestrians for possible threats [3] or social media companies that analyze their networks to identify influencers and information hubs. In order for autonomous systems to interact successfully with unknown multi-agent systems, they must be able to generate an understanding of the internal structure of the system based solely on observations.

*This work was partially supported by ARL CA W911NF-21-2-0021.

[1]Brian Reily is with the Colorado School of Mines and the DEVCOM Army Research Laboratory through ORAU. email: breily@mines.edu

[2]Michael Don and John G. Rogers are with the DEVCOM Army Research Laboratory. email: {michael.l.don2.civ, john.g.rogers59.civ}@mail.mil

[3]Christopher Reardon is with the Department of Computer Science at the University of Denver and the DEVCOM Army Research Laboratory. email: christopher.reardon@du.edu

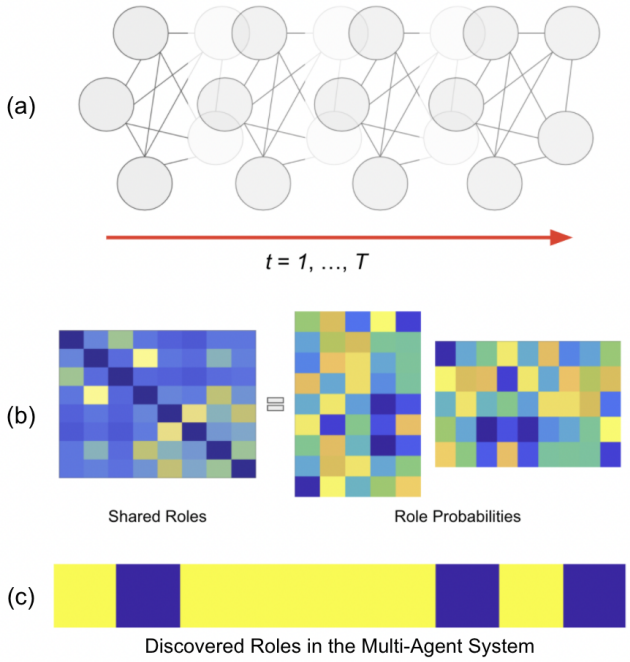


Fig. 1. Our approach discovers roles within a multi-agent system and assigns each agent a role. In (a), a multi-agent system is observed over time, and a unified temporal representation is learned. In (b), a shared role matrix learned from the unified representation is factored into role probability matrices. In (c), the role probability matrix is used to discover roles for each agent and discover the number of roles in the multi-agent system.

A key part of understanding this internal structure is discovering roles within the multi-agent system and identifying which agents play which roles. While recognizing the specific role an agent plays in a multi-agent system requires *a priori* knowledge of the system's structure or purpose, discovering the assignment of agents to roles can be accomplished from observations of the system. Shared roles among agents can provide valuable insight not only into a system's structure, but into the future behavior of the multi-agent system.

While analysis of unknown multi-agent systems has been extensively researched for trajectory and behavior prediction [4], [5], role discovery has seen limited study. Multi-agent systems are often assumed to have a known structure, with roles such as communication hubs [6] or leaders [7] being defined before the system is interacted with. When roles in a multi-agent system are unknown, they have been indirectly addressed as a byproduct of computer vision-based team or group analysis [8], [9], or framed as a node classification

problem in very large static graphs [10]. While recent work has worked to cluster roles and then select clusters [11], the problem of discovering role assignments and the number of roles being played in a unified formulation for unknown dynamic multi-agent systems is largely unaddressed.

In this paper, we introduce a novel unsupervised learning method to discover shared latent roles in a multi-agent system based on observations of it over time. We first present a novel unified temporal representation of a multi-agent system, incorporating the positions and observed attributes of agents at multiple time steps into a temporally weighted approximation. Then, we introduce a role discovery framework based on two regularized optimization formulations. Our approach first learns shared role indicators that minimize the cost of agents sharing a role. We then factor this shared role indicator matrix to discover the probability that an agent is assigned to a specific role. By processing this role probability matrix, we discover which agents actually share a role and also discover the number of roles present in the multi-agent system. We prove that our approach converges to the optimal solution, and show through experimental evaluation on both an existing role recognition dataset and a new role recognition dataset which we introduce that our approach is able to accurately discover roles, outperforming existing approaches.

This paper contains three important contributions.

- First, we present a novel approach for role discovery in an unknown multi-agent system, based on a newly introduced unified temporal representation. Our approach, based on a multi-step regularized optimization framework, is able to accurately identify which agents share which roles. Our approach is also able to simultaneously discover the number of roles present in a multi-agent system, eliminating the need to know this *a priori* about an unknown system.
- Second, we theoretically prove that our approach learns the optimal role assignment and demonstrate its effectiveness through experimental results.
- Finally, we introduce a new dataset for role recognition. This dataset consists of variably sized dynamic multi-agent systems, with observed position changes, communication behaviors, and ground truth role labeling.

II. Related Work

A. Representing Multi-Agent Systems

Multi-agent systems have been modeled in a wide variety of ways. As many of these use models based on known agent types, hierarchies, available behaviors, and other information known about the structure of the multi-agent system [2], they are difficult to apply to unknown multi-agent systems where the only information available is from observations. In this case, graphs become the most used representation, where vertices represent agents and edges represent a relationship between them [12]. Edges have been used to represent a control relationship [13], [14] or the immediate neighborhood of an agent [15], [16]. Edges have also been used to represent communications between agents [6] or the probability of agents staying together as they move over time [17], [18]. Recent work has seen the learning of unified representations from arbitrary input relationships, allowing edges to approximate multiple forms of relationships between agents with a single value [19].

B. Role Discovery in Multi-Agent Systems

In multi-agent systems, role discovery has been approached from the perspectives of computer vision and graph theory. In computer vision-based approaches, social roles have been analyzed from a first person view perspective, where the multi-agent system is considered to be the individuals in the current frame [20], [21]. This has also been performed from videos with wider vantage points, such as surveillance videos [3] or aerial videos [5]. Much of this work has used role assignment as a byproduct of other applications, such as behavior prediction [4] or localization [22]. Additional work has been done to match observed social interactions to known examples [23] or to identify groups which are travelling together in crowded scenes [9].

In graph theory, role discovery has mainly focused on structural roles in large graphs, with the aim of identifying nodes that connect clusters, act as outliers, serve as community centers, and more [10], [24], [25]. Recent work has extended this to 'small' networks, where structure is less influential [26], with the aim of partitioning small social networks [27] or identifying coalitions [28]. While classic graph cut methods such as modularity [29]–[31] have been used to identify communities, new ones have been proposed based on the roles that nodes play. These include extending modularity to identify hubs and outliers [32], detecting communities [33], or using neural networks to classify nodes [34].

While there has been significant research in this area, the problem of discovering role assignments in a multi-agent system over time while simultaneously discovering the number of roles present is an open problem.

III. Our Proposed Approach

Notation. In this paper, matrices are denoted using boldface uppercase letters or italicized uppercase letters, and vectors using boldface lowercase letters. For a matrix $\mathbf{M} = \{m_{ij}\} \in \mathbb{R}^{p \times q}$, we refer to its i-th row as $\mathbf{m}^i$ and its j-th column as $\mathbf{m}_j$, and the element in the i-th row and the j-th column as m_{ij}. For a variable that is representative of a certain point in time, the time is denoted with a parenthetical superscript, as in $\mathbf{x}_i^{(t)}$. A vector of ones is denoted as $\mathbf{1}_n \in \mathbb{R}^n$ and the identity matrix is denoted as $\mathbf{I}_n \in \mathbb{R}^{n \times n}$, or simply $\mathbf{I}$ if the dimensions are clear from context.

A. Preliminaries

We consider a multi-agent system consisting of n agents. Each agent has a known position in space, denoted in $\mathbf{X}^{(t)} \in \mathbb{R}^{d \times n}$, with $\mathbf{x}_i^{(t)} \in \mathbb{R}^d$ representing the d-dimensional position of the i-th agent at time t. We also consider attributes in $\mathbf{Z}^{(t)}$ that describe each agent, with $\mathbf{z}_i^{(t)}$ representing the

i-th agent at time t. An attribute describing an agent could be a visual representation, a behavior, or other possibilities.

Our proposed approach constructs a unified temporal representation of the multi-agent system by approximating the observed positions and attributes over time. We then identify a minimum cost shared role matrix, which is then factored to learn the role assignment probabilities. In doing so, our approach discovers a role assignment for each agent, while also discovering the number of roles that exist in the multi-agent system.

TABLE I
LIST OF NOTATION USED IN OUR PROPOSED ROLE DISCOVERY APPROACH.

Variable	Definition
$\mathbf{x}_i^{(t)}$	Position of agent i at time t.
$\mathbf{z}_i^{(t)}$	Attribute of agent i at time t.
$a_{ij}^{(t)}$	Relationship of agents i and j at time t.
$\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$	Representation of multi-agent system at time t.
$\mathcal{A} \in \mathbb{R}^{n \times n}$	Unified temporal representation of a multi-agent system.
$\mathbf{U} \in \mathbb{R}^{n \times n}$	Shared role indicator matrix.
u_{ij}	Probability that agents i and j share a role.
$\mathbf{W} \in \mathbb{R}^{n \times m}$	Role probability matrix.
w_{ij}	Probability that agent i plays role j.
$\mathbf{r} \in \mathbb{R}^{n}$	Role assignment vector.
$\hat{m}$	Identified number of roles in a multi-agent system.
$\tilde{m}$	Ground truth number of roles in a multi-agent system.

B. Representing the Multi-Agent System

Given the observed positions $\mathbf{X}$ and observed attributes $\mathbf{Z}$ over time, we construct a unified temporal representation of the multi-agent system. We construct a unified representation $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$ at each time step t, where $a_{ij}^{(t)}$ represents the separation relationship between the i-th and j-th agent at time t. Then, these representations at each time step are combined into a unified temporal representation $\mathcal{A}$, representing the entire multi-agent system over $t = 1, \ldots, T$.

First, we incorporate the spatial relationships between agents. Rather than considering the distance between individual agents, we consider the relative distance in the multi-agent system. Intuitively, agents that are both centrally located would play similar roles in the system, just as agents on the exterior would play similar roles. This also incorporates agents that are spatially near each other, as this would result in a similar distance from the centroid. We calculate the centroid $\mathbf{c}^{(t)}$ of the multi-agent system at time t and define the separation between the i-th and j-th agents to be the absolute difference in their distances from this centroid:

$$\mathbf{c}^{(t)} = \frac{1}{n} \sum_{i=0}^{n} \mathbf{x}_i^{(t)} \tag{1}$$

$$d_i^{(t)} = \|\mathbf{x}_i^{(t)} - \mathbf{c}^{(t)}\|_2 \tag{2}$$

$$a_{ij}^{(t)} = \text{abs}(d_i^{(t)} - d_j^{(t)}) \tag{3}$$

We next incorporate the observed attributes of each agent. If no additional attributes are available to describe an agent, then Eq. (3) is solely used to determine the similarity $a_{ij}^{(t)}$. An observed attribute of an agent could represent a variety of different features, such as a histogram representing the color appearance of an agent, or a vector representing its communication with other agents. We again utilize the distance between two attribute representations, adding this to the spatial similarity measure:

$$a_{ij}^{(t)} = \alpha \text{abs}(d_i^{(t)} - d_j^{(t)}) + (1 - \alpha)\|\mathbf{z}_i^{(t)} - \mathbf{z}_j^{(t)}\|_2 \tag{4}$$

Here, we also use hyperparameter α to balance the relative importance between a position $\mathbf{x}_i^{(t)}$ and an attribute $\mathbf{z}_i^{(t)}$, which can assign equal weight or be expert-defined based on the multi-agent system being observed. We note that this part of our proposed approach could also be modified with distance functions that better fit the specific $\mathbf{x}$ and $\mathbf{z}$ forms of different multi-agent system observations.

We now have a unified representation matrix $\mathbf{A}^{(t)}$ describing the relationships between agents at time point t. To integrate this into a unified temporal representation describing the multi-agent system over time, we learn an approximation of all available time steps, with a decay factor to value older observations less than newer observations. This unified temporal representation is defined as $\mathcal{A}$, indicating an overall representation representing the entire time period and without the time step superscript:

$$\min_{\mathcal{A}} \sum_{t=0}^{T} \gamma^{(t)} \|\mathcal{A} - \mathbf{A}^{(t)}\|_F^2 \tag{5}$$

Here, γ is a weighting parameter from a sequence such that $\gamma^{(t+1)} > \gamma^{(t)}) > \gamma^{(t-1)}$. We now have a unified representation of the multi-agent system in $\mathcal{A}$, which incorporates observed relationship representation matrices $\mathbf{A}^{(t)}$ from previous points in time.

C. Discovering Multi-Agent Roles

In a multi-agent system, the number of possible roles m that agents can play is bounded such that $1 < m < n$. If $m = 1$, then all agents share the same role, which does not offer useful insight into the structure of the multi-agent system. Similarly, if $m = n$, then each agent plays a role by itself, which is again not informative about the observed system. In this section, we propose to learn a probabilistic role assignment where we identify the probability that the i-th agent plays the j-th role, enabling the discovery of each agent's role assignment and the number of roles.

First, we recognize that we want to avoid a result where $m = 1$, where each agent is assigned to a role by itself.

Algorithm 1: Role Discovery Algorithm

Input : $\mathbf{X}^{(t)}, t = 1, \ldots, T$: Agent positions for T points in time.
$\mathbf{Z}^{(t)}, t = 1, \ldots, T$: Agent attributes for T points in time (optional).

Output: $\mathbf{r}$: Role assignment vector.
$\hat{m}$: Discovered number of roles present in the multi-agent system.

1: Construct unified representations $\mathbf{A}^{(t)}$ at each time point through Eq. (4) (if $\mathbf{Z}^{(t)}$ is available) or Eq. (3) (if $\mathbf{Z}^{(t)}$ is not available).
2: Learn the optimal unified temporal representation $\mathcal{A}$ to incorporate representations of each time step through Eq. (5).
3: Update $\mathcal{A}$ according to Eq. (6).
4: Initialize ρ such that $1 < \rho < 2$, μ_1 such that $\mu_1 > 0$, and μ_2 such that $\mu_2 > 0$.
5: Learn the optimal shared role indicator matrix $\mathbf{U}$ by minimizing Eq. (7), through the solution described in Section III-D.1.
6: Factor $\mathbf{U}$ to learn the optimal role probability matrix $\mathbf{W}$ by minimizing Eq. (8), through the solution described in Section III-D.2.
7: Discover each agent's role assignment by defining the vector $\mathbf{r}$ from the values in $\mathbf{W}$ through Eq. (9).
8: Discover the number of roles present in the system $\hat{m}$ through Eq. (10).
9: **return** $\mathbf{r}$, $\hat{m}$.

In order to avoid this, we modify our unified representation $\mathcal{A}$ such that each agent is as similar to itself as its most dissimilar relationship:

$$a_{ii} = \underset{j,\ \text{s.t.} j \neq i}{\operatorname{argmax}}\, a_{ij} \tag{6}$$

We now identify shared roles by finding a minimum selection of agent relationships. We define a shared role indicator matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$, where u_{ij} represents the probability that the i-th agent and the j-th agent share a role. We enforce these values to be probabilities by constraining this matrix to be non-negative, with each row and column summing to 1. Further, we also enforce this matrix to be symmetric, so the $u_{ij} = u_{ji}$, meaning that the probability that agent i shares a role with agent j is equal to the probability that agent j shares a role with agent i. We formulate this as a constrained minimization problem:

$$\begin{aligned} &\min_{\mathbf{U}} \ \|\mathbf{U} \odot \mathcal{A}\|_1 \\ &\text{s.t. } \mathbf{U} \geq 0, \mathbf{U} = \mathbf{U}^\top, \mathbf{U}\mathbf{1}_n = \mathbf{1}_n. \end{aligned} \tag{7}$$

Here, $\mathbf{U} \odot \mathcal{A}$ represents the Hadamard product of $\mathbf{U}$ and $\mathcal{A}$, or element-wise multiplication, where $(\mathbf{U} \odot \mathcal{A})_{ij} = u_{ij} a_{ij}$. $\|\cdot\|_1$ represents the ℓ_1-norm of a matrix, or the sum of the absolute values of all elements.

In order to discover the number of roles present in the system, we then factor this shared role indicator matrix into two $n \times m$ matrices, where $m = n - 1$. We introduce role probability matrices $\mathbf{W} \in \mathbb{R}^{n \times m}$ and $\mathbf{V}^\top \in \mathbb{R}^{n \times m}$. While we loosely enforce that $\mathbf{W} = \mathbf{V}^\top$, we further constrain only $\mathbf{W}$, with each row this matrix representing the probability that an agent belongs to each of m possible roles. While this may seem like a repetition of the values indicated in $\mathbf{U}$, this will allow the discovery of the actual number of roles present. We learn $\mathbf{W}$ and $\mathbf{V}$ through a matrix factorization-based approach by approximating the shared role indicator matrix $\mathbf{U}$:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{V}} \ \|\mathbf{W}\mathbf{V} - \mathbf{U}\|_F^2 + \beta \|\mathbf{W} - \mathbf{V}^\top\|_F^2 \\ &\text{s.t. } \mathbf{W} \geq 0, \mathbf{W}\mathbf{1}_m = \mathbf{1}_n. \end{aligned} \tag{8}$$

Here, β controls the importance of ensuring this factorization is symmetric, i.e., that $\mathbf{W} = \mathbf{V}^\top$. Through the constraints that $\mathbf{W} \geq 0$ and $\mathbf{W}\mathbf{1}_m = \mathbf{1}_n$, we ensure each row in $\mathbf{W}$ again represents a probability by constraining values to be non-negative and requiring rows to sum to 1.

After learning the optimal value of $\mathbf{W}$, we are now able to discover each agent's role assignment and the actual number of roles present in the multi-agent system. We define the final role assignment vector $\mathbf{r} \in \mathbb{R}^n$, where r_i indicates the role that the i-th agent is assigned to. We do this by identifying the role with the maximum probability in each agent's row in $\mathbf{W}$:

$$\mathbf{r} = [r_i] = j, \ \text{ such that} \underset{j}{\operatorname{argmax}}\, w_{ij} \tag{9}$$

The final discovered number of roles present in the multi-agent system is denoted as $\hat{m}$, which is defined by:

$$\hat{m} = \text{unique}(\mathbf{r}) \tag{10}$$

where unique$(\cdot)$ is a function returning the number of unique elements in a vector.

D. Solution Algorithm

In this section, we present iterative algorithms to solve for the optimal $\mathbf{U}$ in Eq. (7) and $\mathbf{W}$ in Eq. (8). We then prove that both algorithms converge to the optimal solution for each formulation.

1) Solution for Eq. (7): In this section, we describe a solution to solve for the optimal shared role indicator matrix $\mathbf{U}$. To do so, we introduce a new constraint $\mathbf{U} = \hat{\mathbf{U}}$. We present the solution for $\mathbf{U}$, as the solution for $\hat{\mathbf{U}}$ is similar but less complex and thus omitted for brevity.

Step 1.1: We first incorporate $\mathbf{U} = \hat{\mathbf{U}}$ and constraints from Eq. (7) into the objective function as penalty terms, introducing parameters μ_1, Λ_1, Λ_2, and λ_1:

$$\begin{aligned} &\min_{\mathbf{U},\hat{\mathbf{U}}} \ \|\mathbf{U} \odot \mathcal{A}\|_1 + \frac{\mu_1}{2}\|\hat{\mathbf{U}} - \mathbf{U} + \frac{1}{\mu_1}\Lambda_1\|_F^2 + \\ &\qquad \frac{\mu_1}{2}\|\mathbf{U}\mathbf{1}_n - \mathbf{1}_n + \frac{1}{\mu_1}\lambda_1\|_2^2 + \frac{\mu_1}{2}\|\mathbf{U}^\top - \hat{\mathbf{U}} + \frac{1}{\mu_1}\Lambda_2\|_F^2 \\ &\text{s.t. } \mathbf{U} \geq 0. \end{aligned} \tag{11}$$

Step 1.2: Next, we rewrite the Hadamard product term in Eq. (7) as a trace of a matrix product:

$$\|\mathbf{U} \odot \mathcal{A}\|_1 \longrightarrow \text{trace}(\mathbf{U}^\top \mathcal{A}) \tag{12}$$

Step 1.3: We now take the derivative of the updated Eq. (11) w.r.t. $\mathbf{U}$, and set this equal to 0:

$$\begin{aligned} 0 =& \mathcal{A} + 2\mu_1\mathbf{U} + \mu_1\mathbf{U}\mathbf{1}_n\mathbf{1}_n^\top - \mu_1\hat{\mathbf{U}} - \mu_1\hat{\mathbf{U}}^\top - \\ & - \mu_1\mathbf{1}_n\mathbf{1}_n^\top + \lambda_1\mathbf{1}_n^\top - \Lambda_1 + \Lambda_2^\top \end{aligned} \tag{13}$$

$$\begin{aligned} \mathbf{U} =& \big(\mu_1(\hat{\mathbf{U}} + \hat{\mathbf{U}}^\top + \mathbf{1}_n\mathbf{1}_n^\top) - \lambda_1\mathbf{1}_n^\top + \Lambda_1 - \Lambda_2^\top - \mathcal{A}\big) \\ & \big(2\mu_1\mathbf{I} + \mu_1\mathbf{1}_n\mathbf{1}_n^\top\big)^{-1} \end{aligned} \tag{14}$$

Step 1.4: We know enforce the constraint that $\mathbf{U}$ contain only non-negative terms:

$$\mathbf{U} = \max(\mathbf{U}, 0) \tag{15}$$

Step 1.5: We lastly update the penalty parameters μ, λ_1, and Λ:

$$\lambda_1 = \lambda_1 + \mu_1(\mathbf{U}\mathbf{1}_n - \mathbf{1}_n) \tag{16}$$

$$\Lambda_1 = \Lambda_1 + \mu_1(\hat{\mathbf{U}} - \mathbf{U}) \tag{17}$$

$$\Lambda_2 = \Lambda_2 + \mu_1(\mathbf{U}^\top - \hat{\mathbf{U}}) \tag{18}$$

$$\mu_1 = \rho\mu_1 \tag{19}$$

where ρ is chosen such that $1 < \rho < 2$.

Step 1.6: We then repeat **Steps 1.3** to **1.5** until convergence to find the optimal solution $\mathbf{U}$ for Eq. (7), while performing similar steps by utilizing the derivative of Eq. (11) w.r.t. $\hat{\mathbf{U}}$.

2) Solution for Eq. (8): In this section, we describe a solution to solve for the optimal role probability matrix $\mathbf{W}$. We present only the solution algorithm for $\mathbf{W}$, as the solution for $\mathbf{V}$ is omitted for brevity due to its similarity.

Step 2.1: We again incorporate constraints in Eq. (8) into the objective function as penalty terms:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{V}} \ & \|\mathbf{W}\mathbf{V} - \mathbf{U}\|_F^2 + \beta\|\mathbf{W} - \mathbf{V}^\top\|_F^2 + \\ & \frac{\mu_2}{2}\|\mathbf{W}\mathbf{1}_m - \mathbf{1}_n + \frac{1}{\mu_2}\lambda_2\|_2^2 \\ \text{s.t. } & \mathbf{W} \geq 0. \end{aligned} \tag{20}$$

Step 2.2: We now take the derivative of Eq. (20) w.r.t. $\mathbf{W}$, set this equal to 0, and solve for the update to $\mathbf{W}$:

$$\begin{aligned} 0 =& 2\mathbf{W}\mathbf{V}\mathbf{V}^\top - 2\mathbf{U}\mathbf{V}^\top + 2\beta\mathbf{W} - 2\beta\mathbf{V}^\top + \\ & + \mu_2\mathbf{W}\mathbf{1}_m\mathbf{1}_m^\top - \mu_2\mathbf{1}_n\mathbf{1}_m^\top + \lambda_2\mathbf{1}_m^\top \end{aligned} \tag{21}$$

$$\begin{aligned} \mathbf{W} =& (2\mathbf{U}\mathbf{V}^\top + 2\beta\mathbf{V}^\top + \mu_2\mathbf{1}_n\mathbf{1}_m^\top - \lambda_2\mathbf{1}_m^\top) * \\ & (2\mathbf{V}\mathbf{V}^\top + 2\beta\mathbf{I}_m + \mu_2\mathbf{1}_m\mathbf{1}_m^\top)^{-1} \end{aligned} \tag{22}$$

Step 2.3: We update $\mathbf{V}$ similarly to **Step 2.2**, by taking the derivative of Eq. (11) w.r.t. $\mathbf{V}$, setting it equal to 0, and solving for the update.

Step 2.4: We enforce the constraint that all values in $\mathbf{W}$ be non-negative:

$$\mathbf{W} = \max(\mathbf{W}, 0) \tag{23}$$

Step 2.5: Finally, we update penalty parameters μ and λ_2:

$$\lambda_2 = \lambda_2 + \mu_2(\mathbf{W}\mathbf{1}_m - \mathbf{1}_n) \tag{24}$$

$$\mu_2 = \rho\mu_2 \tag{25}$$

Step 2.6: To solve for $\mathbf{W}$ that minimize Eq. (8), we repeat **Steps 2.2** to **2.5** until convergence, while also updating $\mathbf{V}$ at each iteration.

3) Convergence of Solutions: : We now prove that both the solutions presented in Section III-D.1 and Section III-D.2 both converge to the respective optimal solutions.

Lemma 1: Constrained optimization problems of the form

$$\min f(\mathbf{X}) \text{ s.t. } h(\mathbf{X}) = 0 \tag{26}$$

are able to be transformed into regularized optimization problems through the addition of penalty terms:

$$\min f(\mathbf{X}) + \frac{\mu}{2}\|h(\mathbf{X}) + \frac{1}{\mu}\Lambda\|_F^2 \tag{27}$$

Problems transformed as such via the general Augmented Lagrangian Method (ALM) [35] reduce the objective value of the function iteratively subject to the constraint that $0 < \mu^k < \mu^{k+1}$ is satisfied at each iteration k.

Theorem 1: Eq. (7) and Eq. (8) both converge to the respective optimal solutions $\mathbf{U}$ and $\mathbf{W}$.

Proof: As these solution algorithms are similar, we will discuss only the convergence of the solution for Eq. (7). Eq. (8) converges for the same conditions as Eq. (7). We first note that the solution for Eq. (7) follows the ALM transformation from the form in Eq. (26) to the form in Eq. (27), where the constraints $\mathbf{U} = \mathbf{U}^\top$ and $\mathbf{U}\mathbf{1}_n = \mathbf{1}_n$ are converted to regularization terms (e.g., $h_1(\mathbf{U}) = \mathbf{U}\mathbf{1}_n - \mathbf{1}_n$). Thus, the constraints in our formulation are converted into penalty terms.

Next, we note that μ_1 is initialized such that $\mu_1^k > 0$ for $k = 0$, where k is the iteration. Therefore, at $k = 0$, Lemma 1 holds.

We now consider two possibilities for the value of μ_1^{k+1}. First, we consider if $\mu_1^{k+1} < \mu_1^k$. For this to be the case, then $\mu_1^{k+1}/\mu_1^k < 1$. As we know from rearrangement of Eq. (19), $\mu_1^{k+1}/\mu_1^k = \rho$. As ρ is defined such that $1 < \rho < 2$ in **Line 4** of Algorithm 1, this case cannot occur.

Next, we consider the case where $\mu_1^{k+1} = \mu_1^k$ for some iteration k. For this case to occur, $\mu_1^{k+1}/\mu_1^k = 1 = \rho$. As ρ is defined such that $\rho > 1$ in **Line 4** of Algorithm 1, this case can also not occur.

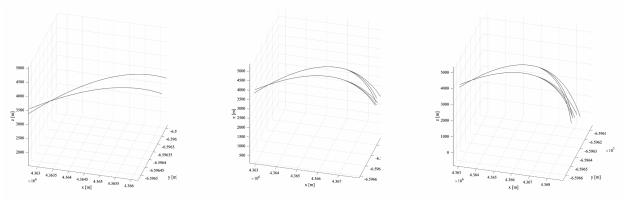
As the case that $\mu_1^{k+1} = \mu_1^k$ cannot occur and the case that $\mu_1^{k+1} < \mu_1^k$ cannot occur, then the case must be that $\mu_1^{k+1} > \mu_1^k$. Given this, for any iteration k Lemma 1 holds and the solution described for Eq. (7) converges to the optimal solution. ■

IV. Experimental Results

A. Experimental Setup

We evaluate our proposed approach on two datasets.

(a) UCLA Aerial Video



(b) Swarm Red Team

Fig. 2. Example data instances from each dataset. Figure 2(a) shows a frame from the UCLA Aerial Video dataset. The dataset provides annotations for each individual's position and role within the scene. Figure 2(b) shows a sample trajectory from our introduced Swarm Red Team dataset, consisting of two hubs deploying four auxiliaries. This dataset provides position data and communication history for each agent at each time step.

- *UCLA Aerial Video* (UCLA) [5]: This dataset[1] consists of drone recordings of interaction among groups of multiple humans, with 12 different events and 18 different roles. An example event would be *Group Tour*, with roles in this scene being *Guide* and *Tourist*. For this dataset, $\mathbf{x} \in \mathbb{R}^2$ corresponds to the human's location in the image and $\mathbf{z}$ describes the visual attributes of the human.
- *Swarm Red Team* (SRT): We introduce this dataset for the purpose of identifying roles in an unknown, observed multi-robot swarm. This simulated dataset consists of ballistic-like trajectory data for multiple robots as they move, with some robots acting as *hubs* and others as *auxiliaries*. The dataset includes 100 instances each for 5 different combinations of hubs and auxiliaries. For this dataset, $\mathbf{x} \in \mathbb{R}^3$ corresponds to the robot's position in space, with latitude, longitude, and altitude recorded in meters. $\mathbf{z}$ describes the communication behavior of the robot. This communication history follows a TDMA protocol, where agents performing different roles broadcast different message content. $\mathbf{z}_i^{(t)}$ would be a binary vector describing this history up to time t, where $z_{ij} \in \{0, 1\}$ indicates whether or not the i-th agent was broadcasting at the j-th point in time. We have made this dataset publicly available[2].

[1]Available at `http://www.stat.ucla.edu/~tianmin.shu/AerialVideo/AerialVideo.html`.

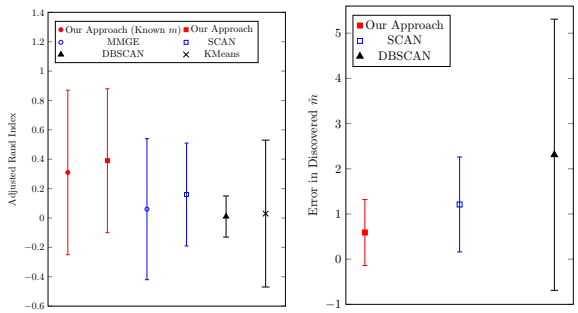
[2]Available at `http://brianreily.com/links/role_discovery_dataset`.

We compare to several other methods, including role recognition methods and clustering approaches, as well an alternate version of our proposed approach.

- *Multimodal Graph Embedding* (MMGE) [12]: We first evaluate against MMGE, a team discovery approach for multi-robot systems. This method embeds multiple graphs describing a single multi-robot system into a unified vector representation for each robot, which can then be clustered into teams. We consider the discovered teams as equivalent to assigned roles for the purposes of evaluation. As this method is capable of incorporating multiple graphs, we use the unified representations $\mathbf{A}^{(t)}$ at each time step as the input graphs. This approach is unable to discover the number of roles, so m is set to the ground truth number of roles for each data instance.
- *Structural Clustering Algorithm for Networks* (SCAN) [32]: We next evaluate against SCAN, a structural role clustering approach for graphs. The SCAN method partitions a graph using a structural similarity measure, identifying clusters, hubs, and outliers. We consider hubs, outliers, and each identified cluster as equivalent to various assigned roles. As SCAN is not designed for a fully connected graph such as our unified temporal representation $\mathcal{A}$, we set below-average weight edges to 0: $a_{ij} = 0$ if $a_{ij} < \bar{\mathcal{A}}$.
- *DBSCAN* [36]: We then evaluate against a state-of-the-art clustering approach in DBSCAN, which we use to cluster $\mathcal{A}$ into roles, while also discovering the number of clusters/roles present.
- *KMeans* [37]: We also evaluate against the KMeans approach, a standard clustering algorithm. As with DBSCAN, the input given is the unified temporal representation $\mathcal{A}$. This approach also is unable to discover the number of roles, so m is set to the ground truth number of roles for each data instance.
- *Our Approach*: Finally, we evaluate the performance of our proposed approach. We also evaluate an alternate version with a *known* value of m, in order to compare more directly to approaches like MMGE or KMeans which cannot identify $\hat{m}$ by themselves.

We evaluate the performance of each method based on the following metrics:

- *Role Assignment*: As a quantitative metric to determine the accuracy of the role assignments in $\mathbf{r}$, we utilize the Adjusted Rand Index [38]. The typical Rand Index quantifies the similarity between two labelings, where a score of 1 indicates the assignments completely match and 0 indicating they do not agree on any element. The Adjusted Rand Index includes an additional factor to adjust for the possibility of chance in the assigned labels, and so values are not restricted to the range of the original method. Higher values indicate a labeling that is closer to ground truth.
- *Error in Discovered* $\hat{m}$: Each instance in both datasets has a ground truth number of roles $\tilde{m}$. As our approach is able to identify the number of roles $\hat{m}$ in a multi-agent



(a) Overall Adjusted Rand Index (b) Overall $\hat{m}$ Error

Fig. 3. Overall metrics for compared approaches on the UCLA Aerial Video dataset. Higher Adjusted Rand Index scores indicate role assignments closer to the ground truth. Lower values for $\hat{m}$ error indicate more accurate identification of the number of roles in a multi-agent system. Markers show mean results, with 1 standard deviation above and below indicated with a bar.

system, we evaluate the error between the identified number of roles versus the ground truth number. For this metric, we use only the approaches which are able to identify a number of labels by themselves (our proposed approach, SCAN, and DBSCAN). We define this metric as $\text{abs}(\tilde{m} - \hat{m})$.

For both metrics, we report the mean of each approach's performance, as well as 1 standard deviation above and below.

B. Results on UCLA Aerial Video

We first evaluate our approach and the comparison approaches on the UCLA Aerial Video dataset. For this dataset, we used events which had more than 1 role being played. As the number of events is large and the number of data instances for each event type is small, we report combined results across the entire dataset, shown in Figure 3.

First, we compare the Adjusted Rand Index scores for all evaluated approaches in Figure 3(a). We can observe that our proposed approach, both with and without a known number of roles m, is able to best discover roles for the individuals in each scene. The version of our approach that must also identify a number of roles $\hat{m}$ actually outperforms the version where m is known *a priori*, suggesting that not restricting the matrix factorization portion of our approach to specific dimensions allows it to more accurately recover connections between agents. Of the compared approaches, SCAN performs the best, providing somewhat informative role assignments. The other compared approaches result in scores near 0, indicating that team discovery and clustering does not translate well to the problem of role discovery.

Next, we compare the ability of our approach, SCAN, and DBSCAN to discover the number of roles in a multi-agent system, seen in Figure 3(b). Here, our proposed approach again performs the best, showing with an error value below 1 that it is able to very accurately map relationships among agents to the number of roles present in the multi-agent system. SCAN again performs the best of the compared approaches. DBSCAN performs the worst, while also being

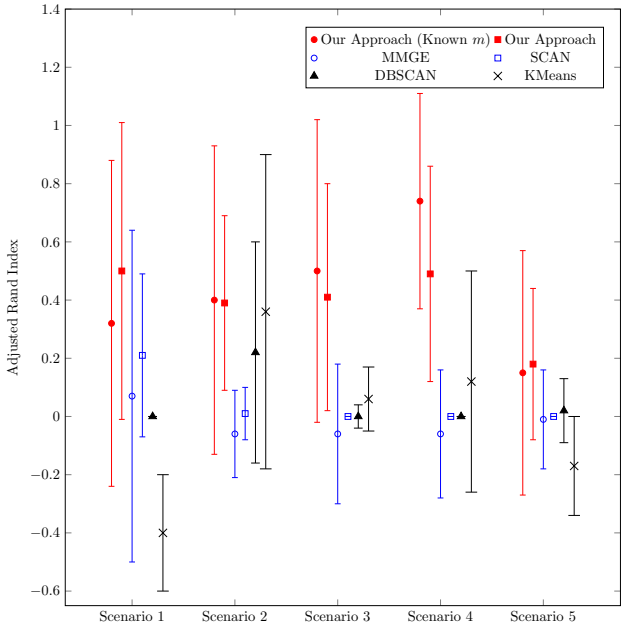


Fig. 4. Results of compared approaches on the problem of role assignment for the Swarm Red Team dataset, scored by the Adjusted Rand Index. Higher scores indicate role assignments closer to the ground truth labelings. Markers show mean results, with 1 standard deviation above and below indicated with a bar.

the most inconsistent with a very high standard deviation in its predicted $\hat{m}$.

C. Results on Swarm Red Team

We next evaluate on the Swarm Red Team dataset, which consists of 5 different scenarios with varying numbers of hubs and auxiliaries. These are specified as:

- *Scenario 1*: 2 hub agents and 2 auxiliary agents.
- *Scenario 2*: 2 hubs and 4 auxiliaries.
- *Scenario 3*: 3 hubs and 2 auxiliaries.
- *Scenario 4*: 3 hubs and 3 auxiliaries.
- *Scenario 5*: 4 hubs and 4 auxiliaries.

These scenarios allow evaluation across a range of multi-agent system sizes and a range of hub-to-auxiliary ratios. Evaluation is done on a portion of the agents' trajectories.

We first report quantitative results on the problem of discovering role assignments, with Adjusted Rand Index scores shown in Figure 4 and broken down by scenario. We can see that there exists significant performance variation across all scenarios. Our approach, both with and without a known m, performs consistently well. There is little variation between the performance of our approach with a set m and without, with both forms performing similarly on Scenarios 2, 3, and 5 and splitting Scenarios 1 and 4. This indicates that our described approach is a capable role assignment method no matter the amount of prior information known about the multi-agent system. In all cases, our approaches generate the best results, often exceeding 1 standard deviation above the mean performance of other approaches. We also observe that the existing team and vertex role discovery approaches MMGE and SCAN only provide useful role assignments

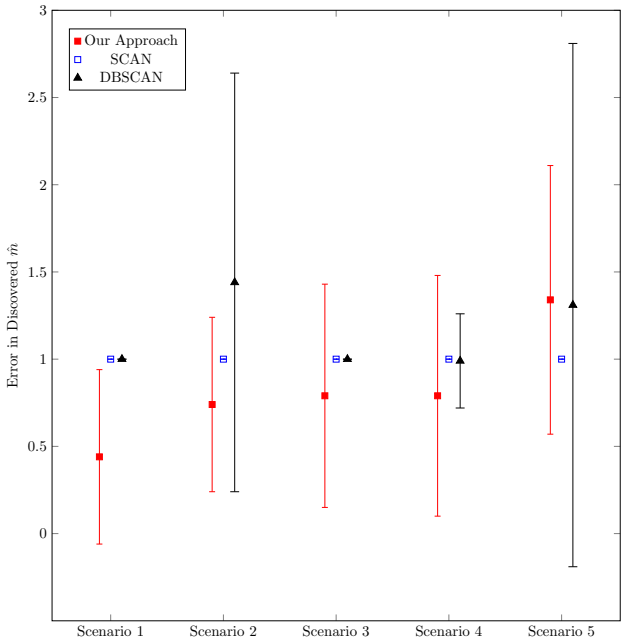
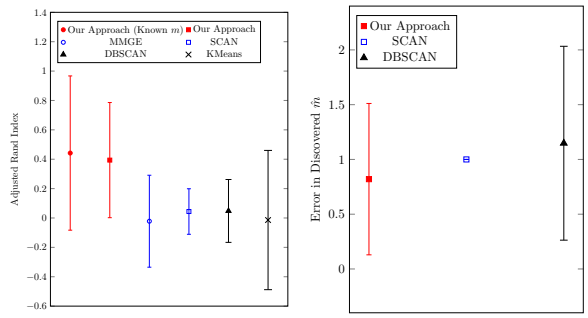


Fig. 5. Results of compared approaches based on the error in the discovered $\hat{m}$ (number of roles identified by the approach) versus $\tilde{m}$ (the ground truth number of roles) for the Swarm Red Team dataset. Lower scores indicate an identified number of roles closer to the ground truth. Markers show mean results, with 1 standard deviation above and below indicated with a bar.

for Scenario 1, the smallest scenario. As the number of agents increases, their performance consistently scores an Adjusted Rand Index of 0, indicating that their original related purposes (multi-robot team identification for MMGE and graph vertex role assignment for SCAN) do not translate well to multi-agent role assignment. While the existing clustering approaches DBSCAN and KMeans do outperform those methods, they significantly underperform compared to our proposed role discovery method.

These observations extend to the performance of these approaches across the entire dataset, as reported in Figure 6(a). Overall, our proposed approach performs well, even when it must also discover the number of roles present. The compared approaches all perform similarly overall to their specific performance on individual scenarios, with poor results near an Adjusted Rand Index of 0.

We next report quantitative results on the error in the discovered $\hat{m}$ in Figure 5, again broken down by scenario. For these results, we compared only methods which are able to identify a number of labels without it being known *a priori*: SCAN, DBSCAN, and our proposed approach. For this dataset, the ground truth number of roles was $\tilde{m} = 2$ for all scenarios. We can see that SCAN consistently achieves an error of 1, with a standard deviation of 0. Due to how SCAN analyzes the structure of the input graph (the unified temporal representation $\mathcal{A}$), it always discovers 3 roles in this dataset. This can be interpreted not as an actual reflection of the data but a limitation of the SCAN algorithm. We can see that DBSCAN has an error near 1 for most scenarios, showing that its ability to discover clusters does translate



(a) Overall Adjusted Rand Index (b) Overall $\hat{m}$ Error

Fig. 6. Overall metrics for compared approaches on the Swarm Red Team dataset. Markers show mean results, with 1 standard deviation above and below indicated with a bar.

well to discovering the number of roles. However, there can be significant variation in its performance, particularly in the larger multi-agent systems in Scenarios 2 and 5. This indicates it could not be relied on to consistently discover the number of roles in a system. Our approach performs the best in all scenarios except Scenario 5, while even then averaging an error of less than 1.5. In all other scenarios the error of our approach is less than 1, showing that our approach is able to consistently identify close to the correct number of roles. We can see that our approach's superior performance holds when considering the dataset overall, as seen in Figure 6(b). Our approach is able to achieve the lowest error, while SCAN still outperforms the DBSCAN approach due to its consistent discovery of 3 roles.

When considered together, Figures 6(a) and 6(b) show that our approach outperforms existing methods in role discovery, both by learning accurate role assignments in a multi-agent system and able by more accurately discovering the number of roles within a multi-agent system.

V. Conclusion

Multi-agent systems are ubiquitous in the real world, and a constant variable for autonomous systems to analyze and interact with. A key capability for autonomous systems is the capability to understand multi-agent systems, and an important part of this is recognizing the roles that agents play. We present a novel role discovery method, based on observations of a multi-agent system over time. We present an approach to first learn a unified temporal representation of a multi-agent system across multiple time steps based on observations of the positions and attributes of each agent. Our approach then learns a shared role indicator matrix that minimizes the cost of association among agents, and then factors this matrix into a role probability matrix that represents the probability of each agent playing each role. From this, our approach discovers the optimal role assignment for each agent and discovers the number of roles present in the multi-agent system. Through extensive experimental evaluation on an existing dataset, as well as the introduction of a novel role recognition dataset, we show that our approach outperforms existing role recognition and clustering methods.

REFERENCES

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, 2018.

[2] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *Artificial Intelligence*, vol. 258, pp. 66–95, 2018.

[3] J. Zhang, W. Hu, B. Yao, Y. Wang, and S.-C. Zhu, "Inferring social roles in long timespan video sequence," in *IEEE International Conference on Computer Vision Workshops*, 2011.

[4] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[5] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S. Chun Zhu, "Joint inference of groups, events and human roles in aerial videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[6] R. Fierro and A. Das, "A modular architecture for formation control," in *IEEE International Workshop on Robot Motion and Control*, 2002.

[7] B. Reily, C. Reardon, and H. Zhang, "Leading multi-agent teams to multiple goals while maintaining communication," in *Robotics: Science and Systems*, 2020.

[8] L. Lyujian, H. Wang, B. Reily, and H. Zhang, "Robust real-time group activity recognition of robot teams," *IEEE Robotics and Automation Letters*, 2021.

[9] W. Ge, R. T. Collins, and R. B. Ruback, "Vision-based analysis of small groups in pedestrian crowds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1003–1016, 2012.

[10] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li, "Rolx: structural role extraction & mining in large graphs," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[11] A. Silva and S. Chernova, "Unsupervised role discovery using temporal observations of agents," in *International Conference on Autonomous Agents and Multiagent Systems*, 2019.

[12] B. Reily, C. Reardon, and H. Zhang, "Representing multi-robot structure through multimodal graph embedding for the selection of robot teams," in *International Conference on Robotics and Automation*, 2020.

[13] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001.

[14] J. P. Desai, "A graph theoretic approach for modeling mobile robot team formations," *Journal of Robotic Systems*, vol. 19, no. 11, pp. 511–525, 2002.

[15] Y. Li and X. Chen, "Stability on multi-robot formation with dynamic interaction topologies," in *IEEE International Conference on Intelligent Robots and Systems*, 2005.

[16] M. C. De Gennaro, L. Iannelli, and F. Vasca, "Formation control and collision avoidance in mobile agent systems," in *IEEE International Symposium on Intelligent Control*, 2005.

[17] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *IEEE Conference on Decision and Control*, 2002.

[18] R. Olfati-Saber, W. B. Dunbar, and R. M. Murray, "Cooperative control of multi-vehicle systems using cost graphs and optimization," in *IEEE American Control Conference*, 2003.

[19] B. Reily and H. Zhang, "Team assignment for heterogeneous multi-robot sensor coverage through graph representation learning," in *International Conference on Robotics and Automation*, 2021.

[20] V. Ramanathan, B. Yao, and L. Fei-Fei, "Social role discovery in human events," in *IEEE conference on Computer Vision and Pattern Recognition*, 2013.

[21] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[22] S. Kwak, B. Han, and J. Hee Han, "Multi-agent event detection: Localization and role assignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[23] R. Li, P. Porfilio, and T. Zickler, "Finding group interactions in social clutter," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[24] P. V. Gupte, B. Ravindran, and S. Parthasarathy, "Role discovery in graphs using global features: Algorithms, applications and a novel evaluation strategy," in *IEEE International Conference on Data Engineering)*, 2017.

[25] G. Dasoulas, G. Nikolentzos, K. Scaman, A. Virmaux, and M. Vazirgiannis, "Ego-based entropy measures for structural representations on graphs," *arXiv preprint arXiv:2102.08735*, 2021.

[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[27] F. Brandt and M. Bullinger, "Finding and recognizing popular coalition structures.," in *International Conference on Autonomous Agents and Multiagent Systems*, 2020.

[28] G. Istrate, C. Bonchis, and C. Gatina, "It's not whom you know, it's what you, or your friends, can do: Coalitional frameworks for network centralities," in *International Conference on Autonomous Agents and Multiagent Systems*, 2020.

[29] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, p. 025102, 2001.

[30] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, p. 026113, 2004.

[31] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[32] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: a structural clustering algorithm for networks," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

[33] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *IEEE International Conference on Data Mining*, 2001.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[35] D. P. Bertsekas, *Constrained optimization and lagrange multiplier methods*. Academic Press, 2014.

[36] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 1996.

[37] S. Vassilvitskii and D. Arthur, "k-means++: The advantages of careful seeding," in *ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[38] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," in *International Conference on Machine Learning*, 2009.