Role-Oriented Dynamic Network Embedding

Ting Pan
College of Intelligence and Computing
Tianjin University
Tianjin, China
tingpan@tju.edu.cn

Wenjun Wang
College of Intelligence and Computing
Tianjin University
Tianjin, China
wjwang@tju.edu.cn

Minglai Shao*
School of New Media and Communication
Tianjin University
Tianjin, China
shaoml@tju.edu.cn

Yueheng Sun
College of Intelligence and Computing
Tianjin University
Tianjin, China
yhs@tju.edu.cn

Pengfei Jiao
School of Cyberspace
Hangzhou Dianzi University
Hangzhou, China
pjiao@hdu.edu.cn

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020276

***Abstract*—Exploring the differences and important patterns of nodes from the perspective of roles has gradually developed into an interesting and important topic in network analysis. However, existing role-oriented network embedding methods focus more on identifying underlying roles for static network, which leads to complex temporal behaviors being overlooked and degraded performance facing dynamic network. The few role analytics methods for dynamic networks either cannot learn general node representations or fail to discovery role transitions of nodes. In this work, we propose a unified framework RDNE (Role-oriented Dynamic Network Embedding) to tackle such challenges, which aim to learn multiple embeddings for individual nodes based on time-varying structural behaviors. Based on regular equivalence, RDNE propagates the structural features over the graph to derive the initial role-oriented representations. Then, it applies capsule network to further model the mapping between nodes and roles, which is the first time capsule network is used for role discovery. For the varying and temporal dependence within dynamic network, we utilize the Gated Recurrent Unit to compute historical information and use historical information to influence the generation of representations at the next snapshot. Comprehensive experiments on both synthetic and real-world networks validate the superiority of the proposed RDNE.**

***Index Terms*—Dynamic Network Embedding, Role Discovery, Graph Convolutional Networks (GCNs), Capsule Network, Gated Recurrent Unit (GRU)**

I. Introduction

Roles of nodes usually reflect their connectivity pattern within complex network. Mining the underlying roles of the network has a great potential for application in network analysis tasks, ranging from similarity search in author collaboration networks [1], network alignment in protein-protein interaction networks [2], to user stitching in messenger Logs [3], etc. In recent years, exploring the differences and important patterns of nodes from the perspective of roles has gradually developed into an interesting and important topic.

Role-oriented network embedding methods are currently the main paradigm for role discovery [4], [5]. The goal is to learn a low-dimensional representation based on its connectivity patterns to characterize each node, so that nodes sharing the same role are embedded in close positions in the latent space. The state-of-the-art role-oriented network embedding methods have made significant progress. The most intuitive methods like RolX [6], apply various matrix factorization technique on the structural feature matrix [7] or structural similarity matrix [8] to obtain node embeddings. In addition, in order to capture stronger notion of structural identity, struc2vec [9] constructs a multi-layer weighted graph to encode the structural similarity of nodes before random walk. DRNE [10] leverages layer normalized LSTM to deep recursive aggregate the embeddings of neighbors, which claims learned node embeddings with regular equivalence. Although these methods show good performance in various downstream tasks, it is worth noting that they are almost designed for static networks, which leads to complex temporal behaviors being overlooked and degraded performance facing dynamic networks.

Dynamic network is an abstract representation of complex system, which records the interaction between entities and their dynamic evolution over time. For example, temporal co-author network [11] documents the authors' collaborations at various periods, temporal wikipedia network [12] describes users editing each other's Talk page in different times. Intuitively, these time-varying behaviors not only bring great changes to the network topology, but also affect the distribution of roles. In other words, nodes are likely to play different roles at different time points. However, as far as we know, this phenomenon has not attracted enough attention. At present, most dynamic network embedding methods are based on the proximity of nodes rather than structural similarity [13], [14]. Therefore, in this work, we focus on role discovery for dynamic networks, which is an urgent and practical problem.

We conduct a detailed investigation on role-based dynamic network embedding. It cannot be said that there is no research in this area, but it is very limited and more importantly, their starting points deviate from our goals. HR2vec [15] extended by struc2vec generates node embeddings by tracking historical information, but it ultimately only creates one embedding

* Corresponding author.

for the same node. That is to say, this method cannot be used to identify roles of nodes in different periods. Similar method includes TEDIC [16] and time-dependent stochastic block models [17]. The former assigns a unique role to a node according to its behaviors over a period of time, such as deception detection and dominance identification. The latter detects the roles of bicycle-sharing stations based on the traffic within and between blocks of stations over the course of a day. Although Evans et al. [18] also model roles' evolution and their dynamics to predict roles' distributions in political networks, their implementation of role detection only simply employs a static embedding method. To the best of our knowledge, relevant to our work include DBMM [19] and DyNMF [20]. But strictly speaking, these two methods are not in the category of network embedding and cannot be generalized to other role-oriented tasks.

Based on the above discussion, for the first time, we propose a unified framework RDNE (Role-oriented Dynamic Network Embedding), which learns a set of node embeddings on each snapshot to describe nodes' roles at various periods. In RDNE, we derive the initial representation preserving role information by propagating the structural features of nodes over the graph, which is different from the traditional feature propagation process that captures the proximity of nodes. Intuitively, nodes with similar connectivity patterns can obtain similar representations by aggregating the structural features of their neighbors if the number of feature propagations is not large, which can be explained from the perspective of regular equivalence [21]. Our experiment also verifies its efficiency. RDNE makes the first attempt to employ capsule network [22] for role discovery. Each capsule is built for each role category, and an iterative routing-by-agreement mechanism is designed to calculate the coupling coefficients between nodes and roles. The capsule with the largest the coupling coefficient is activated. Noting that the capsule network in this work is not completely consistent with the literature [22]. In addition, RDNE leverages the Gated Recurrent Unit (GRU) [23] to analyze the varying and temporal dependence in the latent space and update historical information at each snapshot, which is used for affecting the generation of node embeddings at the next snapshot.

Our main contributions are summarized as follows:

- Strictly speaking, we are the first work to study the roles' evolution over time in terms of time-varying structural behavior under the framework of network embedding. Its goal is to create representation vectors for individual nodes on each snapshot to describe their role information at various times.
- We propose a novel Role-oriented Dynamic Network Embedding method (RDNE) to capture the time-varying structural behaviors of nodes. It is the first attempt to apply structural feature propagation and capsule network for role discovery.
- We carry out comprehensive experiments on both synthetic networks and real-world networks to show the superiority of our proposed model compared with other state-of-the-art methods.

II. Related Work

Role-Oriented Static Network Embedding. Role-oriented network embedding for static networks gain tremendous popularity in recent years [4], [5]. Its embedding mechanisms are mainly divided into three categories: matrix factorization, random walk and deep learning. The matrix factorization-based methods [1], [6], [8], [24]–[27] focus on constructing structural feature matrix or structural similarity matrix, and then applying various matrix factorization techniques on them. For example, RolX [6] applies the non-negative matrix factorization technique on structural feature matrix [7] proposed by Henderson et al. GraphWave [26] leverages the diffusion of a spectral graph wavelet centered at the node to create target matrix. SEGK [1] uses graph kernel to capture the high-order structural similarity of neighborhood subgraphs. The core idea of the random walk based methods [9], [28]–[31] is to create the structural context sequence of nodes via unbiased or biased random walk strategies, and then map the co-occurrence relations to the similarity of node embeddings using language models such as skip-gram. For example, in addition to the struc2vec mentioned in the introduction, struc2gauss [30] selects positive and negative samples from the context sequence constructed by rolesim algorithm [32] to train the model. Role2vec [28] proposes the notion of attributed random walks, which maps the structural features of nodes to roles before random walk. The last type are based on deep learning frameworks [10], [33]–[35]. GAS [34] is the first to leverage structural information to guide the training of graph neural networks. RESD [35] introduces Variational Auto-Encoder (VAE) [36] to learn more robust embeddings.

While these methods are promising in learning structural node embeddings, they can not leverage temporal information to model the dynamic behaviors of nodes and explain how these behaviors affect role distributions of nodes. In this work, in order to address such problem, we use the Gated Recurrent Unit [23] to update the historical information at the current time, and at the next time, propagate the historical information and the structural features of the nodes over the graph.

Role Analytics for Dynamic Networks. We conduct a detailed investigation on role analytics for dynamic networks, and find that the existing works are either inconsistent with our goals or not in the category of network embedding. The former includes HR2vec [15], TEDIC [16] and time-dependent stochastic block models [17] proposed by Jane et al. Their goals are to assign a unique role to a node rather than multiple roles according to its behavior over a period of time. The last one is not the network embedding method. The latter includes DBMM [19] and DyNMF [20], which are currently the most relevant works to our knowledge. DBMM is the first scalable dynamic behavioral mixed-membership model capable of modeling the behavioral transition patterns and trends of nodes. It directly applies RolX [6] to discover roles in each snapshot and then trains a stacked or summary model based on node-role matrices to learn a transition model. DyNMF captures the role transition between previous roles

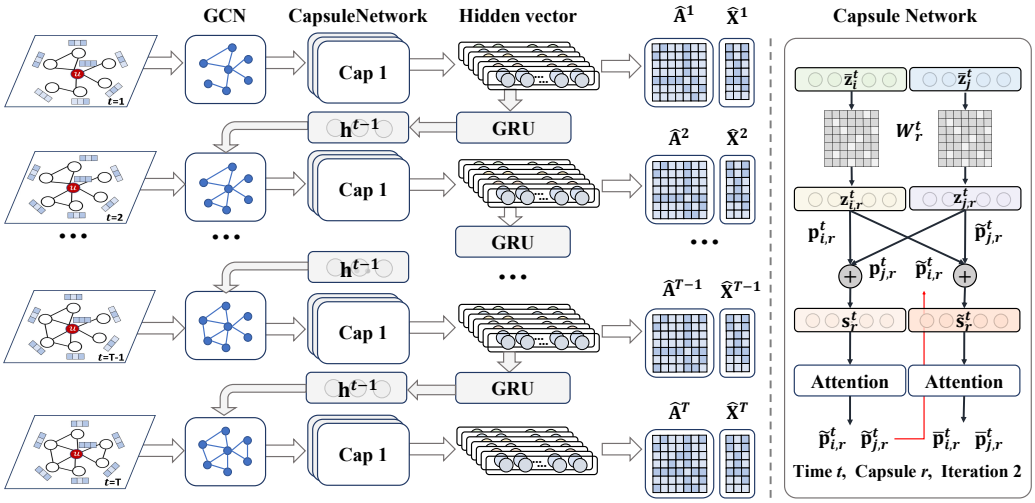

Fig. 1. An overview of the proposed RDNE: (a) Structural feature propagation. (b) Capsule network. (c) Modeling Temporal behaviors.

and current roles by directly introducing a role transition matrix. However, since they aim to learn nodes' membership instead of embedding, these methods are not in the category of network representation and cannot be generalized to other role-oriented tasks. In addition, Evans et al. [18] propose role detection and prediction in dynamic political networks, but they pay more attention to the roles' evolution and their dynamics, rather than focusing on how to design a dynamic role detection algorithm.

In this work, we aim to propose a dynamic network embedding method for analyzing the roles of nodes on each snapshot. To the best of our knowledge, this is the first attempt.

Dynamic Network Embedding. Most dynamic network embedding methods [37], [38] pay their attention on the proximity of nodes to achieve the tasks of link prediction [14], [39]–[41], network reconstruction [42], network clustering [43], anomaly detection [44], etc. The basic assumption behind them is the homogeneity of the network. However, role discovery tasks are regarded as orthogonal to them, where the nodes sharing the same role are usually not directly connected, or even located in different parts in the network. Therefore, these methods show poor performance facing role-oriented tasks.

III. Problem Definition

Let $\mathcal{G} = \{G^1, G^2, \ldots, G^T\}$ be a discrete temporal network with T snapshots of equal time intervals. For each snapshot $G^t = (V^t, E^t)$, where V^t indicates the set of active nodes, E^t reflects their interaction, $|V^t|$ and $|E^t|$ represent the number of nodes and edges, respectively. $\mathbf{A}^t$ is the adjacency matrix and $\mathbf{X}^t$ is the structural feature matrix extracted by ReFeX algorithm [7]. The neighbors of node v_i at time step t is defined as $\mathcal{N}_i^t = \{v_j \mid (i,j) \in E^t\}$ and its degree is defined as $d_i^t = |\mathcal{N}_i^t|$. We assume the set of nodes is constant within T snapshots and the set of edges evolves over time. In this work, our aim is to propose a role-oriented dynamic network embedding method to learn the d-dimension vectors $\{\mathbf{z}^t\}_{1 \le t \le T}$ for individual nodes. The learned embeddings should well capture the nodes' structural behaviors and dynamics, and support various role-oriented tasks, such as visualization, node classification and clustering.

IV. Methodology

In this section, we present the details of **R**ole-oriented **D**ynamic **N**etwork **E**mbedding (RDNE), with a brief overview illustrated in Figure 1.

A. Structural Feature Propagation

A number of the existing approaches take feature popagation to representation learning, among which Graph Convolutional Networks (GCNs) [45] have gained increasing attention and shown promising performance in community detection and link prediction. Yet GCNs are rarely used for role discovery. The related literature [4] states that due to iteratively applying the smoothing operator, the representation vectors derived from GCNs are tied to the proximity of nodes rather than equivalence in roles.

Therefore, GAS [34] applies sum-pooling propagation rule instead of original mean-pooling to encode graph structure. However, we argue that the influencing factors should also include the selection of information propagated. GCNs take the attribute information of nodes as features, and these attribute information generally have certain similarity among nodes with common neighbors, which naturally leads to similar representations when aggregating neighborhood information. For instance, each node in a citation network is associated with the statistics of words that were taken from the corresponding document, and the documents in the same domain are more likely to use the same words [46].

Based on this, we attempt to propagate nodes' structural features over the graph and limit the number of feature propagations for role-oriented representation learning. From the perspective of regular equivalence, we explain why role-oriented embeddings can be derived in this way. Specifically, regular equivalence defines that nodes share the same role

if they are connected to role-equivalent nodes [21]. In other words, the immediate neighbors of nodes that play the same role also share the same role, i.e., having similar structural features. Therefore, when we propagate structural features of nodes over the graph and the number of iterations is not large, nodes with the same role can obtain similar representations.

Specifically, given a node v_i and network snapshot t, We first extract its egonet network $\mathcal{E}_i^t = \left(V_{\mathcal{E}_i^t}, E_{\mathcal{E}_i^t}\right)$, where $V_{\mathcal{E}_i^t} = \{v_i\} \bigcup \{v_j \in V^t \mid (v_i, v_j) \in E^t\}$ and $E_{\mathcal{E}_i^t} = \{(v_i, v_j) \in E^t \mid v_i, v_j \in V_{\mathcal{E}_i^t}\}$. Based on egonet network, we compute the following six structural features with reference to ReFeX [7]:

- The degree of v_i: $\mathbf{X}_{i,1}^t = d_i^t = |\mathcal{N}_i^t|$.
- The number of edges in $\mathcal{E}_i^t$: $\mathbf{X}_{i,2}^t = |E_{\mathcal{E}_i^t}|$.
- The sum of nodes' degrees in $\mathcal{E}_i^t$: $\mathbf{X}_{i,3}^t = \sum_{v_j \in V_{\mathcal{E}_i^t}} \left|d_j^t\right|$.
- The proportion of within-egonet edges to all edges entering and leaving $\mathcal{E}_i^t$: $\mathbf{X}_{i,4}^t = \mathbf{X}_{i,2}^t / \mathbf{X}_{i,3}^t$.
- The proportion of non-egonet edges to all edges entering and leaving $\mathcal{E}_i^t$: $\mathbf{X}_{i,5}^t = \left(\mathbf{X}_{i,3}^t - \mathbf{X}_{i,2}^t\right) / \mathbf{X}_{i,3}^t$.
- The clustering coefficient of v_i: $\mathbf{X}_{i,6}^t = 2c_i^t / \left(\mathbf{X}_{i,1}^t \left(\mathbf{X}_{i,1} - 1\right)\right)$, where c_i^t represents the number of triangles that v_i participates in at time step t.

Then, we encode and propagate the structural features over the graph snapshot to obtain initial node embeddings:

$$\widetilde{\mathbf{Z}}^t = \sigma\left(\widetilde{\mathbf{D}}^{t^{-\frac{1}{2}}} \widetilde{\mathbf{A}}^t \widetilde{\mathbf{D}}^{t^{-\frac{1}{2}}} \phi(\mathbf{X}^t)\mathbf{W}\right) \tag{1}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is an identity matrix, $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}}_{ij}$, $\mathbf{X}$ is the structural feature matrix, $\mathbf{W}$ is trainable weight parameters, $\widetilde{\mathbf{Z}}$ is the output of GCNs, i.e., initial embeddings, ϕ can be any function such as a neural network, σ is an activation function, t represents the corresponding time step. Intuitively, we limit the number of GCNs layer to 1, so as to avoid the similarity between nodes with more connections caused by iterations.

B. Capsule Network

Capsule network has attracted wide attention since it was proposed [22], [47]–[49]. Its idea is that to take capsule as the basic unit of neural network, and use an iterative routing-by-agreement mechanism in the forward propagation to ensure that the output of the lower-level capsule gets sent to an appropriate higher-level capsule. A capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity. Intuitively, if higher-level capsules are instantiated as roles, lower-level capsules are instantiated as nodes, and node embeddings are considered as the outputs of lower-level capsules, then we can model the mapping between nodes and roles through the routing-by-agreement mechanism. Based on this, for the first time, we introduce capsule network in role-oriented representation learning. The structure is shown in the right panel of Figure 1.

Specifically, we first perform affine transformations on the output vector $\bar{\mathbf{z}}_i^t$ of low-level capsule i:

$$\mathbf{z}_{i,r}^t = \mathbf{W}_r^t \bar{\mathbf{z}}_i^t \tag{2}$$

where $\mathbf{W}_r^t$ is a transformation matrix with respect to high-level capsule r, $\widehat{\mathbf{z}}_{i,r}^t$ is called prediction vector. Then, we compute the vector $\mathbf{s}_r^t$ of high-level capsule r by weighted sum of the prediction vectors:

$$\mathbf{s}_r^t = \sum_i \mathbf{p}_{i,r}^t \mathbf{z}_{i,r}^t \tag{3}$$

where $\mathbf{p}_{i,r}^t$ is coupling coefficient between capsule i and capsule r, having $\sum_r \mathbf{p}_{i,r}^t = 1$. Unlike the literature [22], we do not use any non-linear function to activate $\mathbf{s}_r^t$.

Next, we introduce how to use dynamic routing to calculate and update the coupling coefficients. We first initialize $\mathbf{p}_{i,r}^t$ to $\frac{1}{R}$, where R is the number of high-level capsules. Then, we calculate $\mathbf{s}_r^t$ according to Eq.(3). Thereafter, we update the coupling coefficient by measuring the agreement between $\mathbf{z}_{i,r}^t$ and $\mathbf{s}_r^t$ through the attention mechanism. We repeat the above last two operations t times. The attention layer is defined as:

$$\widetilde{\mathbf{p}}_{i,r}^t = softmax(score(\mathbf{z}_{i,r}^t, \mathbf{s}_r^t)) \tag{4}$$

where $score(\cdot)$ can be any scoring function, e.g., additive model, scaled dot-product model, multiplicative model. In fact, the above process can be viewed as network clustering. The high agreement between prediction vector $\mathbf{z}_i^t$ and the vector $\mathbf{s}_r^t$ of capsule r results in a high coupling coefficient between capsule i and capsule r and low coupling coefficients between capsule i and other high-level capsules, which in turn makes the predicted vector with a larger contribution to the vector of capsule r in the next update. In other words, the final $\mathbf{s}_r^t$ can be view of the cluster center of nodes whose role is r, and the final $\mathbf{p}_{i,r}^t$ is the probability of node v_i playing the role r. Based on this, we select the corresponding prediction vector of high-level capsule with the largest coupling coefficient as the final embedding vector of node v_i.

C. Modeling Temporal behaviors

To further analyze the varying and temporal dependence for the dynamic network, we leverage GRU [23] to encode historical information at snapshot t and use it to influence the generation of node embeddings for the next snapshot $t+1$. The detailed update rules of GRU are as follows:

$$\mathbf{u}_i^t = \sigma(\mathbf{W}_u[\mathbf{z}_i^t, \mathbf{h}_i^{t-1}]) \tag{5}$$

$$\mathbf{r}_i^t = \sigma(\mathbf{W}_r[\mathbf{z}_i^t, \mathbf{h}_i^{t-1}]) \tag{6}$$

$$\widetilde{\mathbf{h}}_i^t = \tanh\left(\mathbf{W}[\mathbf{z}_i^t, \mathbf{h}_i^{t-1} \odot \mathbf{r}_i^t]\right) \tag{7}$$

$$\mathbf{h}_i^t = (1 - \mathbf{u}_i^t)\mathbf{h}_i^{t-1} + \mathbf{u}_i^t \widetilde{\mathbf{h}}_i^t \tag{8}$$

where $\mathbf{u}_i^t$ is the update gate vector, $\mathbf{r}_i^t$ is the reset gate vector, $\widetilde{\mathbf{h}}_i^t$ is the new estimated candidate state, $\mathbf{h}_i^t$ and $\mathbf{h}_i^{t-1}$ is the historical information at snapshot t and snapshot $t-1$, respectively, $\mathbf{z}_i^t$ is embedding of node v_i learned from the capsule network. $\mathbf{W}_u$, $\mathbf{W}_r$ and $\mathbf{W}$ are the weight matrices.

At the snapshot $t+1$, we combine the initial embedding $\widetilde{\mathbf{Z}}^{t+1}$ with historical information $\mathbf{H}^t$, and obtain the vector $\bar{\mathbf{Z}}^{t+1}$ used in capsule network through GCNs layer:

$$\widehat{\mathbf{Z}}^{t+1} = \psi\left([\widetilde{\mathbf{Z}}^{t+1}, \mathbf{H}^t]\right) \tag{9}$$

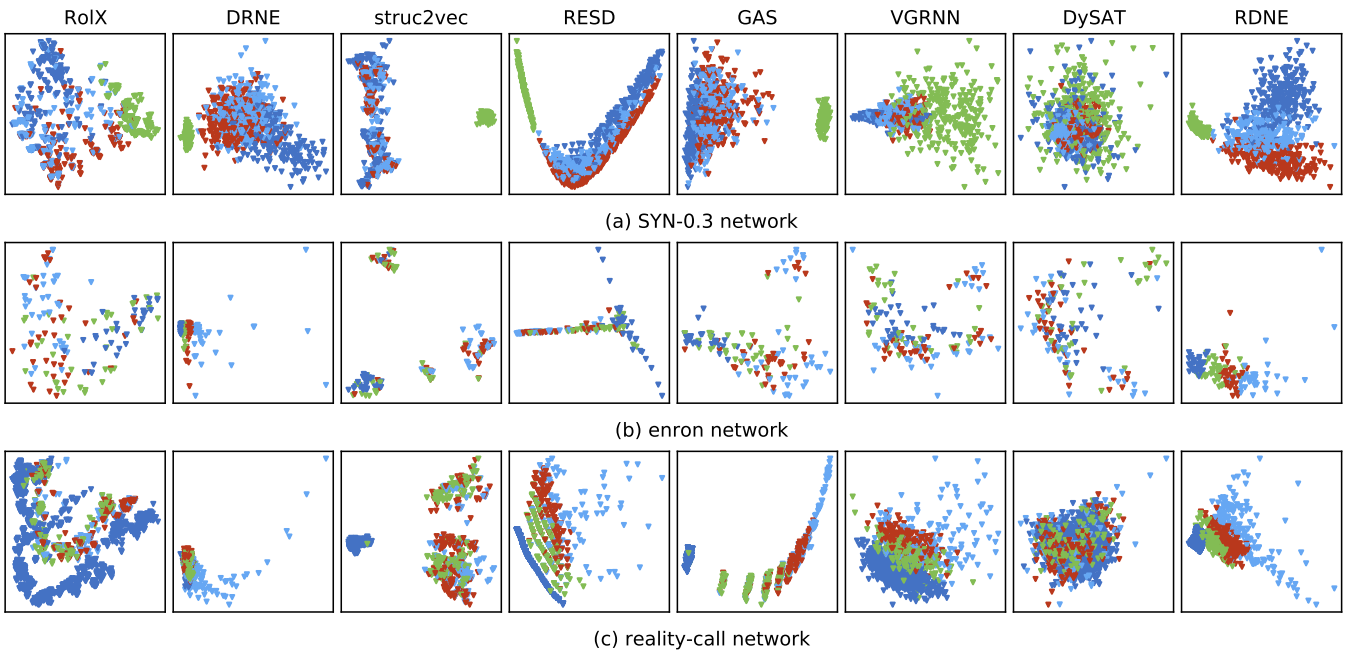

Fig. 2. Visualization results on synthetic network with $\alpha = 30\%$ and two real-world networks: Nodes with the same roles are marked with the same color.

$$\bar{\mathbf{Z}}^{t+1} = \sigma\left(\widetilde{\mathbf{D}}^{t+1^{-\frac{1}{2}}}\widetilde{\mathbf{A}}^{t+1}\widetilde{\mathbf{D}}^{t+1^{-\frac{1}{2}}}\widehat{\mathbf{Z}}^{t+1}\mathbf{W}\right) \tag{10}$$

where $\mathbf{W}$ is the weight matrix and ψ can be any function such as a neural network.

D. Loss Function and Training

The proposed RDNE consists of two training objective. One is to minimize the reconstruction loss of nodes' structural features. Roles represent the connectivity patterns of nodes, and role-oriented network embedding ought to preserve the structural information. The other is to minimize the binary cross-entropy loss of network links. In fact, many network links are driven by roles that nodes play. For instance, officers are likely to receive more emails than lower-ranking employees [8]. Therefore, we think that the learned embeddings should also preserve such interaction information. Given the snapshot t, the entirety loss function are listed as follows:

$$\begin{aligned}\mathcal{L}^t = &\sum_{v_i \in |V^t|} \left\| \mathbf{X}_i^t - \widehat{\mathbf{X}}_i^t \right\|_2^2 \\ &- \sum_{v_i, v_j \in |V^t|} \mathbf{A}_{ij}^t \log(\widehat{\mathbf{A}}_{ij}^t) - (1 - \mathbf{A}_{ij}^t) \log(1 - \widehat{\mathbf{A}}_{ij}^t)\end{aligned} \tag{11}$$

where $\widehat{\mathbf{X}}_i^t = \sigma(\mathbf{W}_x \mathbf{z}_i^t + \mathbf{b}_x)$ and $\widehat{\mathbf{A}}_i^t = \sigma(\mathbf{W}_a \mathbf{z}_i^t + \mathbf{b}_a)$ are the reconstructed structural feature matrix and adjacency matrix by the multilayer perceptron as decoder, respectively. σ is activation function, $\mathbf{W}_x$ and $\mathbf{W}_a$ are the weight matrices, and $\mathbf{b}_x$ and $\mathbf{b}_a$ are the bias vectors.

V. Experiments

We conduct comprehensive experiments on synthetic and real-world dynamic networks to evaluate the performance of our proposed model. The experiments include visualization, node classification, clustering, parameter sensitivity, ablation experiments and case study.

A. Datasets

Synthetic Networks. In this work, we provide a generation mechanism to build dynamic networks with role labels as testing benchmarks. We generate data with $N = 1000$ nodes, $T = 5$ time points and $R = 4$ roles including hub nodes, star-edge nodes, near-cliques nodes and bridge nodes. The distribution of roles and the interaction probability between roles control the connectivity patterns of nodes and the number of edges. The network dynamics is introduced through changing nodes' role labels and reconnecting edges. The hyperparameter α controls the severity of the evolution. We vary the value of α from 10% to 60% to generate six synthetic networks.

Real-world Networks. We use two real-world dynamic network datasets [20]. Enron network records 28920 emails of 146 employees with 9 snapshots. Reality-call network records 65177 calls of 2723 users with 10 snapshots. Roles in these two datasets are divided into four categories based on communication frequency.

B. Baselines

We select several state-of-the-art methods as baselines, which are mainly divided into three categories: (1) Role-oriented static network embedding. Such methods include RolX [6], DRNE [10], struc2vec [9], RESD [35] and GAS [34]. We obtain the corresponding node embeddings by running these methods on each snapshot. We can analyze the advantages of modeling with historical information

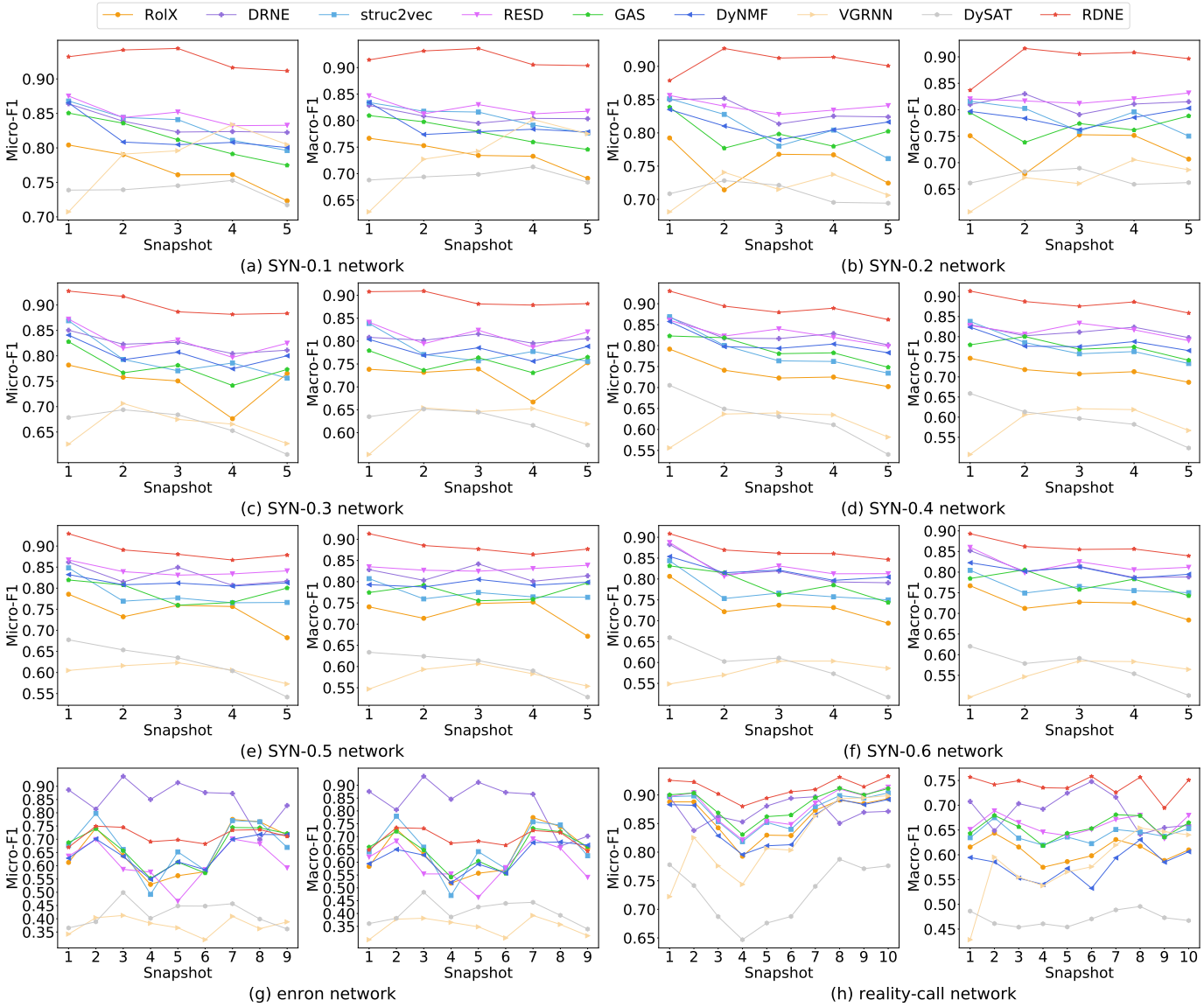

Fig. 3. Classification results on all networks: The horizontal axis represents the time step, and the vertical axis represents the scores of Micro-F1 or Macro-F1.

by comparing with them. (2) Role-oriented method for dynamic networks. Such method includes DyNMF [20]. We can show the effectiveness and generality of the proposed RDNE by comparing with DyNMF. (3) Dynamic network embedding methods. Such methods include VGRNN [40] and DySAT [14]. We verify that traditional proximity-based methods cannot effectively capture the role information and fail to support role-oriented tasks.

C. Experimental Settings

For all baselines except DyNMF, the learned embedding dimension is set to 128, and other parameters are fine-tuned based on default settings in original papers. The outputs of DyNMF are the role distribution vectors of nodes.

For the part of structural feature propagation, we employ single layer neural network with tanh activation function to encode features. For the part of capsule network, we set the number of iterations to 4 and use scaled dot-product model as scoring function. For the part of modeling temporal behaviors, we set the number of GRU layers to 2 and use single layer neural network with ReLU activation function to encode the initial embeddings and historical information. Noting that we share the parameters of some components at different time steps, reducing the number of parameters that need to be learned. The model is trained via Adam optimizer with the learning rate of 0.0001 or 0.0008 for 1000 epochs.

D. Visualization

In this part, we conduct visualization experiments on a synthetic network and two real-world networks to qualitatively evaluate the performance of RDNE against all baselines except DyNMF. The outputs of DyNMF is not suitable for

TABLE I
CLUSTERING RESULTS ON ALL NETWORKS: EACH VALUE IS THE AVERAGE OF ALL THE SNAPSHOTS. THE TWO HIGHEST VALUES ARE SHOWN IN BOLD.

Method	SYN-0.1		SYN-0.2		SYN-0.3		SYN-0.4		SYN-0.5		SYN-0.6		enron		reality-call	
	NMI	silhouette	NMI	silhouette	NMI	silhouette	NMI	silhouette	NMI	silhouette	NMI	silhouette	NMI	silhouette	NMI	silhouette
RolX	0.3437	0.1967	0.3427	0.1792	0.3384	0.1694	0.3649	0.1942	0.3533	0.1755	0.3425	0.1639	0.2819	0.1046	0.1006	0.1283
DRNE	0.5290	**0.5889**	0.5361	**0.5734**	0.5045	**0.5661**	**0.5089**	**0.5711**	**0.5105**	**0.5573**	0.4951	**0.5569**	0.4666	**0.6925**	0.1781	0.5083
struc2vec	0.4370	0.1770	0.4499	0.1638	0.4442	0.1544	0.4380	0.1487	0.3966	0.1454	0.4050	0.1418	**0.4689**	0.3293	0.6554	0.4017
RESD	0.4752	0.5246	0.4671	0.5348	0.4590	0.5353	0.4562	**0.5379**	0.4479	**0.5323**	0.4424	**0.5421**	0.3495	**0.5135**	0.5140	0.6349
GAS	**0.5522**	0.2800	**0.5380**	0.2774	**0.5128**	0.2572	0.4945	0.2550	0.5081	0.2438	**0.5073**	0.2385	0.3435	0.3321	**0.6884**	0.6264
VGRNN	0.4708	0.1521	0.4128	0.1452	0.3767	0.1464	0.3656	0.2059	0.3447	0.2035	0.3338	0.2075	0.1029	0.2870	0.1524	0.0330
DySAT	0.2484	0.0540	0.1950	0.0777	0.1627	0.0929	0.1517	0.1228	0.1332	0.1265	0.1335	0.1408	0.0729	0.1708	0.0069	0.0352
DyNMF	0.3943	0.4711	0.3729	0.4460	0.3633	0.4413	0.3465	0.4323	0.3521	0.4509	0.3373	0.4352	0.3261	0.4573	0.4494	**0.6897**
RDNE	**0.7236**	**0.5423**	**0.6394**	**0.5747**	**0.6345**	**0.5635**	**0.6342**	0.5364	**0.6354**	0.5269	**0.6130**	0.5266	**0.5215**	0.4665	**0.6734**	**0.7026**

this task. Specifically, we embed node representations into a two-dimensional space via the PCA algorithm as shown in Figure 2. VGRNN and DySAT perform poorly on all networks since they are not designed for capturing the structural information of nodes in the network. Although RolX and struc2vec are role-oriented approaches, they do not show the expected results. RESD and GAS achieve that nodes with the same role are clustered on the reality-call network, but not on synthetic network and enron network. It is clearly that the visualization results based on RDNE have more distinct role blocks. Such results indicate RDNE can learn more similar embeddings for nodes with the same role.

E. Node Classification

To qualitatively evaluate the performance of the baselines and the proposed model RDNE, we conduct node classification experiments and take Micro-F1 and Macro-F1 as indicators. In this experiment, for each method and each network, we select 70% of learned embedding vectors to build the training dataset for training the linear logistic regression classifier, and the rest are used for testing. For fairness, we repeat the experiment 20 times and reported the average. The results are shown in Figure 3. It can be seen that our model RDNE exhibits great advantages on six synthetic networks and reality-call network. Although the classification results of RDNE are not as good as DRNE on the enron network, they are still sub-optimal overall. And more importantly, its results are stable on different snapshots, but other methods are not. This proves to some extent that it is beneficial to use historical information for network modeling. Taking a closer look, we also observe that the severity of network evolution has a limited impact on the proposed RDNE. That is to say, RDNE can still maintain good performance even if the network structure changes drastically. In the baselines, VGRNN and DySAT still have the worst scores. Noting that DRNE whose performance is flat in visualization are second only to RDNE in this task. Furthermore, we also find that although DyNMF is designed for dynamic networks, it does not exhibit more powerful classification ability compared to some static methods such as DRNE and RESD, and even ranks last on reality-call networks. We think this is because that it focuses more on modeling role transitions than on discovery.

F. Node Clustering

In this part, we utilize unsupervised clustering experiments to evaluate the performance of RDNE, i.e. using the k-means algorithm to partition nodes into different clusters. We use NMI and silhouette coefficient as evaluation metrics, where the former quantifies the matching degree between the true distribution and the clustering results, and the latter comprehensively considers the compactness of one cluster and separation between clusters. The results are shown in Table I and the two highest values are shown in bold. It is obvious that RDNE has a significantly improvement in NMI scores while maintaining a high level in silhouette coefficient. Other methods, by contrast cannot have the best of both worlds. For instance, the clustering results of GAS can be close to the real distribution, but its clustering boundary is not clear. The results of DRNE are in the opposite way to GAS. Although DyNMF is on par with RolX in NMI score, it has a higher silhouette coefficient. This proves that even just modeling role transitions with historical and current views is better than directly applying static methods on each snapshot. That is to say, capturing the network dynamics is the key to better modeling dynamic networks. The above observations clearly show that more discriminative structured information is preserved in the embeddings generated by our proposed RDNE, making it more effective facing role-oriented tasks.

G. Parameter Sensitivity and Ablation Experiments

The above experiments demonstrate the effectiveness of RDNE. In this part, we conduct parameter sensitivity and ablation experiments to investigate the impact of model parameters and each component on model performance.

Parameter Sensitivity. Based on classification experiments on two real-world networks, we evaluate the impact of dimension of node embedding, the number of iterations of dynamic routing, and the number of GRU layers on model performance.

- Dimension of node embedding. We vary the dimension of node embedding from 2 to 128 as shown in Figure 4. We can observe that the value of dimension has a great influence on its performance, which is manifested in the significant improvement of the classification scores with the increase of the dimension. This is because the

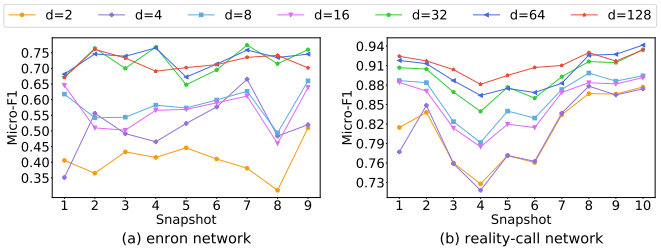

Fig. 4. Parameter sensitivity: Dimension of node embedding.

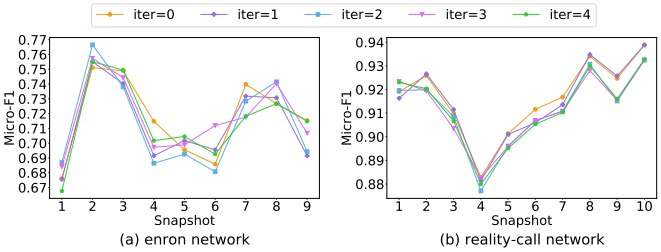

Fig. 5. Parameter sensitivity: The number of iterations for dynamic routing.

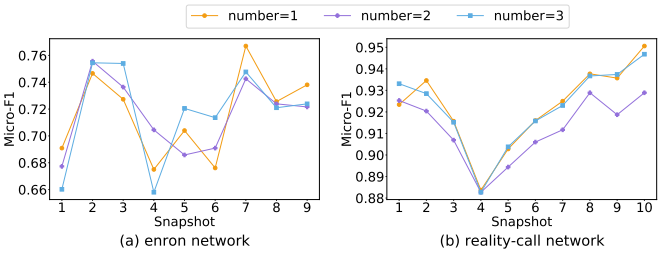

Fig. 6. Parameter sensitivity: The number of GRU layers.

embeddings with higher dimension can preserve more structural information. But when the dimension reaches a certain threshold, its performance tends to be stable. For example, the threshold is 32 on enron network while 128 on reality-call network.

- The number of iterations for dynamic routing. We vary the number of iterations from 0 to 4 as shown in Figure 5, where setting the number of iterations to 0 means taking the mean of the prediction vectors as the vector of the high level capsule. The results indicate that the model converges after very few iterations.
- The number of GRU layers. We vary the number of GRU layers from 1 to 3 as shown in Figure 6. The results show that there are certain differences in the optimal value of the number of GRU layers on different networks. On enron network, the results based on the 2-layer GRU are best overall, but the worst on reality-call network. We consider that this may be caused by the severity of network evolution. However, noting that our model is optimal compared to other baselines even though we use a 2-layer GRU on reality-call network.

Ablation Experiments. The core of the proposed RDNE is structural feature propagation and capsule network. Based on clustering experiments on all networks, we evaluate the impact of each component on model performance. The experimental results are shown in Table II, where the column "Dot⋆" is the default setting for RDNE.

TABLE II
ABLATION RESULTS ON ALL NETWORKS. EACH VALUE IS THE AVERAGE OF ALL THE SNAPSHOTS.

Dataset	Metric	Add	Dot⋆	Mul	no caps	no feat
SYN-0.1	NMI	0.6206	**0.7236**	0.6119	0.6960	0.5095
	silhouette	0.5557	0.5423	**0.5834**	0.5430	0.5655
SYN-0.2	NMI	0.6454	0.6394	0.6255	**0.6625**	0.4118
	silhouette	0.5535	0.5747	**0.6064**	0.5097	0.5020
SYN-0.3	NMI	**0.6484**	0.6344	0.6374	0.6462	0.3705
	silhouette	0.5405	0.5635	**0.6187**	0.4922	0.4599
SYN-0.4	NMI	**0.6379**	0.6342	0.5880	0.6327	0.3150
	silhouette	0.5301	0.5364	**0.5377**	0.4494	0.4478
SYN-0.5	NMI	**0.6487**	0.6354	0.6229	0.6143	0.2821
	silhouette	0.5241	0.5269	**0.5270**	0.4930	0.4275
SYN-0.6	NMI	**0.6289**	0.6130	0.6175	0.5761	0.2418
	silhouette	0.5277	0.5266	**0.5340**	0.4774	0.4214
enron	NMI	0.5171	0.5205	**0.5209**	0.4348	0.2186
	silhouette	0.4552	0.4661	0.4413	0.3755	**0.5652**
reality-call	NMI	0.5409	**0.6735**	0.6431	0.6714	0.2172
	silhouette	0.5808	**0.7025**	0.7007	0.6530	0.4891

[a] ⋆ is the default setting for RDNE.

- Structural feature propagation. To verify the effectiveness of structural feature propagation, we establish a control group that replaces the structural feature matrix with the identity matrix for propagation. The column "no feat" in Table II is the corresponding results. As expected, the scores of experiments without the structural features are much lower than those with structural features. Such results clearly indicate that it is difficult for traditional feature propagation to capture the role information of nodes, but based on regular equivalence, structural feature propagation allows nodes playing the same role to obtain similar representations by aggregating structural features of their neighborhoods, where these neighborhood have similar structural features since they are role-equivalent.
- Capsule network. We design multiple sets of controls in our ablation experiments with respect to capsule network to evaluate its effectiveness, including using additive model, multiplicative model in place of the default scaled dot-product model as the scoring function, and using a single-layer neural network in place of capsule network. Their results are reported in columns "Add", "Mul" and "no caps" of Table II and the highest values are shown in bold. It is obvious that the experiments with using capsule network show stronger clustering ability compared to those without capsule network, especially on two real-world networks. Such results indicate that the node embeddings obtained based on capsule network contains more discriminative role information. In addition, we can see that the selection of scoring function also has a certain impact on model performance. The experiments with using scaled dot-product model have better performance on two real-world networks.

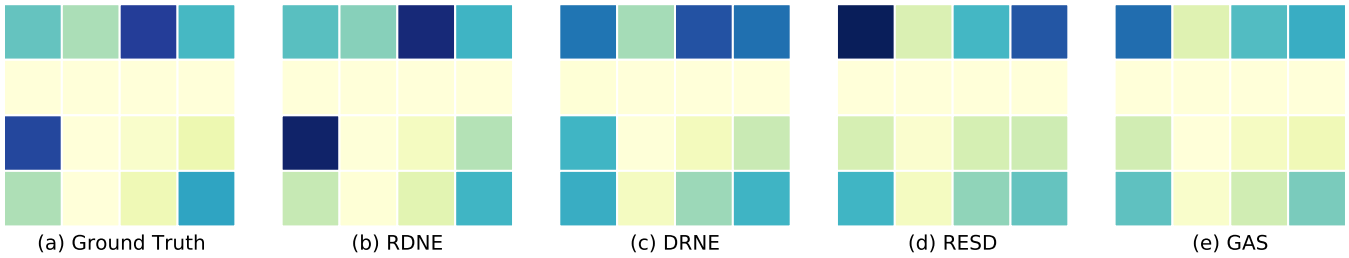

Fig. 7. Case study of role interaction analysis on synthetic network: The visualization of the role interaction strength matrix $\mathbf{\Delta} \in \mathbb{R}^{4\times4}$, where the darker the color, the greater the value of the interaction strength.

H. Case Study: Role interaction analysis

A case study on synthetic network is to demonstrate the effectiveness of the proposed RDNE. Specifically, we apply the k-means algorithm on the learned embeddings to obtain four role clusters, and use the Hungarian algorithm to match roles with the ground truth. After that, we calculate the interaction strength $\mathbf{\Delta}_{kl}$ from role k to role l as follow [30]:

$$\mathbf{\Delta}_{kl} = \begin{cases} \sum_{v_i \in \mathcal{R}_k, v_j \in \mathcal{R}_l} \frac{\mathbf{A}_{ij}}{(|\mathcal{R}_k| \cdot |\mathcal{R}_l|)}, & if \quad k \neq l \\ \sum_{v_i \in \mathcal{R}_k, v_j \in \mathcal{R}_l} \frac{\mathbf{A}_{ij}}{(|\mathcal{R}_k| \cdot (|\mathcal{R}_l|-1))}, & if \quad k = l \end{cases} \tag{12}$$

where $v_i \in \mathcal{R}_k$ represents that node v_i belongs to role k, and $|\mathcal{R}_k|$ is the number of nodes playing the role k. $\mathbf{A}_{ij}$ is the number of connections from node v_i to node v_j. By doing so, we can more intuitively explore whether the learned embeddings can reflect the connectivity patterns of nodes.

We visualize the real role interaction strength matrix $\mathbf{\Delta} \in \mathbb{R}^{4\times4}$ and the results based on the proposed RDNE, DRNE, RESD and GAS respectively as shown in Figure 7. The darker the color, the stronger the interaction strength between the two. We can see that the role interaction strength matrix is asymmetric, that is because a hub node should have a large value of $\mathbf{\Delta}_{kl}$ to be surrounded by star-edge nodes, but a star-edge node should have a small value of $\mathbf{\Delta}_{lk}$ to ensure that it only connect to one hub node.

As can be seen from Figure 7, there is almost agreement between the results based on the proposed RDNE and the true distribution, while DRNE, RESD and GAS all have large deviation. For instance, the interaction strength between the first role is overestimated, and those of between the first role and the third role is underestimated. We consider that the superiority of RDNE benefits from the propagation of structural features over the graph structure. Overall, this experiment indicates the effectiveness of the RDNE in capturing the structural behaviors of nodes.

VI. Conclusion

In recent years, role analysis on complex networks has become a trend. However, until now, there is still a lack of methods that can both learn general node representations for dynamic networks and discover the nodes' role transitions in different periods. In this work, we propose the RDNE, which is a role-oriented dynamic network embedding method designed for discovering nodes' roles in different periods. On the one hand, RDNE proposes structural feature propagation based on regular equivalence and introduces capsule network to generate more discriminative node representations for nodes. On the other hand, it uses GRU to capture the intrinsic relationship between the node's historical behaviors and current behaviors, and thereby influence the generation of node embeddings at the next snapshot. We verify the effectiveness of the proposed model by conducting comprehensive experiments on synthetic and real-world networks.

Acknowledgment

This work is supported by NSFC program (No. 62272338, No. 61902278), the special fund for basic scientific research business expenses of Zhejiang provincial colleges and universities (GK229909299001-008), China Postdoctoral Science Foundation (No. 2022T150470).

References

[1] G. Nikolentzos and M. Vazirgiannis, "Learning structural node representations using graph kernels," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 5, pp. 2045–2056, 2021.

[2] M. Heimann, H. Shen, T. Safavi, and D. Koutra, "Regal: Representation learning-based graph alignment," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 117–126.

[3] D. Jin, M. Heimann, R. A. Rossi, and D. Koutra, "Node2bits: Compact time-and attribute-aware node representations for user stitching," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2019, pp. 483–506.

[4] R. A. Rossi, D. Jin, S. Kim, N. K. Ahmed, D. Koutra, and J. B. Lee, "On proximity and structural role-based embeddings in networks: Misconceptions, techniques, and applications," *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 14, pp. 1–37, 2020.

[5] P. Jiao, X. Guo, T. Pan, W. Zhang, Y. Pei, and L. Pan, "A survey on role-oriented network embedding," *IEEE Transactions on Big Data*, vol. 8, no. 4, pp. 933–952, 2022.

[6] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li, "Rolx: structural role extraction and mining in large graphs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1231–1239.

[7] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos, "It's who you know: graph mining using recursive structural features," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 663–671.

[8] D. Jin, M. Heimann, T. Safavi, M. Wang, W. Lee, L. Snider, and D. Koutra, "Smart roles: inferring professional roles in email networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 2923–2933.

[9] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "Struc2vec: Learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 385–394.

[10] K. Tu, P. Cui, X. Wang, P. S. Yu, and W. Zhu, "Deep recursive network embedding with regular equivalence," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 2357–2366.

[11] D. Xu, W. Cheng, D. Luo, X. Liu, and X. Zhang, "Spatio-temporal attentive rnn for node classification in temporal attributed graphs," in *Twenty-Eighth International Joint Conference on Artificial Intelligence IJCAI-19*, 2019, pp. 3947–3953.

[12] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 4292–4293.

[13] D. Zhu, P. Cui, Z. Zhang, J. Pei, and W. Zhu, "High-order proximity preserved embedding for dynamic networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 11, pp. 2134–2144, 2018.

[14] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dysat: Deep neural representation learning on dynamic graphs via self-attention networks," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 519–527.

[15] L. Wang, C. Huang, Y. Lu, W. Ma, R. Liu, and S. Vosoughi, "Dynamic structural role node embedding for user modeling in evolving networks," *ACM Transactions on Information Systems*, vol. 40, no. 3, pp. 1–21, 2021.

[16] Y. Wang, P. Li, C. Bai, and J. Leskovec, "Tedic: Neural modeling of behavioral patterns in dynamic social interaction networks," in *Proceedings of the Web Conference 2021*, 2021, pp. 693–705.

[17] J. Carlen, J. de Dios Pont, C. Mentus, S.-S. Chang, S. Wang, and M. A. Porter, "Role detection in bicycle-sharing networks using multilayer stochastic block models," *Network Science*, vol. 10, no. 1, pp. 46–81, 2022.

[18] E. Evans, W. Guo, A. Genctav, S. Tari, C. Domeniconi, A. Murillo, J. Chuang, L. AlSumait, P. Mani, and N. El-Zehiry, *Role Detection and Prediction in Dynamic Political Networks*, 2021, pp. 233–252.

[19] R. A. Rossi, B. Gallagher, J. Neville, and K. Henderson, "Modeling dynamic behavior in large evolving graphs," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, 2013, pp. 667–676.

[20] Y. Pei, J. Zhang, G. Fletcher, and M. Pechenizkiy, "Dynmf: Role analytics in dynamic social networks," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 2018, pp. 3818–3824.

[21] R. A. Rossi and N. K. Ahmed, "Role discovery in networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1112–1131, 2015.

[22] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 3859–3869.

[23] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[24] S. Gilpin, T. Eliassi-Rad, and I. Davidson, "Guided learning for role discovery (glrd) framework, algorithms, and applications," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 113–121.

[25] P. V. Gupte, B. Ravindran, and S. Parthasarathy, "Role discovery in graphs using global features: algorithms, applications and a novel evaluation strategy," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, 2017, pp. 771–782.

[26] C. Donnat, M. Zitnik, D. Hallac, and J. Leskovec, "Learning structural node embeddings via diffusion wavelets," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1320–1329.

[27] R. A. Rossi, N. K. Ahmed, E. Koh, S. Kim, A. Rao, and Y. Abbasi-Yadkori, "A structural graph representation learning framework," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 483–491.

[28] N. K. Ahmed, R. A. Rossi, J. B. Lee, T. L. Willke, R. Zhou, X. Kong, and H. Eldardiry, "Role-based graph embeddings," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2401–2415, 2022.

[29] X. Ma, G. Qin, Z. Qiu, M. Zheng, and Z. Wang, "Riwalk: Fast structural node embedding via role identification," in *2019 IEEE International Conference on Data Mining (ICDM)*, 2019, pp. 478–487.

[30] Y. Pei, X. Du, J. Zhang, G. Fletcher, and M. Pechenizkiy, "Struc2gauss: Structural role preserving network embedding via gaussian embedding," *Data Mining and Knowledge Discovery*, vol. 34, no. 4, pp. 1072–1103, 2020.

[31] J. Guo, L. Xu, and J. Liu, "Spine: Structural identity preserved inductive network embedding," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2019, pp. 2399–2405.

[32] R. Jin, V. E. Lee, and H. Hong, "Axiomatic ranking of network role similarity," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 922–930.

[33] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 1150–1160.

[34] X. Guo, W. Zhang, W. Wang, Y. Yu, Y. Wang, and P. Jiao, "Role-oriented graph auto-encoder guided by structural information," in *International Conference on Database Systems for Advanced Applications*, 2020, pp. 466–481.

[35] W. Zhang, X. Guo, W. Wang, Q. Tian, L. Pan, and P. Jiao, "Role-based network embedding via structural features reconstruction with degree-regularized constraint," *Knowledge-Based Systems*, vol. 218, 2021.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014.

[37] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 2648–2720, 2020.

[38] G. Xue, M. Zhong, J. Li, J. Chen, C. Zhai, and R. Kong, "Dynamic network embedding survey," *Neurocomputing*, vol. 472, pp. 212–223, 2022.

[39] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Dyrep: Learning representations over dynamic graphs," in *International Conference on Learning Representations*, 2019.

[40] E. Hajiramezanali, A. Hasanzadeh, N. Duffield, K. Narayanan, M. Zhou, and X. Qian, "Variational graph recurrent neural networks," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 10 701–10 711.

[41] P. Goyal, S. R. Chhetri, and A. Canedo, "Dyngraph2vec: Capturing network dynamics using dynamic graph representation learning," *Knowledge-Based Systems*, vol. 187, p. 104816, 2020.

[42] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro- and macro-dynamics," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 469–478.

[43] J. Li, H. Dani, X. Hu, J. Tang, Y. Chang, and H. Liu, "Attributed network embedding for learning in a dynamic environment," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 387–396.

[44] S. Mahdavi, S. Khoshraftar, and A. An, "Dynnode2vec: Scalable dynamic network embedding," *2018 IEEE International Conference on Big Data (Big Data)*, pp. 3762–3765, 2018.

[45] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[46] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[47] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1165–1174.

[48] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *International Conference on Learning Representations*, 2018.

[49] D. Jin, Z. Li, L. Yang, D. He, P. Jiao, and L. Zhai, "Adversarial capsule learning for network embedding," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 214–221.