Expert Systems With Applications 240 (2024) 122596

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Deep autoencoder architecture with outliers for temporal attributed network embedding

Xian Mo [a,c,*], Jun Pang [b], Zhiming Liu [c]

[a] School of Information Engineering, Ningxia University, Yinchuan 750021, China
[b] Faculty of Science, Technology and Medicine, University of Luxembourg, Esch-sur-Alzette L-4364, Luxembourg
[c] College of Computer & Information Science, Southwest University, Chongqing 400700, China

ARTICLE INFO

Keywords:
Temporal attributed networks
Network embedding
Outlier nodes
Autoencoders

ABSTRACT

Temporal attributed network embedding aspires to learn a low-dimensional vector representation for each node in each snapshot of a temporal network, which can be capable of various network analysis tasks such as link prediction and node classification. In temporal attributed networks, attribute similarities or link structures of certain nodes may deviate from the regular nodes of the community they belong to, which are called community outlier nodes. However, many existing embedding methods consider only the link structures and their attributes of the nodes adhere to the community structure of the network while ignoring outlier nodes, this can affect the embedding performance of the regular nodes. In this paper, we propose a temporal attributed network embedding framework with outliers, based on autoencoders, to solve the problem. In particular, we propose an outlier-aware autoencoder to model the node information, which combines the current snapshot and previous snapshots to jointly learn embedded vectors of nodes in the current snapshot of a temporal network. In feature preprocessing, we propose a simplified higher graph convolutional mechanism to incorporate attribute information into link structure information, which can leverage attribute features into link structure. Experimental results on node classification and link prediction reveal that our model is competitive against various baseline models.

1. Introduction

In recent years, network embedding techniques have attracted more and more researchers' attention (Amara, Taieb, & Aouicha, 2021). Such network embedding techniques have been verified to be very effective in community detection (Bandyopadhyay, Lokesh, & Murty, 2019), node classification (Zou et al., 2021), and link prediction (Jiao et al., 2022). In the real world, however, many networks have rich attributes and temporal information, which we call temporal attributed networks (Liu, Huang, Yu, & Dong, 2021). For instance, users in social attributed networks may contain interest, gender, and age attributes, which may evolve over time. Motivated by the success of network embedding in plain temporal networks for link prediction (Sankar, Wu, Gou, Zhang, & Yang, 2020), node classification (Zou et al., 2021) tasks, etc., several researchers utilize similar ideas for temporal attributed networks (Liu et al., 2021). Nonetheless, there are still two key challenges in the node embedding of temporal attributed networks, namely:

1. How to incorporate effectively attribute information into structure information in temporal attributed networks to solve the highly sparse of structural features in feature preprocessing?
2. In a noisy environment, how to address the outlier nodes in temporal attributed networks and learn more robust embeddings?

Real-life networks often come with missing connections between nodes. Moreover, the rows of an adjacency matrix, representing a network, can only obtain the observed links as they are. Many previous network embedding methods (Meng, Liang, Zhang, Mccreadie, & Ounis, 2020; Wei, Hu, Bai, Xia, & Pan, 2019) consider structure information and attribute information separately, without considering how to incorporate attribute information into structural information to address the first challenge. Social science theories (Miller, Lynn, & James, 2001) suggest that attribute information can be incorporated into structure information as complementary content to enhance the performance of many downstream applications. Especially in sparse networks, attribute information can be very useful complementary content to learn better representation for networks. Hence, it is essential to incorporate attribute information into structural information to gain insights and comprehend the complex behaviors of networks.

In temporal attributed networks, many existing network embedding methods assume that the network's nodes are well-connected in their

* Corresponding author at: School of Information Engineering, Ningxia University, Yinchuan 750021, China.
E-mail addresses: mxian168@nxu.edu.cn (X. Mo), jun.pang@uni.lu (J. Pang), zhimingliu88@swu.edu.cn (Z. Liu).

https://doi.org/10.1016/j.eswa.2023.122596
Received 1 April 2022; Received in revised form 2 November 2023; Accepted 14 November 2023
Available online 16 November 2023
0957-4174/

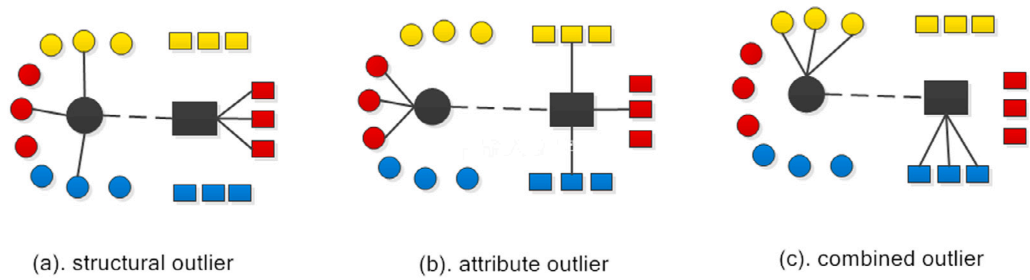


Fig. 1. Three types of outliers.

respective communities and attributes are consistent with the topological structure (Sankar et al., 2020; Wei et al., 2019) overlooking the second challenge. However, in a real-life network, node structures or their attributes may deviate from the property of the community to which they belong. One node in the network has edges connected to other nodes in different communities or its attributes are more similar to the attributes of nodes from different communities. We call this kind of nodes as outlier nodes (Bandyopadhyay, N, Vivek, & Murty, 2020). Recently, some researchers have defined outlier nodes from different perspectives. For example, Ji et al. (2019) define three types of outliers from the multiple views, namely attribute outlier, class outlier, and class-attribute outlier. Huang et al. (2021) define abnormal nodes and subgraphs from nodes' structure and attributes for anomaly detection. Du, Yu, Chu, Jin, and Chen (2022) consider outliers that are mixed within normal object regions or around dense clusters for outlier detection. In real life, networks can often be divided into different communities (Keikha, Rahgozar, & Asadpour, 2018). However, the above methods do not take into account network community structure, i.e., defining the outlier nodes by considering nodes' structure and attributes with respect to the communities. Moreover, these methods are mainly used for outlier detection or anomaly detection, but rarely used for temporal attributed network embedding. Fig. 1 exhibits three kinds of outliers that we involve in temporal attributed networks. We assume that each subgraph in Fig. 1 is a snapshot of a temporal attributed network, where the circles represent nodes, while the rectangles represent the attribute of these nodes. Lines between nodes represent edges of networks, while lines between two attributes represent that two attributes are similar. We adopt different colors to describe different communities. We use the larger black circle and black rectangle to underline the outlier node and its associated attribute respectively. In Fig. 1(a), the larger black node has edges connected to other nodes in different communities, i.e., its structural neighborhood is inconsistent. Hence, the larger black node is considered as a structural outlier node. In Fig. 1(b), the attribute of the larger black node is similar to node attributes of different communities, i.e., its attribute neighborhood is inconsistent. Hence, the larger black node is considered as an attribute outlier node. In Fig. 1(c), the larger black node belongs to a community structurally but it has a different community in the aspect of attribute similarity. Hence, the larger black node is considered as a combined outlier node.

An empirical analysis showed that a few manually seeded outlier nodes in a synthetic static network using the classical embedding algorithm Node2Vec (Grover & Leskovec, 2016) can largely effect the embedding of the regular nodes. The experiment uses a synthetic network containing 3 communities and a total of 60 nodes, which performs Node2Vec (Grover & Leskovec, 2016) on the synthetic network with embedding dimension size equal to 2. The learning node representations perform well by separating the communities far apart. Next, the experiment inserts only 6 outlier nodes (having edges to all the three communities randomly) in the network, this enables these outlier nodes to pull the embeddings from different communities together so that the community cannot be separated and to largely affect the embedding of the regular nodes in the communities (see Bandyopadhyay et al. (2020) for details). For real-life networks, there always exist outlier nodes as pointed by Ding, Li, Bhanushali, and Liu (2019). This reveals the importance of paying attention to outlier nodes when learning node embeddings, as they may seriously affect the embedding performance of the regular nodes. However, there is no previous work explicitly considering the effect of outlier nodes in the temporal attributed network embedding. Hence, it is necessary to consider outlier nodes to gain insights and comprehend the complex behaviors of temporal attributed networks.

To tackle the two identified challenges, we propose a temporal attributed network embedding framework with outliers based on autoencoder (TAOA) for temporal attributed network embedding in an unsupervised way. Our major contributions in this work can be summarized as follows.

- We propose a new model TAOA to learn node embedding in temporal attributed networks. In particular, the model utilizes an outlier-aware autoencoder to model the node information, which combines the current network snapshot and previous snapshots jointly in order to learn embedded vectors for nodes in the network.
- We propose a simplified higher graph convolutional mechanism (SHGC) to preprocess attribute features for each node in each snapshot of temporal attributed networks. The SHGC incorporates attribute information into link structure information, which can leverage attribute information into link structure features.
- Experimental results on node classification and link prediction reveal that our model is competitive against various baseline models.

The rest of this paper is organized as follows. We summarize several related works in Section 2. Section 3 formulates the problem. We present our proposed framework for temporal attributed network embedding in Section 4. Sections 5 and 6 discusses the experimental results and concludes the paper respectively.

2. Related work

Plain network embedding in static and temporal networks has been extensively studied (Amara et al., 2021), hence we introduce related work for attributed network embedding in this section.

2.1. Static attributed network embedding

Because networks contain rich attribute information in real life, it is essential to study attributed network embedding. MNMF (Wang et al., 2017) considers the community structure in network embedding. It assumes that node representations in the same community should be more similar. NECS (Li, Wang, Zhang, Zhang, & Chang, 2019) utilizes the high-order proximity and the community structure to learn the network embedding. Because the above methods are based on matrix operation, their ability to extract high-dimensional nonlinear features is limited.

Recently, some attributed network embedding approaches based on deep neural network have been developed. For instance, GCN (Kipf & Welling, 2017) adopts an efficient variant of convolutional neural networks to learn node embeddings for attributed networks. ANRL (Zhang et al., 2018) adopts deep neural network architectures to learn node embeddings while capturing the high non-linearity features. GAT (Velickovic et al., 2018) leverages masked self-attentional layers to assign different importance to different nodes in static attributed networks. The model (Meng, Liang, Fang, & Xiao, 2019) properly interoperates attribute features of nodes by the neighborhood aggregation procedure using GCN. CSAN (Meng et al., 2020) proposes a variational auto-encoder algorithm for attributed network embedding. The approach HANS proposed by Zhao, Chen, Chen, Zhang, and Tang (2022) fusions the nodes and hierarchical labels via a attention-based fusion module and attributes for network embedding. Finally, CoANE (Hsieh & Li, 2023) uses the convolutional mechanism considering the specific combination between the network attributes and topological structure for attributed network embedding. More recently, self-supervised learning (Lin, Tian, Hou, & Zhao, 2022) has achieved success in graph learning to solves the problem of label scarcity. NCL (Lin et al., 2022) combines neighbors of a node with graph structure and semantic space into contrastive pairs to learn node embeddings. CVAEs (Sun et al., 2023) present a disentangled conditional variational autoencoder with a contrastive learning loss function for explainable recommendation. DCRec (Yang et al., 2023) adopts adaptive conformity-aware augmentation using a debiased contrastive learning to learn node embeddings. IDCL (Wang et al., 2023) constructs a disentangled graph contrastive learning framework using intent-wise contrastive learning. LightGCL (Cai, Huang, Xia, & Ren, 2023) uses singular value decomposition to generate a more robust recommendation model for graph contrastive learning.

In the past few years, some static attributed network embedding methods considering outlier nodes have been proposed, and the main strategy is similar to distributionally robust learning in dealing with outliers (Sadeghi, Ma, Li, & Giannakis, 2021). Sadeghi et al. (2021) utilize distributionally robust semi-supervised learning to handle networks with uncertain node attributes and mismatches between training and testing data distribution exists. DR-DSGD (Issaid, Elgabli, & Bennis, 2022) uses a Kullback–Leibler regularization function to solve a regularized distributionally robust learning problem. Wang, Pun and So (2022) propose robust-against-uncertainties in the observed signals using distributionally robust optimization. However, these methods also do not consider network communities, i.e., node structures or their attributes may deviate from the property of the community to which they belong. Moreover, distributionally robust learning is usually an optimization model rather than a deep learning model and is rarely used in network representation learning. Liang, Jacobs, Sun and Parthasarathy (2018) presents a semi-supervised algorithm to detect outliers, which preserves topological proximity, attributes affinity, and label similarity of nodes, and this algorithm can alleviate noise effects from outliers in a partially labeled attributed network. However, it is not easy to obtain labeled data for real-life networks. Bandyopadhyay et al. (2019) develops an unsupervised algorithm to reduce the effect of outlier nodes. However, the method is based on matrix decomposition with restricted ability to extract the correlation of high dimensional features. DONE (Bandyopadhyay et al., 2020) and AdONE (Bandyopadhyay et al., 2020) adopt a deep autoencoder architecture to minimize the effect of outliers, which update parameters using stochastic gradient descent and adversarial learning in an unsupervised way, respectively. Nonetheless, there is no previous work explicitly considering the effect of outlier nodes in the temporal attributed network embedding. Moreover, these methods above are all limited to deal with static attributed networks because of ignoring temporal information.

2.2. Temporal attributed network embedding

Recently, attributed network embedding approaches on temporal networks have gained a lot of popularity. DANE (Li et al., 2017) utilizes matrix perturbation theory to embed vectors for tackling the problem of attributed network embedding in temporal networks. SLIDE (Li, Cheng, Liang, & Liu, 2018) uses a matrix sketching strategy in temporal attributed network embeddings. Toffee (Ma, Zhang, Lou, Xiong, & Ho, 2021) utilizes the tensor–tensor product operator to encode the cross-time information using tensor decomposition to capture periodic changes in the evolving networks. RDAM (Yao et al., 2021) presents a reinforcement learning-based temporal attribute matrix representation method for network embedding. TPANE (Li & Lai, 2022) utilizes temporal path adjacency measures to capture the temporal dependency between edges. DyHNE (Wang, Lu et al., 2022) captures the semantics and structure of a network using a proposed meta-path-based multi-order relationship. However, matrix factorization methods may bring high computational costs because the networks in real life are sparse and large.

DUWE (Liang, Zhang, Ren and Kanoulas, 2018) proposes a dynamic user and word embedding model to tracks the semantic representation of users and words over time. NetWalk (Yu et al., 2018) encodes the node of temporal networks to vector by clique embedding. T-GCN (Zhao et al., 2020) combines with the gated recurrent unit (GRU) and the graph convolutional network (GCN) to capture spatial–temporal features for traffic prediction. DySAT (Sankar et al., 2020) computes node representations through joint self-attention, and TemporalGAT (Fathy & Li, 2020) adopts GAT and temporal convolutional networks (Bai, Kolter, & Koltun, 2018) to learn temporal network representation. However, none of these methods can be applied to attributed network datasets to verify its performance.

LDANE (Wei et al., 2019), a lifelong learning framework, automatically expands the deep neural networks to capture highly nonlinear features for attributed temporal networks. CDAN (Meng et al., 2020) learns low-dimensional embeddings to capture affinities between nodes and attributes. MTSN (Liu et al., 2021) captures the local high-order structures and temporal evolution using motif-preserving methods for temporal attributed networks. TVAE (Jiao et al., 2022) adopts a variational autoencoder to detect the shift of temporal networks. Additionally, DynGNN (Zhang, Yao, Yao, Huang, & Chen, 2023) embeds recurrent neural networks into graph neural networks to capture more fine-grained network evolving. More recently, some temporal network embedding methods based on self-supervised learning have appeared. GTEA (Xie et al., 2023) aggregates features of neighborings and the corresponding edge embeddings in a self-supervised form for the temporal graph learning. CLDG (Xu et al., 2023) adopts contrastive learning process to learn node embeddings on temporal graphs in unsupervised scenarios. However, all the above methods ignore the outlier nodes when learning node embeddings for temporal attributed networks. Therefore, it is necessary to consider outlier nodes to gain insights and comprehend the complex behaviors of temporal attributed networks.

3. Problem definition

The main notations used in this paper are summarized in Table 1. We then introduce some definitions and formally define the research problem.

Definition 1 (*Attributed Network*). An attributed network is defined as: $G = \langle V, E, H \rangle$, where $V = \{v_1, \dots, v_n\}$ represents a set of nodes, n is the number of nodes, $E \sqsubseteq V \times V$ represents a set of links (edges) among the nodes, e_{ij} means an edge exists between nodes v_i and v_j, $H \in R^{n\times c}$ is the attribute matrix of nodes, and c is the attribute dimensions.

Table 1
Notations used in this paper.

Notations	Descriptions
$G = \langle V, E, H\rangle$	Attributed network
$V = \{v_1, \ldots, v_n\}$	Node set consisting of n nodes
e_{ij}	An edge exists between nodes v_i and v_j
$H \in R^{n\times c}$	Attribute matrix of nodes
c	Attribute dimensions
E_t	Edge set among the nodes in timestamp t
H_t	Attribute matrix of nodes in timestamp t
$\{\widehat{G_1}, \ldots, \widehat{G_t}, \ldots, \widehat{G_T}\}$	A sequence of network snapshots
$t \in \{1, \ldots, T\}$	Timestamps of network snapshots
$C = \{C_1, \ldots, C_K\}$	K communities
$f^t : v_i \to R^k$	A mapping function
k	Embedding dimensions of node v_i
L_t	Symmetrically normalized Laplace matrix
A_t	Adjacency matrix
D_t	Diagonal degree matrix
P	Powers of L_t
γ	Damping factor
$C_t = \{c_1^t, \ldots, c_n^t\}$	Output of n nodes of SHGC
z_i^t $(z_i^t \in R^k)$	Embedded vector of node i
$\dot{o}_i^t$	Outlier node scores for node i

Definition 2 (*Temporal Attributed Network*). The network studied in this paper assumes that the set of nodes is fixed, and the edges between nodes can evolve over time. Therefore, a temporal attributed network can be represented graphically: $G_t = \langle V, E_t, H_t\rangle$, where V represents a set of nodes, E_t represents a set of links (edges) among the nodes in timestamp t, and H_t represents the attribute matrix of nodes in timestamp t. Additionally, a temporal attributed network can generate a sequence of network snapshots $\{\widehat{G_1}, \ldots, \widehat{G_t}, \ldots, \widehat{G_T}\}$, where $t \in \{1, \ldots, T\}$ represents the timestamps.

Definition 3 (*Community*). An attributed network $G = \langle V, E, H\rangle$ can be divided into K communities, i.e., $C = \{C_1, \ldots, C_K\}$ such that $K \ll n$ (n is the number of nodes) and $\cup_{k=1}^{K} C_k = V$, and all the communities C_k $(1 \le k \le K)$ are non-empty, mutually exclusive subsets of V. Each community C_k is a collection of the nodes that have commonalities of the network and is a subgraph of the network.

Definition 4 (*Outlier Nodes*). For each snapshot $\widehat{G_t}$ of a temporal attributed network, we can divide it into K communities, i.e., $C = \{C_1, \ldots, C_k, \ldots, C_K\}$. We assume v_i is a node in the community C_k. If v_i has random edges to nodes from different communities, or the attributes of v_i are similar to attributes of nodes from different communities, or v_i belongs to a community structurally but it has a different community in terms of attribute similarity, for all these cases we define v_i as an outlier node.

Definition 5 (*Temporal Attributed Network Embedding*). We can split a temporal attributed network into a sequence of snapshots $\{\widehat{G_1}, \ldots, \widehat{G_T}\}$ by timestamp t. For each snapshot, we aim to learn a mapping function $f^t : v_i \to R^k$, where $v_i \in V$ and k represents dimensions and $k \ll |V|$. The function f^t preserves the node similarity between v_i and v_j on the topological structure, node attributes and evolution patterns of a given temporal network from timestamp 1 up to t. Moreover, it also needs to decrease the effect of outlier nodes.

4. Our model

In this section, we will focus on our new model, a Temporal Attributed network embedding framework with Outliers, based on deep Autoencoder (TAOA), for temporal attributed network embedding. In our model, we first introduce a simplified higher graph convolutional mechanism (SHGC) to preprocess attribute features for each node of each snapshot in temporal attributed networks (Section 4.1). We then introduce an attributed network embedding framework based on a deep autoencoder to solve the outlier problem for temporal attributed network embeddings (Section 4.2).

4.1. Feature preprocessing

For a temporal attributed network $G_t = \langle V, E_t, H_t\rangle$, the structure features are highly sparse and do not incorporate the attribute information. Inspired by the Social science theories (Miller et al., 2001), attribute information can be incorporated into link structure as a complementary content to enhance the performance of many downstream applications. Moreover, as motivated by the work of simple graph convolution (SGC) (Wu et al., 2019), we propose a simplified higher graph convolutional mechanism (SHGC) to incorporate attribute features into link structure for each node v of each snapshot. Compared with GCN (Kipf & Welling, 2017), SGC removes nonlinearities collapsing weight matrices between consecutive layers. Experimental results in many downstream applications demonstrate that these simplifications do not negatively impact the accuracy. In each layer of SGC, the hidden representations are averaged among neighbors that are one hop away. After passing k layers, a node obtains the node feature information from the k−hop nodes in the graph. The mechanism does not hold the feature information of each layer separately, which may miss some valuable information. Inspired by SGC, we propose SHGC to hold valuable information of each layer in a simple way for each snapshot of a temporal attributed network for dealing the problem. SHGC can mix feature representations of neighbors at various distances to learn neighborhood mixing relationships. In particular, it combines 1-hop to k-hop neighbors in distinct feature spaces to effectively aggregate features of different hops in the network, namely:

$$C_t = \frac{1}{P+1}\sum_{i=0}^{P} \gamma^{P-i} L_t^{P-i} H_t \qquad 0 < \gamma < 1 \tag{1}$$

where $H_t \in R^{n\times c}$ is the attribute matrix of nodes at timestamp t, c is the attribute dimension, and n is the number of network nodes. L_t is a symmetrically normalized Laplace matrix, which reveals the link structure of the network at timestamp t. It can be constructed by the following formula.

$$L_t = D_t^{-\frac{1}{2}} A_t D_t^{\frac{1}{2}} \tag{2}$$

where A_t is an adjacency matrix at timestamp t. D_t is a diagonal degree matrix with $D_t^{mm} = \sum_n A_t^{mn}$, where m represents a row and n represents a column. The D_t^{mm} represents the entry of row m and column m of D_t and the A_t^{mn} represents the entry of row m and column n of A_t. L_t^{P-i} denotes the matrix L_t multiplied by itself $P-i$ times, and $P-i$ is the powers of L_t, ranging from 0 to P. For example, L_t^3 represents 3-hop neighbors in feature spaces in timestamp t. The γ is the damping factor, which holds more information for fewer hops. In this way, it mixes feature representations of neighbors at various distances. C_t is the output representing the final node information, which incorporates attribute features information from immediate to further feature spaces into link structure and reveals attribute features of networks.

Time complexity analysis: The $L_t^{P-i}H$ can be calculated by right-to-left multiplication. More precisely, for example, if we set $P = 3$ and $i = 0$, $L_t^3 H$ can be calculated by $L_t(L_t(L_t H))$. We adopt the sparse matrix with m non-zero entries to store L_t, so the computational time of the $L_t^{P-i}H$ is $O(P \times m \times c)$ for each timestamp t, where the c represents the attribute dimensions of attribute matrix H_t. The γ can be treated as a constant, so the time complexity of the SHGC is $O(P \times m \times c)$.

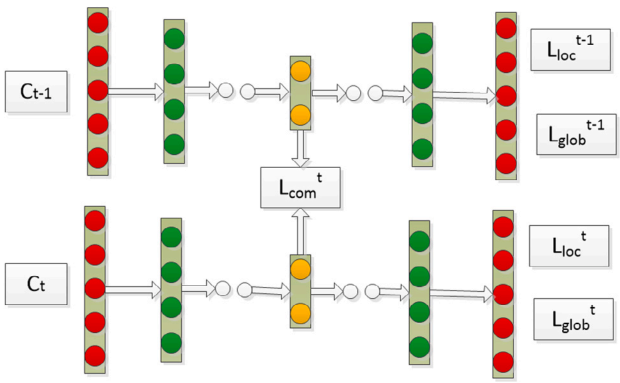


Fig. 2. TAOA architecture.

4.2. Node embedding

To encode the node information, we design the TAOA architecture (see Fig. 2) for temporal attributed network embedding. We adopt two parallel autoencoders to encode and decode the current snapshot and the previous snapshot of the node information for temporal attributed network embeddings. The information of each node of the current snapshot and the previous snapshot incorporate attribute features information into link structure by the SHGC algorithm. For the sake of interpretability and easy to calculate, we only consider two adjacent snapshots in our model, and we will consider more snapshots in future work. Our model consists of an encoder and a decoder, which considers outlier nodes in the reconstruction process. The embedded vector z_i ($z_i \in R^k$) for any node i is obtained from the hidden layer of the model.

Algorithm 1: Procedure of the our method

Input : $G_t = \langle V, E_t, H_t \rangle$: a temporal attributed network ;
T: number of timestamps ;
γ: damping factor;
P: Powers of L_t;
Output: z_i^t: embedding vector of each node i in the timestamp t;

Generating a sequence of snapshots $\{\widehat{G}_1, \dots, \widehat{G}_t, \dots, \widehat{G}_T\}$ from $G_t = \langle V, E_t, H_t \rangle$;
for $t \in \{1, \dots, T\}$ **do**
  Compute C_t based on Eq. (1);
end
Define outlier node scores $\dot{o}_i^t$ based on Eq. (3);
Preprocess the outlier node scores: $o_i^t = \log \frac{1}{\dot{o}_i^t}$;
Construct global structure proximity loss function: Eq. (4) and (5);
Construct local structure proximity loss function: Eq. (6) and (7);
Construct temporal proximity loss function: Eq. (8);
Construct objective function $min\ L_{TAOA}^t$ based on Eq.(4), (5), (6), (7) and (8);
Update $\dot{o}_i^t$ by Eq. (12) and use stochastic gradient descent algorithm to obtain embedding z_i^t of each node i for the timestamp t;

We first introduce outlier node scores of each node i for each snapshot. We denote $\dot{o}_i^t$ as outlier node scores for node i ($i \in 1 \dots N$) in timestamp t. To better understand the outlier node scores, we assume:

$$\sum_{i=1}^{N} \dot{o}_i^t = 1 \qquad o_i^t > 0 \tag{3}$$

For a perfect network (e.g., no inter-community edges in a network, where attributes are perfectly coherent with the link structure), there are no outlier nodes. Hence, we initialize that the outlier node score $\dot{o}_i^t$ for each node i is constant equal to $\frac{1}{N}$, which is a discrete probability distribution and represents the probability of the node i to be an outlier node. We preprocess the outlier node scores by formula $o_i^t = \log \frac{1}{\dot{o}_i^t}$ Hence, the larger the outlier node scores $\dot{o}_i^t$ are, the outlier node scores o_i^t are smaller than the other nodes, and so the contribution to loss function (Eqs. (4), (5), (6), and (7)) from this node would be less. In this way, it can reduce the impact of outlier nodes on regular node embedding.

Based on the outlier node scores, we formulate the loss functions for our model. We first preserve global structure proximity by minimizing reconstruction loss in current timestamp t and previous $t-1$ for each snapshot.

$$L_{glob}^t = \frac{1}{N} \sum_{i=1}^{N} o_i^t \| c_i^t - \hat{c}_i^t \|^2 \tag{4}$$

$$L_{glob}^{t-1} = \frac{1}{N} \sum_{i=1}^{N} o_i^{t-1} \| c_i^{t-1} - \hat{c}_i^{t-1} \|^2 \tag{5}$$

where c_i^t and c_i^{t-1} are ith row of the matrix C_t and C_{t-1} in current timestamp t and in previous timestamp $t-1$ respectively in Section 4.1. $\hat{c}_i^t$ and $\hat{c}_i^{t-1}$ are the reconstructed outputs of the autoencoders for node i in current timestamp t and previous timestamp $t-1$. We adopt the Leaky ReLU nonlinearity function with negative input slope $\alpha = 0.2$ for K layers encoders and decoders. The o_i^t and o_i^{t-1} are the outlier scores of each node i in current timestamp t and previous $t-1$. The smaller the outlier score o_i^t for some outlier node i in current timestamp t, the contribution to loss from this node would be less.

The next component of the loss function is used to preserve the local structure proximity, which indicates that nodes with edges connected should be similar in the embedded space.

$$L_{loc}^t = \frac{1}{N} \sum_{i=1}^{N} o_i^t \frac{1}{|N(i)|} \sum_{j \in N(i)} \| z_i^t - z_j^t \|^2 \tag{6}$$

$$L_{loc}^{t-1} = \frac{1}{N} \sum_{i=1}^{N} o_i^{t-1} \frac{1}{|N(i)|} \sum_{j \in N(i)} \| z_i^{t-1} - z_j^{t-1} \|^2 \tag{7}$$

The z_i^t and z_i^{t-1} are the embeddings of node i in current timestamp t and previous timestamp $t-1$, which can obtain from the hidden layers of encoders. $N(i)$ is the neighbors of node i in the network.

Since the current snapshot topological structure of temporal attributed networks is derived from the previous snapshot topology structure, the embedding vector of the current timestamp t and previous timestamp $t-1$ are highly correlated. Hence we formulate the last component (combining the embedding vector of the current timestamp t and previous timestamp $t-1$) of the loss function as:

$$L_{com}^t = \frac{1}{N} \sum_{i=1}^{N} \| z_i^t - z_i^{t-1} \|^2 \tag{8}$$

Next, we combine Eqs. (4)–(8) and jointly minimize the following objective function using stochastic gradient descent algorithm to obtain embedding z_i^t of each node i for the current timestamp t:

$$minL_{TAOA}^t = a_1 L_{glob}^t + a_2 L_{glob}^{t-1} + a_3 L_{loc}^t + a_4 L_{loc}^{t-1} + a_5 L_{com}^t \tag{9}$$

We adopt the closed-form update rule (Bandyopadhyay et al., 2020) to update $\dot{o}_i^t$ as follows. We update $\dot{o}_i^{t-1}$ in a similar way. Since the loss L_{TAOA}^t is convex when other variables are fixed, we use an alternating minimization technique to update each variable. The Lagrangian of Eq. (9) with respect to the constraint (3) can be written as the following and we ignore the terms that do not contain $\dot{o}_i^t$.

$$\begin{aligned} L = \lambda\Big(\sum_{i=1}^{N} \dot{o}_i^t - 1\Big) + a_1\Big(\frac{1}{N} \sum_{i=1}^{N} o_i^t \| c_i^t - \hat{c}_i^t \|^2\Big) + \\ a_3\Big(\frac{1}{N} \sum_{i=1}^{N} o_i^t \frac{1}{|N(i)|} \sum_{j \in N(i)} \| z_i^t - z_j^t \|^2\Big) \end{aligned} \tag{10}$$

Table 2
Statistics of the three attributed networks datasets.

Datasets	Nodes	Edges	Attributes	Labels	Snapshots
RHNs	55,863	858,490	86	–	7
BlogCatalog	5,196	171,743	8,189	6	10
Flickr	7,575	239,738	12,047	9	10

The λ represents the Lagrangian constant. Equating the partial derivative of Eq. (10), we can obtain the following formula:

$$\dot{o}_i^t = \frac{a_1 \|c_i^t - \hat{c}_i^t\|^2 + a_3 \frac{1}{|N(i)|} \sum_{j \in N(i)} \|z_i^t - z_j^t\|^2}{N\lambda} \tag{11}$$

Using Formula (3), we can thus obtain:

$$\dot{o}_i^t = \frac{a_1 \|c_i^t - \hat{c}_i^t\|^2 + a_3 \frac{1}{|N(i)|} \sum_{j \in N(i)} \|z_i^t - z_j^t\|^2}{\sum_{i=1}^{N} (a_1 \|c_i^t - \hat{c}_i^t\|^2 + a_3 \frac{1}{|N(i)|} \sum_{j \in N(i)} \|z_i^t - z_j^t\|^2)} \tag{12}$$

For clarity, algorithm 1 summarizes the main procedure of the proposed method.

5. Experiments

In this section, we introduce the selected datasets and baseline models. We report the experimental results on temporal attributed networks to demonstrate the performance of our TAOA model.

5.1. Datasets and baseline models

We adopt three attributed networks from diverse fields. We use the Reddit Hyperlink Networks (abbreviated as: RHNs),[1] and two social attributed network,[2] BlogCatalog and Flickr, to demonstrate the effectiveness of the TAOA model. All networks have different features, with the statistics information shown in Table 2.

- **RHNs**: The dataset consists of a hyperlink network on Reddit, where nodes represent subreddits (a subreddit is a community), an edge between two nodes represents that a post between the two subreddits, attributes of each node indicate the posts that create hyperlinks from one subreddit to another, and the network data is collected from Jan 2014 to April 2017. For our experiments, we split it into 7 snapshots by half a year.
- **BlogCatalog**: This is a social network dataset from the BlogCatalog website, where nodes represent users, edges represent user interaction, nodes' attributes indicate a description of a blogger, and labels indicate the topic categories.
- **Flickr**: This is also a social network dataset, where nodes indicate users, edges indicate friendships relationships between users, and labels indicate an interest group.

The BlogCatalog and Flickr are static attributed network datasets. For the BlogCatalog and Flickr datasets, we refer to the settings in Meng et al. (2020) to reconstruct them into two synthetic temporal attributed networks for evaluating our temporal attributed model. More precisely, we generate a sequence of static attributed network snapshots $\{G_1, \ldots, G_T\}$ from the original network, where G_t ($t \in \{1 \ldots T\}$) is a sub-network that sampling from the original network with a fixed size of edges. For our experiments, we split it into 10 snapshots.

Baselines: We select six network embedding baseline models to compare our proposed model.

[1] http://snap.stanford.edu/data/soc-RedditHyperlinks.html.
[2] https://github.com/xhuang31/LANE.

Table 3
Link prediction results (AUC values).

Datasets	Flickr	BlogCatalog	RHNs
AdONE	0.87	0.82	0.88
NetWalk	0.73	0.80	0.78
T-GCN	0.80	0.65	0.76
TemporalGAT	0.88	0.84	0.86
CLDG	0.86	0.89	0.87
TAOA	**0.94**	**0.93**	**0.90**

- **AdONE** (Bandyopadhyay et al., 2020): AdONE adopts deep unsupervised autoencoders based on adversarial learning to minimizes the impact of outliers for static attributed network embedding. AdONE is not specially designed for temporal attributed networks. Since it considers outlier nodes during network embedding, we adopt it as the baseline model. For our datasets, we utilize the model to embed each snapshot into a low-dimensional vector. Gated recurrent unit (GRU) is then adopted to predict the new link for the last network snapshot.
- **NetWalk** (Yu et al., 2018): NetWalk encodes the node of temporal networks to vector by clique embedding. This method is mainly used for anomaly detection, and we use the representation vectors of nodes to predict new links.
- **T-GCN** (Zhao et al., 2020): T-GCN combines with GCN to learn spatial dependence and GRU to learn temporal dependence for temporal network embeddings.
- **TemporalGAT** (Fathy & Li, 2020): It utilizes GATs (Velickovic et al., 2018) and TCNs (Bai et al., 2018) networks to learn representations for temporal networks.
- **DyHNE** (Wang,Lu et al., 2022): It capture the semantics and structure of a network using proposed meta-path-based multiorder relationship.
- **CLDG** (Xu et al., 2023): It adopts contrastive learning process to learn node embeddings on temporal graphs in unsupervised scenarios.

Parameter settings: We set output dimensions is 64 for all datasets in our experiment. The parameters of the baseline methods are tuned optimally. Other settings include: the learning rate of the model was set as 0.0001; the power P was set 2; the damping factor γ was set as 0.8; the layers of the TAOA was set 5. For each dataset, we report average performance over five independent experiments.

5.2. Experimental results

We report our experimental results in this section. We report the performance of our model on the three temporal attributed datasets with five temporal network embedding baseline models for link prediction and two attributed datasets with two attributed network embedding baseline models for node classification.

5.2.1. Link prediction

Temporal network Link prediction desires to estimate the probability of future connection among nodes, which predict the topological structure G_{T+1} at timestamp $T+1$ according to the previously topological structure $\{G_1, \ldots, G_T\}$. we utilize the AUC (J. & X., 2005) to estimate the performance of different baselines. Table 3 compares AUC values over the datasets on the link prediction task.

From Table 3, we can scan that the TAOA model achieves better than all the other baselines in all the three temporal attributed networks. In particular, we use the simplifying higher graph convolutional mechanism (SHGC) to preprocess attribute features for each node of each snapshot in temporal attributed networks. The SHGC incorporate attribute information into link structure information, which can reveal attribute information in link structure. Our model further utilizes an outlier aware autoencoder to model the node information,

Table 4
Results on node classification.

Datasets	BlogCatalog		Flickr	
	Micro_F1	Macro_F1	Micro_F1	Macro_F1
Adone	0.68	0.69	0.56	0.61
DyHNE	0.70	0.68	0.54	0.58
TAOA	**0.74**	**0.74**	**0.57**	0.61

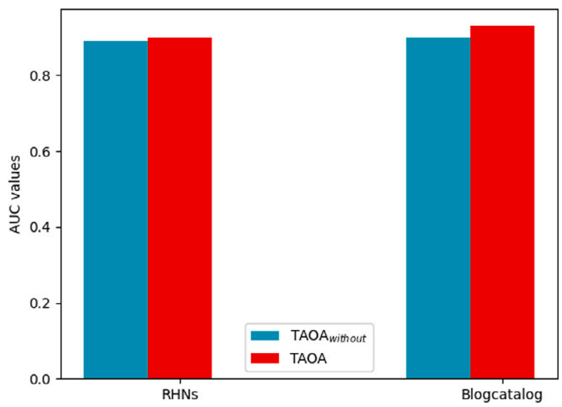


Fig. 3. Effect of outliers.

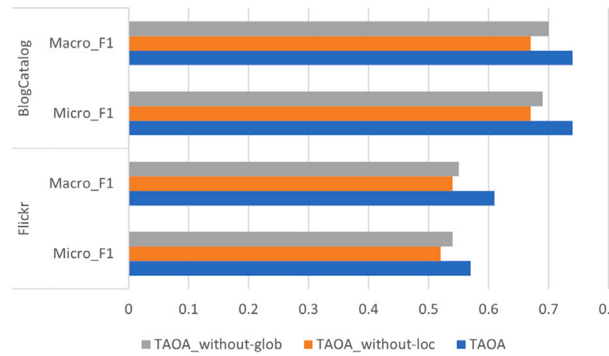


Fig. 4. Effect of loss.

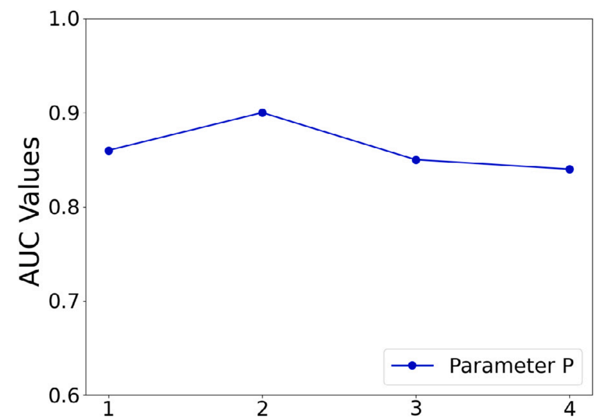


Fig. 5. TAOA's performance on the RHNs dataset when increasing P.

which combines current snapshots and previous snapshots to jointly learn embedded vectors for the current snapshot and can steadily and powerfully model time dependencies. The experimental results also demonstrate that the static attributed network model AdONE is superior to the temporal attributed network model T-GCN and NetWalk, which indicates that it is important and necessary to consider outliers in network embedding. The final result of link prediction demonstrate that our TAOA model can capture the evolving trend to predict links for the future.

5.2.2. Node classification

Node classification is a classical task used to estimate the performance of the learned embedding vectors. We use Micro-F1 and Macro-F1 as metrics that similar to Zhang et al. (2018) to measure the performance. Due to the fact that the BlogCatalog dataset and Flickr dataset contain labeled data, we select them to validate our model. As the DyHNE is the relatively latest node embedding model and AdONE considers outlier nodes during network embedding, we adopt both as our baseline model for node classification tasks. For our experiments, we adopted the embedded vector of the last snapshot for the node classification task.

Table 4 compares Micro_F1 and Macro_F1 values over the two datasets. It can be seen from Table 4, our proposed model TAOA performs the best performance on the node classification task in Flickr and BlogCatalog datasets. In the Flickr dataset, the Macro_F1 value of the TAOA is equal to the AdONE, but the Micro_F1 value of the TAOA is superior to the AdONE. Therefore, the overall performance on the Flickr dataset of our model is superior to the AdONE. The experimental results reveal that our proposed model can learn effective node representations for attributed networks.

5.3. Ablation study

We conducted an ablation study in this section. In particular, we analyze how outliers and each component of the loss functions affect the performance of the TAOA model.

5.3.1. Effect of outliers

We analyze outliers on the RHNs dataset and the BlogCatalog dataset for the link prediction task in this section. We do not consider outliers to prove the performance of the TAOA model defined as $TAOA_{without}$. Specifically, we removed outlier scores o_i^t of each node i for each snapshot t from the TAOA loss function. As shown in Fig. 3, the experimental results show the average AUC value of TAOA is 1% higher than $TAOA_{without}$ on the RHNs dataset, 3% higher than $TAOA_{without}$ on the BlogCatalog dataset. The experimental results repeatedly reveal that it is important and necessary to consider outliers in network embedding.

5.3.2. Effect of loss functions

We analyze how each component of the loss functions affects the performance of the TAOA model. For our experiments, we selected the BlogCatalog and Flickr datasets for the node classification task. When we do not consider the global structure proximity components, we rephrase our TAOA model as $TAOA_{without-glob}$. Specifically, we removed the global structure proximity component L_{glob}^t and L_{glob}^{t-1} from the TAOA loss function. We also consider the case when the local structure proximity components are not explored, and we name such TAOA model as $TAOA_{without-loc}$. Specifically, we removed the local structure proximity component L_{loc}^t and L_{loc}^{t-1} from the TAOA loss function. Since our model deals with temporal attributed networks, we cannot remove the last component L_{com}^t from the TAOA loss function. For our experiments, we also adopted the embedded vector of the last snapshot for the node classification task.

Fig. 4 compares Micro_F1 and Macro_F1 values over the two datasets. It can be seen from Fig. 4, the model TAOA performs the best on node classification task in Flickr and BlogCatalog datasets. We

Fig. 6. TAOA's performance on the RHNs dataset when increasing γ.

also have observed that the overall performance of $TAOA_{without-glob}$ is superior to the $TAOA_{without-loc}$. The reason may be that the local approximation relationship of nodes contains more valuable information about nodes. The experimental results also reveal the effectiveness of our proposed model for attributed network embedding.

5.4. Parameter sensitivity analysis

In this section, we show parameter sensitivity analysis. In particular, we evaluate the power of the SHGC P and the damping factor γ can affect the link prediction performance. Due to the fact that the RHNs dataset is a real temporal attributed network, we select the dataset to perform sensitivity analysis for both parameters. Since P and γ jointly determined the final node information, we analyzed both parameters together. When analyzing P, γ was set to 0.8, and when analyzing γ, P was set to 2.

For the parameter P, we demonstrate the effect of changing this parameter by increasing the power P of the SHGC from 1 to 4, increasing it by 1 at each step. In Fig. 5, we can see that the best result was obtained at $P = 2$. The performance continues to increase as P increases from 1 to 2. Performance decreases slightly as P restarts to increase. For the parameter γ, we demonstrate the effect of changing this parameter by increasing the damping factor γ from 0.5 to 0.9, increasing it by 0.1 at each step. In Fig. 6, we can see that the best result was obtained at $\gamma = 0.8$. The performance continues to increase as γ increases from 0.5 to 0.8. Performance decreases slightly as γ restarts to increase.

6. Conclusions

We have proposed an effective framework TAOA for temporal attributed network embedding, which utilized an outlier aware autoencoder to model the node information. It combined current snapshots and previous snapshots to jointly learn embedded vectors for the current snapshot and can steadily and powerfully model time dependencies. Moreover, we proposed the simplifying higher graph convolutional mechanism (SHGC) to preprocess attribute features for each node of each snapshot in temporal attributed networks. SHGC incorporated attribute information into link structure, which can reveal attribute information in link structure. Experimental results on temporal attributed network datasets reveal that our model is competitive against various baseline models. For the sake of interpretability and easy to calculate, we only consider two adjacent snapshots in our model, and we will consider more snapshots in future work.

Recently, some researchers have defined outlier nodes from different perspectives, such as multi-view outliers. We will extend our work these definitions and consider more previous snapshots to learn jointly the embedding vectors in future work. Moreover, networks usually consist of multiple types of edges and nodes. We will also study network embedding with multi-view outliers for heterogeneous dynamic networks.

CRediT authorship contribution statement

Xian Mo: Conceptualization, Data curation, Methodology, Validation, Writing – original draft, Writing – review & editing. **Jun Pang:** Conceptualization, Methodology, Writing – review & editing. **Zhiming Liu:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The work is financed by the National Natural Science Foundation of China (62306157, 62202320), 62032019, Capacity Development Grant of Southwest University (SWU116007).

References

Amara, A., Taieb, M. A. H., & Aouicha, M. B. (2021). Network representation learning systematic review: ancestors and current development state. CoRR, abs/2109.07583.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv: Learning*.

Bandyopadhyay, S., Lokesh, N., & Murty, M. N. (2019). Outlier aware network embedding for attributed networks. In *Proceedings of the 33th AAAI conference on artificial intelligence* (pp. 12–19).

Bandyopadhyay, S., N, L., Vivek, S. V., & Murty, M. N. (2020). Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of 13th international conference on web search and data mining* (pp. 25–33).

Cai, X., Huang, C., Xia, L., & Ren, X. (2023). LightGCL: Simple yet effective graph contrastive learning for recommendation. In *Proceedings of the 7th international conference on learning representations* (pp. 1–15).

Ding, K., Li, J., Bhanushali, R., & Liu, H. (2019). Deep anomaly detection on attributed networks. In *Proceedings of 2019 international conference on data mining* (pp. 594–602).

Du, X., Yu, J., Chu, Z., Jin, L., & Chen, J. (2022). Graph autoencoder-based unsupervised outlier detection. *Information Sciences*, *608*, 532–550.

Fathy, A., & Li, K. (2020). TemporalGAT: Attention-based dynamic graph representation learning. In *Proceedings of the 24th Pacific-Asia conference on knowledge discovery and data mining* (pp. 413–423).

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 855–864).

Hsieh, I., & Li, C. (2023). CoANE: Modeling context co-occurrence for attributed network embedding. *IEEE Transactions on Knowledge and Data Engineering*, *35*(1), 167–180.

Huang, L., Zhu, Y., Gao, Y., Liu, T., Chang, C., Liu, C., et al. (2021). Hybrid-order anomaly detection on attributed networks. *IEEE Transactions on Knowledge and Data Engineering*, 1–14.

Issaid, C. B., Elgabli, A., & Bennis, M. (2022). DR-DSGD: A distributionally robust decentralized learning algorithm over graphs. CoRR, abs/2208.13810.

J., H., & X., L. C. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*, 299–310.

Ji, Y., Huang, L., He, H., Wang, C., Xie, G., Shi, W., et al. (2019). Multi-view outlier detection in deep intact space. In *Proceedings of the 2019 IEEE international conference on data mining* (pp. 1132–1137).

Jiao, P., Guo, X., Jing, X., He, D., Wu, H., Pan, S., et al. (2022). Temporal network embedding for link prediction via VAE joint attention mechanism. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 7400–7413.

Keikha, M. M., Rahgozar, M., & Asadpour, M. (2018). Community aware random walk for network embedding. *Knowledge Based Systems*, *148*, 47–54.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th international conference on learning representations* (pp. 1–14).

Li, J., Cheng, K., Liang, W., & Liu, H. (2018). Streaming link prediction on dynamic attributed networks. In *Proceedings of the 11th ACM international conference on web search and data mining* (pp. 369–377).

Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., & Liu, H. (2017). Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 26th conference on information and knowledge management* (pp. 387–396).

Li, Z., & Lai, D. (2022). Dynamic network embedding via temporal path adjacency matrix factorization. In *Proceedings of the 31th ACM international conference on information & knowledge management* (pp. 1219–1228).

Li, Y., Wang, Y., Zhang, T., Zhang, J., & Chang, Y. (2019). Learning network embedding with community structural information. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 2937–2943).

Liang, J., Jacobs, P., Sun, J., & Parthasarathy, S. (2018). Semi-supervised embedding in attributed networks with outliers. In *Proceedings of the 2018 SIAM international conference on data mining* (pp. 153–161).

Liang, S., Zhang, X., Ren, Z., & Kanoulas, E. (2018). Dynamic embeddings for user profiling in Twitter. In *Proceedings of the 24th conference on knowledge discovery and data mining* (pp. 1764–1773).

Lin, Z., Tian, C., Hou, Y., & Zhao, W. X. (2022). Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the 7th ACM web conference 2022* (pp. 2320–2329).

Liu, Z., Huang, C., Yu, Y., & Dong, J. (2021). Motif-preserving dynamic attributed network embedding. In *Proceedings of the 21th web conference* (pp. 1629–1638).

Ma, J., Zhang, Q., Lou, J., Xiong, L., & Ho, J. C. (2021). Temporal network embedding via tensor factorization. In *Proceedings of the 30th ACM international conference on information and knowledge management* (pp. 3313–3317).

Meng, Z., Liang, S., Fang, J., & Xiao, T. (2019). Semi-supervisedly co-embedding attributed networks. In *Proceedings of the 32th annual conference on neural information processing systems 2019* (pp. 6504–6513).

Meng, Z., Liang, S., Zhang, X., Mccreadie, R., & Ounis, I. (2020). Jointly learning representations of nodes and attributes for attributed networks. *ACM Transactions on Information Systems*, *38*(2), 1–32.

Miller, M., Lynn, S.-L., & James, M. C. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, *27*, 415–444.

Sadeghi, A., Ma, M., Li, B., & Giannakis, G. B. (2021). Distributionally robust semi-supervised learning over graphs. CoRR, abs/2110.10582.

Sankar, A., Wu, Y., Gou, L., Zhang, W., & Yang, H. (2020). DySAT: Deep neural representation learning on dynamic graphs via self-attention networks. In *Proceedings of the 13th ACM international conference on web search and data mining* (pp. 519–527).

Sun, G., Shen, Y., Zhou, S., Chen, X., Liu, H., Wu, C., et al. (2023). Self-supervised interest transfer network via prototypical contrastive learning for recommendation. In *Proceedings of the 37th AAAI conference on artificial intelligence* (pp. 4614–4622).

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. In *Proceedings of the 6th international conference on learning representations* (pp. 1–12).

Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., & Yang, S. (2017). Community preserving network embedding. In *Proceedings of the 41th AAAI conference on artificial intelligence* (pp. 203–209).

Wang, X., Lu, Y., Shi, C., Wang, R., Cui, P., & Mou, S. (2022). Dynamic heterogeneous information network embedding with meta-path based proximity. *IEEE Transactions on Knowledge and Data Engineering*, *34*(3), 1117–1132.

Wang, X., Pun, Y., & So, A. M. (2022). Distributionally robust graph learning from smooth signals under moment uncertainty. *IEEE Transactions on Signal Processing*, *70*, 6216–6231.

Wang, Y., Wang, X., Huang, X., Yu, Y., Li, H., Zhang, M., et al. (2023). Intent-aware recommendation via disentangled graph contrastive learning. In *Proceedings of the 32th international joint conference on artificial intelligence* (pp. 2343–2351).

Wei, H., Hu, G., Bai, W., Xia, S., & Pan, Z. (2019). Lifelong representation learning in dynamic attributed networks. *Neurocomputing*, *358*, 1–9.

Wu, F., Jr., A. H. S., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. Q. (2019). Simplifying graph convolutional networks. In *Proceedings of the 36th international conference on machine learning* (pp. 6861–6871).

Xie, S., Li, Y., Tam, D. S. H., Liu, X., Ying, Q., Lau, W. C., et al. (2023). GTEA: Inductive representation learning on temporal interaction graphs via temporal edge aggregation. In *Proceedings of the 27th Pacific-Asia conference on knowledge discovery and data mining* (pp. 28–39).

Xu, Y., Shi, B., Ma, T., Dong, B., Zhou, H., & Zheng, Q. (2023). CLDG: Contrastive learning on dynamic graphs. In *Proceedings of the 39th IEEE international conference on data engineering* (pp. 696–707).

Yang, Y., Huang, C., Xia, L., Huang, C., Luo, D., & Lin, K. (2023). Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023* (pp. 1063–1073).

Yao, H., Zhang, B., Zhang, P., Wu, S., Jiang, C., & Guo, S. (2021). RDAM: A reinforcement learning based dynamic attribute matrix representation for virtual network embedding. *IEEE Transactions on Emerging Topics in Computing*, *9*(2), 901–914.

Yu, W., Cheng, W., Aggarwal, C. C., Zhang, K., Chen, H., & Wang, W. (2018). NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM international conference on knowledge discovery and data mining* (pp. 2672–2681).

Zhang, Z., Yang, H., Bu, J., Zhou, S., Yu, P., Zhang, J., et al. (2018). ANRL: Attributed network representation learning via deep neural networks. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 3155–3161).

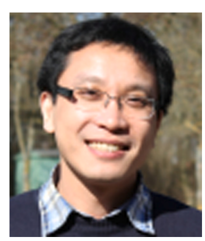
Zhang, C., Yao, Z., Yao, H., Huang, F., & Chen, C. L. P. (2023). Dynamic representation learning via recurrent graph neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(2), 1284–1297.

Zhao, S., Chen, J., Chen, J., Zhang, Y., & Tang, J. (2022). Hierarchical label with imbalance and attributed network structure fusion for network embedding. *AI Open*, *3*, 91–100.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., et al. (2020). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(9), 3848–3858.

Zou, H., Duan, Z., Guo, X., Zhao, S., Chen, J., Zhang, Y., et al. (2021). On embedding sequence correlations in attributed network for semi-supervised node classification. *Information Sciences*, *562*, 385–397.

Xian Mo is currently a Lecturer at the School of Information Engineering, Ningxia University, China. He received his PhD degree from Southwest University in 2021. His principal research interest is in representation learning for dynamic networks. He was a recipient of the National Scholarship in 2019 and the China National Study Abroad Fund in 2020.



Jun Pang is currently a Senior Researcher at University of Luxembourg. He received his PhD degree from Vrije Universiteit Amsterdam in 2004. His principal research interests include spatio-temporal data mining, social network analysis.

Zhiming Liu is currently a Professor at Southwest University. He received his PhD degree from University of Warwick in 1991. His principal research interest is in the area of theory and methods of software development and theory for modelling and analysis of system architecture of Human-Cyber-Physical Systems (HCPS). His well-known work including Model Transformations for Fault-Tolerance, Probabilistic Duration Calculus for Reliability Analysis, rCOS for Formal Model-Driven Development, and Model of Open Hierarchical HCPS Architectures. He has widely published in modeling, analysis and verification of complex software systems.