

Gated graph convolutional network based on spatio-temporal semi-variogram for link prediction in dynamic complex network



Liping Yang^a, Xin Jiang^a, Yiming Ji^a, Hua Wang^b, Ajith Abraham^c, Hongbo Liu^{a,*}

^a College of Artificial Intelligence, Dalian Maritime University, 116026, China

^b Department of Mathematical Sciences, Georgia Southern University Statesboro, GA 30460, USA

^c Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, Auburn, WA 98071, USA

ARTICLE INFO

Article history:

Received 29 September 2021

Revised 2 April 2022

Accepted 12 July 2022

Available online 16 July 2022

Communicated by Zidong Wang

Keywords:

Dynamic network link prediction

GCN

LSTM

Spatio-temporal semi-variogram

ABSTRACT

Link prediction is one of the most important methods to uncover evolving mechanisms of dynamic complex networks. Determining these links raises well-known technical challenges in terms of weak correlation, uncertainty and non-stationary. In this paper, we presented a novel gated graph convolutional network (GCN) based on spatio-temporal semi-variogram (STEM-GCN). It learns spacial and temporal features in order to achieve link prediction in the dynamic networks. In this STEM-GCN model, we first utilized the spatio-temporal semi-variogram to obtain the spacial and temporal correlations with the dynamic networks. Its spacial correlation helped us determine the hyper-parameters of STEM-GCN and speed up its training. The correlation smoothing strategy is also introduced to eliminate the noise through temporal correlation and to improve the accuracy of link prediction. Finally, the network dynamics are captured by propagating the spacial and temporal features between consecutive time steps with stacked memory cell structures. The extensive experiments on real data sets demonstrated the effectiveness of the proposed approach for link prediction in dynamic complex networks.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Link prediction in dynamic networks (graphs) aims to infer the changing of its topological structures and temporal features, which plays a key role in the understanding of network evolution and the relationships between the topology and functions [1–3]. Most real-world networks are naturally dynamic, evolving over time with the addition, deletion, and updating of nodes and links [4]. Their properties, such as weak correlations in time and space, uncertainty, and non-stationary, make it very difficult to characterize the structural relationships (links) among the nodes in the networks. Various graph neural network models were designed to investigate network dynamics [5,6]. The dynamic network is usually represented as a series of continuous network snapshots. Then the model learns the fixed-size embedding of each node through continuous network snapshots and the link prediction is mapped to the nearest neighbor search in the embedding space [7–9]. However, real-world networks often have complex evolving dynamics,

making it extremely challenging to predict the potential structural relationships over time [10,11].

The challenges can be summarized from three aspects. 1) The temporal correlation may exist over periods of various lengths. An example is shown in Fig. 1: there is a link between nodes *A* and *B* at the first two time steps, and it disappears at time $t + 2$. The duration of links is different and uncertain, making weak correlation in time among network snapshots [12,13]. 2) There may exist diverse link patterns among the nodes. In Fig. 1, node *A* ceases contact with its neighbors, while other nodes still keep in touch with neighbors. They influence each other, which usually leads to complex evolution over both time and space [14,15]. 3) The embedding of continuous snapshots is non-stationary. In real-world dynamic networks, the distribution of links in each network snapshot varies. Such successive network snapshots are non-stationary time series in nature [16].

To address the above challenges, many methods combine graph neural networks (GNNs) with long short-term memory (LSTM) networks to capture evolutionary patterns in the dynamic networks. Seo et al. [17] presented a graph convolutional recurrent network (GCRN) combining an LSTM network with ChebNet [18]. Since they assume the links are fixed over time, Chen et al. [19] proposed a novel end-to-end framework GC-LSTM, which is capable of han-

* Corresponding author.

E-mail addresses: liping.yang@dmlu.edu.cn (L. Yang), doctor_jiangxin@sina.com (X. Jiang), jymxmg@foxmail.com (Y. Ji), hwang@georgiasouthern.edu (H. Wang), ajith.abraham@ieee.org (A. Abraham), lhb@dmlu.edu.cn (H. Liu).

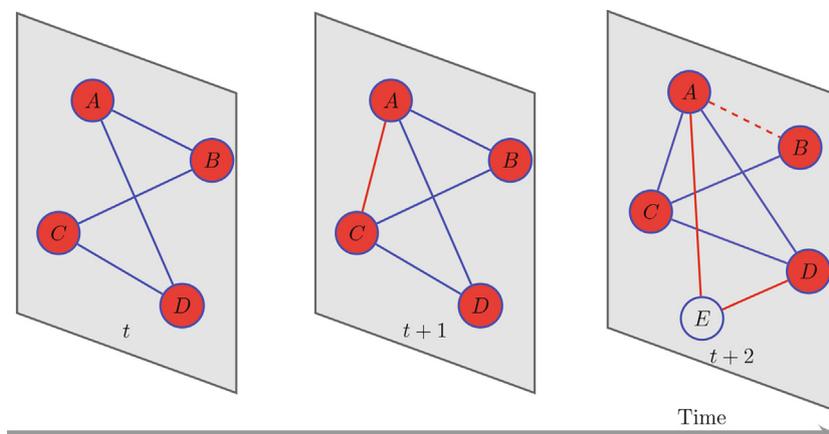


Fig. 1. Evolution of the Dynamic Networks.

dling links appearing or disappearing. Similarly, Chen et al. [20] presented a novel encoder-LSTM-decoder (E-LSTM-D) deep learning model for link prediction in dynamic networks. The encoder is utilized to acquire node features, helping LSTM to learn the evolutionary patterns of snapshot sequences. Recently, the temporal graph attention (TGAT) layer [21] is introduced to efficiently aggregate temporal-topological neighborhood features to learn the time-feature interactions. However, the distribution of links in each network snapshot is very sparse since only a subset of nodes in the real-world dynamic network will generate links in a finite time. Such issue leads to weak correlations between network snapshots, which in turn brings great difficulty to effectively implement link prediction in dynamic networks.

In this paper, we proposed a novel gated graph convolutional network based on spatio-temporal semi-variogram (STEM-GCN) for link prediction in dynamic complex networks. Firstly, we leverage the spatio-temporal semi-variogram to analyze the correlation of dynamic networks. For successive snapshots with large changes in temporal semi-variance, a weak correlation smoothing strategy is introduced to guarantee continuous smoothness. Then, the stacked memory structure cell (MCS) is designed to obtain the hidden state of the current network snapshot by iteratively learning from successive network snapshots. Finally, the probability matrix of links in the current network is generated by using the fully connected neural network to decode the state hidden state. Our main contributions are summarized as follows:

- We analyzed the correlation of successive network snapshots using the spatio-temporal semi-variogram. The model's hyperparameters are determined through spacial correlations. The theoretical analysis is consistent with the experimental results, which significantly speeds up the training of our model.
- Based on the spatio-temporal semi-variogram, a weak correlation smoothing strategy is introduced for continuous snapshots with large changes in temporal semi-variance. This strategy not only preserves the continuity of the network snapshots, but also improves the accuracy of the prediction.
- To reduce the number of parameters and achieve more efficient temporal learning, we introduce the memory cell structure which embeds GCN into the input gate of LSTM. So that GCN can directly obtain the structural features of dynamic networks and share parameters with LSTM.

The remainder of this paper is organized as follows. Section 2 outlines the state-of-the-art methods for dynamic network link prediction. In Section 3, we introduce the definition of dynamic networks and related theories. In Section 4, we present a novel

gated graph convolutional network based on spatio-temporal semi-variogram for link prediction in dynamic networks. Experiments of evaluating the area under the curve (AUC) and error rate (ER) of the STEM-GCN are carried out in Section 5. Finally, we summarize the results and propose future work in Section 6.

2. Related Work

Link prediction has been widely used in social networks, financial networks, and brain networks [22–24], etc. More recently, the dynamical aspects of networks have been incorporated into this task. In this section, we discuss relevant works in dynamic network link prediction from links stream-based approaches and snapshot-based approaches.

The existing links stream-based approaches are point processes which are continuous in time [25–28]. Nguyen et al. [25] suggested using node embedding and the random walk models for learning a time-dependent network representation. Trivedi et al. [26] presented an architecture, Know-Evolve, capturing evolutionary dynamics of both subject and object entities in a temporal knowledge graph using recurrent neural networks and temporal point processes, which captures temporal node interactions in addition to the topological evolution. DyRep [27] extended Know-Evolve with a two-time scale deep temporal point process approach that captures temporal node interactions in addition to the topological evolution. Recently, DynamicTriad [28] imposed the triad to learn the node embeddings while preserving the temporal information. These methods take a pair of nodes as input and utilize recurrent neural networks and temporal point processes to capture evolutionary dynamics. They are advantageous for event time prediction because of the continuous nature of the process. However, such techniques cannot be applied to network snapshots that lack fine-grained timestamps.

The snapshot-based methods are proposed from another perspective. These methods aggregate the links within fixed time intervals into network snapshots, and use GNNs and RNNs or transformer networks to encode structural patterns and temporal patterns respectively. Seo et al. [17] presented graph convolutional recurrent network (GCRN) combining an LSTM network with ChebNet [18]. They use a GCN to acquire node features, helping LSTM to learn the evolutionary patterns. Later, Manessi et al. [29] used a recursive network of LSTM to deal with the time-varying changes of the GNN. They modify the graph convolution layers by adding a skip connection. Recently, EvolveGCN [30] is introduced to process dynamic networks, in which gated recursive unit GRU and LSTM are combined with GCN to capture dynamics, and RNN is

used to estimate the GCN parameters for the future snapshots. One advantage of capturing dynamics in parameters is that it can handle more flexible dynamic data as the nodes do not need to exist all the time. Meanwhile, Chen et al. [31] proposed, for the first time, a novel end-to-end multiple residual recurrent graph neural networks (Res-RGNNs) framework, referred to as “MRRes-RGNN”, for traffic prediction. Wang et al. [32] proposed a novel decomposition of in-cell and inter-cell data to effectively model the Spatio-temporal dependency. Later, a novel temporal convolutional framework TrellisNets (STP-TrellisNets) [33] is introduced for spatial-temporal prediction. The temporal module of STP-TrellisNets employs two TrellisNets in serial to capture both the short- and long-term temporal correlation. Liu et al. [34] presented a novel deep model called “AttConvLSTM”. ConvLSTM keeps spatial information as intact as possible during sequential analysis, and the attention mechanism focuses on variations that cannot be identified by the recurrent module.

In general, most of real-world networks are sparse networks, and the link distribution becomes uneven over time. Some edges may continue to appear at multiple consecutive network snapshots, and the network structure would not change, resulting in weak temporal correlations among snapshots. These approaches do not analyze the correlation between consecutive snapshots. In this paper, we propose a model based on spatio-temporal semi-variogram analysis, which takes relatively long consecutive snapshots as input and is able to automatically learn the global structure of networks.

3. Preliminaries

In this section, we introduce the dynamic network (dynamic graph) link prediction model and spatio-temporal semi-variogram of dynamic networks.

3.1. Problem Formulation

We formally define the problem of dynamic network link prediction. A dynamic network can be represented as a sequence of discrete network snapshots, $\mathbb{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_T\}$, where T is the number of time steps. Each snapshot $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, A_t)$ represents a directed and unweighted graph at time t . Let \mathcal{V}_t be the set of all vertices, and \mathcal{E}_t be the temporal edges within the fixed time intervals. A_t denotes the adjacency matrix of \mathcal{G}_t , where the element $A_t(i, j) = 1$ if there is a directed edge from vertex i to j , and $A_t(i, j) = 0$ otherwise.

For a dynamic network, its network snapshot sequence is $\{A_{t-s}, \dots, A_{t-1}\}$, we consider the link prediction of a dynamic network as a structural sequence modeling problem. The goal is to predict the probability of all links at time t through the evolution information of the precious s time stamps in network snapshot sequence. It can be defined as

$$\tilde{A}_{t,pre} = \underset{A_t}{\operatorname{argmax}} P(\tilde{A}_t | A_{t-s}, \dots, A_{t-1}), \quad (1)$$

where $\{A_{t-s}, \dots, A_{t-1}\}$ represent the adjacency matrices of previous s snapshots, \tilde{A}_t represents the predictive variable of the model $P(\tilde{A}_t | A_{t-s}, \dots, A_{t-1})$, and $\tilde{A}_{t,pre}$ denotes the predictive adjacency matrix of the snapshot at time t .

Consequently, the link prediction problem essentially amounts to the estimation of a function or predictor based on the collected history links. In other words, the function at each time step uses history snapshots to capture network dynamics and can thus predict links with higher precision.

3.2. Spatio-temporal Semi-variogram of Dynamic Networks

A dynamic network is composed of nodes and edges with time series. As it evolves over time, node attributes will be affected by neighbors around them. Therefore, the comprehensive attributes of each node could be further expressed in terms of both the spatial and temporal aspects as follows:

$$\varphi(v_i, t) = \frac{1}{|\Gamma(v_i)|} \sum_{v_j \in \Gamma(v_i)} x_j^t, \quad (2)$$

where x_j^t denotes the attributes of node v_j at time t , and $\Gamma(v_i)$ represents the set of neighboring nodes of node v_i . The correlation between nodes in a dynamic network can be analyzed by semi-variogram [35]. The semi-variogram $\psi(d)$, which describes the spatial dependence of two random processes generated by two nodes v_i and v_j separated at d distance, is defined as

$$\psi(d) = \frac{1}{2N} \sum_{i=1}^N \sum_{j=1}^N (\varphi(v_i, t) - \varphi(v_j, t))^2, \operatorname{dist}(v_i, v_j) = d, \quad (3)$$

where N is the number of the nodes. When considering temporal correlation, the semi-variogram function $\psi(d, \tau)$ should take into account the time interval τ :

$$\psi(d, \tau) = \frac{1}{N} \sum_{i=1}^N (\varphi(v_i, t) - \varphi(v_j, t + \tau))^2, \operatorname{dist}(v_i, v_j) = d. \quad (4)$$

However, corresponding to the degree of nodes, the distribution of nodes in the dynamic networks is irregular. Therefore, we analyze the spatial dependence of node attributes, based on the spatio-temporal semi-variogram of dynamic networks, as follows:

$$\psi(d(\omega), \tau) = \frac{1}{|\phi(d(\omega), \tau)|} \times \sum_{\zeta \in \phi(d(\omega), \tau)} (\varphi(v_i, t) - \varphi(v_j, t^*))^2, \quad (5)$$

where $\phi(d(\omega), \tau) = \{(v_i, v_j, t, t^*) | \operatorname{dist}(v_i, v_j) \in d(\omega), |t - t^*| = \tau\}$ and ζ is (v_i, v_j, t, t^*) . In other words, $\phi(d(\omega), \tau)$ is a set consisting of any pair of nodes that are spatially separated over a distance in the range $d(\omega)$ and are temporally separated over τ . The distance $d(\omega)$ is $1, 2, \dots, k$, where k is the diameter of a network (the maximum distance between any two nodes), which is obtained by Floyd algorithm [36].

We will utilize the spatio-temporal semi-variogram to analyze the spatial and temporal dependence of nodes in the dynamic networks. The number of layers of graph convolution is determined by spatio-temporal semi-variogram to improve the prediction accuracy of the model for the dynamic link prediction tasks.

4. STEM-GCN Model for Dynamic Networks

In this section, we propose a novel STEM-GCN model for link prediction in dynamic networks. First, we introduce a weak correlation smoothing strategy to generate consecutive network snapshot sequences. Then, consecutive network snapshot sequences are fed into the STEM-GCN to learn both structural and temporal features of dynamic networks and predict future network snapshots based on the previously observed ones. Finally, we will define our loss function, demonstrate its optimization, and analyze the time complexity of our algorithm.

4.1. Weak Correlation Smoothing Strategy

Applying consecutive snapshots divided by fixed time intervals, causes poor prediction accuracy since the division makes the temporal correlation among snapshots weak. Therefore, it is important to use spatio-temporal semi-variogram to analyze the temporal

correlation between nodes for consecutive snapshots. A weak correlation smoothing for consecutive snapshots with large correlation fluctuations is implemented to eliminate the noise through temporal correlation. In what follows, we will introduce the weak correlation smoothing strategy in detail.

Firstly, from the dynamic network, we aggregate a series of links within a fixed time interval into network snapshots $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T\}$. The snapshots are represented as adjacent matrices $\{A_1, A_2, \dots, A_T\}$. Secondly, we randomly initialize the attribute x_i^t of each node in the network snapshot. Considering the temporal and spacial dependence of the node, the comprehensive attributes of each node are further expressed by Eq. (2). The spatio-temporal semi-variogram values among snapshots are calculated by Eq. (5). Intuitively, the semi-variogram curve with smooth and small fluctuations indicates that the attributes of the node pairs are well correlated in terms of temporal and spacial dependence. Thirdly, for dynamic networks with large curve fluctuations, we adopt a weak correlation smoothing (WCS) strategy and repartition the network snapshots to improve the accuracy of the model.

More specifically, we smooth the semi-variogram curve and find the extreme points in the curve, record the index T_i of the extreme points through *Index* function. Given the consecutive network snapshot sequences \mathcal{S} , for s network snapshots behind the extreme point, $A_{T_i^p}$ ($p = 1, \dots, s$) are obtained through sampling.

The network snapshots $\{A_{T_{i-1}}, \dots, A_{T_i}, A_{T_i^1}, \dots, A_{T_i^s}\}$ are divided into sequences $\mathcal{S}_{\mathcal{N}_1}$ through the *Slide* function, where \mathcal{N}_1 represents the number of sequences. This function implements a sliding window with a width of s and a sliding step length of s to intercept consecutive network snapshots. Similarly, for s network snapshots before the extreme point, $A_{T_i^p}$ are obtained through sampling from $\{A_{T_i}, \dots, A_{T_{i+1}}\}$, where $p = 1, \dots, s$. The network snapshots $\{A_{T_i^1}, \dots, A_{T_i^s}, A_{T_i}, \dots, A_{T_{i+1}}\}$ are divided into sequences $\mathcal{S}_{\mathcal{N}_2}$ through the *Slide* function. Finally, we concatenate $\mathcal{S}_{\mathcal{N}_1}$ and $\mathcal{S}_{\mathcal{N}_2}$ into $\mathcal{S}_{\mathcal{N}}$. The weak correlation smoothing strategy is provided by Algorithm 1.

Algorithm 1: Weak Correlation Smoothing (WCS)

Require: Dynamic network: $\mathbb{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T\}$, Diameter of a \mathbb{G} : k , Sequence length: s .
Ensure: Consecutive network snapshot sequences $\mathcal{S}_{\mathcal{N}}$.
1: Generate adjacency matrices A_t from \mathbb{G} ;
2: Initialize randomly x_i^t and calculate the comprehensive attributes $\varphi(v_i, t)$ of each node v_i by Eq. (2);
3: Calculate $\psi(d(\omega), \tau)$ by Eq. (5).
4: $\psi(\tau, d(\omega)) \leftarrow \psi_{Smooth}(\tau, d(\omega))$;
5: **for** $\tau = 1$ to T **do**
6: $\xi(\tau, d(\omega)) \leftarrow \frac{\partial \psi(d(\omega), \tau)}{\partial \tau}$;
7: **end for**
8: $\xi(\tau, d(\omega)) \leftarrow \xi_{Smooth}(\tau, d(\omega))$;
9: $T_{list} \leftarrow Index(\xi(\tau, d(\omega)) = 0)$;
10: **for** $T_i \in T_{list}$ **do**
11: **for** $p = 1, \dots, s$ **do**
12: Get $A_{T_i^p}$ by sampling from $\{A_{T_{i-1}}, \dots, A_{T_i}\}$;
13: $\mathcal{S}_{\mathcal{N}_1} \leftarrow Slide\{A_{T_{i-1}}, \dots, A_{T_i}, A_{T_i^1}, \dots, A_{T_i^s}\}$;
14: **end for**
15: $Temp \leftarrow \{A_{T_{i-s}}, A_{T_{i-s+1}}, \dots, A_{T_{i-1}}\}$;
16: **for** $p = 1, \dots, s$ **do**
17: Get $A_{T_i^p}$ by sampling from $\{A_{T_i}, \dots, A_{T_{i+1}}\}$;
18: $\mathcal{S}_{\mathcal{N}_2} \leftarrow Slide\{A_{T_i^1}, \dots, A_{T_i^s}, A_{T_i}, \dots, A_{T_{i+1}}\}$;

a (continued)

Algorithm 1: Weak Correlation Smoothing (WCS)

19: **end for**
20: $\mathcal{S}_{\mathcal{N}} \leftarrow \mathcal{S}_{\mathcal{N}_1} \oplus \mathcal{S}_{\mathcal{N}_2}$;
21: **end for**
22: Return $\mathcal{S}_{\mathcal{N}}$.

4.2. STEM-GCN Model

Our objective is to learn a stable dynamic link prediction model based on spatio-temporal semi-variogram, named STEM-GCN, which predicts future links by simultaneously capturing changes in topology and temporal characteristics of historical snapshots. The overall framework, which is shown in Fig. 2, includes two core parts. One is the stacked Memory Cell Structure (MCS) and the other is the fully connected network. It utilizes the encoder-decoder architecture which takes the smooth network snapshot sequences $\mathcal{S} = \{A_{t-s}, \dots, A_{t-1}\}$ as input and outputs predicted network snapshot $\tilde{A}_{t,pre}$.

4.2.1. Stacked Memory Cell Structure

To reduce the number of model parameters and achieve a more efficient temporal learning, the STEM-GCN model mainly depends on the memory cell structure (MCS), as shown in Fig. 2. Each memory cell structure, with the embedded GCN into the gate of LSTM, can save the structural features of the dynamic networks. The stacked memory cell structures can preserve both structural and temporal features of the long-term networks at the same time, and further capture the evolutionary nature of dynamic networks. Compared with GC-LSTM, the number of trainable parameters could be largely reduced, since the parameters are shared across the GCN and LSTM.

In the MCS, the first step is to introduce a multiplicative forget gate unit to throw away the irrelevant memory contents from the previous cell state. The forget gate is defined as

$$\mathcal{Z}_t^f = \sigma(W_f A_t + U_f \mathcal{H}_{t-1} + b_f), \quad (6)$$

where $A_t \in \mathbb{R}^{N \times N}$ is the input of the MCS at time t , $\mathcal{H}_{t-1} \in \mathbb{R}^{N \times F}$ denotes the output at time $t-1$ and $\sigma(\cdot)$ represents the activation function *sigmoid* defined as $\sigma(x) = 1/(1 + \exp(-x))$. The parameters $W_f \in \mathbb{R}^{N \times F}$, $U_f \in \mathbb{R}^{F \times F}$ and $b_f \in \mathbb{R}^F$ are the weights and bias of the forget gate \mathcal{Z}_t^f , respectively.

Then a multiplicative input gate unit \mathcal{Z}_t^i is introduced to protect the memory contents stored in the cell state from perturbation by irrelevant inputs. GCN is embedded into the input gate to extract the topological structure of each network snapshot at the current time and share the parameters with LSTM to reduce the number of model parameters. The *sigmoid* layer decides how many contents the input contains should be updated by \mathcal{Z}_t^i . The *tanh* layer generates a candidate vector \mathcal{M}_t based on the topological structures of the network snapshot extracted by GCN, and adds it to the cell state. The current memory \mathcal{C}_t can be updated by combination of the cell structure \mathcal{C}_{t-1} at time $t-1$ and network topological structures \mathcal{M}_t at time t . The operation is defined as follows.

$$\begin{aligned} \mathcal{Z}_t^i &= \sigma(W_i A_t + U_i \mathcal{H}_{t-1} + b_i), \\ \mathcal{M}_t &= \tanh(W_m \tilde{D}^{1/2} \tilde{A}_t \tilde{D}^{1/2} A_t + U_m \mathcal{H}_{t-1} + b_m), \\ \mathcal{C}_t &= \mathcal{Z}_t^f * \mathcal{C}_{t-1} + \mathcal{Z}_t^i * \mathcal{M}_t, \end{aligned} \quad (7)$$

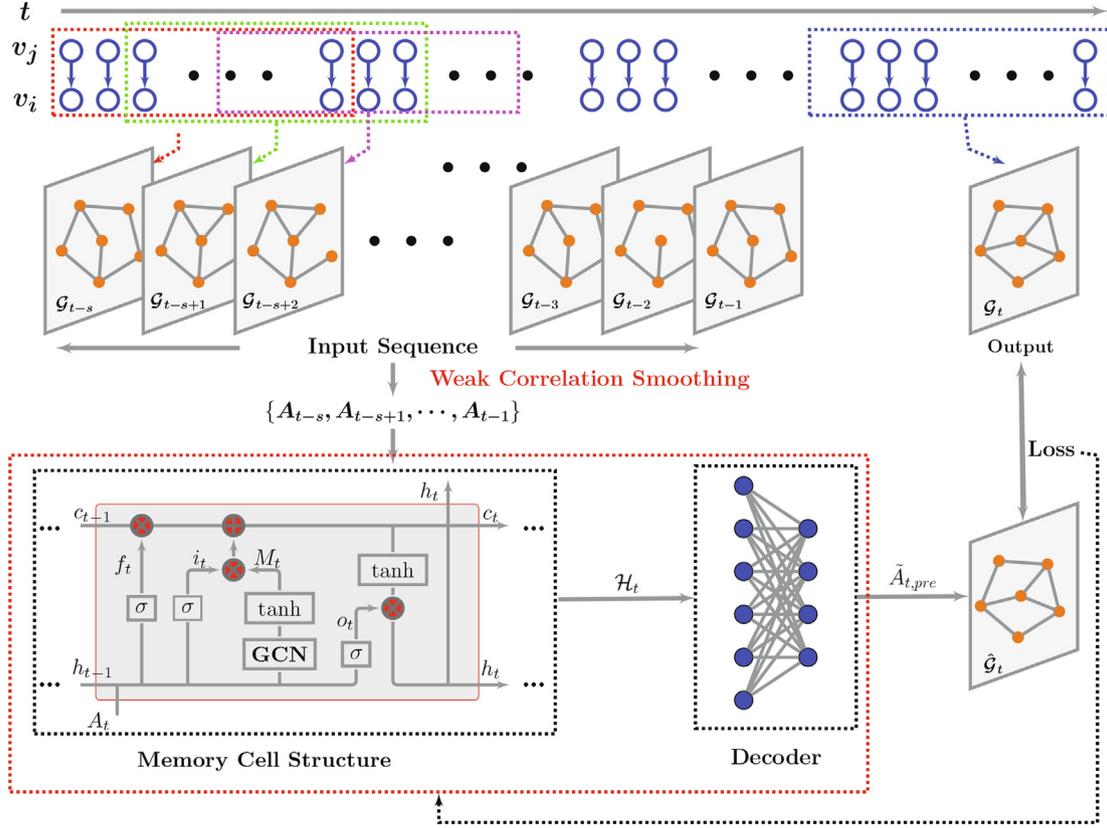


Fig. 2. Overall framework of STEM-GCN model. Given a series of links, we aggregate them within fixed time intervals into network snapshot sequences from the dynamic network. And then the new smooth network snapshot sequences are obtained by analyzing the spatio-temporal semi-variogram of the network snapshots and adopting the weak correlation smoothing (WCS) strategy. Further, given the smooth network snapshot sequences with length s , $\{\mathcal{G}_{t-s}, \mathcal{G}_{t-s+1}, \dots, \mathcal{G}_{t-1}\}$, each network snapshot sequence is transformed into the adjacency matrix $\{A_{t-s}, A_{t-s+1}, \dots, A_{t-1}\}$ as input to the STEM-GCN model. The stacked memory cell structures (MCS) can save both structural features and temporal features, and further predict future links in dynamic networks. Finally, the features \mathcal{H}_t received by the stacked memory cell structures are decoded into adjacency matrix $\tilde{A}_{t,pre}$ through the fully connected network, and the graph $\hat{\mathcal{G}}_t$ is obtained.

where $W_i \in \mathbb{R}^{N \times F}$, $U_i \in \mathbb{R}^{F \times F}$, $W_m \in \mathbb{R}^{N \times F}$ and $U_m \in \mathbb{R}^{F \times F}$ are the weights of the input gate in memory cell structure. Here, $b_i \in \mathbb{R}^F$ and $b_m \in \mathbb{R}^F$ are the corresponding biases; $\tilde{A}_t = A_t + I_N$ is the adjacency matrix of the network snapshot \mathcal{G}_t with added self-connections; I_N is the identity matrix, and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is degree matrix of \tilde{A}_t ; $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ is the normalized transition probability matrix. The memory cell structure can not only store the structural and temporal features of consecutive network snapshots, but also filter out the useless information. The output of the memory cell structure based on \mathcal{G}_t is controlled by the multiplicative output gate \mathcal{O}_t^o . The process is described as

$$\begin{aligned} \mathcal{L}_t^o &= \sigma(W_o A_t + U_o \mathcal{H}_{t-1} + b_o), \\ \mathcal{H}_t &= \mathcal{L}_t^o * \tanh(\mathcal{C}_t), \end{aligned} \quad (8)$$

where $W_o \in \mathbb{R}^{N \times F}$, $U_o \in \mathbb{R}^{F \times F}$ and $b_o \in \mathbb{R}^F$ denote the weight and bias of the output gate in gated graph convolutional network, respectively.

4.2.2. Decoder Architecture

To obtain the final prediction network, we utilize two-layer fully connected network as the decoder. The hidden vector $\mathcal{H}_t \in \mathbb{R}^{N \times F}$ is converted into a network probability matrix. The operation is defined as

$$\tilde{A}_{t,pre} = D_\theta(\mathcal{H}(t)). \quad (9)$$

where D_θ is a $(F, 512, 512, N)$ fully connected neural network with weights θ . Notice that the size of input and output N is the number of nodes in the different dynamic networks. Here, $\tilde{A}_{t,pre} \in [0, 1]$ is the probability matrix. The larger the value, the greater the probability that there is a connected edge between node i and node j . Naturally, $\tilde{A}_{t,pre} = 0$ indicates that there is no connected edge between nodes.

In this paper, given a network snapshot sequence $\mathcal{S} = \{\mathcal{G}_{t-s}, \dots, \mathcal{G}_{t-1}\}$, the adjacency matrix A_{t-s} at time $t-s$ is first input into the memory cell structure, and \mathcal{H}_{t-s} and \mathcal{C}_{t-s} are generated by the memory cell structure processing. Because the cell \mathcal{C}_{t-s} and the hidden state \mathcal{H}_{t-s} reflect different information, we need to consider not only the influence of the neighbor on the hidden state, but also the influence of the neighbor on the state of the cell. Then, the information of \mathcal{H}_{t-s} and \mathcal{C}_{t-s} are taken as the initial state of the memory cell structure. Combined with the structural feature of \mathcal{G}_{t-s+1} extracted by using GCN, it is processed by the memory cell structure and transmitted to the cell at the next moment. Through repeated iterations, the hidden state \mathcal{H}_t of network snapshot sequence \mathcal{S} is obtained. Finally, the network information $\tilde{A}_{t,pre}$ predicted at time t is obtained through the decoder.

4.3. Optimization

In order to verify the accuracy, we compare the predicted network and the real network at time t . The reconstruction error of the output $\tilde{A}_{t,pre}$ and the target matrix A_t is shown as

$$\begin{aligned}\mathcal{L} &= \|\tilde{A}_{t,pre} - A_t\|_F^2 \\ &= \sum_{i=0}^N \sum_{j=0}^N \|\tilde{A}_{t,pre}(i,j) - A_t(i,j)\|_2^2.\end{aligned}\quad (10)$$

It is worth noting, that due to the sparsity of the network, there are far more zero elements in the adjacency matrix than non-zero elements. If the adjacency matrix is used as an input, numerous zeros will be unnecessarily processed. Instead, we improve the loss function to

$$\begin{aligned}\mathcal{L}_{2nd} &= \|\left(\tilde{A}_{t,pre} - A_t\right) \odot \mathcal{B}\|_F^2 \\ &= \sum_{i=0}^N \sum_{j=0}^N \|\left(\tilde{A}_{t,pre}(i,j) - A_t(i,j)\right) \odot \mathcal{B}_{ij}\|_2^2,\end{aligned}\quad (11)$$

where \odot is the Hadamard product (i.e. the corresponding elements are multiplied), $\mathcal{B}_{ij} = 1$ if $A_t(i,j) = 0$, and $\mathcal{B}_{ij} > 1$ for non-zero elements. This way we increase the penalty for nodes with edge connections. In other words, higher weight is given to nodes connected by edges. In order to prevent over-fitting, the \mathcal{L}_2 -norm unit \mathcal{L}_{reg} is added:

$$\mathcal{L}_{reg} = \frac{1}{2} \sum_{k=1}^K \|W_d^{(k)}\|_F^2 + \frac{1}{2} \sum_{l=1}^L \left(\|W_e^{(l)}\|_F^2 + \|W_i^{(l)}\|_F^2 + \|W_M^{(l)}\|_F^2 + \|W_o^{(l)}\|_F^2 \right).\quad (12)$$

Combining Eqs. (11) and (12), we then use the mixed loss function

$$\mathcal{L}_{mix} = \mathcal{L}_{2nd} + \gamma \mathcal{L}_{reg},\quad (13)$$

where γ is a tradeoff parameter.

To optimize the proposed model, we minimize the mixed loss function defined as Eq. (13). We first perform forward propagation to obtain the loss, and then do back propagation to update all the parameters. For the LSTM-based STEM-GCN models, the weights of the network are updated through time back propagation. After obtaining the partial derivatives of the parameters. We optimize the model using Adaptive Moment Estimation (Adam) [37]. The full algorithm is presented in Algorithm 2.

Algorithm 2: Training of STEM-GCN

Require: Dynamic network: $\mathbb{G} = \{\mathcal{G}^1, \mathcal{G}^2, \dots, \mathcal{G}^T\}$, Sequence length: s , Number of sequence: \mathcal{N} , Parameters of MCS: W, b , Parameters of decoder: θ .

Ensure: Parameters of STEM-GCN.

1: Generate adjacency matrices A_t from \mathbb{G} ;

2: $\mathcal{L}_{\mathcal{N}} \leftarrow \text{WCS}\{A_1, \dots, A_T\}$;

3: **for** $n = 1$ to \mathcal{N} **do**

4: **for** $t = 1$ to s **do**

5: $\mathcal{H}(t) \leftarrow \text{MCS}(A_t)$;

6: **end for**

7: $\tilde{A}_{t,pre} \leftarrow D_{\theta}(\mathcal{H}(t))$;

8: $\mathcal{L}_{2nd} \leftarrow \|\left(\tilde{A}_{t,pre} - A_t\right) \odot \mathcal{B}\|_F^2$;

9: $\mathcal{L}_{mix} \leftarrow \mathcal{L}_{2nd} + \gamma \mathcal{L}_{reg}$;

10: $W, b, \theta \leftarrow \text{Adam}(\mathcal{L}_{mix}, W, b, \theta)$;

11: **end for**

12: Return W, b, θ .

4.4. Complexity Analysis

The complexity of STEM-GCN, like that of many other deep learning models, mostly depends on the weights of the model. A simple STEM-GCN model consists of GCN layer whose hidden size

is l , LSTM layer whose hidden size is m_1 , and the MLP containing m_2 layers with m_3 hidden units, then the complexity is obtained by

$$O(n^2) \sim n^2 l + 4(nm_1 + n^2 + m_1) + m_2 m_3 n,\quad (14)$$

where n is the number of nodes of the \mathcal{G}_t , $n^2 l$ is the complexity of GCN, $4(nm_1 + n^2 + m_1)$ is the complexity of the LSTM, and $m_2 m_3 n$ is the complexity of the MLP. The complexity varies with the structure of the model. Although there are many parameters, a test under GPUs acceleration can be completed in a few seconds.

5. Experiments

In this section, we present a comprehensive collection of experiments to demonstrate the performance of STEM-GCN. Our discussion includes a variety of datasets, compared methods, evaluated metrics, and tasks.

5.1. Datasets

To verify the effectiveness of our STEM-GCN, we conduct our experiments on both synthetic and real-world dynamic networks. The detailed statistics of the six datasets are summarized in Table 1.

5.1.1. Synthetic Dynamic Network

From existing network evolution models [38,39], some regular dynamic networks can be generated to use as standard datasets to judge the performance of the model. We call the synthetic data set the BA dynamic network. The method to generate dynamic networks based on the classic BA model [40] is as follows.

Assume that the number of generated nodes is 100, and each node has 5 attributes. The dynamic network is expressed as $\mathcal{G}(t) = \{\mathcal{V}(t), \mathcal{E}(t), \mathcal{X}(t)\}$ at time t , where $\mathcal{V}(t)$ and $\mathcal{E}(t)$ are the sets of vertices and edges, respectively. Here, $\mathcal{X}(t) \in R^{100 \times 5}$ indicates the feature matrix of the node. Considering that the link is related to the mutual influence between nodes, we denote by $\Psi(v_i)$ the comprehensive influence of node v_i . That is,

$$\Psi(v_i) = \sum_{m=1}^5 x_m^i + \frac{1}{|\Gamma(v_i)|} \times \sum_{v_j \in \Gamma(v_i)} \sum_{m=1}^5 x_m^j,\quad (15)$$

where $\Gamma(v_i)$ is the first-order neighbor nodes of node v_i , and x_m^i is the attribute values of node v_i . Since the comprehensive influence of the node determines the probability the node being selected into the newly added edge, the probability $P(v_i)$ is defined as

$$P(v_i) = \frac{\Psi(v_i)}{\sum_{v_j \in \mathcal{V}} \Psi(v_j)},\quad (16)$$

where $P(v_i)$ represents the probability of connecting to node v_i when a new node appears. The larger the value of $P(v_i)$ is, the easier it is to generate an edge between the new node and v_i . As new nodes emerge, a dynamic network is generated through this probabilistic preference. The network structure changes at each moment

Table 1
Basic statistics of the six datasets.

Dataset	Nodes	Edges	Degree	Timespan(days)
BA	100	50.6 K	753	100
RADOSLAW	167	82.9 K	993.1	271.4
HYPertext	113	20.8 K	368	2.5
ENRON	151	10 K	133	164.5
SFHH-CONF	403	70.3 K	348	2.0
FB-FORUM	899	33.7 K	74	164.5

are related to the attribute values of the nodes. The attribute values of each node can be generated by different random functions.

5.1.2. Real-World Dynamic Networks

(1) HYPERTEXT [20]: It is a dynamic network of face-to-face interpersonal communication. The data are collected through the wireless devices carried by people. Each node represents a person, and a link indicates the active contacts between people during 20-s intervals.

(2) ENRON and RADOSLAW [19]: Both of them are email networks, and each node represents an employee in a mid-sized company. Whenever an email is sent from one person to another, a link will appear. ENRON records email interactions for nearly six months and RADOSLAW lasts for nearly nine months.

(3) SFHH-CONF [41]: It is a human contact dynamic network where nodes represent humans and edges between them represent proximity. This data set describes the face-to-face interactions of 403 participants at the 2009 SFHH conference in Nice, France.

(4) FB-FORUM [19]: These data are attained from a Facebook-like online forum of students at the University of California at Irvine, in 2004. It is an online social network where nodes are users and edges represent interactions between them. The records were for more than five months.

Before training, the data is partitioned into several network snapshots, each of them containing data from a particular time window. Since the length of the time window for different datasets may vary according to the timespan, in order to obtain enough samples, in this paper, we divide the dynamic networks into 100 network snapshots in ascending order of timestamp, and use 16 consecutive snapshots as a set of samples, with the previous 15 snapshots as input and the last one as the output. Therefore, a total of 85 samples are obtained, of which the first 65 samples are used as the training set, and the last 20 samples are used as the test set.

5.2. Baseline

To verify the effectiveness of the STEM-GCN, we compared it with five baselines on link prediction of dynamic networks, which are introduced as follows.

- (1) GCRN [17]: It utilizes GCN to acquire node representations, passes them into LSTM and uses a fully connected network to decode the future network structures.
- (2) GCN + GRU [42,43]: Similar to GCRN for dynamic network link prediction, it also leverages GCN to extract node representations, which are used as input of LSTM to capture dynamics.
- (3) GC-LSTM [20]: It is a framework of encoder and decoder model for end-to-end dynamic network link prediction, in which two GCN are embedded into the LSTM to learn temporal and spatial feature.
- (4) E-LSTM-D [19]: It uses an encoder to embed the adjacency matrices into the vectors. The representations for the sequences of graphs are learned from the stacked LSTM module placed right behind the encoder, which will be further decoded to the real next network by a decoder.
- (5) TGAT [21]: It uses the self-attention mechanism and the novel functional time encoding technique to learn node embeddings as functions of time.

5.3. Evaluation Metrics

We used two different metrics to evaluate the performance of our STEM-GCN model and other baseline methods in the link prediction task. One of them is the area under the curve (AUC) [44], which is defined as

$$AUC = \frac{n + 0.5n'}{n}, \quad (17)$$

where n is the number of independent comparisons in which missing links and non-existent links are randomly selected to compare their score. Among these comparisons, n' times the existing link gets a higher score than the nonexistent link and $n - n'$ times they get the same score. The AUC can be interpreted as the probability that a randomly chosen missing link is given a higher score than a randomly chosen nonexistent link.

The other metric we use is error rate (ER) [19], which is defined as the ratio of the number of mis-predicted links to the total number of truly existing links. That is,

$$ER = \frac{N_{false}}{N_{true}}, \quad (18)$$

where N_{false} and N_{true} are the number of mis-predicted links and actually existing links, respectively.

5.4. Experimental Results

In our experiments, we first utilized the spatio-temporal semi-variogram to analyze the spatial and temporal changes in the dynamic networks. To validate the effectiveness of our model, we implemented ablation experiments on the two proposed methods MGCN without the weak correlation strategy and STEM-GCN. Then, we compared them with five state-of-the-art methods including GCRN [17], GCN + GRU [42,43], GC-LSTM [20], E-LSTM-D [19], TGAT [21] on the evaluation metrics AUC and ER. Finally, we evaluated the performance of the above models on tasks of link prediction, network visualization, and network reconstruction.

5.4.1. Dynamic Networks Analysis based on Spatio-temporal Semi-variogram

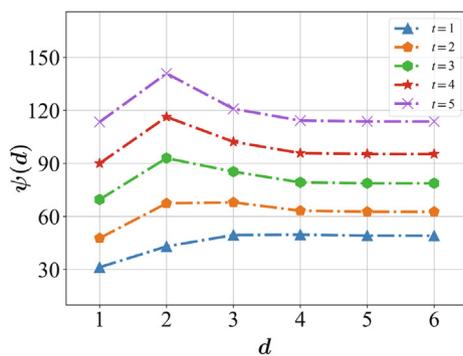
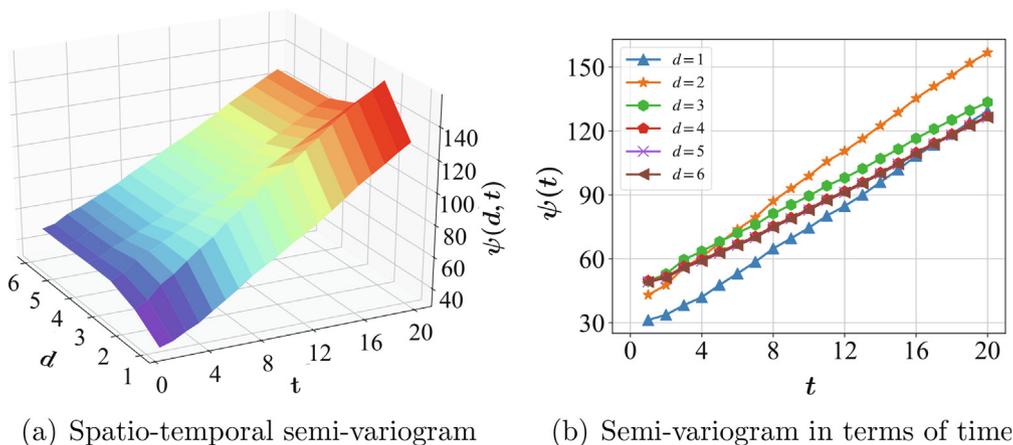
The objective of the spatio-temporal semi-variogram analysis on dynamic networks is to evaluate the spatial and temporal correlation in network snapshots. Aiming at such spatio-temporal analysis and taking corresponding measures for weak correlation will improve the accuracy of dynamic network link prediction.

In the first part, we utilized the spatio-temporal semi-variogram to analyze the spatial dependencies and temporal correlations in the synthetic BA dynamic network. The BA network is generated in Section 5.1.1, consisting of 20 network snapshots with time stamps. The change of spatio-temporal semi-variogram is shown in Fig. 3, which is the correlation of network snapshots with different counting time window lengths and distances.

Fig. 3(a) illustrated spatio-temporal semi-variogram of the BA network, which displays the correlation in the spatial distance and time directions. In the time t direction, all the values of the semi-variogram show continuous and slow increases in semi-variance for time steps of 0 to 20. Based on the definition of the semi-variogram, the small value indicates the high dependence between nodes separated at distance and time steps. It is interesting to notice that the semi-variogram value slowly grows along the time axis when $d = 1$, which suggests that the attributes of each node are highly related to the change of the combined value of connection information within 1 hop and its history. In the spatial distance d direction, one can notice that the value of semi-variogram increases continuously from 1 to 2, and is then steadily approaching 6. This indicates that each node is less affected by neighbor nodes whose distance is between 3 and 6 hops.

Fig. 3(b) illustrates the change of semi-variogram with different distances over time, i.e. the temporal correlation. It could be observed that, as time increases, the values of the semi-variogram at different distances are linear. Therefore, the temporal correlation between the node pairs is good.

The spatial dependence of the comprehensive attributes of the nodes can be observed in Fig. 3(c). The distance at which the model



(c) Semi-variogram in terms of spatial distance

Fig. 3. Spatio-temporal semi-variogram on BA.

is first flattened is known as the range. The location of the sample closer to the range is spatially autocorrelated, while the location farther from the range is not. After the shortest distance between the nodes exceeds 2, the value of the semi-variogram will remain unchanged, which indicates that the mutual influence of the two nodes is small. The model in this paper uses a two-layers graph convolution operation.

In the second part, we utilize the spatio-temporal semi-variogram to analyze the spatial dependencies and temporal correlations in the five real-world dynamic networks. In this paper, we

divide the dynamic network into 100 network snapshots at a fixed time interval, the spatio-temporal semi-variogram changes with different counting time window lengths and distances are shown as the subgraphs *b*. Since the real-world networks are very sparse, in order to more intuitively reflect the change of spatio-temporal semi-variogram, the subgraph *a* shows a spatio-temporal semi-variogram that divides the dynamic network into 20 network snapshots at a fixed time interval.

In the dynamic network HYPERTEXT, the change of spatio-temporal semi-variogram is shown Fig. 4. As evident in Fig. 4(b),

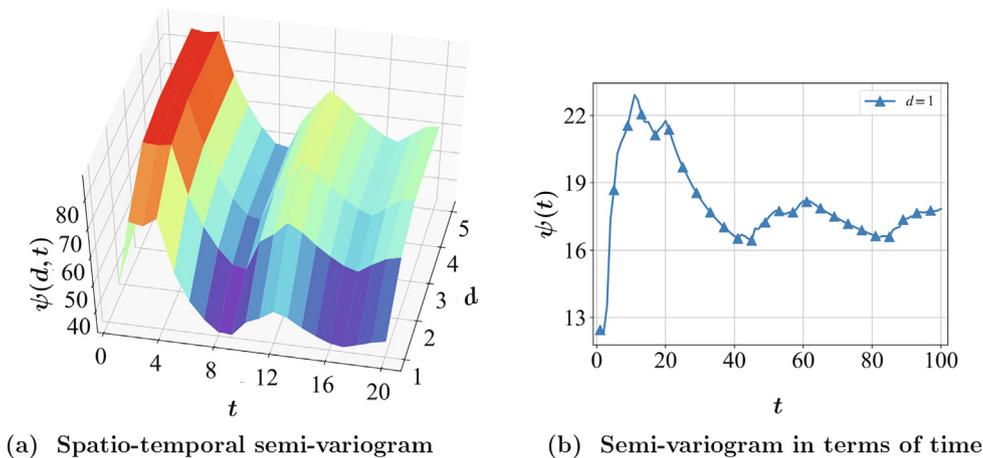


Fig. 4. Spatio-temporal semi-variogram on HYPERTEXT.

there is only one curve, indicating that the actual network is very sparse and the diameter of each network snapshot is one. As time increases, the semi-variogram value at different distances increases rapidly in the first 13 time steps, then decreases rapidly at the time of 40, and then fluctuates continuously. The curve of spatio-temporal semi-variogram fluctuates greatly, indicating that the temporal correlation between nodes is poor. As the dynamic network HYPERTEXT continues to add new members over time and establishes isolated interactions, in order to more intuitively reflect the change of spatio-temporal semi-variogram of the dynamic network, we divide the dynamic network into 20 network snapshots, and the change of spatio-temporal semi-variogram is shown in Fig. 4(a). The behavior is consistent with the Fig. 4(b), and the fluctuation is large. The dynamic network HYPERTEXT is unevenly distributed in space and changes irregularly over time.

Fig. 5 illustrates the change of spatio-temporal semi-variogram in the dynamic network ENRON. As evident in Fig. 5(b), there are two curves, indicating that the diameter of each network snapshot is 2. The value of semi-variogram increases continuously with time from 1 to 90 and then decreases to till 100. Its change is almost linear. As depicted in Fig. 5(a), the change of spatio-temporal semi-variogram of 20 network snapshots is similar to Fig. 5(b). Note that, as the size of the network snapshot increases, the diameter of the network increases to 4. This implies that email interactions continue to increase over time.

Fig. 6 illustrates the change of spatio-temporal semi-variogram in the dynamic network RADOSLAW. There are five curves in Fig. 6(b), that is, the diameter of each network snapshot is 5. Compared with the first two dynamic networks, RADOSLAW is denser. The value of semi-variogram increases rapidly to 47 during the time from 1 to 17 and then slowly drops to about 34 with slight fluctuations. Fig. 6(a) shows the change of spatio-temporal semi-variogram of 20 network snapshots. The change trend is consistent with Fig. 6(b). We also note that even if the size of the network snapshot changes, the diameter of the network remains unchanged. It may be caused by the interaction of email and the small number of new members during the short time.

In the dynamic network SFHH-CONF, the change of spatio-temporal semi-variogram is shown in Fig. 7. There are six curves in Fig. 7(b), and the diameter of each network snapshot is 6. The value of semi-variogram increases rapidly to 15 with time from 1 to 10 and then continue to fluctuate slightly until the 30th time step. From 15 to 30 time steps, the curve value increases to the highest point 22, and then slowly decreases until the 80th time step. The value of semi-variogram increases continuously with time from 81 to 86 and then decreases to approximately 100. Fig. 7(a) illustrates the change of spatio-temporal semi-variogram of 20 network snapshots. The trend of change is roughly the same as the Fig. 7(b). Similar to dynamic network RADOSLAW, even if the size of the network snapshot changes, the diameter of the network remains unchanged.

In the dynamic network FB-FORUM, the change of spatio-temporal semi-variogram is shown in Fig. 8. There are two curves in Fig. 8(b), with the diameter of each network snapshot being 7. The value of semi-variogram increases continuously with time

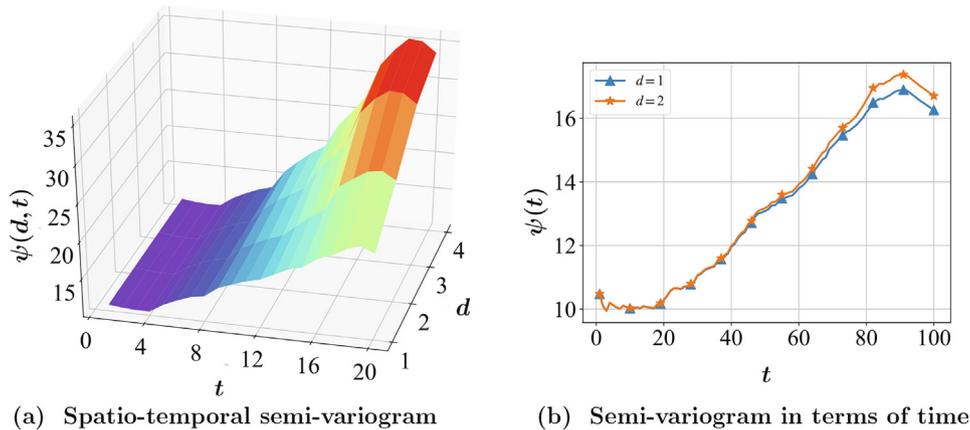


Fig. 5. Spatio-temporal semi-variogram on ENRON.

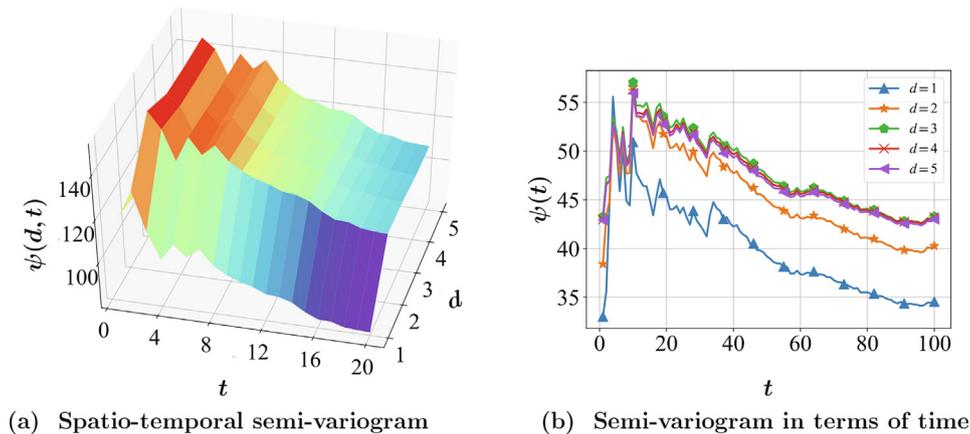


Fig. 6. Spatio-temporal semi-variogram on RADOSLAW.

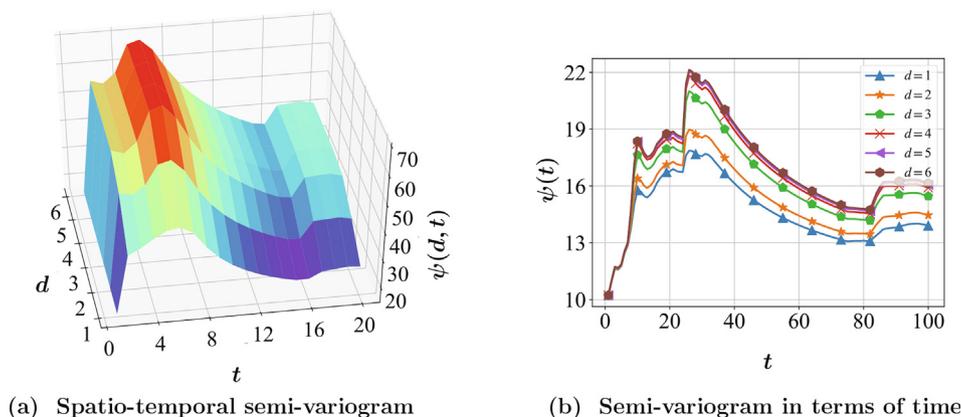


Fig. 7. Spatio-temporal semi-variogram on SFHH-CONF.

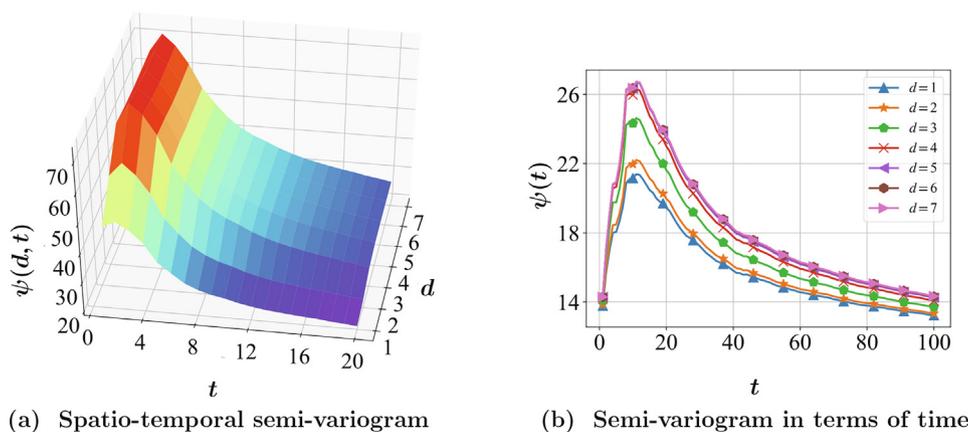


Fig. 8. Spatio-temporal semi-variogram on FB-FORUM.

from 1 to 10 and then decreases to approximately 100. Fig. 8(a) presents the change of spatio-temporal semi-variogram of 20 network snapshots. The change trend is consistent with the Fig. 8(b). Again, the size of the network snapshot changes while the diameter of the network remains unchanged.

5.4.2. Link Prediction

In this section, we present the performance results of various models for link prediction on different datasets. For each dynamic network, using the graph embeddings trained on network snapshots up to time step t , link prediction predicts the connections between nodes at time step $t + 1$. This approach has been widely used in evaluating the quality of dynamic node representations to predict the temporal evolution of network structures. Each model is trained on the input snapshots $\{\mathcal{G}_{t-s}, \dots, \mathcal{G}_{t-1}\}$, to obtain the latest graph embeddings $\tilde{A}_{t,pre}$. In our experiments, for each epoch, we feed 15 historical snapshots, $\{\mathcal{G}_{t-15}, \dots, \mathcal{G}_{t-1}\}$ to STEM-GCN in order to obtain \mathcal{G}_t .

To investigate the prediction performance, we reported the average values of the three performance metrics for all the 20 samples. AUC is the area under the receiver operating characteristic (ROC) curve and has the advantage of being independent of the classification threshold. The ROC curves of six methods in all datasets are presented in Fig. 9, where FPR represents false positive rate and TPR represents true positive rate. We can see that our method can achieve better performance than the other five methods. These results show that the performance of link prediction has been

greatly improved after taking both the topological and temporal features into account. It is interesting to note that the FPR value is almost zero on all data sets. This is because the real data sets are sparse networks, and most of the samples are negative examples (non-existent edges). Consequently, the predicted adjacency matrix has only a few non-zero entries, resulting in FPR being zero.

Table 2 depicts the performance of various models for link prediction on different datasets. This table shows that our methods STEM-GCN and MGCN (without the weak correlation strategy) all have higher AUC values and lower ER values. For synthetic network BA, the AUC value of our methods, close to 87%, is the highest, and the ER value of 0.27 is the lowest. This is because the dynamic network BA is evolved from specific rules, and the semi-variogram of BA is linear with respect to temporal correlation and spatial distance. Since the dynamic networks HYPERTEXT and SFHH-CONF have highly dynamic edge connections and short duration, it is difficult to sample a complete network structure at any time. As a result, the AUC of these two data sets is less than 80%, and the error rate is higher than other data sets. Compared with other benchmark methods, our method STEM-GCN has an AUC of 81% and an increase of about 10% in the dynamic network ENRON. Since ENRON is very sparse, this shows that STEM-GCN is more suitable for sparse networks.

The semi-variogram fluctuates greatly with respect to temporal correlation and spatial distance, indicating that the temporal correlation between nodes is weak. For real-world dynamic networks RADOSLAW, HYPERTEXT, SFHH-CONF and FB-FORUM with large fluctuations in the spatio-temporal semi-variogram curve, we

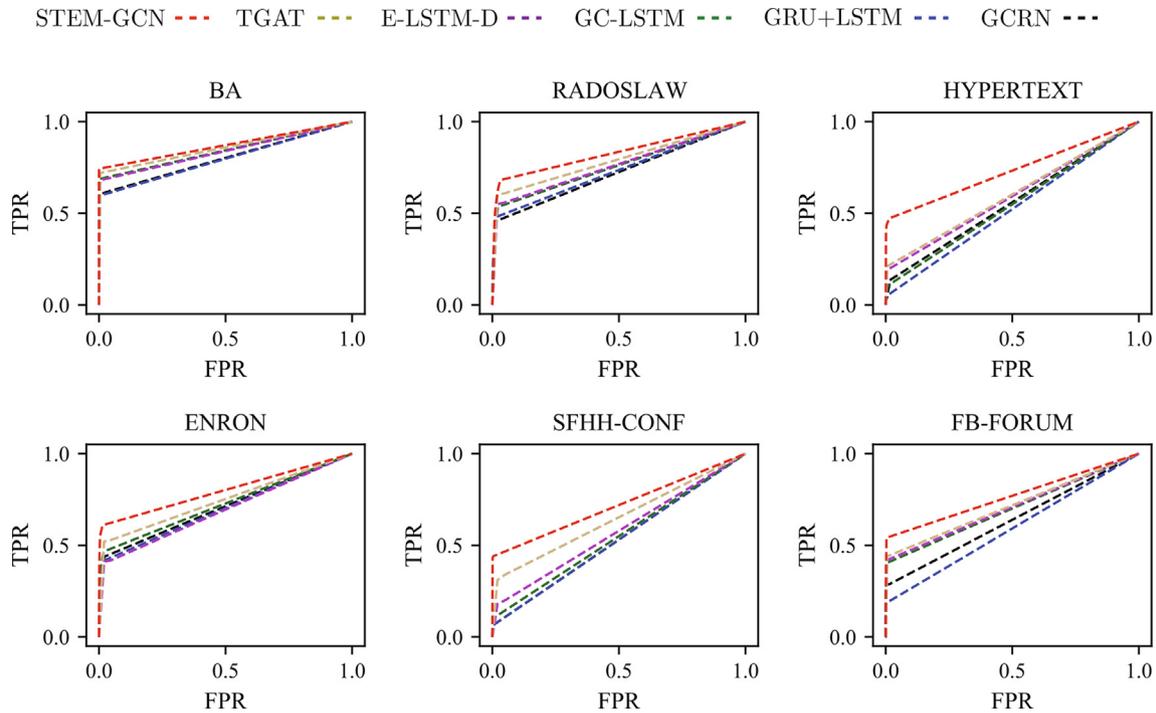


Fig. 9. ROC curves of different methods.

Table 2
Performance of link prediction on AUC and ER.

Metric	Method	BA	RADOSLAW	HYPERTEXT	ENRON	SFHH-CONF	FB-FORUM
AUC	GCRN	0.8031	0.7224	0.5595	0.7137	0.5307	0.6331
	GRU + LSTM	0.7985	0.7327	0.5227	0.7064	0.5332	0.5908
	GC-LSTM	0.8444	0.7573	0.5490	0.7258	0.5503	0.6939
	E-LSTM-D	0.8314	0.7341	0.5529	0.7062	0.5931	0.7595
	TGAT	0.8463	0.8165	0.5660	0.7813	0.6947	0.7636
	MGCN	0.8546	0.8201	0.7060	0.8133	0.7153	0.7694
	STEM-GCN	0.8689	0.8298	0.7309	0.8053	0.7303	0.7804
ER	GCRN	0.3964	0.9528	1.2214	1.0237	1.0297	1.0872
	GRU + LSTM	0.4062	0.9398	1.1023	1.1472	1.0238	1.1429
	GC-LSTM	0.3142	1.0205	1.0878	1.0302	1.0201	1.1045
	E-LSTM-D	0.4022	1.0013	1.0748	1.0302	1.0406	1.0986
	TGAT	0.3982	0.8983	0.9577	0.9504	1.0479	1.1042
	MGCN	0.2753	0.8882	0.8029	0.8802	0.7918	1.0640
	STEM-GCN	0.2656	0.8769	0.7872	0.8535	0.7741	0.9854

introduce a weak correlation smoothing strategy and make link predictions. It is found that our method STEM-GCN has improved AUC by 1% and ER has been reduced by 2%, compared to the previous method MGCN. Note that after the introduction of the weak correlation smoothing strategy, the AUC value of the ENRON dynamic network drops by 1%, because the spatio-temporal semi-variogram curve is approximately linear, and the introduction of the correlation smoothing strategy may destroy the temporal correlation. Although AUC has dropped, the error rate has dropped by 3%. In summary, the STEM-GCN proposed in this paper is more accurate than other approaches in predicting future changes in the network.

Moreover, for the 20 test samples with $\mathcal{G}_{65+\lambda}$, we test dynamic link prediction performances with AUC and ER, obtained by STEM-GCN. Here 65 is the number of snapshots in the training set, and λ represents the number of snapshots between the current predicted one and the 65th one, varying from 1 to 20. As λ varies, we can see how long STEM-GCN can predict network evolution

with satisfying performance. The results are shown in Fig. 10. We can see that, with the increase of λ , the AUC decreases, while the ER increases. This shows that for most dynamic networks, it is indeed relatively difficult to make long-term predictions of the structure. Note that, for BA, the changing trends of prediction performances are relatively stable. Maybe because the synthetic network structure evolves regularly, making the collection of snapshots easy to predict. Interestingly, for ENRON and FB-FORUM, AUC steadily rises with the increase of λ over time interval, which may be because the networks are sparser than others. These results explain why we can achieve better performances on sparse dynamic networks.

5.4.3. Beyond Link Prediction

Network Reconstruction. A good low-dimensional representation of the nodes can accurately reconstruct the network. Our STEM-GCN model learns low-dimensional representations of each node in the process of link prediction for the task of network reconstruction.

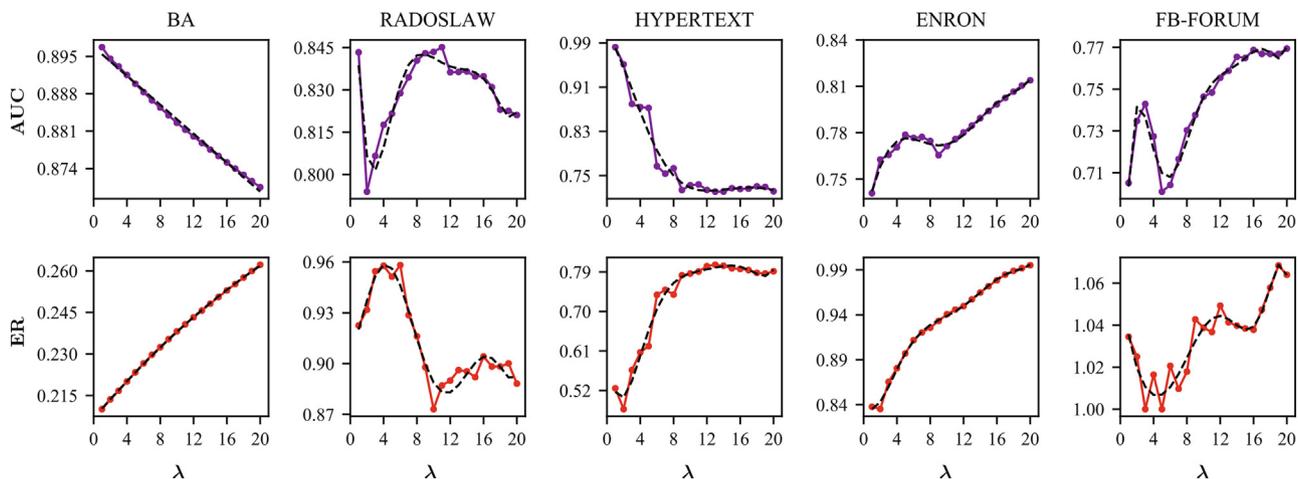


Fig. 10. Dynamic link prediction performances on the AUC and ER.

tion. Through the decoder in the STEM-GCN model, the edges between pairs of nodes are reconstructed from a low-dimensional representation. We then represent the connection probability of the edge according to the structural similarity of the corresponding node pair. The probabilities of the edges are sorted in descending order and the ratio of real links in the top k edges is obtained as the reconstruction precision, where k is 2500.

Table 3 shows the Mean Average Precision (MAP) [45] on our datasets. It is observed that STEM-GCN outperforms all baselines. In the BA network, the average MAP of STEM-GCN is 99.8% while the best method has 98.7%. In the RADOSLAW and ENRON networks, compared with the optimal method, the average MAP of STEM-GCN is 86.28% and 83.68%, showing an increase of 1%. In the HYPERTEXT network, the average MAP of STEM-GCN is 89.64% while the optimal method has 84.2%. In the SFHH-CONF network, the average MAP of STEM-GCN is 98.72% while the opti-

mal method has 87.4%. In the FB-FORUM networks, compared with the optimal method, the average MAP of STEM-GCN is 84.24%, showing an increase of 3%.

Network Visualization. One important application of graph embedding is network visualization. We carry out our experiments on karate club data, and the known network structure is shown in Fig. 11(a). First, the \mathcal{G}_{t-1} is obtained by randomly removing ten links from the original network \mathcal{G}_t . Then, we utilize the network \mathcal{G}_{t-1} to infer \mathcal{G}_t . Finally, the low-dimensional embeddings for each node are generated by STEM-GCN during the process of link prediction. We take the embeddings as input, and the output of the stacked memory cell structure is visualized by method PCA [46], as shown in Fig. 11(b). It is observed that the nodes of the same classes are closely connected, while those of the different classes are far apart. This verifies the excellent transferability of the proposed STEM-GCN model. STEM-GCN can be used for link prediction

Table 3
Average MAP of network reconstruction.

Method	BA	RADOSLAW	HYPERTEXT	ENRON	SFHH-CONF	FB-FORUM
GCRN	0.7888	0.5548	0.5384	0.5940	0.5784	0.6216
GRU + LSTM	0.7996	0.5204	0.5008	0.5756	0.5796	0.6212
GC-LSTM	0.9872	0.8592	0.8420	0.8188	0.8740	0.8008
E-LSTM-D	0.9723	0.8363	0.8517	0.8023	0.9120	0.8119
TGAT	0.9856	0.8496	0.8639	0.8108	0.9341	0.8165
STEM-GCN	0.9980	0.8628	0.8964	0.8368	0.9872	0.8424

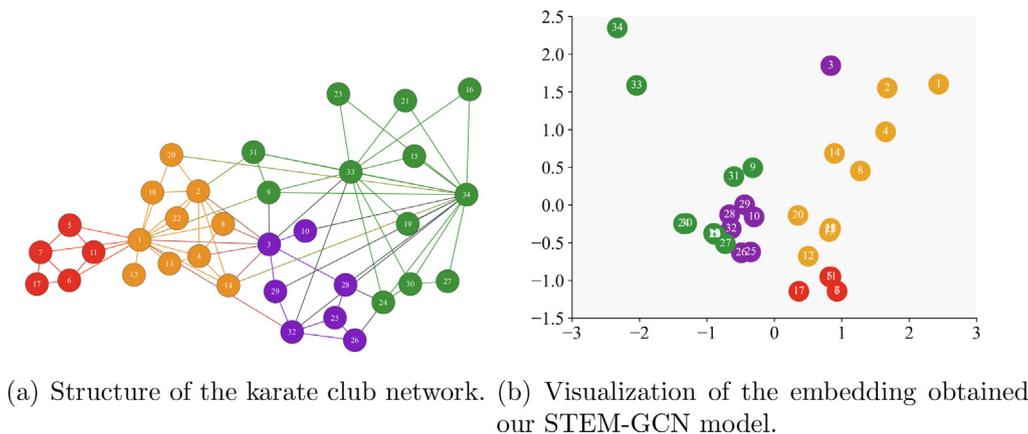


Fig. 11. Visualization of the karate club network.

tasks, and the embedding vector obtained by link prediction can also be used to effectively solve the node classification problem.

5.5. Parameter Sensitivity

The performance of our STEM-GCN model is mainly determined by the model’s structure. We test the model with different numbers of GCN-layers and decoder-layers to prove the validity of the structures. In addition, we use the spatio-temporal semi-variogram to model the dynamic networks and determine the number of graph convolution layers (GCN-layers). The essence of the spatio-temporal semi-variogram operation, in terms of space, is to aggregate the attributes of different-order neighbor. And the evolution of semi-variogram across different distances is equivalent to different GCN-layers. Next, we will investigate their influences on the model performance.

The performances of the AUC and ER with different GCN-layers are shown in Fig. 12. Fig. 12(a) shows the AUC with different GCN-layers. One can notice that the AUC increases as the number of

GCN-layers increases from 1 to 2. However, the AUC starts to decrease when the number of GCN-layers increased from 3 to 4. Fig. 12(b) shows ER of different GCN-layer. As the number of GCN-layers increases from 1 to 2, the ER decreases; when the number of GCN-layers increased from 3 to 4, ER begins to increase. Therefore, we employ 2 as the optimal number of GCN-layers in our model as the scores for the AUC are highest and the values of ER are lowest. At the same time, the semi-variogram analysis also tells us that the performance of the model is the best when the number of GCN-layers is 2. The theoretical analysis is consistent with the experiments, so we choose a two-layer GCN in the previous experiments.

The performances of the AUC and ER with different decoder-layers are shown in Fig. 13. Fig. 13(a) shows the AUC with different decoder-layers. The AUC increases with an increase in the number of decoder-layers from 1 to 2, while the AUC starts to decrease when the number of decoder-layers increases from 3 to 4. Fig. 13 (b) shows the ER of different decoder-layers. As the number of decoder-layers increases from 1 to 2, the ER tends to flatten. Also,

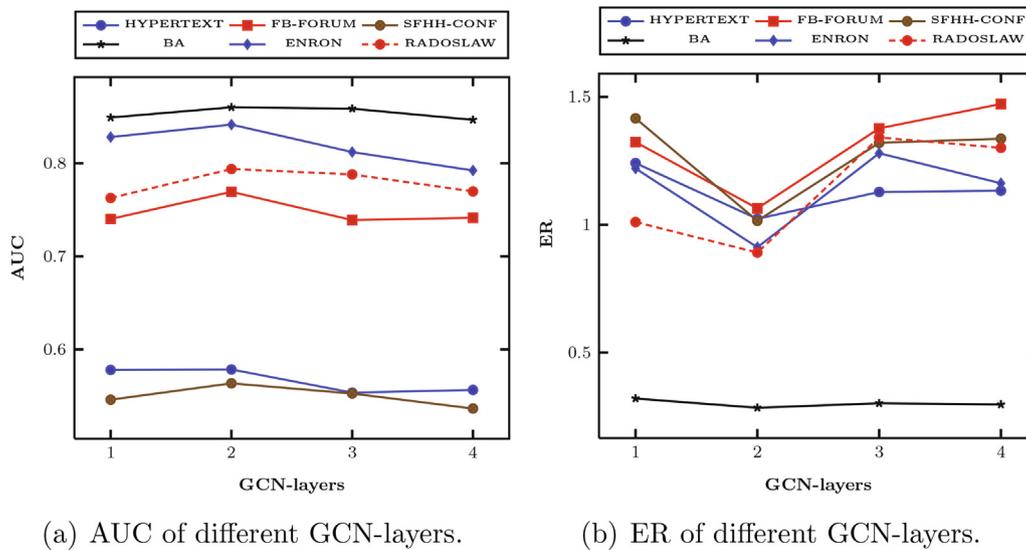


Fig. 12. AUC and ER of different GCN-layers.

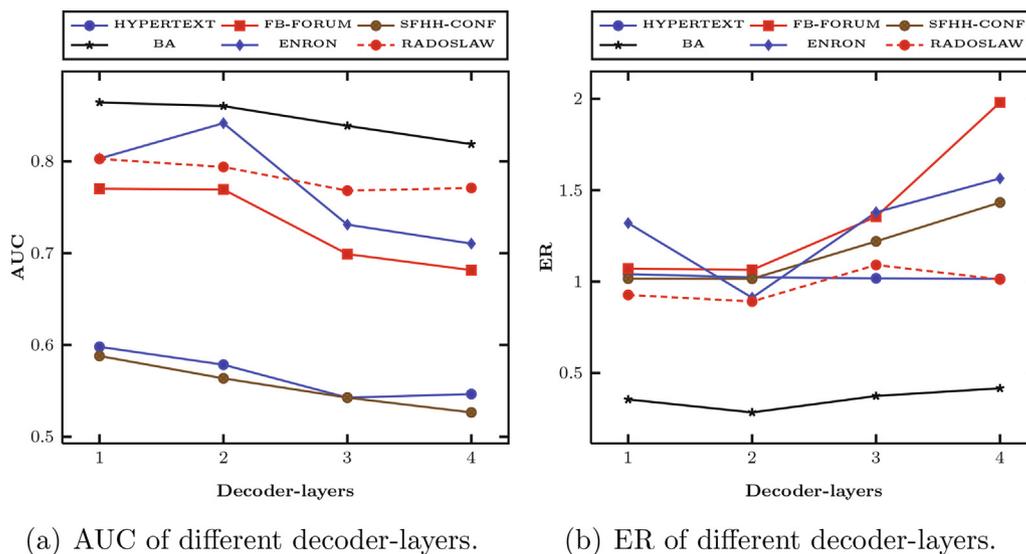


Fig. 13. AUC and ER of different decoder-layers.

ER increases the most when decoder-layers are from 3 to 4. Therefore, we choose 2-fully connected layers in the decoder. Moreover, it seems that, on the synthetic data set, AUC is the largest and ER is the smallest because the synthetic data set is a regular data set with the edge-first connection.

6. Conclusions and Future Works

In this paper, based on the practical challenges, we proposed a novel STEM-GCN model for dynamic network link prediction, which overcomes the limitations of capturing temporal information and the high number of model parameters. For different real-world dynamic networks, we take advantage of spatio-temporal semi-variogram for analyzing the correlation of network snapshots in time and space. We find that the correlation of network snapshots is non-stationary. Aiming at network snapshots sequences which have weak correlation in time, the developed STEM-GCN framework introduced the weak correlation smoothing strategy and also improved the accuracy of link prediction by 1% (compared with MGCN without aforesaid strategy). Moreover, STEM-GCN reduced the number of model parameters and achieved a more efficient temporal learning by memory cell structure. The experimental results demonstrated that the proposed model can capture temporal patterns on synthetic and real-world datasets, and outperformed other well-known state-of-the-art methods in dynamic link prediction. In general, our approach shows promise for revealing the evolving mechanism of real-world dynamic networks.

CRedit authorship contribution statement

Liping Yang: Conceptualization, Data curation, Writing - original draft. **Xin Jiang:** Conceptualization, Methodology, Software. **Yiming Ji:** Methodology, Software, Validation. **Hua Wang:** Writing - review & editing. **Ajith Abraham:** Writing - review & editing. **Hongbo Liu:** Writing - review & editing, Supervision.

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.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 62176036, 61772102) and the Liaoning Collaborative Fund (Grant No. 2020-HYLH-17).

References

- [1] L. Lü, T. Zhou, Link prediction in complex networks: A survey, *Physica A: Statistical Mechanics and Its Applications* 390 (6) (2011) 1150–1170.
- [2] A.K. Singh, K. Lakshmanan, PILHNB: Popularity, interests, location used hidden naive bayesian-based model for link prediction in dynamic social networks, *Neurocomputing* 461 (2021) 562–576.
- [3] Z. Wang, Y. Lei, W. Li, Neighborhood attention networks with adversarial learning for link prediction, *IEEE Transactions on Neural Networks and Learning Systems* 32 (8) (2020) 3653–3663.
- [4] C.-M. Ting, S.B. Samdin, M. Tang, H. Ombao, Detecting dynamic community structure in functional brain networks across individuals: A multilayer approach, *IEEE Transactions on Medical Imaging* 40 (2) (2021) 468–480.
- [5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, *IEEE Transactions on Neural Networks and Learning Systems* 32 (1) (2021) 4–24.
- [6] T. Ahmad, L. Jin, X. Zhang, L. Lin, G. Tang, Graph convolutional neural network for action recognition: A comprehensive survey, *IEEE Transactions on Artificial Intelligence* 2 (2) (2021) 128–145.
- [7] P. Cui, X. Wang, J. Pei, W. Zhu, A survey on network embedding, *IEEE Transactions on Knowledge and Data Engineering* 31 (5) (2019) 833–852.
- [8] G. Spadon, S. Hong, B. Brandoli, S. Matwin, J.F. Rodrigues-Jr, J. Sun, Pay attention to evolution: Time series forecasting with deep graph-evolution learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021) 1–17, <https://doi.org/10.1109/TPAMI.2021.3076155>.
- [9] A. Sankar, Y. Wu, L. Gou, W. Zhang, 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.
- [10] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, H. Liu, Graph learning: A survey, *IEEE Transactions on Artificial Intelligence* 2 (2) (2021) 109–127.
- [11] Y. Xiong, Y. Zhang, X. Kong, Y. Zhu, NetCycle+: A framework for collective evolution inference in dynamic heterogeneous networks, *IEEE Transactions on Knowledge and Data Engineering* 30 (8) (2018) 1547–1560.
- [12] M. Yang, J. Liu, L. Chen, Z. Zhao, X. Chen, Y. Shen, An advanced deep generative framework for temporal link prediction in dynamic networks, *IEEE Transactions on Cybernetics* 50 (12) (2020) 4946–4957.
- [13] L. Wang, V. Narayanan, Y.-C. Yu, Y. Park, J.-S. Li, A nested two-stage clustering method for structured temporal sequence data, *Knowledge and Information Systems* 63 (7) (2021) 1627–1662.
- [14] D. Zhu, P. Cui, Z. Zhang, J. Pei, W. Zhu, High-order proximity preserved embedding for dynamic networks, *IEEE Transactions on Knowledge and Data Engineering* 30 (11) (2018) 2134–2144.
- [15] Y. Wang, Y.-Y. Chang, Y. Liu, J. Leskovec, P. Li, Inductive representation learning in temporal networks via causal anonymous walks, in: *Proceedings of the International Conference on Learning Representations*, 2021, pp. 1–22.
- [16] P.B. Weerakody, K.W. Wong, G. Wang, W. Ela, A review of irregular time series data handling with gated recurrent neural networks, *Neurocomputing* 441 (2021) 161–178.
- [17] Y. Seo, M. Defferrard, P. Vandergheynst, X. Bresson, Structured sequence modeling with graph convolutional recurrent networks, in: *Proceedings of the International Conference on Neural Information Processing*, 2018, pp. 362–373.
- [18] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.
- [19] J. Chen, J. Zhang, X. Xu, C. Fu, D. Zhang, Q. Zhang, Q. Xuan, E-LSTM-D: A deep learning framework for dynamic network link prediction, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51 (6) (2021) 3699–3712.
- [20] J. Chen, X. Wang, X. Xu, GC-LSTM: graph convolution embedded LSTM for dynamic network link prediction, *Applied Intelligence* (2021) 1–16.
- [21] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, K. Achan, Inductive representation learning on temporal graphs, in: *International Conference on Learning Representations*, 2020, pp. 1–19.
- [22] L. Zhu, D. Guo, J. Yin, G.V. Steeg, A. Galstyan, Scalable temporal latent space inference for link prediction in dynamic social networks, *IEEE Transactions on Knowledge and Data Engineering* 28 (10) (2016) 2765–2777.
- [23] L. Yang, Y. Yang, G.B. Mgaya, B. Zhang, L. Chen, H. Liu, Novel fast networking approaches mining underlying structures from investment big data, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51 (10) (2020) 6319–6329.
- [24] K. Liu, H. Liu, T.E. Ward, H. Wang, Y. Yang, B. Zhang, X. Wu, Self-adaptive skeleton approaches to detect self-organized coalitions from brain functional networks through probabilistic mixture models, *ACM Transactions on Knowledge Discovery from Data* 15 (5) (2021) 1556–4681.
- [25] G.H. Nguyen, J.B. Lee, R.A. Rossi, N.K. Ahmed, E. Koh, S. Kim, Continuous-time dynamic network embeddings, in: *Proceedings of the Companion Proceedings of the The Web Conference*, 2018, pp. 969–976.
- [26] R. Trivedi, H. Dai, Y. Wang, L. Song, Know-Evolve: Deep temporal reasoning for dynamic knowledge graphs, in: *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3462–3471.
- [27] R. Trivedi, M. Farajtabar, P. Biswal, H. Zha, DyRep: Learning representations over dynamic graphs, in: *Proceedings of the International Conference on Learning Representations*, 2019, pp. 1–25.
- [28] L. Zhou, Y. Yang, X. Ren, F. Wu, Y. Zhuang, Dynamic network embedding by modeling triadic closure process, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 571–578.
- [29] F. Manessi, A. Rozza, M. Manzo, Dynamic graph convolutional networks, *Pattern Recognition* 97 (2020) 107000.
- [30] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, C.E. Leiserson, EvolveGCN: Evolving graph convolutional networks for dynamic graphs, in: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020, pp. 5363–5370.
- [31] C. Chen, K. Li, S.G. Teo, X. Zou, K. Wang, J. Wang, Z. Zeng, Gated residual recurrent graph neural networks for traffic prediction, in: *Proceedings of the AAAI conference on artificial intelligence*, 2019, pp. 485–492.
- [32] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, C. Peng, Spatio-temporal analysis and prediction of cellular traffic in metropolis, *IEEE Transactions on Mobile Computing* 18 (9) (2018) 2190–2202.
- [33] J. Ou, J. Sun, Y. Zhu, H. Jin, Y. Liu, F. Zhang, J. Huang, X. Wang, Stp-trellisnets: Spatial-temporal parallel trellisnets for metro station passenger flow prediction, in: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1185–1194.
- [34] C.H. Liu, C. Piao, X. Ma, Y. Yuan, J. Tang, G. Wang, K.K. Leung, Modeling citywide crowd flows using attentive convolutional LSTM, in: *Proceedings of the IEEE 37th International Conference on Data Engineering*, 2021, pp. 217–228.

- [35] L. Fang, X. Cheng, H. Wang, L. Yang, Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks, *IEEE Internet of Things Journal* 5 (4) (2018) 3091–3101.
- [36] B. Kumar, E. Reddy, Modified Floyd-Warshall algorithm for cache management in information centric network, *International Journal of Intelligent Engineering and Systems* 13 (1) (2020) 146–155.
- [37] L. Zhang, Z. Li, Y. Hu, C. Smith, E.M.G. Farewik, R. Wang, Ankle joint torque estimation using an EMG-Driven neuromusculoskeletal model and an artificial neural network model, *IEEE Transactions on Automation Science and Engineering* 18 (2) (2021) 564–573.
- [38] A. Barrat, M. Barthélemy, A. Vespignani, Modeling the evolution of weighted networks, *Physical Review E* 70 (6) (2004) 066149.
- [39] Z. Bu, Z. Xia, J. Wang, C. Zhang, A last updating evolution model for online social networks, *Physica A: Statistical Mechanics and Its Applications* 392 (9) (2013) 2240–2247.
- [40] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, *Science* 286 (5439) (1999) 509–512.
- [41] R. Rossi, N. Ahmed, The network data repository with interactive graph analytics and visualization, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 1–15.
- [42] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: *International Conference on Learning Representations*, 2017, pp. 1–14.
- [43] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in: *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 1–9.
- [44] F. Cheng, X. Zhang, C. Zhang, J. Qiu, L. Zhang, An adaptive mini-batch stochastic gradient method for AUC maximization, *Neurocomputing* 318 (2018) 137–150.
- [45] S. Ding, R. Li, Forecasting the sales and stock of electric vehicles using a novel self-adaptive optimized grey model, *Engineering Applications of Artificial Intelligence* 100 (2021) 104148.
- [46] J. Zhang, M. Chen, X. Hong, Nonlinear process monitoring using a mixture of probabilistic PCA with clusterings, *Neurocomputing* 458 (2021) 319–326.



Liping Yang is currently a Ph.D. candidate in the College of Artificial Intelligence, Dalian Maritime University, China. Her research interests include complex network, big data, and graph neural network.



Xin Jiang is currently a Ph.D. candidate in the College of Artificial Intelligence, Dalian Maritime University, China. His research interests include complex network, reinforcement learning, and graph neural network.



Yiming Ji is currently a postgraduate in the College of Artificial Intelligence, Dalian Maritime University, China. His research interests include reinforcement learning, recurrent neural network, and graph neural network.



Hua Wang received his PhD from University of South Carolina in 2005. After visiting University of Florida for three years, he joined Georgia Southern University in 2008, where he has been working ever since. Professor Wang's research interests mainly include extremal graph theory, enumerative combinatorics, combinatorial number theory, deep learning, etc.



Ajith Abraham received Ph.D. degree in Computer Science from Monash University, Melbourne, Australia (2001) and a Master of Science Degree from Nanyang Technological University, Singapore (1998). Ajith's research and development experience includes nearly 30 years in the Academia and Industry. He works in a multi-disciplinary environment involving machine (network) intelligence, cyber security, sensor networks, Web intelligence, scheduling, data mining and applied to various real world problems. He is an author/co-author of 1,400 + publications and some of the works have also won best paper awards at International conferences and also received several citations. Since 2008, Dr. Abraham was the Chair of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing (2008–2021) and served as a Distinguished Lecturer of IEEE Computer Society representing Europe (2011–2013). Dr. Abraham was the editor-in-chief of *Engineering Applications of Artificial Intelligence (EAAI)* and serves/served the editorial board of over 15 International Journals indexed by Thomson ISI.



Hongbo Liu received his three level educations (B. Sc., M. Sc., Ph.D.) at the Dalian University of Technology, China. He is with the College of Artificial Intelligence, Dalian Maritime University. Professor Liu's research interests are in cognitive computing, machine learning, big data, etc. He participates and organizes actively international conference and workshop and international journals/publications. He received the New Century Excellent Talents Award in 2010.