GAEN: Graph Attention Evolving Networks

Min Shi[1] , **Yu Huang**[1] , **Xingquan Zhu**[1] , **Yufei Tang**[1] , **Yuan Zhuang**[2] and **Jianxun Liu**[3]

[1]Dept. of Computer & Elec. Engineering and Computer Science, Florida Atlantic University, USA
[2] State Key Lab of Info. Eng. in Surveying, Mapping and Remote Sensing, Wuhan University, China
[3]School of Computer Science and Engineering, Hunan University of Science and Technology, China
{mshi2018, yhwang2018, xzhu3, tangy}@fau.edu, {zhy.0908, ljx529}@gmail.com

Abstract

Real-world networked systems often show dynamic properties with continuously evolving network nodes and topology over time. When learning from dynamic networks, it is beneficial to correlate all temporal networks to fully capture the similarity/relevance between nodes. Recent work for dynamic network representation learning typically trains each single network independently and imposes relevance regularization on the network learning at different time steps. Such a snapshot scheme fails to leverage topology similarity between temporal networks for progressive training. In addition to the static node relationships within each network, nodes could show similar variation patterns (*e.g.*, change of local structures) within the temporal network sequence. Both static node structures and temporal variation patterns can be combined to better characterize node affinities for unified embedding learning. In this paper, we propose Graph Attention Evolving Networks (GAEN) for dynamic network embedding with preserved similarities between nodes derived from their temporal variation patterns. Instead of training graph attention weights for each network independently, we allow model weights to share and evolve across all temporal networks based on their respective topology discrepancies. Experiments and validations, on four real-world dynamic graphs, demonstrate that GAEN outperforms the state-of-the-art in both link prediction and node classification tasks.

1 Introduction

Networks are ubiquitous to organize real-world data and systems, such as social networks, citation networks, communication networks, and protein interaction networks [Zhang *et al.*, 2020]. The past few years have witnessed many successful graph-based applications such as text classification [Yao *et al.*, 2019], image recognition [Chen *et al.*, 2019] and traffic prediction [Zhao *et al.*, 2019], which are largely benefited from the fruitful network embedding techniques. Given a network, the key to designing a successful network embedding framework lies in the high-efficient mechanism to fully capture the complex relationships (*e.g.*, structure and feature similarities) among nodes, such that nodes with close relationships are represented with similar embedding vectors.

Traditional methods, including DeepWalk [Perozzi *et al.*, 2014] and Node2vec [Grover and Leskovec, 2016], perform truncated random walks over the entire network to capture the neighborhood relationships between nodes. Recent work mainly focuses on the graph neural networks [Wu *et al.*, 2020], where the idea is that nodes can aggregate features from their respective neighbors (*e.g.*, first-order and second-order) to preserve structure relationships in the embedding space with multiple graph convolution kernels or filters [Kipf and Welling, 2017; Veličković *et al.*, 2017]. While existing methods are designed to characterize relationships between nodes from different views and levels, *i.e.*, local and global neighborhood relationships, for quality node representation learning, they are specialized in modeling static networks with invariable node sets and topology structures.

In reality, most real-world networked systems are in an evolving process [Fournier-Viger *et al.*, 2020], where network nodes and structures are continuously changing over time. For example, the dynamic social network arises from new edges and nodes being frequently created when new actors join the network or new friendships are being established. Analogically, new Web pages and their links are interminably added and old ones are deleted, resulting in a highly dynamic Web page citation network. In the biological domain, the physical interactions among proteins will change with the age of proteins, resulting in an evolving protein-protein interaction network [Hegele *et al.*, 2012]. In the dynamic environment, the network evolution trajectory can be recorded and streaming networked data can be organized in a temporal sequence composed of multiple varying networks at different time steps. Compared with a single static network, dynamic networks inherently reflect more complex relationships between nodes, *i.e.*, the node structure evolutionary patterns. As a result, in addition to the structural node relationships observed from the most recent static network, the correlations between the current network with all its historical networks within a temporal sequence can be modeled to enhance the representation learning of dynamic networks.

To capture network dynamics, existing methods typically impose some regularization/constrain on embedding learning of networks at different time steps. For example, in the litera-

ture [Seo *et al.*, 2018; Manessi *et al.*, 2020], Long Short-Term Memory (LSTM) networks are used to model a sequence of temporal networks, *i.e.*, each node can utilize its past node features for enriched embedding learning, where the static network at each time step is modeled by independent Graph Convolutional Networks (GCN) [Kipf and Welling, 2017]. Similarly, dynamic attention network [Sankar *et al.*, 2019] is also proposed to capture temporal dynamics where each node can use its historical representations, while each static network is modeled based on a respective Graph Attention Networks (GAT) [Veličković *et al.*, 2017]. Despite both network dynamics and structures can be preserved, above methods fail to consider the following two attractive properties of dynamic networks:

- **Similar Network Topologies:** While a network G_t at time t evolves to network G_{t+1} at time $(t+1)$, the network topologies between G_t and G_{t+1} are usually similar. Instead of learning the two networks independently, weights of the learning model (*e.g.*, GCN) for G_t can be evolved to the model for G_{t+1}, thereby making the training across all temporal networks more efficient.
- **Temporal Variation Patterns:** While a network changes over time, nodes would demonstrate some temporal variation (*e.g.*, change of local structures) patterns, *i.e.*, adding or removing a link with a neighborhood, observed from a fixed number of sequential evolution steps. Such pattern information can be explicitly modeled in that nodes with similar patterns being close in the embedding space.

To this end, in this paper we propose a novel network representation learning framework, Graph Attention Evolving Networks (GAEN), for dynamic networks organized as a sequence of evolving networks at different time steps. Instead of learning the static networks at each time step independently as most existing methods do, GAEN learns to evolve and share model weights across all temporal networks based on a Gated Recurrent Unit (GRU) [Cho *et al.*, 2014] by learning the topological discrepancies between networks. To model temporal node variation patterns, we first stack each network (*e.g.*, adjacency matrix) with all its historical networks to form a three-way tensor, from which node temporal variation pattern similarities are derived based on the tensor factorization technique. Then, the pattern information is explicitly incorporated into the respective neighborhood attention calculation for enhanced network representation learning at each dynamic time point. Experimental results on four real-world datasets in both dynamic link prediction and node classification tasks demonstrate the effectiveness of the proposed approach.

It is necessary to mention that a recent work [Pareja *et al.*, 2020], along a similar direction, proposes to evolve GCN weights with a recurrent neural network by taking the node features at different time steps as inputs. In comparison, our model considers the topological discrepancies between temporal networks to evolve and share multi-head graph attention network learning weights. In addition, to the best of our knowledge, this is the first work to explicitly represent and incorporate dynamic node variation patterns for learning dynamic graph attention networks.

In summary, our contribution is threefold: 1) We propose a novel graph attention network called GAEN for learning temporal networks; 2) We propose to evolve and share multi-head graph attention network weights by using a GRU to learn the topology discrepancies between temporal networks; and 3) We propose to represent temporal node variation patterns and explicitly incorporate pattern information for enhanced neighborhood attention and network embedding learning.

2 Related Work

Graph Neural Networks. Inspired by the success of Convolutional Neural Networks (CNN) which typically focus on the grid-like data, Graph Neural Networks (GNN) for learning irregular networked data have seen an explosion in attention over the past few years [Wu *et al.*, 2020]. The incredible embedding learning ability of GNN lies in various efficient graph convolution filters. GCN adopts a spectral-based filter where each node generates the representation by aggregating features from its neighborhoods [Kipf and Welling, 2017]. Graph Attention Networks (GAT) extend to learn additional importance weights so that important neighborhoods will be given higher attention [Veličković *et al.*, 2017]. GraphSAGE learns a set of aggregation functions for each node to flexibly aggregate information from neighborhoods of different hops. [Hamilton *et al.*, 2017]. To select the appropriate convolution filter for a specific graph learning task, some researchers propose the GNN architecture search which can identify the optimal one from a set of candidate models [Shi *et al.*, 2020].

Temporal Network Learning. Recently, some works seek to extend existing GNN models to handle temporal networks, where node sets and edge connections can evolve over time. STGCN employs a GCN layer to model the spatial dependency among network nodes at each time step, followed by a CNN layer to capture the temporal dependencies of networks at different time steps [Yu *et al.*, 2017]. DySAT designs a self-attention mechanism in two directions namely structural neighborhood and temporal dynamics to respectively capture node dependency relationships within and between networks [Sankar *et al.*, 2019]. EvolveGCN adapts GCN along the temporal dimension by using a recurrent neural network to evolve the GCN learning weights [Pareja *et al.*, 2020]. In comparison, in this paper we focus on evolving the graph attention networks with multiple attention heads. Different from existing works that typically enforce some regularization on the multiple dynamic networks to capture the temporal node relationships, we propose to first represent the temporal node variation patterns and then explicitly incorporate them into each time step for enhanced node relationships modeling and network embedding learning.

3 Problem Definition

In this work, we extend graph attention networks to study an evolutionary graph attention network learning problem. An evolving network can be represented as a sequence of graph snapshots $\mathbb{G} = \{G_1, \cdots, G_T\}$, where T is the number of time steps. Each snapshot at time step $t \in [1, T]$

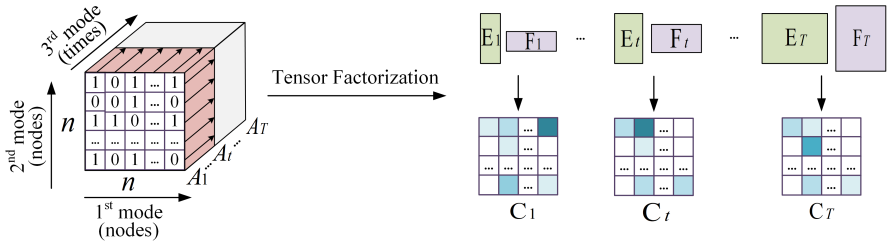


Figure 1: Calculation of pairwise node variation pattern similarities based on the PARAFAC model. For each graph G_t, a three-way tensor $\mathbf{S}_t$ is built by stacking all the adjacency matrices till time t, encoding the correlations of three modes, namely nodes, nodes and times.

is an undirected graph represented by $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{X}_t)$, where $\mathcal{V}_t = \{v_i\}_{i=1,\cdots,|\mathcal{V}_t|}$ is a set of unique nodes, $\mathcal{E}_t = \{e_{ij}\}_{i,j=1,\cdots,|\mathcal{V}_t|}$ is a set of edges which can be encoded in an adjacency matrix $\mathbf{A}_t \in \mathbb{R}^{|\mathcal{V}_t|\times|\mathcal{V}_t|}$, and $\mathbf{X}_t \in \mathbb{R}^{|\mathcal{V}_t|\times d}$ contains all $|\mathcal{V}_t|$ nodes and their d-dimensional feature vectors.

The objective of this paper is to map graph $G_t \in \mathbb{G}$ to a low-dimensional embedding space $\mathrm{O}_t \in \mathbb{R}^{|\mathcal{V}_t|\times h}$ while taking all its historical graphs $G_1, G_2, \cdots, G_{t-1}$ into consideration. It is worth noting that node sets of underlying graphs may be different. In this case, we are able to uniform node sets of these graphs by aligning them to the largest graph with maximum number of nodes n. Therefore, in the following we assume that adjacency matrices for graphs at different time steps have the same size $n \times n$. In this work, we focus on evolving the graph attention networks to capture the dependency relationships across all input graphs and meanwhile explicitly preserve node variation or changing patterns derived from the temporal graph evolution sequence.

4 The Proposed Method

The proposed approach includes two major parts. First, we propose to extract temporal node variation pattern similarities from the respective graph topology evolution sequence based on the non-negative tensor factorization. Then, node variation pattern similarities are explicitly incorporated into the graph evolving attention network learning, where a neural model, GAEN shown in Fig.2, is proposed to evolve graph attention network weights across all temporal graphs. Details of the above processes are illustrated in the following sections.

4.1 Node Variation Pattern Similarity Calculation

Nodes often demonstrate implicit variation patterns as graph evolves over time [Fournier-Viger *et al.*, 2020]. We are interested in the variation of graph structures, where graph nodes in the dynamic process of creating and removing connections with neighborhoods could exhibit similar structure changing patterns. For a given graph G_t at time $t \in [1, T]$, its current topology structure together with all past topology structures can be organized in a three-way array or tensor by $\mathbf{S}_t = [\mathbf{A}_0, \mathbf{A}_1, \cdots \mathbf{A}_t] \in \mathbb{R}^{n\times n\times t}$, where the structural dependencies between nodes are encoded in the frontal slice $\mathbf{S}_t[::t] \in \mathbb{R}^{n\times n}$ and the structure variations of each node v_i are encoded in the respective horizontal slice $\mathbf{S}_t[i::] \in \mathbb{R}^{n\times t}$ along the time axis. Then, as shown in Fig.1 we aim to calculate the pairwise node variation pattern similarities by modeling above structure dependencies and variations simultaneously with tensor factorization (TF) [Kolda and Bader, 2009].

TF has been widely utilized to model high-order variable correlations and further discover temporal variation patterns of objects [Takeuchi *et al.*, 2017]. In this paper, we consider non-negative TF (e.g., both input and output are non-negative) based on the PARAFAC model given that its factorization results are deterministic [Kolda and Bader, 2009].

Definition 1 (rank-one): *A tensor $\mathbf{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ of order N has rank-one if it can be written as the outer product (notation $\circ$) of N vectors, i.e., $\mathbf{T} = \boldsymbol{b}^{(1)} \circ \boldsymbol{b}^{(2)} \circ \cdots \circ \boldsymbol{b}^{(N)}$ or $\mathbf{T}_{i_1,i_2,\cdots,i_N} = \boldsymbol{b}^{(1)}_{i_1}\boldsymbol{b}^{(2)}_{i_2}\cdots\boldsymbol{b}^{(N)}_{i_N}$ for all $1 \le i_n \le I_N$.*

Formally, TF aims to decompose tensor $\mathbf{S}_t \in \mathbb{R}^{n\times n\times t}$ ($t \in [1, T]$) into a sum of component rank-one tensors as:

$$\mathbf{S}_t = \sum_{r=1}^{R} \mathbf{e}_r \circ \mathbf{q}_r \circ \mathbf{f}_r = [[\mathbf{E}_t, \mathbf{Q}_t, \mathbf{F}_t]], \ s.t.\ \mathbf{e}_r, \mathbf{q}_r, \mathbf{f}_r \ge 0 \quad (1)$$

where R is a parameter denoting the number of latent factors. $\mathbf{E}_t = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_R] \in \mathbb{R}^{n\times R}$, $\mathbf{Q}_t = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_R] \in \mathbb{R}^{n\times R}$ and $\mathbf{F}_t = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_R] \in \mathbb{R}^{t\times R}$ are latent factor matrices, *i.e.*, combination of rank-one components, where $\mathbf{E}_t$ and $\mathbf{Q}_t$ are equivalent since $\mathbf{S}_t$ is symmetric for undirected graphs. $\mathbf{E}_t$ or $\mathbf{Q}_t$ can be interpreted as that nodes along the 1^{st} or 2^{nd} mode (shown in Fig.1) are conceptually mapped in a shared space with R latent factors, while $\mathbf{F}_t$ reveals the variation patterns of these latent factors along the time axis (3^{rd} mode). Then, we further derive the structural variation patterns $\mathbf{H}^1_t$ (1^{st} mode) and $\mathbf{H}^2_t$ (2^{nd} mode) of all nodes as:

$$\mathbf{H}^1_t = \mathbf{E}_t\mathbf{F}_t^T, \mathbf{H}^2_t = \mathbf{Q}_t\mathbf{F}_t^T = \mathbf{E}_t\mathbf{F}_t^T \quad (2)$$

where T denotes transpose. Finally, the pairwise node variation pattern similarities $\mathbf{C}_t \in \mathbb{R}^{n\times n}$ are computed by:

$$\mathbf{C}_t = \mathrm{softmax}(\mathbf{H}^1_t\mathbf{H}^{2^T}_t) \quad (3)$$

4.2 Temporal Graph Attention Network Learning

In this part, we incorporate the node variation pattern similarities for structure-based graph attention network embedding learning, where the inputs are node features $\mathbf{X}_t$ and node variation pattern similarities $\mathbf{C}_t$ for each single graph G_t. Assume $\mathcal{N}_{v_i} = \{v_j \in \mathcal{V}_t : e_{i,j} \in \mathcal{E}_t\}$ denotes the set of immediate neighborhoods of every node v_i in graph G_t, the output embedding vector $\mathbf{h}_{v_i} \in \mathbb{R}^h$ for v_i is computed as:

$$\mathbf{h}_{v_i} = \mathrm{ReLU}(\sum_{v_j\in\mathcal{N}_{v_i}} \alpha_{ij}\mathbf{C}_{t,ij}\mathbf{W}_t\mathbf{X}_{t,j}) \quad (4)$$

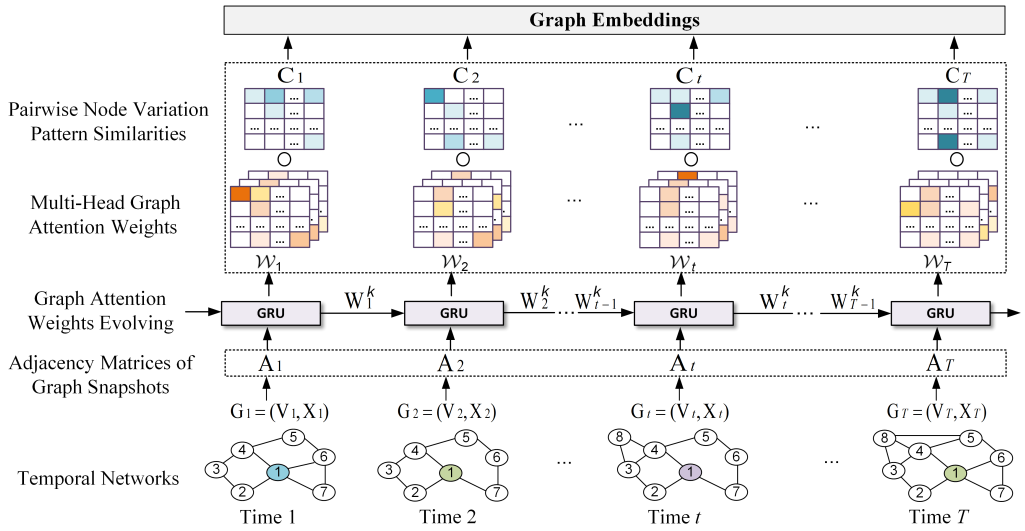


Figure 2: The proposed GAEN model for evolving graph attention networks and embedding learning.

where $\mathbf{X}_{t,j} \in \mathbb{R}^d$ is the input features of node v_j, $\mathbf{W}_t \in \mathbb{R}^{h \times d}$ is a shared weight transformation applied to each node in graph G_t, $\mathbf{C}_{t,ij}$ is the variation pattern similarity (e.g., till time step t) between v_i and v_j, and $\mathrm{ReLU}(\cdot)$ is an activation function, *i.e.*, $\mathrm{ReLU}(x) = max(0, x)$. α_{ij} indicates the attention weight or importance of node v_j's features to node v_i, which is computed with a single-layer Feedforward Neural Network (FNN) parameterized by $\mathbf{a} \in \mathbb{R}^{2h}$:

$$\alpha_{ij} = \frac{\exp(\mathrm{ReLU}(\mathbf{a}^T[\mathbf{W}_t\mathbf{X}_{t,i}||\mathbf{W}_t\mathbf{X}_{t,j}]))}{\sum_{v_j \in \mathcal{N}_{v_i}} \exp(\mathrm{ReLU}(\mathbf{a}^T[\mathbf{W}_t\mathbf{X}_{t,i}||\mathbf{W}_t\mathbf{X}_{t,j}]))} \quad (5)$$

where $||$ is the concatenation operation. Note that α_{ij} has been normalized by a softmax over all neighborhoods of v_i which makes $\alpha_{ij} \neq \alpha_{ji}$. Similar to the GAT [Veličković *et al.*, 2017], multi-head attentions can be computed and averaged to stabilize the node embedding learning by:

$$\mathbf{h}_{v_i} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{ReLU}\Big(\sum_{v_j \in \mathcal{N}_{v_i}} \alpha_{ij}\mathbf{C}_{t,ij}\mathbf{W}_t^k\mathbf{X}_{t,j}\Big) \quad (6)$$

where K is the number of attention heads and $\mathbf{W}_t^k \in \mathbb{R}^{h \times d}$ is a learnable weight parameter associated with the k^{th} attention head for all nodes in graph G_t.

Based on Eq. (6), we can obtain the embedding space $\mathrm{O}_t = \{\mathbf{h}_i\}_{i=1,\cdots,n} \in \mathbb{R}^{n \times h}$ for each graph $G_t \in \mathbb{G}$ with preserved static graph structures (*e.g.*, multi-head graph attentions) and temporal node structure variation patterns. Note that the FNN weight parameter $\mathbf{a}$ is shared across all graph snapshots, and the trainable weights associated with each single graph snapshot G_t are $\mathcal{W}_t = \{\mathbf{W}_t^k\}_{k=1,\cdots,K}$ which solely lead to varying attention values in Eq. (5) and node embeddings in Eq. (6) for graph snapshots at different time steps. Instead of learning $\mathcal{W}_t$ for each graph snapshot from scratch, we propose to evolve the weight learning by taking the structure discrepancies between adjacent graphs along the time axis into consideration.

Algorithm 1: Training the GAEN model

Input : Graph sequence $\mathbb{G} = \{G_1, \cdots, G_T\}$ and node variation patterns $\mathbb{C} = \{\mathbf{C}_1, \cdots, \mathbf{C}_T\}$
Output: Embeddings of each graph G_t : $\mathrm{O}_t \in \mathbb{R}^{n \times h}$

Initialize the number of attention heads K, multi-head attention weights $\mathcal{W}_1$ and training epoch I
for $j \in [1, I]$ **do**
  $\mathrm{O}_1 \leftarrow$ compute node embeddings based on Eq. 6
  for $t \in [2, T]$ **do**
    for $k \in [1, K]$ **do**
      $\mathbf{Z}_t^k \leftarrow \mathrm{sigmoid}(\mathbf{P}_Z^k\mathbf{A}_t + \mathbf{U}_Z^k\mathbf{W}_{t-1}^k + \mathbf{B}_Z^k)$
      $\mathbf{R}_t^k \leftarrow \mathrm{sigmoid}(\mathbf{P}_R^k\mathbf{A}_t + \mathbf{U}_R^k\mathbf{W}_{t-1}^k + \mathbf{B}_R^k)$
      $\widetilde{\mathbf{W}}_t^k \leftarrow \tanh(\mathbf{P}_W^k\mathbf{A}_t + \mathbf{U}_W^k(\mathbf{R}_t^k \circ \mathbf{W}_{t-1}^k) + \mathbf{B}_W^k)$
      $\mathbf{W}_t^k \leftarrow (1 - \mathbf{Z}_t^k) \circ \mathbf{W}_{t-1}^k + \mathbf{Z}_t^k \circ \widetilde{\mathbf{W}}_t^k$
      $\mathcal{W}_t \leftarrow \mathbf{W}_t^k$
    end
    $\mathrm{O}_i \leftarrow$ compute node embeddings based on Eq. 6
  end
  Minimize the binary cross-entropy loss $\mathcal{L}$ in Eq. 8
end

4.3 Multi-Head Graph Attention Weight Evolution

As shown in Fig. 2, we use an independent Gated Recurrent Unit (GRU) to evolve the network weight $\mathbf{W}_t^k \in \mathcal{W}_t$ for each graph attention head. The input is the adjacency matrix $\mathbf{A}_t$ for every graph snapshot G_t and the hidden state is $\mathbf{W}_t^k$ which evolves between adjacent graph snapshots as:

$$\mathbf{W}_t^k = \mathrm{GRU}(\mathbf{A}_t, \mathbf{W}_{t-1}^k) \quad (7)$$

The main idea for the design is to learn the structure difference of the current graph G_t against its previous graph G_{t-1}, thereby updating the attention network weights for G_t.

4.4 Algorithm Training and Explanation

To train the graph attention weight evolving and embedding learning modules, nodes are forced to have similar embed-

ding with their neighborhoods for every single graph. Similar to [Sankar *et al.*, 2019], the set of neighborhoods $\mathcal{N}_{v_i}^t$ for each node v_i in the graph at time t are obtained by performing fixed-length random walks on G_t, *i.e.*, nodes within the same random walk are neighborhoods. Finally, we minimize the similarities between nodes with their neighbors (positive samples) through minimizing a binary cross-entropy loss:

$$\mathcal{L} = \sum_{t=1}^{T}\sum_{i=1}^{n}\Big(\sum_{v_j \in \mathcal{N}_{v_i}^t} -\log(\text{sigmoid}(\langle \mathbf{h}_{v_i}, \mathbf{h}_{v_j}\rangle)) \\ -\beta \sum_{v_j^{'} \in \mathcal{P}_{v_i}^t} \log(1-\text{sigmoid}(\langle \mathbf{h}_{v_i}, \mathbf{h}_{v_j^{'}}\rangle))\Big) \tag{8}$$

where $\langle\cdot\rangle$ represents the inner product, $\mathcal{P}_{v_i}^t$ is a set of negative samples that are not neighbored with node v_i, and β is set to balance the positive and negative samples.

The detailed training procedure of GAEN is summarized in Algorithm 1. The weight evolving process based on GRU (Eq. 7) is completed through lines 6 and 9, where $\mathbf{P}_{\cdot}^k \in \mathbb{R}^{h\times n}$, $\mathbf{U}_{\cdot}^k \in \mathbb{R}^{h\times h}$ and $\mathbf{B}_{\cdot}^k \in \mathbb{R}^{h\times d}$ are learnable GRU weights for the k^{th} graph attention head. The above process requires that the column dimension of the GRU input ($\mathbf{A}_t \in \mathbb{R}^{n\times n}$) matches the number of node features d. For graphs without content features, graph features can be viewed as a diagonal matrix where d equals to n. In other situations where $d \neq n$, we introduce an additional learnable weight matrix $\mathbf{P}_A^k \in \mathbb{R}^{n\times d}$ to transform the input as:

$$\mathbf{A}_t^{'} = \mathbf{A}_t\mathbf{P}_A^k \tag{9}$$

5 Experiments

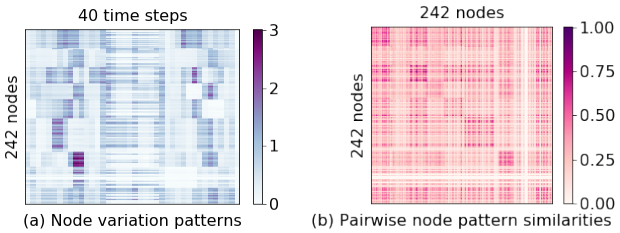
This section evaluates GANE using link prediction and node classification tasks based on the learned network embeddings.

Datasets. We adopt four temporal networks **Enron** [Klimt and Yang, 2004], **UIC** [Panzarasa *et al.*, 2009], **Primary School** [Stehlé *et al.*, 2011] and **DBLP**[1] summarized in Table 1. Each dataset is composed of a set of temporal graph snapshots at multiple time steps. Enron and UIC are communication networks, where in Enron the communication links are email interactions between core employees and in UIC the links represent messages sent between online social users. Primary School represents the face-to-face contact networks, where the network at each time step was constructed from the interactions among students and teachers with an interval of 20 consecutive seconds. There are 10 student classes and 1 teacher class. DBLP represents co-author networks between authors from 5 research areas. The features are extracted from the titles and abstracts of authors' publications.

Compared Methods. We compare against five strong baseline methods, including **GCN** [Kipf and Welling, 2017] and **GAT** [Veličković *et al.*, 2017] that perform embedding learning on static graph snapshots, **GAT-GRU** which uses GAT to learn different snapshots with capturing the network dependencies based on GRU, **EvolveGCN** [Pareja *et al.*, 2020] which uses GCN to learn each single snapshot and meanwhile evolves weights between GCN networks based on the GRU, and **DySAT** [Sankar *et al.*, 2019] which learns network embeddings by jointly employing self-attention layers along two dimensions: structural neighborhood and temporal dynamics. In addition, we develop a variant method **GAEN**$_{basic}$ without explicitly incorporating the pairwise node pattern similarities compared with **GAEN**.

# of	Enron	UIC	Primary School	DBLP
Nodes	143	1890	242	6606
Edges	2347	16822	20019	42815
Features	–	–	–	100
Time steps	10	13	40	10
Labels	–	–	11	5

Table 1: Statistics information of the temporal networks.



(a) Node variation patterns (b) Pairwise node pattern similarities

Figure 3: The node variation patterns on Primary School.

Tasks and Experimental Setup. We evaluate the link prediction performance on all four datasets using Accuracy (Acc) and Area Under the Curve (AUC) metrics. We train an external logistic regression classifier to predict the existence of links at time t based on the embedding features learned from previous networks up to time $t-1$. We compare the node classification accuracy (Acc) on two datasets *Primary School* and *DBLP* as they have label information. Similarly, we train a logistic regression classifier to classify nodes into different categories based on embedding features learned from previous networks up to time t. For link prediction, 20% of the links are used as validation to fine-tune the hyper-parameters, and the remaining are split as 25% and 75% for training and test. For node classification, 20% of nodes are used for validation. Then, 30% and 70% of the remaining nodes are respectively used for training. To train the model, the number of attention heads is set as 8, the hidden dimension in GRU networks is set as 128 and the learning rate is set as 1e-4. For detailed parameter settings, please refer to the GitHub link[2].

5.1 Link Prediction

Table 2 reports the latest time step link prediction performance of different methods. We conclude the following two main observations. 1) GAEN achieves better Acc and AUC performance than other methods on all four datasets. This is mainly attributed to two reasons: First, GAEN is efficient to capture the inter-dependency relationships between nodes across networks at different time steps, which can be observed from that GAEN$_{basic}$ generally performs better than other similar models, including DySAT, EvolveGCN and

[1] https://dblp.uni-trier.de

[2] https://github.com/codeshareabc/GAEN

Datasets	Enron		UIC		Primary School		DBLP	
Metrics	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
GCN [Kipf and Welling, 2017]	0.7575	0.8077	0.7384	0.7893	0.8910	0.9395	0.7746	0.8025
GAT [Veličković *et al.*, 2017]	0.7417	0.7983	0.7690	0.7702	0.7927	0.8709	0.9125	0.9339
GAT-GRU	0.7016	0.7514	0.7329	0.7701	0.8865	0.9481	0.8872	0.9360
EvolveGCN [Pareja *et al.*, 2020]	0.7142	0.7759	0.7350	0.7891	0.8877	0.9478	0.8235	0.8678
DySAT [Sankar *et al.*, 2019]	0.7930	0.8635	0.7324	0.8215	0.9064	0.9487	0.9141	0.9502
GAEN$_{basic}$	0.8076	0.8950	0.7282	0.7901	0.9098	0.9532	0.9074	**0.9607**
GAEN	**0.8144**	**0.9077**	**0.7443**	**0.8409**	**0.9106**	**0.9642**	**0.9231**	0.9587

Table 2: Link prediction results on Enron, UIC, Primary School, and DBLP.

Datasets	Primary School		DBLP	
Train ratios	30%	70%	30%	70%
GCN	0.4705	0.5038	0.5662	0.5671
GAT	0.4823	0.5115	0.5331	0.5430
GAT-GRU	0.4705	0.5038	0.5883	0.6103
EvolveGCN	0.6073	0.6654	0.6564	0.6657
DySAT	0.6311	0.6591	0.6369	0.6702
GAEN$_{basic}$	0.5900	0.6423	0.6676	0.6788
GAEN	**0.6464**	**0.6731**	**0.6784**	**0.6918**

Table 3: Node classification results on Primary School and DBLP.

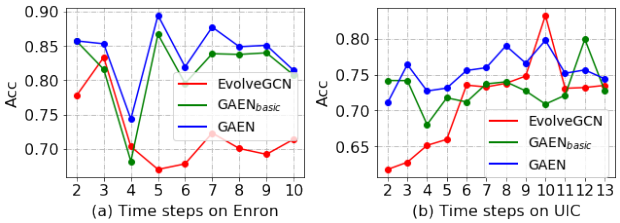


Figure 4: Link prediction performance at various time steps.

GAT-GRU. Second, GAEN performs better than GAEN$_{basic}$ in most cases, which indicates that the pairwise node pattern similarities explicitly preserved in GAEN are helpful to enhance the node relation modeling, *i.e.*, Fig. 3 shows the visualization of node structure variation patterns and the respective node pattern similarities on primary school; 2) Compared with EvolveGCN that evolves the learning weights of GCN by considering the discrepancies of node features between adjacent graph snapshots, GAEN$_{basic}$ adopts GAT and evolves the GAT model weights by considering the discrepancies of node structures between adjacent graph snapshots. Table 2 shows that GAEN$_{basic}$ outperforms EvolveGCN, possibly because that graph snapshots are different in their topological structures and it is essential to learn their structural discrepancies while evolving the learning weights.

To further evaluate the three methods, EvolveGCN, GAEN$_{basic}$, and GAEN, specifically designed to evolve the learning weights, we compared their link prediction performance at various time steps shown in Fig. 4. We can observe that GAEN$_{basic}$ performs better on Enron and demonstrates higher robustness on UIC when compared with EvolveGCN. In addition, GAEN is generally superior to GAEN$_{basic}$, confirming the benefits of explicitly extracting and preserving the node pattern similarities in this paper.

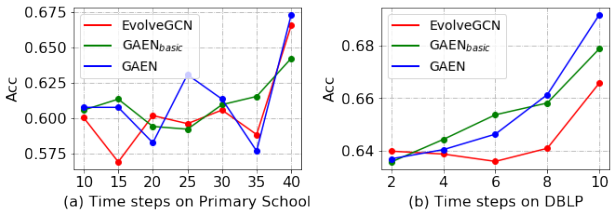


Figure 5: Node classification performance at various time steps.

5.2 Node Classification

Table 3 reports the node classification performance on the two labeled networks, where the node classification accuracy is much lower compared with the respective link prediction results in Table 2. This is because the compared methods are trained to force nodes to have similar embedding with their linked neighborhoods (*e.g.*, Eq. 8), therefore the training scheme would bias the link prediction task. We can observe that GAEN outperforms all baseline methods, which asserts the benefits of preserving temporal node pattern similarities in the model. Similarly, the comparison results in Fig.5 demonstrate the effectiveness and robustness of GAEN for evolving graph attention networks.

6 Conclusion

Networked systems often work in evolving environments with changing network topology over time. It is beneficial to capture underlying dynamics for network learning, especially when the nodes in the evolution process demonstrate strong relevance to the change of network structures. In this paper, we advanced graph neural networks to explicitly incorporate node variation pattern similarities for enhanced node embedding learning. We proposed a novel framework with evolving graph attention networks across different time points, where a GRU network is used for progressive multi-head attention weights learning by taking node structure differences between adjacent networks into consideration. Experiments and comparisons, using link prediction and node classification, demonstrated the effectiveness of the proposed approach against several baseline methods.

Acknowledgements

This research is sponsored by the US National Science Foundation through Grant Nos. IIS-2027339, IIS-1763452, CNS-1828181, and OAC-2017597.

References

[Chen *et al.*, 2019] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

[Fournier-Viger *et al.*, 2020] Philippe Fournier-Viger, Ganghuan He, Chao Cheng, Jiaxuan Li, Min Zhou, Jerry Chun-Wei Lin, and Unil Yun. A survey of pattern mining in dynamic graphs. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1372, 2020.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[Hegele *et al.*, 2012] Anna Hegele, Atanas Kamburov, Arndt Grossmann, Chrysovalantis Sourlis, Sylvia Wowro, Mareike Weimann, Cindy L Will, Vlad Pena, Reinhard Lührmann, and Ulrich Stelzl. Dynamic protein-protein interaction wiring of the human spliceosome. *Molecular cell*, 45(4):567–580, 2012.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[Klimt and Yang, 2004] Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *In CEAS 2004 - First Conference on Email and Anti-Spam*, 2004.

[Kolda and Bader, 2009] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[Manessi *et al.*, 2020] Franco Manessi, Alessandro Rozza, Mario Manzo, and and. Dynamic graph convolutional networks. *Pattern Recognition*, 97:107000, 2020.

[Panzarasa *et al.*, 2009] Pietro Panzarasa, Tore Opsahl, and Kathleen M Carley. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *Journal of the American Society for Information Science and Technology*, 60(5):911–932, 2009.

[Pareja *et al.*, 2020] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B Schardl, and Charles E Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *AAAI*, pages 5363–5370, 2020.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, Steven Skiena, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proc. of the ACM SIGKDD Intl. Conf.*, pages 701–710, 2014.

[Sankar *et al.*, 2019] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dynamic graph representation learning via self-attention networks. *arXiv preprint arXiv:1812.09430*, 2019.

[Seo *et al.*, 2018] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

[Shi *et al.*, 2020] Min Shi, David A Wilson, Xingquan Zhu, Yu Huang, Yuan Zhuang, Jianxun Liu, and Yufei Tang. Evolutionary architecture search for graph neural networks. *arXiv preprint arXiv:2009.10199*, 2020.

[Stehlé *et al.*, 2011] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8):e23176, 2011.

[Takeuchi *et al.*, 2017] Koh Takeuchi, Yoshinobu Kawahara, and Tomoharu Iwata. Structurally regularized non-negative tensor factorization for spatio-temporal pattern discoveries. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 582–598. Springer, 2017.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[Yao *et al.*, 2019] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proc. of the 33rd AAAI Conf. on Artificial Intelligence*, pages 7370–7377, 2019.

[Yu *et al.*, 2017] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[Zhang *et al.*, 2020] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6:3–28, 2020.

[Zhao *et al.*, 2019] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2019.