When Convolutional Network Meets Temporal Heterogeneous Graphs: An Effective Community Detection Method

Yaping Zheng, Xiaofeng Zhang, *Member, IEEE*, Shiyi Chen, Xinni Zhang, Xiaofei Yang, *Member, IEEE*, and Di Wang, *Member, IEEE*

Abstract—Community detection has long been an important yet challenging task to analyze complex networks with a focus on detecting topological structures of graph data. Essentially, real-world graph data is generally heterogeneous which dynamically varies over time, and this invalidates most existing community detection approaches. To cope with these issues, this paper proposes the temporal-heterogeneous graph convolutional networks (THGCN) to detect communities using the learnt feature representations of a set of temporal heterogeneous graphs. Particularly, we first design a heterogeneous GCN component to represent features of heterogeneous graph at each time step. Then, a residual compressed aggregation component is proposed to learn temporal feature representations extracted from two consecutive heterogeneous graphs. These temporal features are considered to contain evolutionary patterns of underlying communities. To the best of our knowledge, this is the first attempt to detect communities from temporal heterogeneous graphs. To evaluate the model performance, extensive experiments are performed on two real-world datasets, i.e., DBLP and IMDB. The promising results have demonstrated that the proposed THGCN is superior to both benchmark and the state-of-the-art approaches, e.g., GCN, GAT, GNN, LGNN, HAN and STAR, with respect to a number of evaluation criteria.

Index Terms—Graph convolutional network, heterogeneous graph, temporal graph, community detection

1 Introduction

Community detection has long been an important yet challenging task to analyze complex networks with a focus on detecting topological structures of homogeneous graphs with flourishing results [1], [2]. However, the real-world graph data is generally heterogeneous which dynamically varies over time, and this poses a great challenge to most existing community detection approaches.

Recently, graph neural network (GNN) based approaches [3], [4], [5] have demonstrated superior ability to represent graph data for downstream tasks such as node classification [3], link prediction and community detection [6]. Among these community detection approaches, the line graph neural network (LGNN) [7] achieves the state-of-the-art model performance, although it is not proposed for heterogeneous graph data. In the proposed approach, LGNN casts the community detection problem to node-wise classification task by applying permutation equivalence rule. To avoid ambiguity, our community detection task from heterogeneous graph data is also to classify graph nodes. To further address characteristics of temporal heterogeneous graph data, we consider following three kinds of evolutionary patterns of community members: (i) varying members; (ii) periodically appeared members; and (iii) emerging members.

We take DBLP data as an example to illustrate these patterns which are plotted in Fig. 1. In this figure, three consecutive momentary graphs are given and there are three communities, i.e., "Deep Learning", "Machine Learning" and "Security and Privacy". For *varying members*, it refers to the case that existing graph nodes change their communities from time to time. For instance, author a_7 leaves "Security and Privacy" and joins "Deep Learning" from time t_1 to time t_2. For *periodically appeared members*, it refers to those existing graph nodes whom periodically appear in a community, e.g., author a_1 appears in community "Deep Learning" at time t_1 and t_3, respectively. As for *emerging members*, it refers to those new graph nodes who appear in a community of current graph.

As aforementioned, community detection task in this paper is to classify each graph node and thus it is desired to learn effective node feature representations. Apparently, this effective feature representation should contain both "static" features and "dynamic" features. The "static" features are extracted from both *emerging nodes* as well as those "static" nodes, i.e., nodes have not changed their historical communities. The "dynamic" features are to extract evolutionary behaviors of both *varying members* and *periodically appeared members*. And these evolutionary features could be extracted from interactions between consecutive momentary graphs. To well address aforementioned issues, we thus propose this temporal-heterogeneous graph convolutional networks (THGCN) for community detection task. Particularly, we first calculate adjacency matrices of temporal heterogeneous graphs, and then embed both graph structural information and node features for each momentary graph respectively. For "dynamic" feature embedding, inspired by [8], we propose the Residual Compressed Aggregation Component (ResCAC) to utilize predefined meta-paths to sample correlated heterogeneous nodes across consecutive temporal graphs. This sampling step is analogous to edge conversion step in LGNN with the merit of preserving temporal community information. After that, this component interacts node features of different temporal graphs and then compresses these interactive feature representations into a low-dimensional feature space. For "static" feature embedding, a linear layer of residual connection [9] is introduced to aggregate features of current graph. At last, a specific community detection loss function is proposed. The major contributions of this paper can be summarized as follows:

- We propose the THGCN model. To the best of our knowledge, this is among the first attempts to perform community detection on a set of temporal heterogeneous graphs.
- We sample nodes from consecutive heterogeneous graphs via predefined meta-paths, and then design a residual compressed aggregation component to capture evolutionary behaviors of time-varying community members.
- We perform extensive experiments on two real-world datasets and the promising results have demonstrated that the proposed THGCN is superior to both baseline and the state-of-the-art approaches.

The rest of this paper is organized as follows. Section 2 reviews related work and then we formulate the problem in Section 3. The proposed THGCN approach is illustrated in Section 4. Experimental results are reported in Section 5 and we conclude the paper in Section 6.

2 Related Work

As is well known, graph neural networks have demonstrated superior ability to represent graph data for various analysis tasks [3],

- *Yaping Zheng, Xiaofeng Zhang, Shiyi Chen, and Xinni Zhang are with the School of Computer Science, Harbin Institute of Technology, Shenzhen 150001, China. E-mail: 18s051034@stu.hit.edu.cn, zhangxiaofeng@hit.edu.cn, {chenshiyi.hit, zhangxinni.hit}@gmail.com.*
- *Xiaofei Yang is with the University of Macau, Zhuhai 999078, China. E-mail: xiaofei.hitsz@gmail.com.*
- *Di Wang is with Nanyang Technological University, Singapore 639798, Singapore. E-mail: wangdi@ntu.edu.sg.*

Manuscript received 20 August 2020; revised 24 May 2021; accepted 24 June 2021. Date of publication 9 July 2021; date of current version 10 January 2023.
(Corresponding author: Xiaofeng Zhang.)
Recommended for acceptance by J. Tang.
Digital Object Identifier no. 10.1109/TKDE.2021.3096122

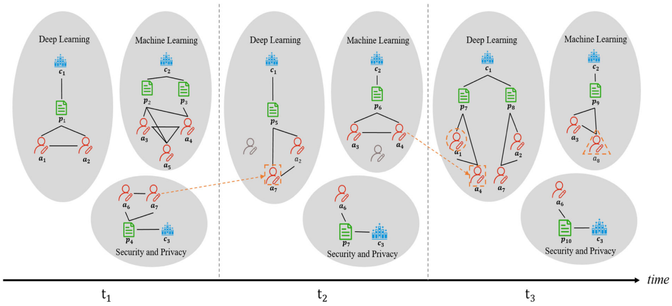


Fig. 1. An illustrating example of three kinds of evolutionary patterns of community members, i.e., (i) varying members; (ii) periodically appeared members; and (iii) emerging members.

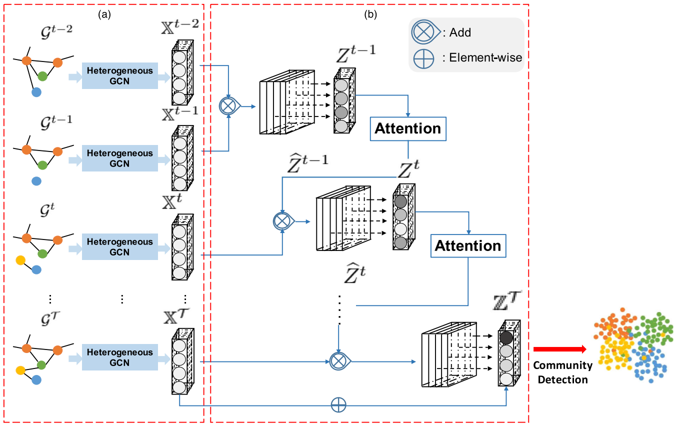


Fig. 2. The framework of the proposed THGCN. It consists of two components: (a) Heterogeneous GCN component respectively embeds each temporal heterogeneous graph; (b) ResCAC aggregates both static and dynamic features for community detection task.

[4], [10]. However, only a few research attempts have been made towards community detection task.

In the literature, a graph generative model [10], [11] is proposed which reconstructs graph structural information as well as graph node attributes in an unsupervised manner. Similarly, a graph clustering method is proposed in [12] which designs modularity-based loss function and employs the Bethe Hessian operator to enhance the representative ability of graph nodes to detect communities. Alternatively, several supervised approaches have been proposed. MRFasGCN [13], as a semi-supervised graph convolutional network model, is proposed to combine Markov random field (MRF) with the original GCN model to utilize node semantics for community detection problem. The Line Graph Neural Networks (LGNN) [7] designs a polynomial function to calculate adjacency matrices of a set of line graphs using the proposed non-backtracking operators to detect communities from homogeneous graph data. To model temporal graph data, several approaches have been proposed [14], [15] but with a focus on homogeneous graph data. To analyze temporal heterogeneous graph, the HDGAN [16] is proposed which learns the node embeddings by applying three-level of attention mechanism including structural-level, semantic-level and time-level attentions. The decent DyHNE [17] is proposed to first preserve the first- and second-order proximities of each momentary graph and then adopts a hyper parameter to balance the contribution of two consecutive graphs. Unfortunately, these pioneer attempts are made towards node classification task instead of community detection task. Therefore, it is desired to develop a novel approach which directly detects communities from a family of temporal heterogeneous graphs.

3 Preliminaries

In this section, we formulate the problem, i.e., community detection from temporal heterogeneous graph.

3.1 Preliminaries

Let $G = (V, E)$ denote an undirected graph, where $V = \{v_{ij}\}$ and $E \subset V \times V$ denote node set and edge set, respectively. Let T_V and T_E respectively denote the set of node types and edge types, if $|T_V| + |T_E| > 2$, G is a *heterogeneous graph*. The corresponding adjacency matrix is defined as follows.

Definition 1. Adjacency Matrix of Heterogeneous Graph. *An adjacency matrix is defined as $A_{HG} = \{A_{ij} | i \in [1, m], j \in [1, n]\}$, where A_{ij} is an element entry, $m = |T_V|$ is the number of node types and $n = |T_E|$ is the number of edge types.*

Note that A_{ij} denotes an adjacency matrix of the corresponding homogeneous graph if we fix node type i and edge type j. Accordingly, the degree matrix of heterogeneous graph G is defined as $D_{HG} = diag(D_{ii})$, where $D_{ii} = \sum_j A_{ij}$. As aforementioned, this paper utilizes predefined meta-paths to sample nodes across consecutive heterogeneous graphs. Generally, a meta-path δ is defined as $\delta = a_1 \xrightarrow{e_1} a_2 \xrightarrow{e_2} \ldots \xrightarrow{e_{n-1}} a_n$, where $a_1, a_2, \ldots, a_n$ and $e_1, e_2, \ldots, e_{n-1}$ represent heterogeneous nodes and edges, respectively.

3.2 Problem Formulation

Let $\mathbb{G} = (\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^{\mathcal{T}})$ denote a set of temporal heterogeneous graphs, where $\mathcal{T}$ is the number of time steps. Let $\mathcal{G}^t = (V^t, A^t_{HG}, X^t)$ denote the heterogeneous graph at time t, V^t denote the node set of $\mathcal{G}^t$, $X^t \in \mathbb{R}^{N^t \times D}$ represent feature matrix of V^t where N^t represents the number of nodes of $\mathcal{G}^t$ and D is the feature dimension of X^t, and $\mathcal{C}^{\mathcal{T}} = \{1, .., C\}$ denotes a set of community labels at time step $\mathcal{T}$. Similar to LGNN [7], the community detection task in this paper is to label nodes in $\mathcal{G}^{\mathcal{T}}$ using the learnt features of a set of temporal heterogeneous graphs $\mathbb{G}$ by minimizing below loss function, written as

$$Loss = \mathcal{L}(\mathcal{C}^{\mathcal{T}} = \{1, .., C\} | \mathbb{G}). \tag{1}$$

4 The Proposed THGCN Approach

As aforementioned, the proposed THGCN is to detect communities from a set of temporal heterogeneous graphs. In the proposed approach, we first embed node features as well as their graph structural information, globally calculated on heterogeneous graph $\mathcal{G}^t$, into low-dimensional feature space. Then, a neural network component called Residual Compressed Aggregation Component (ResCAC) is proposed to aggregate the embedded features of consecutive temporal graphs. After acquiring these evolutionary features, we aggregate them with "static" features of $\mathcal{G}^{\mathcal{T}}$ to detect communities contained in $\mathcal{G}^{\mathcal{T}}$, and the corresponding framework of the proposed THGCN is depicted in Fig. 2.

4.1 Heterogeneous GCN Component

To embed both graph structural information and node features, we propose this heterogeneous GCN component which convolutes node features with the features of all its one-hop neighbors. According to [3], our convolutional operation is defined as

$$X_{i+1} = \sigma[\widehat{D}_{HG}^{-\frac{1}{2}}(A_{HG} + I)\widehat{D}_{HG}^{-\frac{1}{2}} X_i W_i], \tag{2}$$

where $\widehat{D}_{HG} = \sum_j (A_{ij} + I_{ij})$, W_i is the weight matrix, X_i is the feature matrix of the i^{th} layer, σ is the ReLU activation function, A_{HG} denotes the heterogeneous adjacency matrix which is generally used to represent relationship of heterogeneous nodes. The output of this component is denoted as $\mathbb{X} \in \mathbb{R}^{N \times d}$, and ResCAC takes the generated $\mathbb{X}$ as its input.

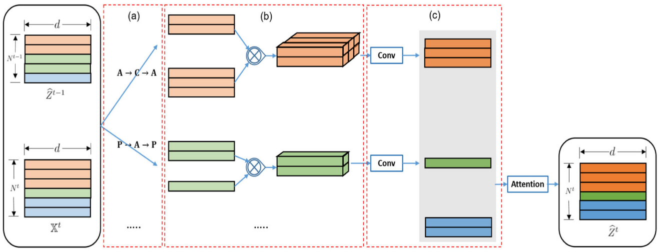


Fig. 3. Details of the proposed ResCAC component. (a) First, we separately sample two feature representation matrices $\mathcal{Z}_\delta^{t-1}$ and $\mathcal{X}_\delta^t$ using the same meta-path ($P \rightarrow A \rightarrow P$, etc.). (b) Second, we acquire a 3D tensor H_θ^t by Hadamard product and concatenate operation. (c) Third, a one-dimensional convolution is applied to compress the tensor. Finally, an attention component is applied on Z^t to differentiate weights of each row (i.e., node) in the feature matrix.

4.2 Residual Compressed Aggregation Component

The proposed ResCAC consists of three operations, i.e., meta-path based sampling operation, Hadamard product operation and compression operation, and this component is performed in order.

First, we utilize meta-paths to sample nodes from consecutive temporal graphs. For instance, according to the meta-path δ: $P \rightarrow A \rightarrow P$, we first sample those nodes (type of 'A') co-occurred in $\mathcal{G}^{t-1}$ and $\mathcal{G}^t$. Then, we separately sample nodes (type of 'P') to build feature matrices, i.e., $\mathcal{X}_\delta^t \subset \mathbb{X}^t$ (for P in $A \rightarrow P$ in $\mathcal{G}^t$) and $\mathcal{Z}_\delta^{t-1} \subset \widehat{Z}^{t-1}$ (for P in $P \rightarrow A$ in $\mathcal{G}^{t-1}$) and the dimensions of $\mathcal{X}_\delta^t$ and $\mathcal{Z}_\delta^{t-1}$ are respectively $N_P^t \times d$ and $N_P^{t-1} \times d$, as illustrated in Fig. 3a, where $\widehat{Z}^{t-1} \subset \mathbb{R}^{N^{t-1} \times d}$ denote feature representations of nodes generated by ResCAC at time t-1.

Second, we perform Hadamard product operation on two sub feature matrices $\mathcal{X}_\delta^t$ and $\mathcal{Z}_\delta^{t-1}$ and then concatenate them to generate a 3D tensor, which preserves interactive information of "varying" community members. The feature matrix after Hadamard product and concatenation is given as

$$\mathbb{H}_\delta^t = \mathop{\Big\|}_{i=1}^{N_\delta^{t-1}} \mathop{\Big\|}_{j=1}^{N_\delta^t} \mathcal{Z}_\delta^{t-1}(i,*) \circ \mathcal{X}_\delta^t(j,*), \tag{3}$$

where $2 \leq t \leq \mathcal{T}$, $\mathbb{H}_\delta^t \in \mathbb{R}^{N_\delta^{t-1} \times N_\delta^t \times d}$ denotes a 3D tensor, $\|$ is the concatenation operator and $\circ$ denotes the Hadamart product. Note that each 3D tensor is generated for each meta-path δ to preserve the temporal information between $\mathcal{G}^{t-1}$ and $\mathcal{G}^{t-1}$. So, the number of $\mathbb{H}_{\delta_k}^t$ is the same as the number of predefined meta-paths denoted as K.

Last, a compression operation is proposed to perform on $\mathbb{H}_{\delta_k}^t$, as illustrated in right-hand side grey rectangle in Fig. 3c. To this end, the one-dimensional convolutional operation is performed on each slice of $\mathbb{H}_{\delta_k}^t$ along the direction of feature dimension. We denote the slice j of $\mathbb{H}_{\delta_k}^t$ as $h_{\delta_{k,j}}^t \in \mathbb{R}^{N_{\delta_k}^{t-1} \times d}$, and thus the 3D tensor is compressed into a 2D tensor Z_δ^t. Then, we add static feature matrix $\mathbb{X}$ to the temporal feature Z^t. The above process can be formulated as

$$Z^t = \mathbb{X}^t + \mathop{\Big\|}_{k=1}^{K} \mathop{\Big\|}_{j=1}^{N_{\delta_k}^t} \sigma(h_{\delta_{k,j}}^t * \widetilde{W}_{k,j}^t), \tag{4}$$

where $W_{k,j}^t \in \mathbb{R}^{1 \times N_{\delta_k}^{t-1}}$ is the parameter matrix and σ is sigmoid activation function. To differentiate the weight of different rows in the feature representation matrices, an attention component is naturally applied on Z^t, and the corresponding coefficient matrix is computed as

$$\widehat{A}^t = softmax(\widehat{W} tanh(\widehat{V} Z^t)), \tag{5}$$

where $\widehat{W} \in \mathbb{R}^{d_a}$ and $\widehat{V} \in \mathbb{R}^{d_a \times d}$ are parameter matrices of attention component. Consequently, the weighted output is now re-calculated as

$$\widehat{Z}^{t-1} = \widehat{A}^t Z^{t-1} \in \mathbb{R}^{N^t \times d}. \tag{6}$$

To capture static features of current graph, a natural choice is to adopt the ResNet [9] structure. Thus, the final feature representation $\mathbb{Z}^\mathcal{T}$ after the ResNet component is written as

$$\mathbb{Z}^\mathcal{T} = \widehat{Z}^\mathcal{T} + \mathbb{X}^\mathcal{T}. \tag{7}$$

Algorithm 1. The Algorithm of the Proposed THGCN

Input: A set of temporal heterogeneous graphs
$\mathbb{G} = (\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^\mathcal{T})$, Node feature $\{X^t, t \in [1, \mathcal{T}]\}$,
Heterogeneous adjacency matrix $\{A_{HG}^t, t \in [1, \mathcal{T}]\}$,
Meta-path $\{\delta_k, k \in [1, K]\}$.
Parameter: Weight matrix $\{W_0, W_1, \widetilde{W}, \widehat{W}\}$,
The number of nodes N^t at time step t.
Output: The feature representations $\mathbb{Z}^t$.

Heterogeneous GCN embedding component:
for $X^t \in \{X^1, X^2, \ldots, X^\mathcal{T}\}$ **do**
  $\widehat{A}_{HG}^t \leftarrow \widehat{D}_{HG}^{-\frac{1}{2}}(A_{HG}^t + I)\widehat{D}_{HG}^{-\frac{1}{2}}$;
  $\mathbb{X}^t \leftarrow \widehat{A}_{HG}^t[ReLU(\widehat{A}_{HG}^t X^t W_0)]W_1$;
end for
Residual Compressed Aggregation component:
for $\mathbb{X}^t \in \{\mathbb{X}^2, \ldots, \mathbb{X}^\mathcal{T}\}$ **do**
  for $\delta_k \in \{\delta_1, \delta_2, \ldots, \delta_K\}$ **do**
    Sample feature representation matrix $\mathcal{Z}_{\delta_k}^{t-1}, \mathcal{X}_{\delta_k}^t$;
    $\mathbb{H}_{\delta_k}^t \leftarrow \|_{i=1}^{N_{\delta_k}^{t-1}} \|_{j=1}^{N_{\delta_k}^t} \mathcal{Z}_{\delta_k}^{t-1}(i,*) \circ \mathcal{X}_{\delta_k}^t(j,*)$;
    $Z^t \leftarrow \mathbb{X}^t + \|_{k=1}^K \|_{j=1}^{N_{\delta_k}^t} \sigma(h_{\delta_{k,j}}^t * \widetilde{W}_{k,j}^t)$;
  end for
  for $i \in N^t$ **do**
    Calculate the weight coefficient $\widehat{a}_i^t$ in attention module;
  end for
  $\widehat{A}^t \leftarrow \|_{i=1}^{N^t} \widehat{a}_i^t$; $\widehat{Z}^t \leftarrow \widehat{A}^t Z^t$;
end for
return $\mathbb{Z}^\mathcal{T} \leftarrow \widehat{Z}^\mathcal{T} + \mathbb{X}^\mathcal{T}$;

4.3 Community Detection Component

In this subsection, we define a loss function for community detection task. To simplify the problem, we assume the underlying communities are non-overlapping. According to [7], node labels should satisfy the permutation equivalence rule, and thus the loss function should be defined according to the permutations of node labels. Particularly, a softmax $(\cdot)$ function is applied on the output feature representations $\mathbb{Z}^\mathcal{T}$. Let $\hat{c}_i$ denote the predicted community label of node i, c_i denote the ground truth community label and $\mathcal{S}_\mathcal{C}$ denote the permutations, the loss function of the proposed model is defined as

$$\mathcal{L} = \mathop{\mathrm{INF}}_{\pi \in \mathcal{S}_\mathcal{C}} - \sum_{i \in V} \pi(c_i) \log(\hat{c}_i), \tag{8}$$

where π is a set of permutations of community labels in $\mathcal{S}_\mathcal{C}$. Assume that $\Phi : X \rightarrow \mathcal{C}$ is the true mapping function from X to the ground truth labels $\mathcal{C}$. And $\Omega : X \rightarrow \hat{\mathcal{C}}$ is our mapping function from X to the predicted labels $\hat{\mathcal{C}}$. By substituting these functions into Eq. (8), we have

$$\mathcal{L} = \mathop{\mathrm{INF}}_{\pi \in \mathcal{S}_\mathcal{C}} - \sum_{i \in V} \Phi(\pi(x_i)) \log\left[\Omega(\pi(x_i))\right]. \tag{9}$$

To optimize this equation is equivalent to optimize the second term. As the number of communities in a real-world graph is relatively large, it is desired to cluster label set $\mathcal{C}$ into $\hat{\mathcal{C}}$ sub groups to further reduce computational cost.

TABLE 1
The Statistics of DBLP and IMDB Datasets

Dataset	Nodes	Edges	Features	Communities
DBLP	20,919	117,074	174	3
IMDB	10,114	55,924	1,213	5

TABLE 2
The Relations and Meta-Paths Defined in Each Dataset

Dataset	Relations	Meta-paths
DBLP	paper-author	$P \rightarrow A \rightarrow P$
	paper-conference	$P \rightarrow C \rightarrow P$
	author-conference	$A \rightarrow C \rightarrow A$
IMDB	movie-actor	$M \rightarrow A \rightarrow M$
	movie-director	$M \rightarrow D \rightarrow M$
	actor-director	$A \rightarrow D \rightarrow A$
	director-actor	$D \rightarrow A \rightarrow D$

4.4 Complexity Analysis

The calculation of heterogeneous GCN component is similar to [3], and thus its time complexity is $O(|E|Cd)$, where $|E|$ is the number of edges contained in all temporal graphs and $C \times d$ denotes the dimension of weight matrix W. Let $\bar{\Phi}_1$ and $\bar{\Phi}_2$ denote the size of two sub feature matrices sampled from two consecutive temporal graphs (refer to Section 4.2). The corresponding Hadamard product and convolutional compression operation could be executed in parallel on each meta-path. Accordingly, the time complexity of ResCAC is estimated as $O(\mathcal{T}\bar{\Phi}_1\bar{\Phi}_2 d + (\mathcal{T}-1)\bar{N}F_1F_2)$, where $F_1 \times F_2$ denotes the size of attention coefficient matrix, and $\bar{N}$ is the average number of nodes contained in all temporal graphs. As $\bar{\Phi}_1\bar{\Phi}_2 < \bar{N}$, the overall time complexity of the proposed THGCN could be estimated as $O(|E|Cd + |V|d + |V|F_1F_2)$, which is linear to the number of nodes and edges, and thus we can conclude that the proposed approach is a scalable one.

5 Experiments

In this section, we first introduce the experimental datasets, baseline models and experimental settings. Then, extensive experiments are performed on these datasets to answer following research questions.

- RQ1: Whether the proposed THGCN outperforms state-of-the-art graph neural network based community detection approaches or not?
- RQ2: How the label rate of training data affect the model performance?
- RQ3: Whether the community detection results are good or not?

5.1 Datasets and Evaluation Criteria

A number of widely adopted evaluation criteria for community detection task are chosen to evaluate the model performance including *Accuracy*, *NMI*, *Modularity*, *ARI*, *Macro-F1* and *Micro-F1*. Two real-world heterogeneous datasets, i.e., DBLP[1] and IMDB[2] dataset are adopted in the experiments.

- *DBLP* is a monthly updated citation network dataset and each year is treated as a time step. To construct the heteregeneous graph, we extract three types of nodes, i.e., paper (P), author(A) and conference(C) and choose three meta-paths.
- *IMDB*is one of the most widely adopted datasets for heterogeneous graph analysis task. It consists of three types of data, i.e., “director”, “actors” and “Movie Release Date”. Accordingly, three types of nodes are extracted, i.e., movie (M), director(D) and actor(A), and four meta-paths.

The statistics of these datasets are reported in Tables 1 and 2, respectively.

5.2 Baseline Methods

For performance comparison, both baseline methods as well as the state-of-the-art approaches are evaluated in the experiments, and we briefly review these approaches as follows.

- *GCN* [3] is considered as a benchmark graph convolutional network model, originally proposed for semi-supervised classification on homogeneous graph.
- *GAT* [5] is proposed for modeling homogeneous graph by applying a hidden self-attention layer to assign different weight to different node features.
- *GNN* and *LGNN* [7] are state-of-the-art community detection approaches originally proposed for homogeneous graphs.
- *HAN* [8] is proposed to model heterogeneous graph data by discovering both node-level and semantic-level information via designed attention mechanism.
- *STAR* [18] is a temporal approach to learn feature representations of temporal attributed graph by a designed grated recurrent unit (GRU) network.

5.3 Experimental Settings

It is well noticed that aforementioned approaches could not be directly applied to temporal heterogeneous graph. To customize these approaches, we first merge all momentary graphs to form a global one for fair comparison and then we evaluate each approach using homogeneous adjacency matrix in the experiments. Furthermore, to evaluate how the historical temporal graphs could affect community detection task, we respectively convolute past 3, 5 and 7 temporal graphs in an iterative manner to generate three versions of THGCN denoted as THGCN-3T, THGCN-5T and THGCN-7T. The heterogeneous GCN has two layers and d is set to the number of underlying communities. All approaches are implemented using PyTorch and optimized by Adam with a learning rate of 0.001.

5.4 Community Detection Results (RQ1)

We evaluate all methods as well as our THGCN-3T, THGCN-5T and THGCN-7T for community detection task and report the corresponding experimental results in Table 3. It is noticed that the model performance of all approaches on DBLP data significantly outperforms those on IMDB data. The possible reasons might be as follows. First, *authors* in DBLP data usually belongs to a specific research area, whereas *actors* or *directors* might belong to different communities. Second, the number of node features in IMDB data is nearly 7 times that of DBLP data which makes it more difficult to detect communities from IMDB data.

DBLP Results. We observe that THGCNs significantly outperform all other approaches w.r.t. all evaluation criteria. Particularly, the *NMI* score is improved by 17 percent when compared with the second best model, i.e., GCN. Furthermore, the *ACC* and *Macro-F1* scores of THGCNs indicate that the proposed approach could well detect communities. Compared with structural information embedding based approaches like GCN, THGCNs embed both structural and temporal features which could explain their superior performance. We also notice that HAN achieves the second best score on ACC criterion.

IMDB Results. Similar observations could be found from IMDB results. However, LGNN achieves the best *ARI* score while the rest

1. https://dblp.uni-trier.de/
2. https://www.imdb.com/

TABLE 3
Community Detection Results for all Compared Methods (%)

Methods	DBLP					IMDB				
	ACC	NMI	Modularity	ARI	Macro-F1	ACC	NMI	Modularity	ARI	Macro-F1
GAT	92.80	80.15	60.11	82.13	93.93	50.14	41.45	30.27	23.60	60.95
GNN	92.20	79.59	60.33	85.48	94.50	60.31	48.73	30.27	25.94	71.09
LGNN	91.40	76.73	61.09	83.37	94.22	58.47	29.46	26.02	**28.33**	51.90
HAN	94.13	51.84	58.72	42.51	94.96	61.48	50.56	40.29	24.10	70.93
STAR	84.51	57.08	60.50	58.56	84.64	59.00	34.43	42.53	23.29	53.05
THGCN-3T	**98.93**	**94.32**	61.12	**96.83**	**98.91**	**66.00**	**51.17**	43.38	27.20	**74.33**
THGCN-5T	98.13	90.78	61.51	94.36	98.13	64.67	48.77	**44.98**	26.17	71.12
THGCN-7T	97.87	89.88	**61.58**	93.55	97.87	65.33	50.84	44.79	26.69	73.19

The proposed THGCN-3T, THGCN-5T and THGCN-7T respectively denotes how many temporal graphs are convoluted.

evaluation results of LGNN are far from satisfying. The THGCNs achieve the second best ARI results. One possible reason is that LGNN employs a so-called "non backtracking" operator which can well preserve graph structural information across multiple time steps, whereas THGCNs cannot extract deeper structural information which is restricted by the limited length of predefined meta-paths. Moreover, we observe that longer time intervals could not further enhance model performance, as seen in the results of THGCN-5T and THGCN-7T. The possible reason for this observation is that the embedding features might get oversmoothing if more temporal graph data are convoluted. Detailedly, it is known that the model performance will gradually decrease with the number of GNN layers increases [19]. Essentially, the proposed ResCAC component is to convolute two consecutive graphs together, which is equivalent to introduce an extra layer of GNN model to some extent. Thus, if more temporal graphs are convoluted, the node embeddings will become close to each other.

Ablation Study. To demonstrate the effectiveness of the proposed ResCAC component and the heterogeneous component, we respectively remove these components, denoted as THGCN w/o R and THGCN w/o H-R, and report the results in Tables 4 and 5. It is noticed that after removing the ResCAC component, i.e., the temporal component, the model performance (THGCN w/o R) significantly decreases. If further removing the heterogeneous component, the model performance of THGCN w/o R-H is worse than that of THGCN w/o R which demonstrates the effectiveness of the proposed components. It is also noticed that THGCN w/o R achieves the better Modularity value. The possible reason is that without the temporal component, the model only considers the graph at the last time step, and thus is able to well capture the densely connected nodes. On the contrary, if considering the evolutionary patterns, the evolving graph nodes at past time steps will be counted to predict communities and these introduced dynamic features might undermine the contribution of current graph nodes, and thus results in a lower Modularity value.

5.5 Effect of Training Label Rate (RQ2)

To investigate the effect of training label rate, we choose *ACC* and *NMI* criteria to evaluate all approaches on DBLP dataset. In this experiment, we respectively choose training label ratio to 10, 20, 40, 60 and 80 percent and report the corresponding results in Fig. 4. From Fig. 4, THGCN-3T is constantly superior to the rest approaches and THGCN-5T is the second best one. It is also noticed that the NMI value of HAN and STAR are not good enough. The possible reason might be these models are proposed to model either heterogeneous graph or temporal graph, but cannot simultaneously model heterogeneous and temporal graph data. This verifies the effectiveness of the proposed approach.

TABLE 4
Ablation Study for THGCN on DBLP

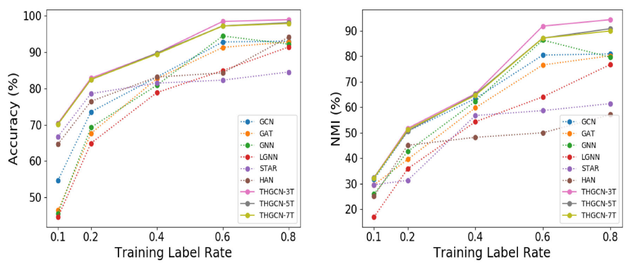
Methods	ACC	NMI	Modularity	ARI	Macro-F1
THGCN-3T	**98.93**	**94.32**	61.12	**96.83**	**98.91**
THGCN w/o R	95.33	80.94	**62.04**	86.20	95.35
THGCN w/o R-H	93.00	80.90	61.31	86.20	95.31

5.6 Visualization Results (RQ3)

To further demonstrate the model performance, we visualize community detection results of the original GCN, HAN, LGNN and THGCN, as plotted in Figs. 5 and 6. Apparently, the THGCN achieves the best visualization results on both datasets especially on IMDB. From the visualization results on IMDB, it is noticed that the five communities discovered by THGCN could be well separated and spread over the data space, whilst the "red" community in Figs. 5a and 5b still mixes up with the rest communities. Meanwhile, the "green", "yellow" and "purple" communities in LGNN overlap with each other, and thus is hard to separate. HAN can well separate "red" community but cannot separate the "blue" and "orange" communities. We observe that from visualization results on DBLP, all approaches could well separate the discovered communities. However, both GCN and THGCN could spread these communities over 2D plane, and THGCN is slightly better than GCN. This observation verifies the effectiveness of the proposed THGCN.

TABLE 5
Ablation Study for THGCN on IMDB

Methods	ACC	NMI	Modularity	ARI	Macro-F1
THGCN-3T	**66.00**	**51.17**	**43.38**	**27.20**	**74.33**
THGCN w/o R	62.33	45.58	43.34	22.24	67.82
THGCN w/o R-H	56.11	44.56	33.79	21.31	60.82



(a) Accuracy (b) NMI

Fig. 4. Effect of training label rate on DBLP dataset.

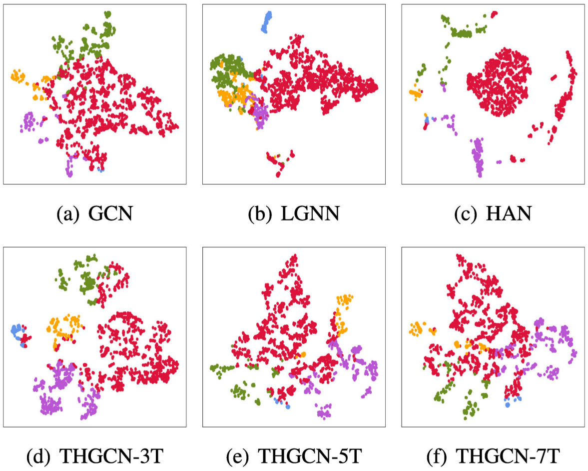


Fig. 5. Visulaization results on IMDB dataset.

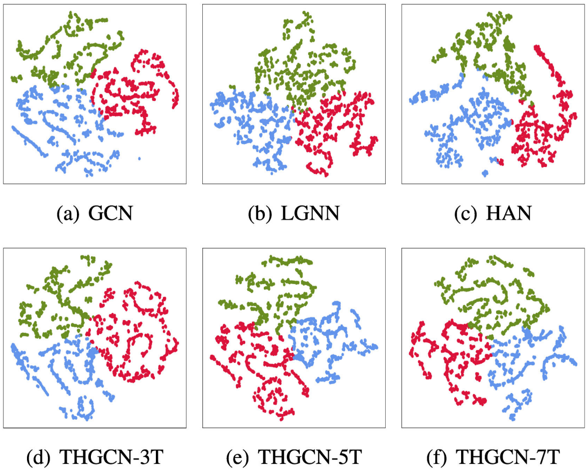


Fig. 6. Visulaization results on DBLP dataset.

6 Conclusion

In this paper, we propose a novel temporal-heterogeneous graph convolutional networks (THGCN) for community detection task. Particularly, we carefully design two neural network components, i.e., heterogeneous GCN component and ResCAC. The proposed heterogeneous GCN component is respectively applied on each momentary heterogeneous graph. Then, the proposed ResCAC component is iteratively applied on two consecutive graphs to well preserve the evolutionary features across several temporal graphs in addition to capturing static features of community member nodes. Extensive experiments are performed on several real-world datasets and the promising results demonstrate that the proposed THGCN is superior to the state-of-the-art approaches with respect to a number of widely adopted evaluation criteria.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grants 2018YFB1003800 and 2018YFB1003804, the National Natural Science Foundation of China under Grants 61872108, and the Shenzhen Science and Technology Program under Grant No. JCYJ20200109113201726 and JCYJ20170811153507788.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.
[2] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004, Art. no. 026113.
[3] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
[4] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[6] Y. Li, Y. Wang, T. Zhang, J. Zhang, and Y. Chang, "Learning network embedding with community structural information," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2937–2943.
[7] Z. Chen, X. Li, and J. Bruna, "Supervised community detection with line graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2019.
[8] X. Wang *et al.*, "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, 2019, pp. 2022–2032.
[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[10] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *Proc. 35th Conf. Assoc. Advancement Artif. Intell.*, 2021, pp. 10049–10057.
[11] B. Zhang, Z. Yu, and W. Zhang, "Community-centric graph convolutional network for unsupervised community detection," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 3515–3521.
[12] I. Lobov and S. Ivanov, "Unsupervised community detection with modularity-based attention model," in *Proc. Int. Conf. Learn. Representations*, 2019.
[13] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet Markov random fields: Semi-supervised community detection in attribute networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 152–159.
[14] Z. Liu, C. Huang, Y. Yu, P. Song, B. Fan, and J. Dong, "Dynamic representation learning for large-scale attributed networks," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1005–1014.
[15] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro- and macro-dynamics," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 469–478.
[16] Q. Li, Y. Shang, X. Qiao, and W. Dai, "Heterogeneous dynamic graph attention network," in *Proc. IEEE Int. Conf. Knowl. Graph*, 2020, pp. 404–411.
[17] X. Wang, Y. Lu, C. Shi, R. Wang, P. Cui, and S. Mou, "Dynamic heterogeneous information network embedding with meta-path based proximity," *IEEE Trans. Knowl. Data Eng.*, pp. 1–15, May 11, 2020, doi: 10.1109/TKDE.2020.2993870.
[18] D. Xu, W. Cheng, D. Luo, X. Liu, and X. Zhang, "Spatio-temporal attentive RNN for node classification in temporal attributed graphs," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3947–3953.
[19] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *Proc. Int. Conf. Learn. Representations*, 2021.