Continuous-Time Link Prediction via Temporal Dependent Graph Neural Network

Liang Qu[*]
Huaisheng Zhu[*]
qul@mail.sustech.edu.cn
11611725@mail.sustech.edu.cn
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China

Qiqi Duan
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
11749325@mail.sustech.edu.cn

Yuhui Shi[†]
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
shiyh@sustech.edu.cn

ABSTRACT

Recently, graph neural networks (GNNs) have been shown to be an effective tool for learning the node representations of the networks and have achieved good performance on the semi-supervised node classification task. However, most existing GNNs methods fail to take networks' temporal information into account, therefore, cannot be well applied to dynamic network applications such as the continuous-time link prediction task. To address this problem, we propose a Temporal Dependent Graph Neural Network (TDGNN), a simple yet effective dynamic network representation learning framework which incorporates the network temporal information into GNNs. TDGNN introduces a novel Temporal Aggregator (TDAgg) to aggregate the neighbor nodes' features and edges' temporal information to obtain the target node representations. Specifically, it assigns the neighbor nodes aggregation weights using an exponential distribution to bias different edges' temporal information. The performance of the proposed method has been validated on six real-world dynamic network datasets for the continuous-time link prediction task. The experimental results show that the proposed method outperforms several state-of-the-art baselines.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*.

KEYWORDS

graph neural networks, link prediction, network representation learning, dynamic networks

ACM Reference Format:
Liang Qu, Huaisheng Zhu, Qiqi Duan, and Yuhui Shi. 2020. Continuous-Time Link Prediction via Temporal Dependent Graph Neural Network. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380073

[*]Both authors contributed equally to this research.
[†]Corresponding author.

1 INTRODUCTION

Network data are ubiquitous in many domains such as social networks [12, 16], academic citation and collaboration networks [6, 9], protein-protein interaction networks [8, 14] and so on [1]. Generally, most links of these networks are continuously evolving over time. Predicting their dynamic behaviours plays an important role in many real-world applications, such as, recommending new friends in the social networks [10] or suggesting goods to customers in the E-commerce networks [33].

However, there are three main challenges to implement the above tasks in real-world applications: 1) Networks are non-Euclidean structural data to which many classic methods (e.g. Support Vector Machine [2] and Convolutional Neural Network [18]) cannot be well applied; 2) With the number of nodes increasing, the dimension of networks increases dramatically which needs expensive computing and storage resources; 3) The real-world network structures are dynamically evolving over time, which makes it difficult to capture implicit network features.

To address the aforementioned problems, in recent years, many network representation learning methods [32, 34] have been proposed. They map the nodes of the networks into the low-dimension real-valued vectors and preserve the networks structural information as much as possible. Most of these methods focus on the static networks and ignore the network temporal information. For example, Figure 1(a) shows an user-items network by linking the user and the items browsed. The traditional static network representation learning methods, as shown in Figure 1(b), consider the links of network independently and ignore the network temporal information. But in the real world, the users may browse many similar items in a short period of time according to their interests. Thus, as shown in Figure 1(c), the user is more likely to browse the mobile phone at time t because of the recent browsing history, that is, the link at $t-1$ should have a larger impact on predicting the link at t than the link at $t-5$.

Built on the recent success of GNNs for static network representation learning [31], in order to extend them to the dynamic

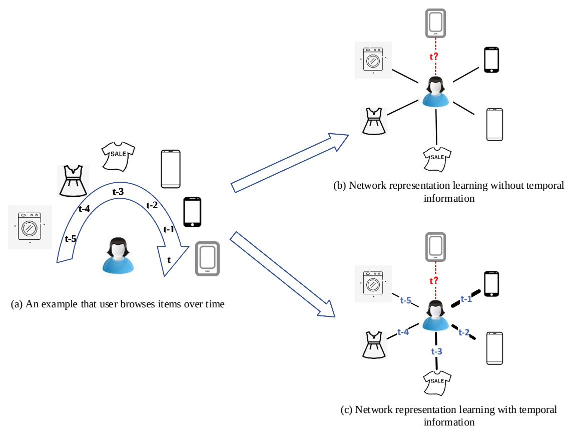

t-5
t-4
t-3
t-2
t-1
t
SALE
(a) An example that user browses items over time
t?
(b) Network representation learning without temporal information
(c) Network representation learning with temporal information

Figure 1: (a) An user-items network by linking the user and the items browsed. (b) The network representation learning method without temporal information. (c) The network representation learning method with temporal information.

networks applications such as continuous-time link prediction problem [26], we propose a Temporal Dependent Graph Neural Network (TDGNN) which incorporates network temporal information into GNNs. Specifically, the first step of TDGNN is to obtain the representation vectors of the two target nodes. Different from the current GNN methods which obtain the target node representations by aggregating their neighbor nodes features with the same weights. The proposed method introduces the Temporal Aggregator (TDAgg) which assigns the aggregation weights of the neighbor nodes features using an exponential distribution to bias the different edges' temporal information, that is, the neighbor nodes with closer connection time t to the target nodes have larger aggregation weights. The second step is to obtain the edge representation using Edge Aggregator (EdgeAgg) which maps the two target node representation vectors into the edge representation vectors, and in this work, we introduce and compare six different EdgeAggs.

The contributions of the paper can be summarized as follows:

- We propose a novel dynamic network representation learning framework named Temporal Dependent Graph Neural Network (TDGNN), an extended GNN incorporating network nodes' features and edges' temporal information via the new aggregation function TDAgg which can effectively obtain dynamic networks' node representations.
- We introduce and compare six different edge aggregation functions EdgeAggs which can effectively extend classic GNNs based node representations learning methods to learn the network edges representation.
- We validate the proposed method on six real-world dynamic networks for continuous-time link prediction tasks, and the experimental results show that TDGNN outperforms several existing state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 will formulate the essential definitions and knowledge about this work; Section 3 will give a detailed introduction to the proposed method; The experimental settings and results will be provided in Section 4, followed by related works and conclusion in Section 5 and 6 respectively.

2 PRELIMINARIES

In this section, we will provide several essential definitions and some related knowledge about this work.

2.1 Basic Definitions

- **Definition 1:** (*Dynamic Network/Graph*). A dynamic network $G = (V, E^T, T)$ consists of nodes set V, edges set E^T and time set T, where $v_i \in V$ represents the node in the network. $e^t_{i,j} \in E$ represents the edge linking nodes v_i and v_j at time t. $t \in T$ represents the link generating at time t.
- **Definition 2:** (*Network Node Representations Learning (NNRL)*). For a given network $G = (V, E^T, T)$, NNRL aims to map each node $v_i \in V$ into a real-valued vector $\vec{h}_v \in \mathbb{R}^d$, where d is node representations dimension.
- **Definition 3:** (*Network Edge Representations Learning (NERL)*). For a network $G = (V, E^T, T)$, NERL aims to map each edge $e^t_{i,j}$ into a real-valued vector $\vec{s}_e \in R^k$, where k is the edge representations dimension.
- **Definition 4:** (*Continuous-Time Link Prediction (CTLP)*). For a given network $G = (V, E^T, T)$ and a target edge/link $e^{t_0}_{i,j}$ linking node pair (v_i, v_j) at observed time $t_0 \in T$. CTLP aims to predict whether (v_i, v_j) will generate a new link or not when $t > t_0$.

2.2 Graph Neural Networks (GNNs)

Most existing GNNs methods can only capture the network structural information, however, they loss the network temporal information. In order to describe how to incorporate network temporal information into GNN models, we use GCN [16], one of the most representative GNNs, as an example. It is worth noting that the proposed method can easily be generalized to other GNNs.

The GCN is the approximation of spectral graph convolutions [13] method which highly relies on graph Laplacian as follows [16]:

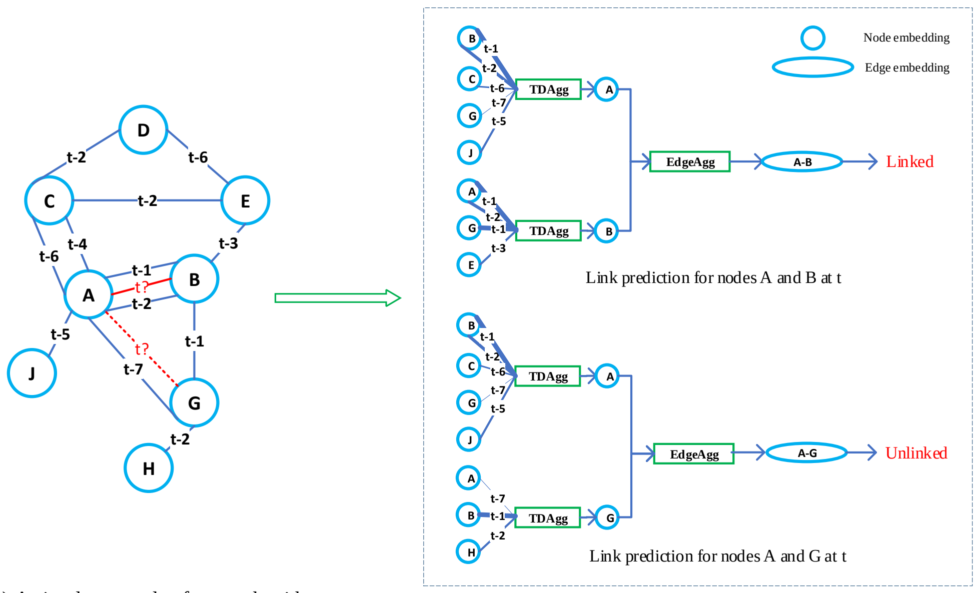
$$L = I - D^{-1/2} A D^{-1/2} \tag{1}$$

where L is the real symmetric positive semi-definite matrix, I is identity matrix, A is adjacent matrix and D is degree matrix.

For a given static undirected network $G = (V, E)$, the node $v_i \in V$ representation $\vec{h}^l_v$ at the l-th layer can be obtained by GCN as follows [16]:

$$\vec{h}^l_v = \sigma(W_l \sum_{u \in N(v) \cup v} \frac{\vec{h}^{l-1}_u}{\sqrt{|N(u)||N(v)|}}) \tag{2}$$

where σ is a non-linear activation function, W_l is parameters of GCN at the l-th layer and $N(u)$ is the neighbor nodes set of the target node v_i. From equation (2), we can see that GCN assigns the same weights to the neighbor node representations $\vec{h}^{l-1}_u$, therefore, it ignores the edges information in the network, especially for the dynamic networks containing the temporal information.



(a) A simple example of network with evolving links

(b) TDGNN

Figure 2: (a) A simple example of network with evolving links, and the links connecting node pairs (A,B) and (A,G) are to be predicted whether they will generate new links or not at time t (b) The proposed method consists of a novel aggregation function Temporal Aggregator (TDAgg) and the edge aggregation function EdgeAgg.

3 TEMPORAL DEPENDENT GRAPH NEURAL NETWORK (TDGNN)

To address the above problems, we propose TDGNN, a novel GNN framework, which can effectively capture both the networks' structural and temporal information.

Figure 2 provides an overview of TDGNN. The input to the TDGNN is a dynamic graph $G = (V, E^T, T, X)$ with edge temporal information, where $X = \{\vec{x}_1, \vec{x}_2 ..., \vec{x}_n\}, \vec{x}_i \in R^P$ is the node features, n is the number of nodes and P is the dimension of node features. In particular, as shown in Figure 1(a), the node pairs (v_A, v_B) can consist of multiple edges which generate at different time $t-1$ and $t-2$. The outputs of TDGNN are new node representation $H = \{\vec{h}_1^t, \vec{h}_2^t, ..., \vec{h}_n^t\}, \vec{h}_i^t \in R^d$ and edge representation $S = \{\vec{s}_1^t, \vec{s}_2^t, ..., \vec{s}_q^t\}, \vec{s}_j^t \in R^m$ at time t, where d and m are the dimension of node and edge representation respectively.

First, in order to introduce how TDGNN can learn the network structural and temporal information, for simplicity, we describe it in a single-layer forward TDGNN context and it can be easily generalized to the multi-layer networks. Assume that the node v is the target node, and the initial node representation $\vec{h}_v^0 = \vec{x}_v$. The new node representation at time t can be calculate by $TDAgg$ as follows:

$$\vec{h}_v^k = \sigma\Big(\sum_{u \in N_t(v) \cup v} \alpha_{vu}^t W \vec{h}_u^{k-1}\Big) \tag{3}$$

where k represents the k-th layer of the TDGNN, σ is a nonlinear activation function such as ReLU, $N_t(v)$ represents the neighbor nodes set of the target node v at time t, W is the learnable shared weight matrix and the neighbor nodes aggregating weights α_{vu}^t at time t can be calculated as follows:

$$\alpha_{vu}^t = \frac{e^{t-t_{v,u}}}{\sum_{u \in N_t(v) \cup v} e^{t-t_{v,u}}} \tag{4}$$

where $t_{v,u} \in T$ represents the edge generating time between node v and u.

Table 1: The Edge Aggregation Functions

EdgeAgg	Definition
Average(Ave)	$\frac{h_v+h_u}{2}$
Hadamard(Had)	$h_v \circ h_u$
Weighted-L1(W-L1)	$\lvert h_v - h_u \rvert$
Weighted-L2(W-L2)	$\lvert h_v - h_u \rvert^2$
Activation(Act)	$ReLU([h_v, h_u])$
Concatenation(Con)	$[h_v, h_u]$

Then, for the target edge $e_{v,u}^t$ between node pair (v, u) to be predicted at time t, the edge representation $\vec{s}_{v,u}^t$ can be obtain by the edge aggregation function $EdgeAgg$: $R^d \times R^d \rightarrow R^m$. In this work, we have selected and compared six common edge aggregation functions [11] summarized as Table 1.

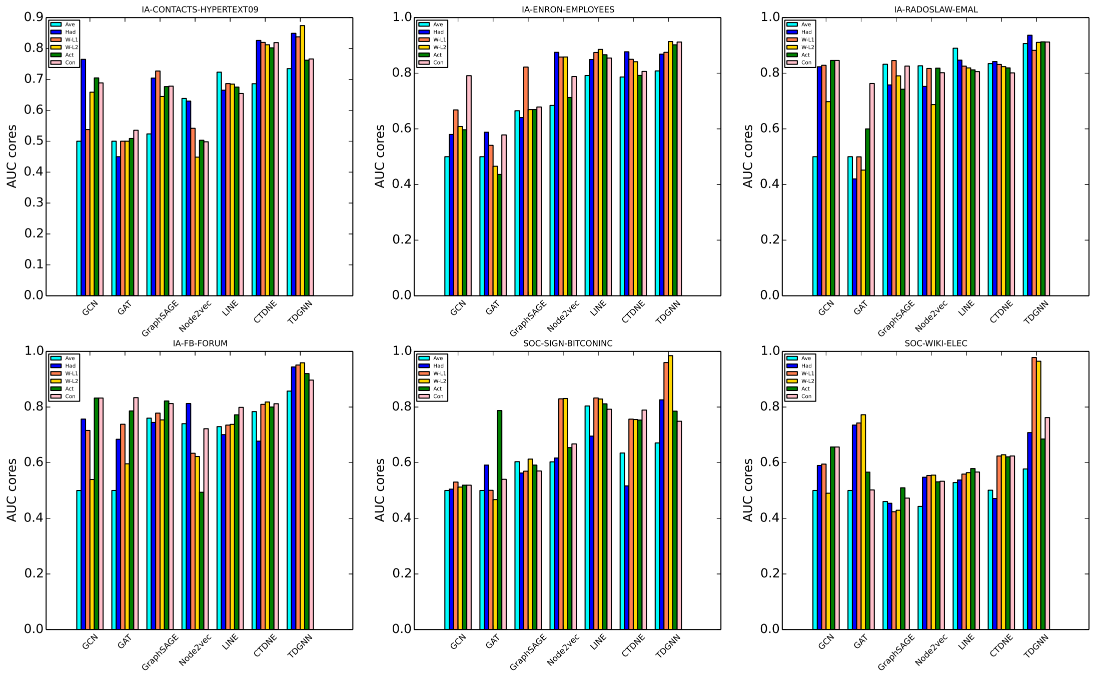


Figure 3: The Model Area Under Curve (AUC) Scores On Continuous-time Link Prediction Tasks with Various EdgeAggs on various datasets.

4 EXPERIMENTS

In order to validate the effectiveness of the proposed method, we conduct continuous-time link prediction experiments on six real-world dynamic network datasets. The sensitiveness of hyper parameter node representation dimension has also been analyzed to show the robustness of our proposed method. Moreover, We visualize the network representation learning process over the time. The experimental settings and results will be introduced as follows.

4.1 Datasets

We select six real-world dynamic network datasets under various scenarios (e.g. the social networks and the email communication networks) from NetworkRepository [25], and all these networks consists of a sequence of edges/links with timestamps. The statistic information of these datasets are summarized as Table 2. In particular, for continuous-time link prediction tasks, to generate a set of links with labels to train and test, for all datasets, first, we sort the links by time (ascending) and select the top 75% as training data, and remaining 25% as testing data. For both training data and testing data, we randomly sample an equal number negative edges from the networks.

4.2 Baselines

We compare the proposed TDGNN method with six state-of-the-art network representation learning algorithms including three static network representation learning methods (i.e. node2vec, LINE and graph convolutional network (GCN)) and three dynamic network representation learning methods (i.e. GraphSAGE, graph attention network (GAT) and CTDNE).

Table 2: The statistic information of the datasets. ($|V|$ is the number of nodes; $|E|$ is the number of edges.)

Datasets	$\|V\|$	$\|E\|$
IA-CONTACTS-HYPERTEXT09	113	20.8K
IA-ENRON-EMPLOYEES	151	50.5K
IA-RADOSLAW-EMAIL	167	82.9K
FB-FORUM	899	33.7K
SOC-SIGN-BITCONINC	5.8K	35.5K
SOC-WIKI-ELEC	7.1K	107K

- **GCN**: GCN [16] is one of the most representative GNNs based network representation learning method. It is a semi-supervised learning method which can directly operate on the network structural data.
- **GraphSAGE**: GraphSAGE [12] is an extended GNNs which can learn the node representations for previous unseen data. It provides three various aggregators including mean aggregator, LSTM aggregator and Pooling aggregator. For each datasets, we select the aggregators with the best performance on the tasks.
- **Node2vec**: Node2vec [11] uses a biased random walk based nodes sampling strategy to generate many node sequences,

and these sequences can be input into the skip-gram model to obtain the node representation.

- **LINE**: LINE [29] can also effectively obtain the node representation of the networks ,which uses a carefully designed objective function to preserve the local and global networks structural information,
- **CTDNE**: CTDNE [22] is also a biased random walk based node representation method. Unlike Node2vec, CTDNE incorporates the networks temporal information into the nodes sampling strategy, which can be applied to the dynamic networks.
- **GAT**: GAT [30] is also a GNNs based network representation learning method, which uses the attention mechanism to learn the nodes aggregating weights, and can deal with inductive as well as transductive tasks.

4.3 Continuous-time link prediction

The experimental results of continuous-time link prediction tasks are shown as Figure 3. For each method, we have selected and compared six EdgeAggs, and we repeat experiments 10 times and use Area Under Curve (AUC) scores as evaluation metric for all datasets. In general, we observe that: (1) The proposed method TDGNN can outperform all the comparison methods across all datasets. For example, on SOC-WIKI-ELEC dynamic network dataset, comparing with the best competitor GAT(W-L2), TDGNN is 20% higher in terms of AUC score. (2) For task-dependent network representation learning methods (i.e. GCN, GAT, GraphSAGE and TDGNN), the weighted based EdgeAggs could obtain better results than other EdgeAggs. For task-independent network representation learning method (i.e. Node2vec, LINE and CTDNE), the average and hadamard EdgeAggs are better. (3) It is worth noting that GCN and GAT are the methods with the averaging and self-adaption neighbor node features aggregation weights respectively, and our proposed TDGNN shows that using edges temporal information based exponential distribution to weight the aggregation weights could obtain the better results on dynamic networks for continuous-time link prediction task.

4.4 Network Layouts

In order to better show that TDGNN can well capture network temporal information, we select IA-CONTACTS-HYPERTEXT09 dynamic network to visualize the node pairs evolving over time in embedding space. Specifically, first all the nodes of network are embedded into a 128-dimension space with different network representation learning methods (i.e. GCN, Node2vec and TDGNN) at three sampled time t_1, t_2, t_3, then utilizes the t-SNE [20] map the 128-dimension vectors into a 2-dimension space for the better visualization. For a clearer illustration, we only randomly select 6 node pairs with evolving edges from the entire network, which is shown as Figure 4. Intuitively, when the two nodes have a link, they should be closer than when they are not linked. We can clearly observe that: (1) the static network representation learning method Node2vec is hard to capture the network temporal evolving process. (2) GCN can capture the change of the network, but it cannot well preserve the node pairs structural information. (3) The proposed method TDGNN can well capture the network structural and temporal information. For example, the distance between node pair (v_{10} and v_{11}) goes from near to far to near again, because the link generates, losts and generates again at time t_1, t_2, t_3.

4.5 Hyperparameter sensitivity analysis

The crucial hyperparameter of the proposed method TDGNN is the dimension of node representation. In order to analyze its influence to TDGNN, we select three dynamic network datasets (i.e. FB-FORUM, IA-ENRON-EMPLOYEES AND IA-CONTACTS-HYPERTEXT09), we investigate the continuous-time link prediction performance in terms of AUC score with various hyperparameter values $\{32, 64, 128, 256\}$. The experimental results are shown as Figure 5. We can observe that the dimension of node representation has small impacts on the TDGNN, which means the proposed method has good robustness.

5 RELATED WORKS

The main related works to the proposed method are network representation learning techniques on static networks and dynamic networks.

5.1 Static Network Representation Learning

Static network representation learning methods can be categorized into the task-independent methods [3] and the task-dependent methods [35]. As for the former, the representative methods include DeepWalk [24] and Node2vec [11] which are inspired by the skip-gram in word2vec [21]. They utilize the biased and unbiased truncated random walk to learn the latent node representations of a network respectively, and the obtained node representations can be input into many off-the-shelf methods for various downstream tasks such as node classification, link prediction, node clustering and etc..

As for the latter, built on the success of deep learning [19] techniques operating on regular Euclidean structural data such as images [5] and texts [4], GNN [28] and Graph Convolutional Network (GCN) [17] are proposed to operate on irregular non-Euclidean structural network/graph data. They learn the node representations through aggregating neighbor nodes feature information of the target nodes and the loss functions can contain downstream tasks information.

5.2 Dynamic Network Representation Learning

Similar to the static network representation learning methods, the dynamic network representation learning can also be categorized into task-dependent methods and task-independent methods.

Inspired by the ideas of CTDNE [22] which introduces a framework based on the idea of using temporal walks (temporally valid neighbors) for learning temporally valid embeddings with temporally valid information, the proposed method also utilizes temporal neighbors, but in a different way, as the proposed method uses this idea to extend aggregation functions to obey time as well. [15] extends the idea of CTDNE using feature-based temporal walks to obtain the node embeddings. [7] introduces an extended skip-gram approach to learn the representation of new nodes and update the original node representations. [36] proposes DynamicTriad which

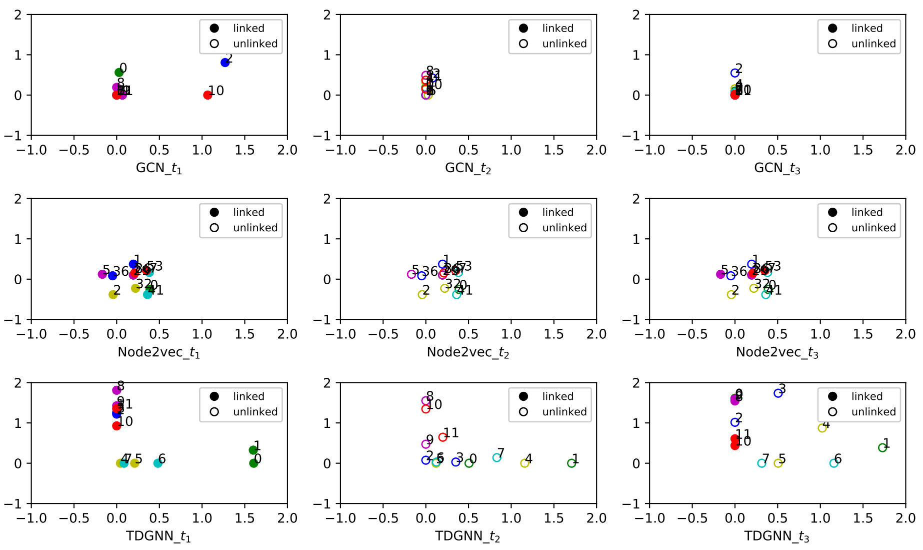


Figure 4: The 2-dimension t-SNE visualization of network node pairs with different methods (i.e. GCN, Node2vec and TDGNN) at three sampled time t_1, t_2, t_3. Each node pair is shown by various colors. The solid circles represent the node pair with a link at sampled time, and the hollow circles represent the node pair without a link at sampled time.

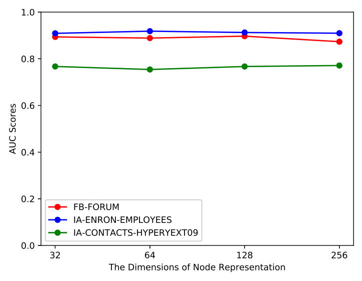


Figure 5: The dimension of node representation sensitivity analysis.

focus on triadic structure properties of the networks and can preserve the network structural evolving information.

For the task-dependent methods, [27] present DySAT, a method that uses self-attention mechanisms to learn node representations on dynamic networks. [33] introduces BrustGraph that is a framework for capturing the bursty events in a network. Other method like EvolveGCN [23] combines GNN with a recurrent neural network to evolve the graph model itself over time.

6 CONCLUSION

In this paper, we propose a simple yet effective dynamic network representation learning method named TDGNN, which bridges the gap between the task-independent network representation learning methods losing the downstream tasks information and task-dependent methods ignoring the network temporal information. It extends the GNNs based network representation learning methods to the dynamic networks by leveraging the edges temporal information. In particular, we propose a novel neighbor node features aggregator TDAgg, and it assigns the neighbor nodes aggregation weights using an exponential distribution to bias different edges' temporal information. Additionally, in order to extend the method to the edge related task (e.g. continuous-time link prediction), we select and compare six common edge aggregation functions which can obtain the edge representation.

In order to validate the effectiveness of the proposed method. We conduct the experiments including continuous-time link prediction, network layouts and hyperparameter sensitivity analysis on six real-world dynamic network datasets. The experimental results show that TDGNN can outperform several state-of-the-art baselines and has good robustness. Furthermore, the visualization results show that TDGNN can well capture the networks structural and temporal information.

In the future, we plan to explore whether we can use different distributions to calculate the node features aggregation weights. We also intend to explore the reasons behind the various EdgeAggs with various performances, and how TDGNN can be applied to the heterogeneous information networks which consist of multi-type of nodes or edges.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0804002, in part by the National Science Foundation of China under Grant 61761136008, in part by the Shenzhen Peacock Plan under Grant KQTD2016112514355531, and in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant 2017ZT07X386.

REFERENCES

[1] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. 2017. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. *arXiv:1709.07604 [cs]* (Sept. 2017). http://arxiv.org/abs/1709.07604 arXiv: 1709.07604.

[2] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, Article 27 (May 2011), 27 pages. https://doi.org/10.1145/1961189.1961199

[3] Haochen Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2018. A Tutorial on Network Embeddings. *CoRR* abs/1808.02590 (2018). arXiv:1808.02590 http://arxiv.org/abs/1808.02590

[4] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[6] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*. ACM Press, Halifax, NS, Canada, 135–144. https://doi.org/10.1145/3097983.3098036

[7] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. 2018. Dynamic Network Embedding: An Extended Approach for Skip-gram based Network Embedding.. In *IJCAI*. 2086–2092.

[8] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6530–6539. http://papers.nips.cc/paper/7231-protein-interface-prediction-using-graph-convolutional-networks.pdf

[9] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore Meta-paths in Heterogeneous Information Networks for Representation Learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*. ACM Press, Singapore, Singapore, 1797–1806. https://doi.org/10.1145/3132847.3132953

[10] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning Image and User Features for Recommendation in Social Networks. *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), 4274–4282.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. *CoRR* abs/1607.00653 (2016). arXiv:1607.00653 http://arxiv.org/abs/1607.00653

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. *arXiv:1706.02216 [cs, stat]* (June 2017). http://arxiv.org/abs/1706.02216 arXiv: 1706.02216.

[13] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30, 2 (2011), 129–150.

[14] Donald J. Jacobs, A.J. Rader, Leslie A. Kuhn, and M.F. Thorpe. 2001. Protein flexibility predictions using graph theory. *Proteins: Structure, Function, and Bioinformatics* 44, 2 (2001), 150–165. https://doi.org/10.1002/prot.1081 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.1081

[15] Di Jin, Mark Heimann, Ryan Rossi, and Danai Koutra. 2019. node2bits: Compact Time-and Attribute-aware Node Representations for User Stitching. *arXiv preprint arXiv:1904.08572* (2019).

[16] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* (Sept. 2016). http://arxiv.org/abs/1609.02907 arXiv: 1609.02907.

[17] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016). arXiv:1609.02907 http://arxiv.org/abs/1609.02907

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[20] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. Curran Associates Inc., USA, 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959

[22] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-time dynamic network embeddings. In *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 969–976.

[23] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E. Leisersen. 2019. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. *CoRR* abs/1902.10191 (2019). arXiv:1902.10191 http://arxiv.org/abs/1902.10191

[24] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 701–710. https://doi.org/10.1145/2623330.2623732

[25] Ryan Rossi and Nesreen Ahmed. 2015. The network data repository with interactive graph analytics and visualization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[26] Sina Sajadmanesh, Jiawei Zhang, and Hamid R. Rabiee. 2017. Continuous-Time Relationship Prediction in Dynamic Heterogeneous Information Networks. *CoRR* abs/1710.00818 (2017). arXiv:1710.00818 http://arxiv.org/abs/1710.00818

[27] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2018. Dynamic Graph Representation Learning via Self-Attention Networks. *arXiv preprint arXiv:1812.09430* (2018).

[28] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[29] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 1067–1077.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv:1710.10903 [cs, stat]* (Oct. 2017). arXiv: 1710.10903.

[31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2019. A Comprehensive Survey on Graph Neural Networks. *CoRR* abs/1901.00596 (2019). arXiv:1901.00596 http://arxiv.org/abs/1901.00596

[32] Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2018. Deep Learning on Graphs: A Survey. *arXiv:1812.04202 [cs, stat]* (Dec. 2018). http://arxiv.org/abs/1812.04202 arXiv: 1812.04202.

[33] Yifeng Zhao, Xiangwei Wang, Hongxia Yang, Le Song, and Jie Tang. 2019. Large Scale Evolving Graphs with Burst Detection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4412–4418. https://doi.org/10.24963/ijcai.2019/613

[34] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *arXiv:1812.08434 [cs, stat]* (Dec. 2018). http://arxiv.org/abs/1812.08434 arXiv: 1812.08434.

[35] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *CoRR* abs/1812.08434 (2018). arXiv:1812.08434 http://arxiv.org/abs/1812.08434

[36] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. 2018. Dynamic network embedding by modeling triadic closure process. In *Thirty-Second AAAI Conference on Artificial Intelligence*.