Towards Explainable Representation of Time-Evolving Graphs via Spatial-Temporal Graph Attention Networks

Zhining Liu*
liuzhininguestc@gmail.com
University of Electronic Science and Technology of China

Dawei Zhou*
davidchouzdw@gmail.com
University of Illinois at Urbana-Champaign

Jingrui He
jingrui.he@gmail.com
University of Illinois at Urbana-Champaign

ABSTRACT

Many complex systems with relational data can be naturally represented as dynamic processes on graphs, with the addition/deletion of nodes and edges over time. For such graphs, network embedding provides an important class of tools for leveraging the node proximity to learn a low-dimensional representation before using the off-the-shelf machine learning models. However, for dynamic graphs, most, if not all, embedding approaches rely on various hyper-parameters to extract spatial and temporal context information, which differ from task to task and from data to data. Besides, many regulated industries (e.g., finance, health care) require the learning models to be interpretable and the output results to meet compliance. Therefore, a natural research question is how we can jointly model the spatial and temporal context information and learn a unique network representation, while being able to provide interpretable inference over the observed data. To address this question, we propose a generic graph attention neural mechanism named *STANE*, which guides the context sampling process to focus on the crucial part of the data. Moreover, to interpret the network embedding results, *STANE* enables the end users to investigate the graph context distributions along three dimensions (i.e., nodes, training window length, and time). We perform extensive experiments regarding quantitative evaluation and case studies, which demonstrate the effectiveness and interpretability of *STANE*.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

Network embedding; temporal networks; graph attention

ACM Reference Format:
Zhining Liu, Dawei Zhou, and Jingrui He. 2019. Towards Explainable Representation of Time-Evolving Graphs via Spatial-Temporal Graph Attention Networks. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3357384.3358155

1 INTRODUCTION

Network embedding [5, 11] has recently attracted a surge of research interest in a myriad of high impact domains, ranging from social networks [8] to collaborative networks [17], from knowledge graphs [9] to protein-protein networks [2]. In contrast to the conventional graph analytic tools, network embedding leverages the node proximity to learn a low-dimensional network representation, based on which a variety of off-the-shelf machine learning models can be easily applied for graph mining tasks such as node classification [7], link prediction [5], community detection [8, 14] and rare category analysis [12, 13].

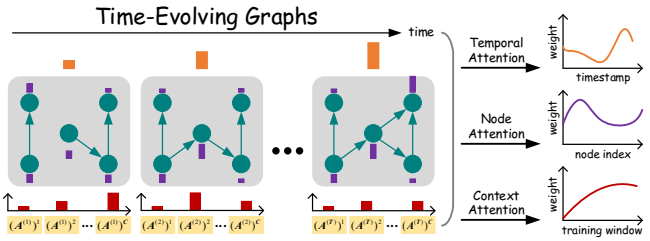

Figure 1: An outline of *STANE*. The embedded chart includes the attention distributions over nodes (purple), training window length (red) and timestamps (orange) dimensions, which allows the analysts to investigate the importance of node, time, training window length.

However, most real-world networks are intrinsically evolving over time. Compared with the static setting, the dynamic network evolution is more complex and noisy as the nodes and edges may appear, vanish, or even reappear. Several initial attempts (e.g., [4, 15, 17]) have been made to solve the dynamic network embedding problem, which often introduce some hyper-parameters (e.g., arbitrary length random walk [4]) to extract the spatial-temporal context information. Such hyper-parameters may have a huge impact on the performance of downstream applications(e.g., training window length in [4]). On the other hand, it is unclear how to jointly model the extracted context information from the spatial domain (e.g., which node is more important?) and time domain (e.g., which snapshot of the time-evolving graph contains crucial dynamic patterns?). Furthermore, many real systems with highly regulated processes (e.g., finance, health care) often require the learning models to be interpretable and the output results to meet compliance [3]. In this case, a user-friendly model with interpretable inferences can help analysts investigate the malicious patterns and largely reduce the workload of analyzing the raw data.

To address the aforementioned challenges, we propose a generic learning framework named *STANE*, which aims to learn a unique representation for dynamic networks and provide comprehensive interpretable inferences for the end users. In particular, instead of extracting context via random walks, we introduce the expectation of the co-occurrence matrix of the dynamic graphs. In addition, we develop a spatial-temporal neural attention mechanism to estimate the above co-occurrence matrix and guide the embedding algorithm to focus on the context information with high importance. At last, benefiting from the attention mechanism, we are able to conduct

*Both authors contributed equally to this research.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358155

fine-grained analysis on the node embedding through aggregating attention parameters along different dimensions (i.e., nodes, training window length and time).

The major contributions of this paper are as follows:

- **Problem.** We formally define the problem of explainable time-evolving graph representation and identify its unique challenges arising from real applications.
- **Algorithm.** We propose a generic learning framework for dynamic network embedding, which is able to (1) jointly model the spatial and temporal context information without extra hyper-parameters, and (2) provide interpretable inferences over the observed dynamic graphs.
- **Evaluations.** We perform extensive experiments and case studies on six real data sets, showing that the proposed algorithm achieves consistent improvement in the prediction performance with good interpretability.

The rest of the paper is organized as follows. In Section 2, we introduce the problem definition and our proposed framework *STANE*. Experimental results and literature review are presented in Section 3 and Section 4, before we conclude the paper in Section 5.

2 PROPOSED MODEL

In this section, we start from the problem definition, then we introduce the proposed method for the dynamic network embedding.

2.1 Problem Definition

Suppose we are given a evolving graph $\widetilde{G} = \{G^{(1)}, \ldots, G^{(T)}\}$, i.e., $G^{(t)} = (V^{(t)}, E^{(t)}), t = 1, \ldots, T$, that can be presented as a series of *time-evolving adjacency matrices*, i.e., $\boldsymbol{A^{(1)}}, \ldots, \boldsymbol{A^{(T)}}$. For the sake of exposition, we assume the numbers of nodes of different snapshots $G^{(t)}$ are fixed, which leads to a fixed node set $\widetilde{V}$ with $|\widetilde{V}| = n$; if not, we can reserve rows/columns with zero padding if necessary. In addition, since the information for a single time slice may be too sparse, analysts typically want to study a larger portion of the observed data to capture interesting patterns and structures. In this paper, we preprocess the data in the form of increasing time-evolving graphs, where each *time-evolving adjacency matrix* $\boldsymbol{A^{(t)}}$ aggregates the information from timestamp 1 to t. With the above notations, we formally define our problem as follows:

Problem 1. *Explainable Time-Evolving Graph Representation*

Input: *(i) a time-evolving graph $\widetilde{G}$, (ii) a user-specific graph embedding dimension d.*

Output: *(i) a graph representation $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$ that captures both spatial and temporal graph context in $\widetilde{G}$, (ii) importance inferences regarding nodes, training window length and timestamps.*

2.2 A Generic Learning Framework

The central goal of this work is to learn a generic network embedding for time-evolving graphs, which is able to jointly encode the spatial and temporal context distribution into a unique representation and provide interpretable inferences over the dynamic graph elements (e.g., nodes, timestamps). To achieve this goal, we need to take consideration of the following aspects. First (*C.1*), our framework needs to be capable of learning hyper-parameters with respect to context sampling (e.g., co-occurrence matrix $\boldsymbol{D}$, the training window length k, the sampling timestamp t) in order to be automatically trained on different graphs. Second (*C.2*), to obtain the comprehensive representation of the observed time-evolving graphs, we aim to jointly model the spatial and temporal context

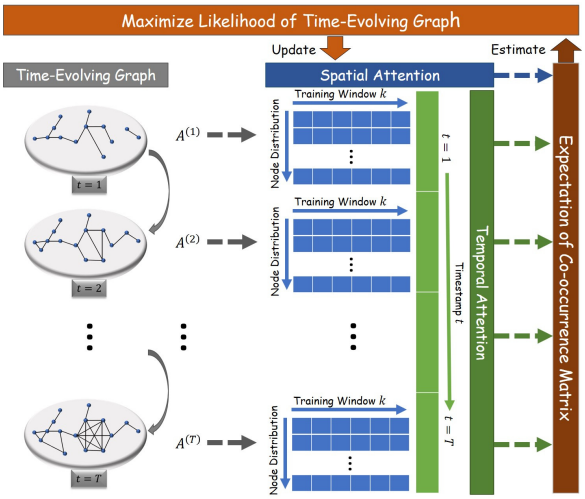

Figure 2: An illustration of the proposed framework. On the left-hand side, the time-evolving graph $\widetilde{G}$ that changes over T timestamps. On the right-hand side, the spatial attention module (i.e., colored in blue) and the temporal attention module (i.e., colored in green) are jointly trained to simulate the context sampling process by computing the expectation of co-occurrence matrix $\mathbb{E}[\widetilde{D}; \widetilde{Q}]$.

within a unique optimization scheme. Third (*C.3*), in addition to the performance, we also require our model to be explainable.

Fig. 2 presents an overview of the proposed *STANE* framework, where the spatial context and temporal context are jointly extracted via a neural attention mechanism. In particular, given a time-evolving graph $\widetilde{G} = \{G^{(1)}, \ldots, G^{(T)}\}$, the whole process can be separated into three steps: (1) instead of generating random walks by simulation[5], we estimate the expectation of co-occurrence matrix $\widetilde{\boldsymbol{D}}$ of the time-evolving graph with a neural attention mechanism. then, (2) the spatial-temporal attention module and node embeddings are jointly trained within a unified optimization process, aiming to maximize the likelihood of the observed $\widetilde{G}$; at last, (3) the end users can investigate the graph context distributions by aggregating the learned attention weights. Next, we dive into the details of *STANE* in the following three aspects.

Learning the spatial-temporal context distribution. As pointed out by [1], random walk based network embedding approaches (e.g., [5]) actually construct a co-occurrence matrix $\boldsymbol{D}$, of which expectation is written as:

$$\mathbb{E}[\boldsymbol{D}; C] = \sum_{k=1}^{C} w_k \cdot \boldsymbol{P} \times (\boldsymbol{M})^k \tag{1}$$

where C is the largest walk length; the definition of w_k varies with different methods (e.g., w_k is defined as the probability of node with distance k from anchor node to be selected in [5]); $\boldsymbol{P}$ is the diagonal matrix of the prior distribution $\boldsymbol{p}$, i.e., $\boldsymbol{P} = diag(\boldsymbol{p})$, with setting $P(v, v)$ as the number of walks starting at anchor point v; $\boldsymbol{M}$ is the transition probability matrix $\boldsymbol{M} = diag(\boldsymbol{A} \times \boldsymbol{1}_n)^{-1} \times \boldsymbol{A}$.

Here, we generalize the expectation of co-occurrence matrix $\mathbb{E}[\boldsymbol{D}]$ to the dynamic setting. Instead of introducing new hyper-parameters of prior distribution regarding nodes (i.e., v), training window length (k) and timestamp (t), we estimate the expectation of $\widetilde{\boldsymbol{D}}$ of time-evolving graph $\widetilde{G}$ via a neural attention mechanism

with trainable attention parameters $\widetilde{\boldsymbol{Q}} = \{\boldsymbol{Q^{(1)}}, \boldsymbol{Q^{(2)}}, \ldots, \boldsymbol{Q^{(T)}}\}$, that is

$$\mathbb{E}[\widetilde{\boldsymbol{D}}; \widetilde{\boldsymbol{Q}}] = \sum_{t=1}^{T} \sum_{v=1}^{|\widetilde{V}|} \sum_{k=1}^{C} B^{(t)}(v,k)(\boldsymbol{M^{(t)}})^k \tag{2}$$

where $\boldsymbol{Q^{(t)}} \in \mathbb{R}^{n \times C}$, $t = 1, \ldots, T$, is the context distribution matrix to capture the dynamic network context information with respect to nodes and training window length at timestamp t; $\boldsymbol{B^{(t)}} \in \mathbb{R}^{n \times C}$, $t = 1, \ldots, T$, is the normalized context distribution matrix at timestamp t, i.e., $B^{(t)}(v,k) = \frac{Q^{(t)}(v,k)}{\sum_{t=1}^{T} \sum_{v=1}^{|\widetilde{V}|} \sum_{k=1}^{C} Q^{(t)}(v,k)}$; $\boldsymbol{M^{(t)}}$ is the transition probability matrix at timestamp t. To be specific, we replace the hyper-parameter w_k and $\boldsymbol{P}$ in Eq. 1 with the trainable attention weight matrix $\boldsymbol{Q^{(t)}}$ at each timestamp t, where each entry $Q^{(t)}(v,k) = w_k \cdot P(v,v)$ indicates the importance factor of network context within k distance to v at the timestamp t.

Maximizing the graph likelihood of time-evolving graphs. Many existing temporal network embedding approaches [4, 15, 17] treat temporal context (i.e., which timestamp is important?) and spatial context (i.e., which region of the graph is important and how large it is?) as two independent information sources, thus these methods fail to fully investigate the fine-grained context information of dynamic graphs (e.g., given two node-context pairs (v_1, c_1) and (v_2, c_2), which one is more important in the k^{th}-order ego-network of a given anchor point u at a specific timestamp t). In order to thoroughly investigate such fine-grained context information in the dynamic setting, we propose to jointly extract the spatial context and temporal context by maximizing the likelihood of time-evolving graphs. In particular, the overall objective function of our *STANE* framework is formulated as follows

$$\begin{aligned} \min_{Z, \widetilde{Q}} & - \prod_{v,c \in \widetilde{V}} log[\sigma(\boldsymbol{z_v^T z_c})^{\mathbb{E}[\widetilde{D}; \widetilde{Q}]} \cdot (1 - \sigma(\boldsymbol{z_v^T z_c})^{1[\boldsymbol{A^{(T)}}=0]}] \\ & + \alpha \sum_{t=1}^{T} \sum_{v=1}^{|\widetilde{V}|} \sum_{k=1}^{C} |Q^{(t)}(v,k)| \end{aligned} \tag{3}$$

where $\mathbb{E}[\widetilde{\boldsymbol{D}}; \widetilde{\boldsymbol{Q}}]$ is defined in Eq.2, $\boldsymbol{z_v}$ and $\boldsymbol{z_c}$ are the embedding vectors of anchor node v and context node c respectively, and α is a hyper-parameter to balance the impact of the regularization term on the overall objective function. In particular, the first term corresponds to the graph likelihood estimator of the observed time-evolving graphs. Note that, since the last snapshot $G^{(T)}$ aggregates all the information from the initial timestamp to the very last timestamp, we are considering all the nodes and edges that have been added to the graph over time. The second term corresponds to the sparse regularizer (i.e., L_1 norm) that is designed to select the key context information from $\widetilde{G}$.

Interpretation via Spatial-Temporal Attention. For various network analytic tasks, the interpretability of the model is essential for understanding the logic behind the graph data. To be specific, given a time-evolving graph $\widetilde{G}$, the end users may want to investigate the attention distribution of various dimensions (e.g., node dimension and time dimension). It is natural to exploit the learned attention parameters $\widetilde{\boldsymbol{B}}$ to fulfill the interpretation requirements. While the burden of using the raw attention $\widetilde{\boldsymbol{B}}$ to decipher the importance of the nodes and timestamps comes from the high-dimensionality of $\widetilde{\boldsymbol{B}} \in \mathbb{R}^{T \times n \times C}$. To accommodate this issue, we adopt the aggregation function $f_{agg} : \mathbb{R}^{a \times b \times c} \rightarrow \mathbb{R}^q (q \in \{a, b, c\})$ to aggregate along two of three dimensions of $\widetilde{\boldsymbol{B}}$, e.g., $f_{agg}(\widetilde{\boldsymbol{B}}) = \left[\frac{\sum_j \sum_k \widetilde{B}(1,j,k)}{\sum_i \sum_j \sum_k \widetilde{B}(i,j,k)}, \ldots, \frac{\sum_j \sum_k \widetilde{B}(a,j,k)}{\sum_i \sum_j \sum_k \widetilde{B}(i,j,k)}\right]$. In this way, we can estimate the attention distributions of node, training window length and time by compressing $\widetilde{\boldsymbol{B}}$ into vector representations.

3 EXPERIMENTAL RESULTS

In this section, we evaluate our proposed *STANE* framework regarding effectiveness and interpretability on six real dynamic networks.

3.1 Experiment Setup

Datasets: The statistics and brief information of all datasets used in our experiments are summarized in Table 1.

Table 1: Statistics of datasets

Name	#Nodes|#Edges|#Classes|#T	Description
DBLP	1909 | 8237 | 4 | 3	DBLP citation network [11]
FB	1899 | 61734 | - | 5	Facebook social network [6]
SO	3262 | 19926 | 2 | 5	Stack Overflow comment network [11]
IAR	6809 | 52050 | - | 5	communication network [6]
WIKI	7118 | 107071 | 2 | 6	who-votes-on-whom network [6]
IAE	10106 | 50632 | - | 6	bipartite graph of people [6]

Comparison Methods: Our proposed method is compared to five baselines: **DeepWalk**[5], **WYS**[1], **TNE**[16], **Traid**[15] and **HTNE**[17]. In particular, DeepWalk and WYS are static network embedding methods, while TNE, Traid and HTNE are recent network embedding approaches that designed for dynamic graphs.

3.2 Effectiveness Analysis

We evaluate the effectiveness of the *STANE* on the task of link prediction and node classification, by comparing with five baseline methods across six datasets.

Link Prediction. The experiment of link prediction is designed to predict the probability of whether two nodes are connected by an edge at the last timestamp T, as the last snapshot $G^{(T)}$ aggregates all the previous information. Besides, a fraction (50%) of edges in $G^{(T)}$ is removed, which ends with two set $E^{(T)}_{train}$ and $E^{(T)}_{test}$. Then node embeddings are learned from the remaining edges (i.e., $E^{(T)}_{train}$ and $E^{(t)}, t = 1, \ldots, T-1$). With the same number of non-existent edges sampled from $G^{(T)}$, we calculate ROC AUC to report the performance of each method, which is shown in Table 2. We observe that *STANE* consistently outperform all the five baseline methods across all the six datasets.

Table 2: Link prediction results.

	Datasets					
Methods	DBLP	FB	SO	IAR	WIKI	IAE
DeepWalk	0.670	0.656	0.579	0.875	0.794	0.555
WYS	0.841	0.952	0.844	0.954	0.963	0.843
TNE	0.610	0.708	0.736	0.881	0.719	0.656
Triad	0.522	0.601	0.575	0.762	0.502	0.515
HTNE	0.844	0.913	0.607	0.911	0.844	0.792
STANE	**0.882**	**0.966**	**0.888**	**0.997**	**0.982**	**0.898**

Node Classification. This task to predict the label of the node based on its embedding. We train a logistic classifier with a fraction (50%) of graph nodes and predict the labels of the rest of nodes

based on the learned node embeddings. Note that the experiments are performed on the three datasets (i.e., DBLP, SO, WIKI) with label information. In Table 3, we report the micro-F1 to measure the performance of the methods. In general, *STANE* outperforms all the baseline methods in most cases.

Table 3: Node classification results.

	Methods					
Datasets	DeepWalk	WYS	TNE	Triad	HTNE	STANE
DBLP	0.718	0.727	0.612	0.511	0.737	**0.757**
SO	0.610	0.614	0.592	0.591	**0.632**	0.617
WIKI	0.701	0.699	0.690	0.553	0.715	**0.718**

3.3 Interpretability Analysis

Here, we investigate the interpretability of *STANE* on DBLP dataset. The distribution of the context attention and time attention is shown in Fig. 3, where the largest training window length is 5 and the total timestamps are 3. In summary, we have the following observations: (1) the last snapshot $G^{(3)}$ is more important than the previous two snapshots, which is easy to follow as the last snapshot aggregates all the nodes and edges in the previous timestamps; (2) the high-order proximity plays the most important role in presenting the observed time-evolving networks. To further verify the above observations, we test *STANE* based on $\mathbb{E}[\widetilde{D}; \widetilde{Q}]$ which is estimated solely from one of the three snapshots, respectively. The evaluation result of each snapshot is [0.527,0.540,0.848] (ROC AUC for the link prediction) and [0.501,0.612,0.720] (micro-F1 for the node classification), respectively. This suggests that among the three snapshots, $G^{(3)}$ is the most important one, which is consistent with the observed distribution of attention weights.

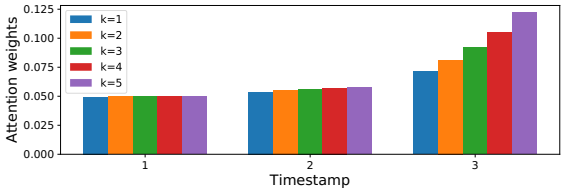

Figure 3: Attention weights over time and walk length (k).

4 RELATED WORKS

There is a growing interest in encoding the temporal patterns of time-evolving graphs into embedding representations. For example, TNE [16] generates the embedding based on the non-negative matrix factorization of a series of time-evolving adjacency matrices with the smoothness constraint; Traid [15] focuses on modeling how to derive a closed triad from an open triad; [4] proposes to generate temporal random walks in increasing order of edge times to embed the continuous-time network in a unique representation; HTNE [17] studies on the neighborhood formation sequence through Hawkes process to capture the influence of historical neighbors on the current neighbors; There is also another line of works based on graph convolution for attributed graph sequence, most of which are applied for traffic prediction [10]. However, it is still an open question that how to incorporate and balance the extracted network context information from the spatial domain (e.g., the network structures) and the temporal domain (e.g., the evolution of the network over time). In this paper, we develop a unified attention mechanism to jointly explore the spatial and temporal context from the time-evolving networks with a learned importance factor in terms of each node v, training window length k, and timestamps t.

5 CONCLUSION

In this paper, we propose a temporal network embedding (*STANE*) framework. By parameterizing the co-occurrence matrix with trainable parameters to balance spatial and temporal context information, *STANE* successfully encodes structural and temporal patterns within the time-evolving graphs into node embeddings. In addition, we present that through detailed analysis on the attention parameters, we could achieve a better understanding of the node embeddings and the evolution of the temporal networks. Extensive experiments on several real-world networks demonstrate the effectiveness of the proposed method. In the future, it is of interest to introduce structured attention and to study the scalability of the proposed method via a batched training scheme.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation under Grant No. IIS-1552654 and Grant No. IIS-1813464, the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-02-00, United States Air Force and DARPA under contract number FA8750-17-C-0153, and an IBM Faculty Award. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A Alemi. 2018. Watch Your Step: Learning Node Embeddings via Graph Attention. In *NIPS*.
[2] Carlo Vittorio Cannistraci, Gregorio Alanis-Lobato, and Timothy Ravasi. 2013. Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics*.
[3] W. Knight. 2017. The financial workd wants to open AI's black boxes. Intelligent Machines.
[4] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. 2018. Continuous-time dynamic network embeddings. In *WWW BigNet*.
[5] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*. ACM.
[6] Ryan Rossi and Nesreen Ahmed. 2015. The network data repository with interactive graph analytics and visualization. In *AAAI*.
[7] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. International World Wide Web Conferences Steering Committee.
[8] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding.. In *AAAI*.
[9] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes.. In *AAAI*.
[10] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.
[11] Dawei Zhou, Jingrui He, Hongxia Yang, and Wei Fan. 2018. Sparc: Self-paced network representation for few-shot rare category characterization. In *SIGKDD*. ACM.
[12] Dawei Zhou, Arun Karthikeyan, Kangyang Wang, Nan Cao, and Jingrui He. 2017. Discovering rare categories from graph streams. *Data mining and knowledge discovery*.
[13] Dawei Zhou, Kangyang Wang, Nan Cao, and Jingrui He. 2015. Rare category detection on time-evolving graphs. In *ICDM*. IEEE.
[14] Dawei Zhou, Si Zhang, Mehmet Yigit Yildirim, Scott Alcorn, Hanghang Tong, Hasan Davulcu, and Jingrui He. 2017. A local algorithm for structure-preserving graph cut. In *SIGKDD*. ACM.
[15] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. 2018. Dynamic Network Embedding by Modeling Triadic Closure Process. In *AAAI*.
[16] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. 2016. Scalable temporal latent space inference for link prediction in dynamic social networks. *TKDE*.
[17] Yuan Zuo, Guannan Liu, Hao Lin, Jia Guo, Xiaoqian Hu, and Junjie Wu. 2018. Embedding Temporal Network via Neighborhood Formation. In *SIGKDD*. ACM.