Graph Structural-topic Neural Network

Qingqing Long*
Key Laboratory of Machine Perception (Ministry of Education), Peking University
qingqinglong@pku.edu.cn

Yilun Jin*
The Hong Kong University of Science and Technology
yilun.jin@connect.ust.hk

Guojie Song†
Key Laboratory of Machine Perception (Ministry of Education), Peking University
gjsong@pku.edu.cn

Yi Li
Peking University
liyi2015@pku.edu.cn

Wei Lin
Alibaba Group
yangkun.lw@alibaba-inc.com

ABSTRACT

Graph Convolutional Networks (GCNs) achieved tremendous success by effectively gathering local features for nodes. However, commonly do GCNs focus more on node features but less on local graph structures, especially high-order structural patterns within the neighborhood. However, such local structural patterns have been shown to be indicative of node properties in numerous fields. In addition, it is not just single patterns, but the distribution over all these patterns matter, because networks are complex and the neighborhood of each node consists of a mixture of various nodes and structural patterns. Correspondingly, in this paper, we propose *Graph Structural-topic Neural Network*, abbreviated GraphSTONE, a GCN model that utilizes topic models of graphs, such that the structural topics capture indicative graph structures broadly from a probabilistic aspect rather than focusing on merely a few structures. Specifically, we build topic models upon graphs using anonymous walks and Graph Anchor LDA, an LDA variant that selects significant structural patterns first, so as to alleviate the complexity and generate structural topics efficiently and concisely. In addition, we design multi-view GCNs to unify node features and structural topic features and utilize structural topics to guide the aggregation. We evaluate our model through both quantitative and qualitative experiments, where our model exhibits not only promising performance and high efficiency, but also clear interpretability.

CCS CONCEPTS

• **Networks** → *Network structure*; • **Information systems** → *Collaborative and social computing systems and tools.*

KEYWORDS

Graph Convolutional Network, Local Structural Patterns, Topic Modeling

*These authors contributed equally to the work.
†Corresponding Author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '20, August 23–27, 2020, Virtual Event, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

ACM Reference Format:
Qingqing Long, Yilun Jin, Guojie Song, Yi Li, and Wei Lin. 2020. Graph Structural-topic Neural Network. In *26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Graphs[1] are intractable due to their irregularity and sparsity. Fortunately, Graph Convolutional Networks (GCNs) succeed in learning deep representations of graph vertices and attract tremendous attention due to their performance and scalability.

While GCNs succeed in extracting local features from a node's neighborhood, it should be noted that they primarily focus on node features, leaving them less capable of exploiting local structural properties of nodes. Specifically, uniform aggregation depicts one-hop relations, leaving higher-order structural patterns within the neighborhood less attended. Moreover, it has also been shown that [21] deep GCNs are able to learn little other than degrees and connected components, which further underscores such inability. However, higher-order local structural patterns of nodes, such as network motifs [19] do provide insightful guidance towards understanding the networks. For example, in social networks, network motifs around a node will shed light on social relationships [6] and dynamic behaviors [28].

There have been several works that utilize high order structural patterns for neighborhood aggregation, including [14]. However, in [14] only a few motifs are selected for each node for convolution, which we consider inadequate. In most cases the higher-order neighborhood of a node consists of nodes with a mixture of characteristics, leading to possibly many structural patterns within the neighborhood of a single node. Consequently, selecting a few local structural patterns would be insufficient for GCNs to fully characterize a node's neighborhood.

We illustrate such a concept using Fig. 1 which shows the neighborhoods of Manager X and Professor Y with three types of relations: family, employees, and followers. Family members know each other well, while employees form hierarchies and followers may be highly scattered and do not know each other. It can be seen that although both networks contain all three relations, a manager generally leads a larger team, while a professor is more influential

[1]In this paper we interchangeably use terms *network* and *graph*.

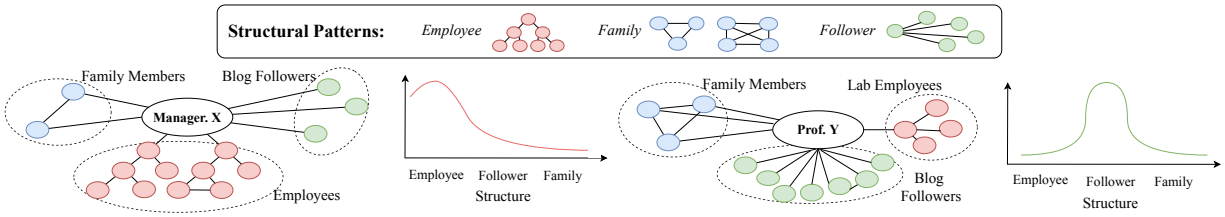


Figure 1: An example of distributional difference of structural patterns in social networks. A manager generally leads a bigger team, while a professor is more influential and is followed by more people. While both networks contain the same type of relations and structural patterns, the distributions over them are different.

and has more followers. As a result, although structural patterns like clusters, trees and stars appear in both neighborhoods, a significant difference in their distributions can be observed. Consequently, it is *the distribution of structural patterns*, rather than individuals, that is required to precisely depict a node's neighborhood.

Topic modeling is a technique in natural language processing (NLP) where neither documents nor topics are defined by individuals, but distributions of topics and words. Such probabilistic nature immediately corresponds with the *distribution of structural patterns* required to describe complex high-order neighborhoods of networks. Consequently, we similarly model nodes with *structural topics* to capture such differences in distributions of structural patterns. For example in Fig. 1, three structural topics characterized by clusters, trees and stars can be interpreted as family, employees and followers respectively, with Manager X and Prof. Y showing different distributions over the structural topics.

We highlight two advantages of topic models for modeling structural patterns. On one hand, probabilistic modeling captures the distributional differences of local structural patterns for nodes more accurately, leading to better complement over node features captured by GCNs. On the other hand, compared with previous works that directly deal with high-order structures, the structural topics are low dimensional representations of structural patterns, thus possessing less variance and leading to better efficiency.

However, several major obstacles stand in our path towards leveraging topic modeling of structural patterns to enhance GCNs:

(1) **Discovering Structural Patterns** is itself complex. Specifically, previous works [15] generally focus on pre-defined structures, which may not be flexible enough to generalize well on networks with varying nature. Also, many structural metrics require pattern matching, whose time consumption would barely be acceptable for GCNs.
(2) **Topic Modeling for Graphs** also requires elaborate effort, as graphs are relational while documents are independent samples. Consequently, adequate adaptations should be made such that the structural topics are technically sound.
(3) **Leveraging structural properties in GCNs** requires unifying node features with structural features of nodes. As they depict different aspects of a node, it would take elaborate designs of graph convolutions such that each set of features would act as a complement to the other.

In response to these challenges, in this paper we propose **Graph Structural Topic Neural Network**, abbreviated **GraphSTONE**, a GCN framework featuring topic modeling of graph structures. Specifically, we capture structural topics via anonymous walks [18] and Graph Anchor LDA. On one hand, anonymous walks are a flexible and efficient metric of depicting structural patterns, which only involve sampling instead of matching. On the other hand, we propose graph anchor LDA, a novel topic modeling algorithm that pre-selects "anchors", i.e. representative patterns, which will be emphasized during the topic modeling. By doing so, we are relieved of the overwhelming volume of structural patterns and can thus focus on relatively few key structures. As a result, concise structural topics can be generated with better efficiency.

In addition, we design multi-view graph convolutions that are able to aggregate both node features and structural topic features simultaneously, and utilize the extracted structural topics to guide the aggregation. Extensive experiments are carried out on multiple datasets, where our model illustrates competence in performance, efficiency and also interpretability.

To summarize, we make the following contributions.

(1) We propose structural topic modeling on graphs that capture distributional differences over local structural patterns on graphs, which to the best of our knowledge, is the first attempt to utilize topic modeling on graphs.
(2) We enable topic modeling on graphs through anonymous walks and a novel Graph Anchor LDA algorithm, which are both flexibie and efficient.
(3) We propose a multi-view GCN unifying both node features with structural-topic features, which we show are complementary to each other.
(4) We carry out extensive experiments on multiple datasets, where GraphSTONE outperforms its competitive baselines. In addition, we carry out visualization on synthetic datasets, where the results provide intuitive understandings of both Graph Anchor LDA and GraphSTONE.

2 RELATED WORKS

2.1 Graph Neural Networks (GNNs)

Recent years have witnessed numerous works focusing on deep architectures over graphs [7, 12, 26], among which the GCNs are those receiving the most attention. GCNs are generally based on neighborhood aggregation, where the computation is carried out by sampling nodes within a node's neighborhood before aggregation.

Although neighborhood aggregation makes GCNs equally powerful as the Weisfeiler-Lehman isomorphism test [20, 25], common neighborhood aggregations generally refer to node features only,

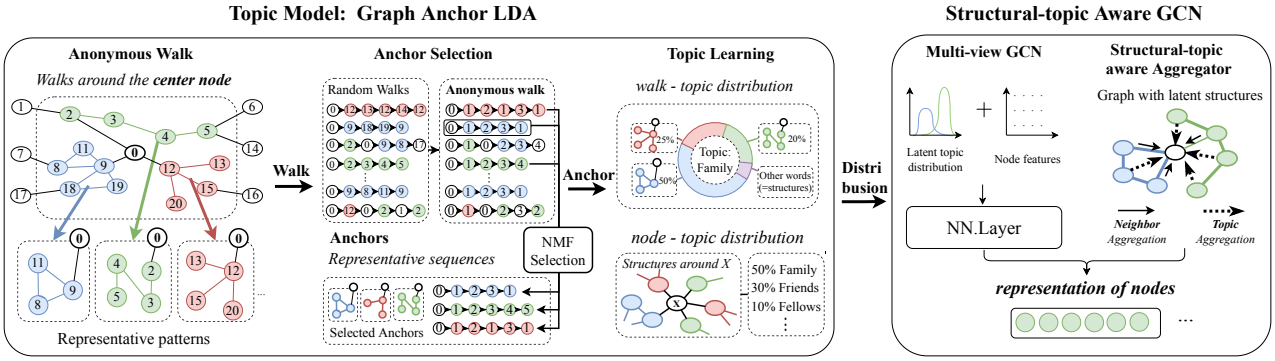


Figure 2: An overview of GraphSTONE. GraphSTONE consists of two major components: (a) Graph Anchor LDA, (b) Structural topic aware multi-view GCN.

leaving them less capable in capturing complex neighborhood structures. Such weakness is also shown in theory. For example, [17] states that GCNs should be sufficiently wide and deep to achieve subgraph detection, while [21] demonstrates that deep GCNs can learn little other than degrees and connected components.

To complement, many works have been focusing on GCNs with high-order local structural patterns. For example, [14] selects indicative motifs within a neighborhood before applying attention, which we consider to be insufficient, while our work focuses on distributions over structures rather than individual structures. [10] captures local structural patterns via anonymous walks, which can capture complex structures yet suffers from poor efficiency. By comparison, our solution using topic models is more efficient.

2.2 Modeling Graph Structures

There are many previous works on depicting graph structure properties using metrics such as graphlets and shortest paths [4, 23]. However, they commonly require pattern matching, which is hardly affordable in large, real-world networks. In addition, these models are constrained to extract pre-designed structural patterns, which are not flexible enough to depict real-world networks with different properties. More recent works including [8, 10] explore local structures on graphs based on anonymous random walks [18].

Several works in network embedding also exploit network structures to generate node representations, such as [5, 22]. However, their focuses are generally singular in that they do not refer to node features, while our model is able to combine both graph structures and node features through GCNs.

2.3 Topic Modeling

Topic modeling in NLP is a widely used technique aiming to cluster texts. Such models assign a distribution of *topics* to each document, and a distribution of words to each topic to provide low-dimensional, probabilistic descriptions of documents and words.

Latent Dirichlet Allocation (LDA) [3], a three-level generative model, embodies the most typical topic models. However, although prevalent in NLP [11, 16], LDA has hardly, if ever, been utilized in non-i.i.d. data like networks. In this work, we design a topic model on networks, where structural topics are introduced to capture distributional differences over structural patterns in networks.

3 MODEL: GRAPHSTONE

In this section we introduce our model *Graph Structural-topic Neural Network*, i.e. *GraphSTONE*. We first present the topic modeling on graphs, including general ideas and details, before presenting the structural multi-view graph convolution.

Fig. 2 gives an overview of our model GraphSTONE. Anonymous random walks are sampled for each node to depict local structures of a node. Graph Anchor LDA is then carried out on anonymous walks for each node, where we first select "anchors", i.e. indicative anonymous walks through non-negative matrix factorization, with theoretical results. After obtaining the walk-topic and node-topic distributions, we combine these structural properties with original node features through a multi-view GCN which outputs representations for each node.

3.1 Topic Modeling for Graphs

3.1.1 Anonymous Walks. We briefly introduce anonymous walks here and refer readers to [8, 18] for further details.

An anonymous walk is similar to a random walk, but with the exact identities of nodes removed. A node in an anonymous walk is represented by the first position where it appears. Fig. 2 (a) provides intuitive explanations of anonymous walks which we find appealing. For example, $w_i = (0, 9, 8, 11, 9)$ is a random walk starting from node 0, and its anonymous walk is defined as $w_i = (0, 1, 2, 3, 1)$. It is highly likely that it is generated through a triadic closure.

We present the following theorem to demonstrate the property of anonymous walks in depicting graph structures.

Theorem 1. [18] Let $B(v, r)$ be the subgraph induced by all nodes u such that $dist(v, u) \leq r$ and P_L be the distribution of anonymous walks of length L starting from v, one can reconstruct $B(v, r)$ using $(P_1, ..., P_L)$, where $L = 2(m + 1)$ and m is the number of edges in $B(v, r)$.

Theorem 1 underscores the ability of anonymous walks in capturing local structures of nodes in a highly general manner. Therefore,

we take each anonymous walk as a basic pattern for describing graph structures[2].

3.1.2 Formulation. We formulate topic modeling on graphs in our paper as follows.

Definition 1 (Topic Modeling on Graphs)**.** Given a graph $G = \langle V, E\rangle$, a set of possible anonymous walks of length l as $\mathcal{W}_l$, and the number of desired structural topics K, a topic model on graphs aims to learn the following parameters.

- A node-topic matrix $R \in \mathbb{R}^{|V|\times K}$, where a row R_i corresponds to a distribution with R_{ik} denoting the probability of node v_i belonging to the k-th structural topic.
- A walk-topic matrix $U \in \mathbb{R}^{K\times|\mathcal{W}_l|}$ where a row U_k is a distribution over $\mathcal{W}_l$ and U_{kw} denotes the probability of $w \in \mathcal{W}_l$ belonging to structural topic k.

In addition, we define the set of anonymous walks starting from v_i as D_i, with $D_i = N$ as the number of walks to sample.

The formulation is an analogy to topic modeling in NLP, where structural patterns correspond to *words*, and the sets of walks starting from each node correspond to *documents*. By making the analogy, nodes are given probabilistic depictions over their distributions of local structural patterns, and structural topics would thus consist of structural pattern distributions that are indicative towards node properties (social relations in Fig. 1, for example).

According to LDA constraints in NLP [2], we introduce Lemma 1 to show that the topic model in networks can indeed be learned.

Lemma 1. There is a polynomial-time algorithm that fits a topic model on a graph with error ϵ, if N and the length of walks l satisfy

$$\frac{N}{l} \geq O\left(\frac{b^4K^6}{\epsilon^2p^6\gamma^2|V|}\right)$$

where K is the number of topics, $|V|$ is the number of nodes. b, γ and p are parameters related to topic imbalance defined in [3], which we assume to be fixed.

Proof. We first introduce the general idea underlying the lemma. In topic models in NLP, it is generally assumed that the length of each document $|D_i|$ as well as the vocabulary $\mathcal{W}$ is fixed, while the size of the corpus $|\mathcal{D}|$ is variable. Marked differences exist in graphs, where the number of nodes $|\mathcal{D}| = |V|$ is fixed, while our anonymous walk sets and samples are variable-sized. Hence we focus on N, l instead of $|\mathcal{D}|$.

[2] gives a lower bound on the number of documents such that the topic model can be fit, namely

$$|\mathcal{D}| = |V| \geq \max\left\{O\left(\frac{\log n \cdot b^4K^6}{\epsilon^2p^6\gamma^2N}\right), O\left(\frac{\log K \cdot b^2K^4}{\gamma^2}\right)\right\},$$

where $n = |\mathcal{W}_l|$ is the vocabulary size. As the latter term is constant, we focus on the first term.

We then get the bound of N and $|\mathcal{W}_l|$, namely

$$\frac{N}{\log|\mathcal{W}_l|} \geq O\left(\frac{b^4K^6}{\epsilon^2p^6\gamma^2|V|}\right).$$

[2]Although we do not explicitly reconstruct $B(v, r)$, such theorem demonstrates the ability of anonymous walks to represent structural properties.

The number of anonymous walks increases exponentially with length of walks l [8], i.e.,

$$\log|\mathcal{W}_l| = \Theta(l).$$

Consequently we reach Lemma 1. □

3.1.3 Graph Anchor LDA. A large number of different walk sequences will be generated due to the irregularity of graphs, among which many may not be indicative as illustrated by the theory of motifs [19]. If sequences are regarded separately, we would be encountered with a huge "vocabulary", compromising the efficiency of our model, since the topic model may focus on meaningless sequences and ignore more important structural patterns.

Unfortunately, while in NLP the concept of *stopwords* is utilized to remove meaningless words, no such results exist in networks to remove such walk sequences. Consequently, we propose to select highly indicative structures first, where we call "anchors", before moving on to further topic modeling.

Specifically, we define the walk-walk co-occurrence matrix $M \in \mathbb{R}^{|\mathcal{W}_l|\times|\mathcal{W}_l|}$, with $M_{i,j} = \sum_{v_k\in V}\mathbb{I}(w_i \in D_k, w_j \in D_k)$, and adopt the method of non-negative matrix factorization (NMF) [13] to extract such anchors

$$\begin{aligned} &H, Z = \arg\min \|M - HZ\|_F^2 \\ &s.t. \quad H, Z^T \in \mathbb{R}^{|\mathcal{W}_l|\times\alpha}, H, Z \geq 0. \end{aligned} \tag{1}$$

We iteratively update H and Z, before finding the anchors through $\mathbf{A}_k = \arg\max(Z_k), k = 1, ...\alpha$, where $\mathbf{A}$ is the set of indices for anchors with $|\mathbf{A}| = \alpha$. Intuitively, by choosing the walks with largest weights, we are choosing the walks most capable of interpreting the occurrence of other walks, i.e. influential and indicative walks. We later show theoretically that the selected walks are not only indicative of walk co-occurrences but also the underlying topics.

Based on the anchors we picked, we move forward to learn the latent structural topic distribution. [1] presents a fast optimization for LDA with anchors as primary indicators and ordinary non-anchors providing auxiliary information. We get the walk-topic distribution $U \in \mathbb{R}^{K\times|\mathcal{W}_l|}$ through optimizing

$$\arg\min D_{KL}\left(Q_i \| \sum_{k\in\mathbf{A}} U_{ik}\mathrm{diag}^{-1}(Q\vec{1})Q_{A_k}\right), \tag{2}$$

where Q is the re-arranged walk co-occurrence matrix with first α rows and columns corresponding to the anchors $\mathbf{A}$, and Q_{A_k} is the row of Q for the k-th anchor.

In addition we define node-walk matrix as $Y \in \mathbb{R}^{|V|\times|\mathcal{W}_l|}$ with Y_{iw} denoting the number of occurrences of w in D_i. We then get the node-topic distribution $R \in \mathbb{R}^{|V|\times K}$ through $YU^\dagger$, where $U^\dagger$ is the pseudo-inverse of U.

3.1.4 Theoretical Analysis. We here provide a brief theoretical analysis of our Graph Anchor LDA in its ability to recover "anchors" of not only walk co-occurrences but also topics. We first formalize the notion of "anchors" via the definition of separable matrices.

Definition 2 (p-separable matrices)**.** [2] An $n \times r$ non-negative matrix C is p-separable if for each i there is some row $\pi(i)$ of C that has a single nonzero entry $C_{\pi(i),i}$ with $C_{\pi(i),i} \geq p$.

Algorithm 1 Algorithm of GraphSTONE

Require: Graph $G = (V, E, X)$, number of latent topics K
Ensure: walk-topic distribution matrix U, node-topic distribution R, node embeddings Φ with latent topic information

$M \leftarrow$ Walk co-occurences(G)
Form $\bar{M} = \{\bar{M}_1, \bar{M}_2, ..., \bar{M}_V\}$, the normalized rows of M.
$\mathbf{A} \leftarrow$ Graph Anchor LDA ($\bar{M}, K$) (Eq. 1) to find key structural indicators.
$U, R \leftarrow$ RecoverLatentTopic ($M, \mathbf{A}$) (Eq. 2)
$\Phi \leftarrow$ Structural-topic aware GCN (G, U, R, X) (Eq. 4)
return U, R, Φ

Specifically, if a walk-topic matrix U is separable, we call the walks with non-zero weights "anchors". We then present a corollary derived from [2] indicating that the non-negative matrix factorization is indeed capable of finding such anchors.

Corollary 1. Suppose the real walk-node matrix (i.e. the real walk distribution of each node) is generated via $Y = U\Lambda$, where $U\Lambda$ is the real walk-topic matrix and Λ is a matrix of coefficients, both non-negative. We define $\Sigma = \mathbb{E}[YY^T] = \mathbb{E}[U\Lambda\Lambda^T U^T]$ and $\hat{\Sigma}$ as an observation of Σ. For every $\varepsilon > 0$, there is a polynomial time algorithm that factorizes $\hat{\Sigma} \approx \hat{U}\hat{\Phi}$ such that $\|\hat{U}\hat{\Phi} - \Sigma\|_1 \leq \varepsilon$. Moreover, if U is p-separable, then the rows of $\hat{U}$ almost reconstruct the anchors up to an error of $O(\varepsilon)$.

Specifically, the walk co-occurrence matrix M serves as an estimate $\hat{\Sigma}$ of the walk covariance Σ in Corollary 1. Corollary 1 emphasizes that our Graph Anchor LDA algorithm is indeed able to select representative structural patterns within each structural topic. In reality the topics are generally non-separable, where we will empirically show the effectiveness of anchor selection.

3.1.5 Discussion. We discuss the difference between Graph Anchor LDA and community detection, both of which can be used for dimensionality reduction on graphs. While community detection focuses on dense connections, Graph Anchor LDA focuses on the distribution of local structures, which will assign similar topics to structurally similar, but not necessarily connected nodes (e.g. u and v in Fig. 3). We will experimentally show such a difference in Fig. 4 by comparing Graph Anchor LDA with MNMF [9].

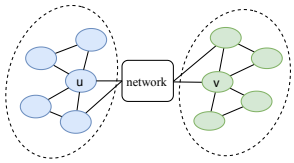


Figure 3: An example of two nodes (u and v) that are structurally similar, but belongs to two communities. Graph Anchor LDA will assign similar structural topics for u and v.

3.2 Structural-topic Aware GCN

We then introduce our design of graph convolutions which fuses structural-topic features along with node features.

3.2.1 Multi-view GCN. As the two types of features come from different domains, utilizing one to complement the other would be ideal. Inspired by the idea of boosting, we introduce two parallel GCNs dealing with them correspondingly. Specifically, let $h_{i,n}$ and $h_{i,s}$ denote the outputs of GCNs for node i learned with node features and structural topic features as inputs, we apply a nonlinear neural network layer for these two vectors.

$$h_i^{(L)} = (\mathbf{W} \cdot \mathrm{ReLU}([h_{i,n}^{(L)} \otimes h_{i,s}^{(L)}]) + \mathbf{b}) \tag{3}$$

where $h_i^{(L)}$ is the final (layer L) output vector of the multi-view GCN for node i, and $\otimes$ can be arbitrary vector operations. Here we take $\otimes$ as concatenation.

We construct the input structural topic features for node i from two parts: a) a vector indicating the distribution of anchors $\mathbf{A}$ around center node, and b) the distribution of topics R_i for node i.

3.2.2 Structural-topic Aware Aggregator. Prevalent GCNs gather all neighbors with equal contributions, while GAT, although assigning different weights on neighbors, takes only node features but not structural properties into account. Consequently, we propose to utilize the learned structural topics to guide the aggregation of neighbors. We utilize the scheme of sampling the neighbors based on the similarities of their structural topics, such that the neighborhood information can be aggregated in a way that illustrates *homophily*, i.e. similar nodes influence each other more.

$$h_i^{(k)} = \mathrm{AGGREGATE}\left(\left\{\frac{R_i^T R_j}{\sum_j R_i^T R_j} h_j^{(k-1)}, v_j \in N(v_i)\right\}\right), \tag{4}$$

where $R_i^T R_j$ is the topic similarity between node i and its neighbor j. We take identical AGGREGATE as GraphSAGE while being flexible to more sophisticated methods.

Finally, we adopt the unsupervised objective of GraphSAGE [7] for learning the output vectors.

$$\mathcal{L} = -\log[\sigma(h_i^{(L)T} h_j^{(L)}] - q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log[\sigma(h_i^{(L)T} h_n^{(L)})] \tag{5}$$

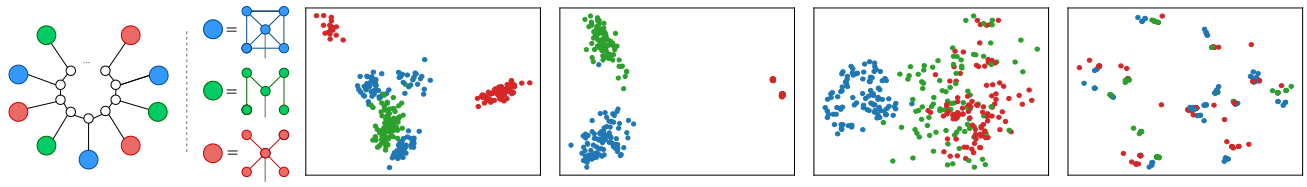
where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function, $h_i^{(L)}$ is the representation vector of v_i after the multi-view GCN, $N(v_i)$ denotes the "neighborhood" of v_i, i.e. vertices co-occuring with v_i in random walks, and $P_n(v)$ is the noise distribution for negative sampling.

As a summary, we show the pseudo code of GraphSTONE in Algorithm 1.

4 EXPERIMENTS

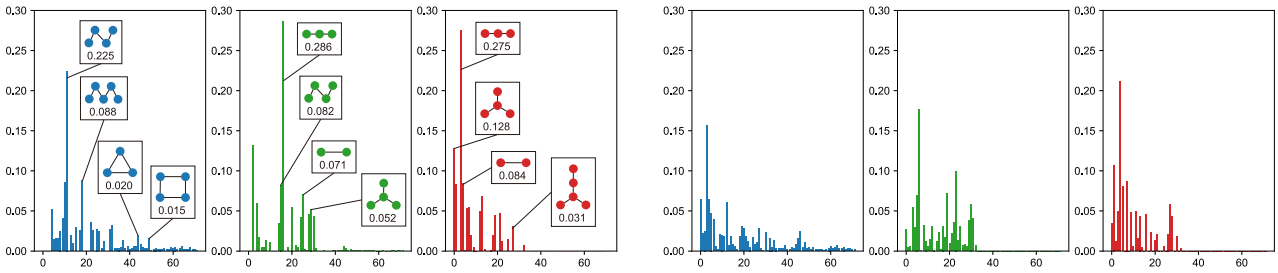
In this section, we introduce our empirical evaluations of the model GraphSTONE. We first introduce experimental settings, before results on various tasks. Specifically, our evaluation consists of:

- Quantitative evaluation, including link reconstruction and node classification.
- Qualitative evaluation, including visualization tasks that illustrate the results of Graph Anchor LDA and GraphSTONE in a straightforward manner.
- Self evaluation, including analysis on model parameters and model components.



(a) Illustration of $G(n)$ (b) Graph Anchor LDA (c) GraphSTONE (d) GraLSP (e) MNMF

Figure 4: Visualization of structural topics, and results by various models on $G(n)$. Graph Anchor LDA and GraphSTONE are able to more clearly mark the differences between local structural patterns than GraLSP and MNMF.



(a) Walk-topic distribution by Graph Anchor LDA (b) Walk-topic distribution by ordinary LDA

Figure 5: Visualization of walk-topic distributions by Graph Anchor LDA (left) and ordinary LDA (right). Graph Anchor LDA generates sharper walk-topic distributions, and amplifies indicative structural patterns within each structural topic.

4.1 Experimental Setup

We first introduce the datasets, comparison methods as well as settings for experimental evaluation.

Datasets We utilize citation networks (Cora, Pubmed [12]), social network (Aminer [27]) and protein interaction network (PPI) to evaluate our model. Dataset statistics are listed in Table 1.

Datasets	Type	$\|V\|$	$\|E\|$	# Classes
Cora	Citation	2,708	5,429	7
AMiner	Social	3,121	7,219	4
Pubmed	Citation	19,717	44,338	3
PPI	Protein	14,755	228,431	121

Table 1: Dataset statistics and properties.

Baselines We take the following novel approaches in representation learning as baselines.

- **Structure models**, focusing on structural properties of nodes. Here we choose a popular model Struc2Vec [22].
- **GNNs**, including GraphSAGE, GCN [12] and GAT [24]. We train these models using the unsupervised loss of Eqn. 5.
- **GraphSTONE (nf)**. We take the outputs of Graph Anchor LDA as inputs of GCN to directly verify how the extracted structural topics on networks contribute to better GCN modeling. We denote this variant as GraphSTONE (nf). Note that GraphSTONE (nf) does not take raw node features as inputs.

Settings We take 64-dimensional embeddings for all methods, and adopt Adam optimizer with a learning rate of 0.005. For GNNs, we take 2-layer networks with a hidden layer sized 100. For skip-gram optimization (Eqn. 5), we take $N = 100, l = 10$, window size as 5 and the number of negative sampling $q = 8$. For models involving neighborhood sampling, we take the number for sampling as 20. We leave the parameters of other baselines as default mentioned in corresponding papers. In addition, we take $K = 5$ for GraphSTONE.

We also introduce two settings for node classification tasks.

- **Transductive setting.** We allow all models access to the whole graph, i.e. all edges and node features. We apply this setting for Cora, AMiner and Pubmed and PPI.
- **Inductive setting.** The test nodes are unobserved during training. We apply this setting on PPI, where we train all GNNs on 20 graphs and directly predict on appointed two test graphs, same as in [7].

4.2 Proof-of-concept Visualization

As we propose a new problem – topic modeling on graphs, we first show a result on visualization to intuitively explain its results. We carry out visualization on a synthetic dataset as a simple proof-of-concept. We design such dataset $G(n)$ using nodes with three basic structures: one dense cluster, one T-shaped and one star, and then connect n such structures interleavingly on a ring. We show an illustration of $G(n)$ in Fig. 4(a). For a clear illustration, we replace each structure with a colored dot. Apparently, the nodes in $G(n)$

Input	Model	Cora		AMiner		Pubmed	
		AUC	Recall@0.5	AUC	Recall@0.5	AUC	Recall@0.5
No features	Struc2Vec	54.29	54.38	47.55	47.63	53.14	53.14
	GraLSP	66.28	66.38	65.40	65.50	57.62	57.63
	GCN	74.60	74.71	71.98	72.07	59.20	59.22
	GraphSTONE (nf)	**92.44**	**92.56**	**89.87**	**89.91**	**87.47**	**87.48**
Features	GCN	94.14	94.26	94.47	94.55	92.23	92.25
	GAT	94.66	94.78	95.24	95.34	92.36	92.38
	GraLSP	94.39	94.51	94.85	94.89	90.83	90.84
	GraphSAGE	95.30	95.42	94.92	95.02	91.52	91.54
	GraphSTONE	**96.37**	**96.70**	**95.94**	**96.06**	**94.25**	**94.27**

Table 2: Results of link reconstruction on different datasets.

Input	Model	Cora				AMiner				Pubmed				PPI			
		Macro-f1		Micro-f1		Macro-f1		Micro-f1		Macro-f1		Micro-f1		Macro-f1		Micro-f1	
		30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%
No features	Struc2Vec	17.55	18.92	29.07	31.34	23.17	21.80	36.11	38.44	31.29	31.31	41.50	41.49	12.89	13.53	40.49	40.74
	GraLSP	58.86	63.62	60.88	64.45	43.19	43.03	45.85	45.92	38.89	38.84	45.88	46.01	10.19	10.72	37.65	37.88
	GCN	11.65	11.94	32.30	32.83	14.86	16.81	41.24	42.51	35.07	36.51	46.56	47.83	8.75	9.08	36.70	37.46
	GraphSTONE (nf)	**70.75**	**71.83**	**72.73**	**73.52**	**57.11**	**56.70**	**58.21**	**58.91**	**56.87**	**58.88**	**60.47**	**60.69**	10.28	11.20	38.93	38.96
Features	GCN	79.84	81.09	80.97	81.94	65.02	67.33	64.89	66.72	76.93	77.21	76.42	77.49	12.57	12.62	40.40	40.44
	GAT	79.33	82.08	80.41	83.43	68.76	69.10	67.92	68.16	76.94	76.92	77.64	77.82	11.91	11.97	39.92	40.10
	GraLSP	82.43	83.47	83.67	84.61	68.82	70.15	69.12	69.73	**81.21**	**81.38**	**81.43**	**81.52**	11.34	11.89	39.55	39.80
	GraphSAGE	80.52	81.90	82.13	83.17	67.40	68.32	66.59	67.54	76.61	77.24	77.36	77.84	11.81	12.41	39.80	40.08
	GraphSTONE	**83.01**	**84.62**	**83.88**	**85.60**	**69.37**	**71.16**	**69.51**	**69.93**	78.61	78.87	79.53	81.03	**15.55**	**15.91**	**43.60**	**43.64**

Table 3: Macro-f1 and Micro-f1 scores of transductive node classification.

possess three distinct structural properties (with the ring excluded), which can be regarded as three structural topics. We then obtain representation vectors from Graph Anchor LDA, GraphSTONE, GraLSP and MNMF, and plot them on a 2d plane.

As shown in Fig. 4(b) and 4(c), both Graph Anchor LDA and GraphSTONE cluster the three topics of nodes clearly, which shows that topic modeling does capture indicative structural patterns and marks the difference between neighborhoods of nodes. The results achieved are made more astonishing as GraLSP (Fig. 4(d)) fails to cluster nodes in a satisfactory manner, showing that the probabilistic modeling of topic models is indeed critical. In addition, it can be shown that MNMF, a community-aware network embedding algorithm, largely ignores the structural similarity between nodes and fails to cluster nodes appropriately.

Moreover, we visualize the walk-topic distributions generated by Graph Anchor LDA in Fig. 5(a), and compare them with those by ordinary LDA, where x-axis denotes indices of anonymous walks and y-axis denotes the corresponding probability. It can be shown that the anchors selected by our Graph Anchor LDA are not only indicative of "topics", but are also in accordance with the actual graph structures. Moreover, the walk-topic distributions generated by Graph Anchor LDA are indeed sharper than those by ordinary LDA, underscoring the need for selecting anchors. We hence conclude that Graph Anchor LDA is highly interpretable in summarizing graph structures and would potentially lead to a better understanding of the impact of graph structures in information networks.

Model	Macro-f1	Micro-f1
Struc2Vec	-	-
GCN	12.15	40.85
GAT	12.31	39.76
GraLSP	12.59	40.81
GraphSAGE	11.92	40.05
GraphSTONE	**18.14**	**46.02**

Table 4: Inductive node classification results on PPI.

4.3 Quantitative Evaluations

We conduct experiments on link reconstruction and node classification to evaluate GraphSTONE quantitatively.

4.3.1 Link Reconstruction. We conduct link reconstruction using Cora, AMiner, and Pubmed, but not PPI because PPI contains 20 disjoint graphs. We sample a proportion of edges from the initial network, which are used as positive samples, along with an identical number of random negative edges. It should be noted that the positive edges *are not* removed from training. We take the inner product of embedding vectors as the score for each sample, which is used for ranking. Samples with scores among top 50% are predicted as "positive" edges. We report AUC and Recall as metrics.

The results are shown in Table 2. It can be showed that by combining structural features with original node features, our model is

able to generate representations that more accurately summarizes the original network. It is even more remarkable that based on structural features alone, GraphSTONE(nf) is able to significantly outperform the counterparts without the help of raw node features. Although Struc2Vec considering some global network structures, its poor performance in link reconstruction reveals its shortage in capturing the simplest structure – edges. By comparison, GraphSTONE and GraphSTONE (nf) achieve a more elaborate balance among simple structures, like edges, and complex structures.

4.3.2 Node Classification. We then carry out node classification according to the corresponding setting of each dataset. For the transductive setting, different fractions of nodes are sampled randomly for testing, leaving the rest for training. For the inductive setting, we take embedding vectors of nodes from the 20 training graphs for training, and those from the two test graphs for testing. We use Logistic Regression as the classifier and take the macro and micro F1 scores for evaluation in both settings. The results are averaged over 10 independent runs.

Transductive classification Results of transductive node classification are shown in Table 3. It is generally shown that compared to competitive baselines, GraphSTONE is able to achieve satisfactory results. In addition, GraphSTONE (nf) significantly outperforms GCN without feature input, hence verifying that the structural topic information extracted by topic modeling does contribute significantly to more accurate representations of nodes.

Inductive classification Results of inductive classification are shown in Table 4. It should be noted that algorithms that do not refer to node features are not inductive and therefore not listed. It is shown that GraphSTONE also outperforms competitive baselines in inductive setting, which shows that the structural topic information extracted generalizes well across networks of the same type.

Model	Macro-f1	Micro-f1
GCN	81.46	82.61
GraphSTONE (node)	82.33	83.55
GraphSTONE (degree)	81.29	82.36
GraphSTONE (rw)	82.35	82.85
GraphSTONE (no anchors)	83.12	83.39
GCN-concat	81.43	82.62
GraphSTONE	**84.21**	**85.13**

Table 5: Ablation studies of GraphSTONE.

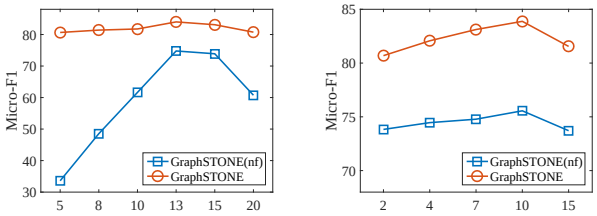
4.4 Model Analysis

We carry out tests on our model itself, including ablation studies and efficiency. Unless specified, we use node classification to reflect the performance of the model.

4.4.1 Ablation studies. We carry out thorough ablation studies and show the results in Table 5. Specifically, we focus on the two parts of GraphSTONE, topic modeling and GCN correspondingly.

First, we delve deeper into the design of our Graph Anchor LDA by comparing it with several variants. On one hand, we vary the analogy of "words" by substituting anonymous walks with three other units of structures: individual nodes, distribution of node degrees in D_i and raw random walks without anonymity. We denote the three variants as **GraphSTONE (node)**, **GraphSTONE (degree)** and **GraphSTONE (rw)**. On the other hand, we verify the contribution of "anchors" by comparing with GraphSTONE without anchors, referred to as **GraphSTONE (no anchors)**. It can be shown that GraphSTONE (node), GraphSTONE (degree) and GraphSTONE (rw) all fail to outperform GraphSTONE, probably because individual nodes and degrees only depict one-hop structures, and raw random walks are too sparse to observe co-appearance. By comparison, GraphSTONE (no anchors) performs better than the previous three due to its use of anonymous walks, but worse than GraphSTONE, which endorses the contribution of anchors in topic modeling and corresponds with Fig. 5(a).

In addition, we validate the effectiveness of our structural multi-view GCN by comparing GraphSTONE with **GCN-concat**, a GCN with its input node features concatenated with structural topic features. It can be shown that with our designs of multi-view graph convolution, GraphSTONE enables node and structural features to complement each other, while simple aggregation fails to do so.



(a) Walk length l (b) Number of topics K

Figure 6: Parameter analysis of GraphSTONE.

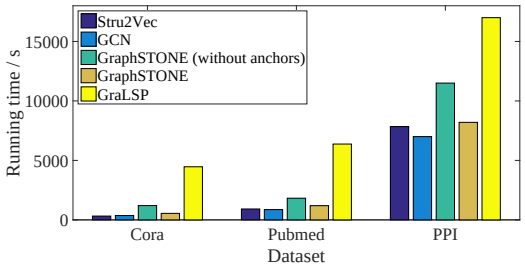
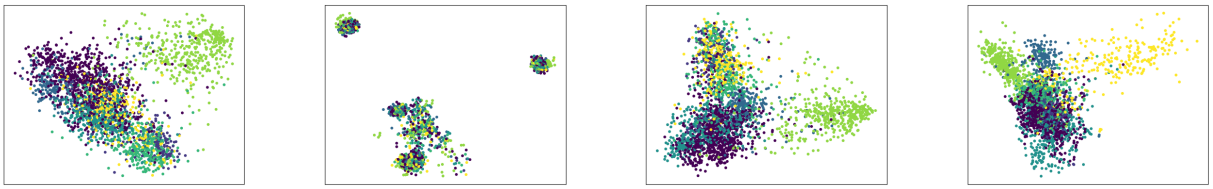


Figure 7: Running time on different datasets.

4.4.2 Parameter Analysis. We analyze two parameters in our model, namely the length of walks l and the number of topics K and plot the performances in Fig. 6. As can be shown, performances peak at certain choices for both parameters, while unduly set values for both parameters, either too big or too small, will compromise the model performance. By comparison, our model is less sensitive to the number of topics K than the length of walks l, with larger l indicating more complex local structures to consider.

4.4.3 Efficiency. We compared the efficiency of GraphSTONE with several baselines: Struc2Vec, GCN and GraLSP, as we anticipate the



(a) GraphSAGE (b) Struc2Vec (c) GraLSP (d) GraphSTONE

Figure 8: Visualization of representation vectors from various algorithms in 2D space.

introduction of "anchors" would lead to better efficiency. We train all models and report the time needed for convergence on a single machine equipped with a GPU with 12GB RAM.

Results are presented in Fig.7. It can be shown that although GraLSP captures structural patterns and is indeed competitive, it suffers from poor efficiency with a massive number of parameters, leaving it not suitable for large networks. By comparison, GraphSTONE is highly scalable to large networks without compromising performance. It is even more stunning that GraphSTONE barely takes more time than GCN, further endorsing its efficiency.

4.5 Visualization

We finally carry out visualization on real-world datasets to qualitatively evaluate our model. We learn the representation vectors on Cora, which are then reduced to 2-dimensional vectors using PCA. We select three representative models: Struc2Vec (structure models), GraphSAGE (GNNs) and GraLSP (structure GNNs), along with our model to compare.

The results are shown in Fig.8, where different colors correspond to 7 labels in Cora. It is observed that GraphSTONE clusters nodes in a compact and separable manner, especially on certain colors (yellow, green and blue), compared with other methods.

5 CONCLUSION

In this paper, we present a GCN framework that captures local structural patterns for graph convolution, called GraphSTONE. To the best of our knowledge, it is the first attempt on topic models on graphs and GCNs. Specifically, we observe that the distributions, rather than individuals of local structural patterns are indicative towards node properties in networks, while current GCNs are scarcely capable of modeling. We then utilize topic modeling, specifically Graph Anchor LDA to capture the distributional differences over local structural patterns, and multi-view GCNs to incorporate such properties. We demonstrate that GraphSTONE is competitive, efficient and interpretable through multiple experiments.

For future work, we seek to extend the topic modeling on networks towards featurized structural topics,f which we hope will provide more inclusive and precise depictions over networks.

REFERENCES

[1] Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*. 280–288.

[2] Sanjeev Arora, Rong Ge, and Ankur Moitra. 2012. Learning topic models–going beyond SVD. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 1–10.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Karsten M Borgwardt and Hans-Peter Kriegel. 2005. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*. IEEE, 8–pp.

[5] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2018. Learning structural node embeddings via diffusion wavelets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1320–1329.

[6] Mark S Granovetter. 1977. The strength of weak ties. In *Social networks*. Elsevier, 347–367.

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

[8] Sergey Ivanov and Evgeny Burnaev. 2018. Anonymous Walk Embeddings. In *International Conference on Machine Learning*. 2191–2200.

[9] Di Jin, Xinxin You, Weihao Li, Dongxiao He, Peng Cui, Françoise Fogelman-Soulié, and Tanmoy Chakraborty. 2019. Incorporating network embedding into markov random field for better community detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 160–167.

[10] Yilun Jin, Guojie Song, and Chuan Shi. 2020. GraLSP: Graph Neural Networks with Local Structural Patterns. *AAAI* (2020).

[11] Noriaki Kawamae. 2019. Topic Structure-Aware Neural Language Model: Unified language model that maintains word and topic ordering by their embedded representations. In *The World Wide Web Conference*. ACM, 2900–2906.

[12] Thomas Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference of Learning Representations*.

[13] Daniel D Lee and H Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (1999), 788.

[14] John Boaz Lee, Ryan A Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. 2019. Graph Convolutional Networks with Motif-based Attention. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 499–508.

[15] Lin Liu, Lin Tang, Libo He, Shaowen Yao, and Wei Zhou. 2017. Predicting protein function via multi-label supervised topic model on gene ontology. *Biotechnology & Biotechnological Equipment* 31, 3 (2017), 630–638.

[16] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[17] Andreas Loukas. 2020. What graph neural networks cannot learn: depth vs width. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1l2bp4YwS

[18] Silvio Micali and Zeyuan Allen Zhu. 2016. Reconstructing markov processes from independent and anonymous experiments. *Discrete Applied Mathematics* 200 (2016), 108–122.

[19] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 5594 (2002), 824–827.

[20] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4602–4609.

[21] Kenta Oono and Taiji Suzuki. 2020. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1ldO2EFPr

[22] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 385–394.

[23] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. 2009. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*. 488–495.

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

[25] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[26] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 974–983.

[27] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. 2019. DANE: Domain Adaptive Network Embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

[28] 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*.