Graph Clustering with Embedding Propagation

Carl Yang
Emory University
Atlanta, USA
j.carlyang@emory.edu

Liyuan Liu
University of Illinois, Urbana Champaign
Urbana, USA
ll2@illinois.edu

Mengxiong Liu
Carnegie Mellon University
Pittsburgh, USA
mengxiol@andrew.cmu.edu

Zongyi Wang
Google Inc.
Mountain View, USA
zongyiwang@google.com

Chao Zhang
Georgia Institute of Technology
Atlanta, USA
chaozhang@gatech.edu

Jiawei Han
University of Illinois, Urbana Champaign
Urbana, USA
hanj@illinois.edu

***Abstract*—In the past decade, the amount of attributed network data has skyrocketed, and the problem of identifying their underlying group structures has received significant attention. By leveraging both attribute and link information, recent state-of-the-art network clustering methods have achieved significant improvements on relatively clean datasets. However, the noisy nature of real-world attributed networks has long been overlooked, which leads to degraded performance facing missing or inaccurate attributes and links. In this work, we overcome such weaknesses by marrying the strengths of clustering and embedding on attributed networks. Specifically, we propose GRACE (*GRAph Clustering with Embedding propagation*), to simultaneously learn network representations and identify network clusters in an end-to-end manner. It employs deep denoise autoencoders to generate robust network embeddings from node attributes, propagates the embeddings in the network to capture node interactions, and detects clusters based on the stable state of embedding propagation. To provide more insight, we further analyze GRACE in a theoretical manner and find its underlying connections with two canonical approaches for network modeling. Extensive experiments on six real-world attributed networks demonstrate the superiority of GRACE over various baselines from the state-of-the-art. Remarkably, GRACE improves the averaged performance of the strongest baseline from 0.43 to 0.52, yielding a 21% relative improvement. Controlled experiments and case studies further verify our intuitions and demonstrate the ability of GRACE to handle noisy information in real-world attributed networks.**

***Index Terms*—network clustering, representation learning, influence propagation**

I. Introduction

Nowadays, attributed networks have been widely leveraged as a ubiquitous model for real-world interactive systems, ranging from social and academic datasets, to biochemical pathways and the Web [1]–[5]. Particularly, network clustering has been shown important to various relevant tasks including the discovery of functionally related objects [6], the study of interactions among modules [7], the inference of missing attributes [8], and the prediction of unobserved connections [9], [10].

Typically, traditional clustering algorithms solely focus on analyzing either the topological structures of networks [11]–[13], or the contents of nodes [14]–[16]. Recent advances on attributed network clustering have provided clear evidence on the importance of integrating both kinds of information [8], [17]–[22]. Accordingly, in this work, we focus on the clustering of attributed networks, which is an urgent and practical problem with much potential. However, properly integrating attributes and links for network clustering is a challenging task.

On one hand, while node attributes can be leveraged to alleviate the sparsity of linkage data, attributes themselves in a network can be very sparse and noisy. As shown in Table I, our investigation into six real-world networks shows that many social network users leave important attributes as blank in their profiles, and many papers in the citation networks only cover a small portion of keywords in the entire vocabulary. Meanwhile, users may fill in random fake information, and papers may use different terminology even for the same topics. Therefore, traditional models assuming high-quality simple attributes can become ineffective in capturing the deep and complex semantics hidden in the sparse noisy attributes in real-world attributed networks [8], [17]–[22].

On the other hand, although there has been substantial work on network embedding to capture the link structures, most of them do not consider network attributes at all [23]–[32], whereas others either only employ shallow models that cannot handle complex noisy attributes [2], [3], [33]–[35], or leverage heavy neural networks in a supervised way where labeled data are indispensable [1], [36]–[39]. More importantly, all existing network embedding algorithms including [2], [3], [28]–[30], [32], [34], [35] do not jointly learn embedding and clustering, but rather separate the two processes, so the representations they produce are not optimized towards the particular task of network clustering.

In this work, we propose GRACE (*GRAph Clustering with Embedding propagation*), for network clustering based on both node attributes and link structures. For the first time, deep embedding and unsupervised clustering are jointly trained on attributed networks in an end-to-end fashion. By combining a powerful attribute embedding module based on deep denoise autoencoders [40], [41] and a principled embedding propagation module based on the influence propagation theory

[20], [21], [42], we compute the novel *dynamic embedding* which effectively reveals the network clustering structures while being robust to noisy attributes and links. Moreover, by leveraging a self-training clustering module [14], [43], the embedding and clustering results are refined in a closed loop to further enhance each other.

We summarize the contributions in this work as follows:

1) Based on denoising autoencoder and influence propagation techniques, we design an embedding propagation process that jointly performs network clustering and embedding. In the process, both node attributes and link structures are leveraged to generate high-quality robust node representations tailored for network clustering.
2) To provide more insight into network clusters and corroborate our model design, we theoretically analyze the principle of influence propagation. We demonstrate that our leverage of embedding propagation leads to a mathematical model essentially related to two canonical network modeling approaches. However, our model is more powerful and efficient without sampling or supervision.
3) We conduct extensive experiments on six standard public real-world datasets including both social networks and citation networks with attributes. The performance of GRACE is superior compared with various state-of-the-art community detection and network embedding algorithms. A series of controlled experiments and case studies are further conducted to analyze the quality of learned representations towards network clustering.

II. Related Work

Network Community Detection. Most traditional network community detection algorithms are unsupervised, based on either the link density assumption alone or jointly with the attribute homogeneity assumption. The former assumes that communities should be constructed by densely connected nodes, and the latter requires that nodes in communities should also share similar attributes. Among various algorithms, the most widely used ones are based on heuristic metrics such as modularity [11], [12] and maximal clique [46]. They fundamentally leverage the link density assumption. As for attribute homogeneity, some firstly augment the network with attribute links before computing the clustering [19], [47], while others attempt to find a network partition that yields the minimum encoding cost based on information theory [48], [49]. [18], [22], [50] leverage both link density and attribute homogeneity by assuming communities to be formed by densely connected homogeneous nodes. [20] leverages content propagation on networks to detect communities by firstly propagating the attributes among neighboring nodes. These shallow models are efficient even with large networks, but are still incapable of working with sparse noisy attributes.

As a novel treatment to high-dimensional sparse attributes with noise in real-world networks, in this work, for the first time, the power of deep embedding is combined with the interpretation of network clustering as a consequence of embedding propagation, to yield high-quality embeddings optimized for network clustering.

Network Node Embedding. The objective of network embedding is to find a compact node representation that captures the proximities among nodes on the networks [23]–[25], [51]–[53]. The propagation of embeddings has been studied on networks, but existing works do not jointly consider the embedding of networks jointly with a clustering objective [54]–[57] Several recent works have studied the embedding of networks jointly with clustering [58], [59], but they do not consider attributes in networks. On the other hand, embedding algorithms developed towards attributed networks mainly employ shallow neural networks for attribute modeling and do not jointly learn network clustering [1]–[3], [33]–[35], [60]–[62], so the representations they produce are less robust to noises in real-world networks and are not optimized towards the clustering objective.

To deeply integrate node attributes and link structures, deep neural networks such as CNN [63], [64] and RNN [65], [66] have also been adopted. For instance, to leverage the CNN model, [37] uses graph labelling to select an ordering of nodes, upon which fixed-sized receptive fields can then be computed, [67], [68] utilize spectral filtering to emulate receptive fields in the Fourier domain, and [38], [69], [70] treat convolution as neighborhood matching in the spatial domain. Meanwhile, RNN has been utilized to embed sequences of nodes such as information cascade paths [39], [71], [72]. These models are often deep and powerful in exploring high-dimensional noisy attributes as well as complex network structures, but they usually assume a supervised or semi-supervised learning scenario, and their performance largely depends on the amount of labeled data.

In this work, we aim at jointly learning embedding and clustering on attributed networks to allow the mutual enhancement between them in a fully unsupervised fashion. To the best of our knowledge, this is the first work that combines representation learning and clustering on attributed networks.

III. Overview

The goal of this work is unsupervised clustering on attributed networks. We summarize the main challenges of this task into the following three aspects.

- Node attributes are high-dimensional, sparse and noisy.
- Node interactions are also noisy and hard to integrate.
- No supervision is available to guide exploration.

To deal with all challenges above, we develop GRACE (*GRAph Clustering with Embedding propagation*), which inherently combines deep embedding with influence propagation to integrate network node attributes and link structures. A self-training clustering framework is adopted to further improve the performance.

A. Overall Framework

Input. We are given a network modeled by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{C}\}$, where $\mathcal{V}$ is the set of n nodes (*e.g.*, users on social networks,

Type	Dataset	#Attr. dim.	Attr. sparsity	#Nodes	#Links	#Clusters
Social Networks	Facebook [22]	1,283	93.49%	4,039	88,234	193
	Gplus [22]	15,907	98.85%	107,614	13,673,453	468
	Twitter [22]	216,839	96.73%	81,306	1,768,149	4,065
Citation Networks	Cora [44]	1,433	98.73%	2,708	5,429	7
	Citeseer [44]	3,703	99.14%	3,312	4,715	6
	PubMed [45]	500	89.98%	19,717	44,338	3

TABLE I: **Summary of statistics of six real-world attributed networks.**

papers in citation networks, *etc.*), and $\mathcal{E}$ is the set of m observed links among the nodes in $\mathcal{V}$. $\mathcal{A}$ is the set of observed node attributes associated with $\mathcal{V}$, where each $\mathbf{a}_i$ is the set of κ features on node v_i. $\mathcal{C}$ is a set of ground-truth cluster memberships, where $\mathbf{c}_{ik} = 1$ means node v_i is in the kth ground-truth cluster. For unsupervised graph clustering, $\mathcal{C}$ is only used for evaluation.

Output. Our model jointly learns a deep embedding $\tilde{\mathcal{X}}$ and a soft clustering $\mathcal{Q}$ on $\mathcal{V}$. Each $\mathbf{x}_i \in \tilde{\mathcal{X}}$ is a distributed representation of node v_i, capturing a prominent nonlinear combination of its original features $\mathbf{a}_i$ under the consideration of v_i's contexts in its neighborhood $\mathcal{N}_i$ on the network. The definition of $\mathcal{N}_i$ will be introduced in the following subsection (Section IV.3), together with our specific treatment to $\mathcal{X}$ based on the principle of influence propagation on networks. Each $\mathbf{q}_i \in \mathcal{Q}$ is a K-dimensional probability distribution, where K is the number of clusters to be detected and $\sum_k q_{ik} = 1, \forall i$, characterizing the probability of node v_i belonging to the kth cluster.

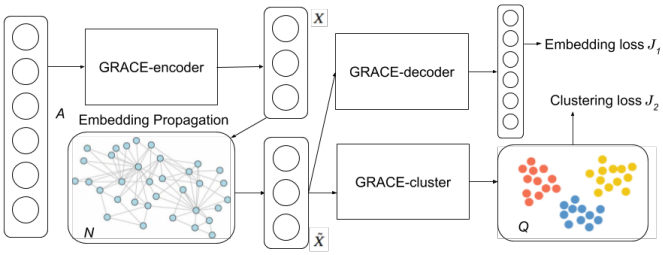


Fig. 1: The overall architecture of GRACE.

Architecture. Figure 1 illustrates the overall architecture of our GRACE framework. We take the input $\mathcal{A}$ for all nodes $\mathcal{V}$ and compute their nonlinear deep embedding $\mathcal{X}$. To integrate network attributes and links for robust representation learning, we propagate $\mathcal{X}$ in the network based on the structures of neighborhoods $\mathcal{N}$ to generate the dynamic network embedding $\tilde{\mathcal{X}}$, which is used to reconstruct the original node attributes with an embedding loss $\mathcal{J}_1$. At the meantime, a soft clustering loss $\mathcal{J}_2$ with a self-training mechanism is optimized based on $\tilde{\mathcal{X}}$ to produce high-quality graph clustering results $\mathcal{Q}$. Therefore, we have the overall loss function

$$\mathcal{J} = \mathcal{J}_1 + \lambda \mathcal{J}_2, \tag{1}$$

where λ is a weighting parameter. By minimizing Eq. 1, we can jointly optimize the deep embedding and soft clustering modules and allow them to mutually enhance each other in a closed loop.

IV. GRACE

In this section, we introduce the details of different modules in GRACE, and discuss how to jointly learn network clustering and embedding in an end-to-end manner.

A. Deep Embedding of Sparse Noisy Attributes

Table I illustrates the nature, dimensionality and sparsity of attributes in several real-world network datasets. Such high dimensionality and sparsity are not surprising, because users are known to be reluctant in filling out many attributes and papers only cover their limited vocabularies. Moreover, users are free to fill in fake attributes and papers can contain different terminologies, which further make attributes in networks noisy and inaccurate.

In our situation, it is desirable to capture attributes precisely, because they often become the signatures of network clusters. For example, in a football fans' club, a popular player identified by the attributes of his/her tweets is likely to be the center of a community, surrounded by fans posting their semantically related tweets.

However, we note that existing network embedding algorithms are ineffective with sparse noisy attributes because they usually start from preserving the link structures among non-attributed nodes [23]–[25], and then incorporate attributes as augmented nodes [33], text feature matrices [60] or bag-of-word vectors [36]. In such ways, attributes are shallowly modeled as auxiliary information, and the deep semantics within them can not be fully explored.

To deal with such a challenge, we get inspired by recent successes in deep learning for feature composition [40], [41]. Specifically, we are interested in leveraging deep embedding to discover the latent representations of node attributes in networks, which, in an ideal case, are low-dimensional, dense and robust to noise. To this end, we employ deep denoise autoencoder (DAE) [40], [41], which has been proven advantageous in capturing the intrinsic features within high-dimensional sparse noisy inputs in an unsupervised fashion.

To leverage the power of DAE, given a node v_i's original feature vector $\mathbf{a}_i$, we firstly apply a GRACE-encoder, which consists of multiple layers of fully connected feedforward neural networks with Rectified Linear Unit (ReLU) activations. The neural networks are in decreasing sizes and after them we get a Z-dimensional latent representation $\mathbf{x}_i$ as

$$\mathbf{x}_i = \mathbf{f}_e^H(\dots \mathbf{f}_e^2(\mathbf{f}_e^1(\mathbf{a}_i))\dots), \text{where} \tag{2}$$

$$\mathbf{f}_e^h(\mathbf{x}) = \text{ReLU}(\mathbf{W}_e^h \text{Dropout}(\mathbf{x}) + \mathbf{b}_e^h). \quad (3)$$

H is the number of hidden layers in GRACE-encoder. Θ_e is the set of parameters in the H encoder layers.

To ensure that $\mathbf{x}_i$ captures the important information in $\mathbf{a}_i$, we compute the reconstruction $\tilde{\mathbf{a}}_i$ of $\mathbf{a}_i$ through stacking a GRACE-decoder and apply random dropout to each of both encoding and decoding layers [73]. The GRACE-decoder also consists of multiple layers of fully connected feedforward neural networks with ReLU activations. The sizes of neural networks are in an increasing order, exactly the opposite as in GRACE-encoder. So we have

$$\tilde{\mathbf{a}}_i = \mathbf{f}_d^H(\dots \mathbf{f}_d^2(\mathbf{f}_d^1(\mathbf{x}_i))\dots), \text{where} \quad (4)$$

$$\mathbf{f}_d^h(\mathbf{x}) = \text{ReLU}(\mathbf{W}_d^h \text{Dropout}(\mathbf{x}) + \mathbf{b}_d^h). \quad (5)$$

The number of hidden layers in GRACE-decoder is also H. Θ_d is the set of parameters in the H decoder layers.

After decoding, an attribute reconstruction loss is

$$\mathcal{J}_1 = \sum_{i=1}^{n} l(\mathbf{a}_i, \tilde{\mathbf{a}}_i), \quad (6)$$

which is a summation over all nodes in $\mathcal{V}$. Depending on the nature of the attributes in the network, l can be implemented either as a cross entropy (for binary features, such as bag-of-word) or a mean squared error (for continuous features, such as TF-IDF scores [74]).

B. Graph Clustering with Embedding Propagation

The major difference between network clustering and general clustering lies in the availability of structural information, such as users' friendships on social platforms and papers' citations in publication datasets. Such links, while providing crucial information not necessarily captured by node attributes, are hardly leveraged by existing content embedding frameworks.

To reliably find clusters (or communities) on networks, we study the nature of clusters and model them under the consideration of network dynamics– nodes on networks are constantly interacting with each other, sending and receiving influences among themselves. Inspired by [8], [20], [75], we assume that clustering is a consequence of such influence propagation. Accordingly, they should be modeled based on the consistency of propagated influences on the network when they reach a stable state. Unlike traditional influence propagation studies that model a single influence factor that corresponds to node activations [76], [77], we assume Z-dimensional latent factors propagated on the network, which jointly influence the original K-dimensional node attributes.

To model the network dynamics regarding influence propagation among nodes, we compute the *dynamic embedding* $\tilde{\mathcal{X}}$ of nodes $\mathcal{V}$ by propagating the attribute embedding $\mathcal{X}$ in the network according to the link structures. Considering the standard influence propagation model [78], we use an adjacency matrix W to record the link structures in $\mathcal{G}$. For simplicity, we assume 0-1 weights and undirected links, while the model generalizes trivially to weighted directed graphs. To ensure that every node can receive influence from at least one node, we also assume every node can propagate to itself. So we have

$$w_{ij} = \begin{cases} 1, & \text{if } e_{ij} \in \mathcal{E}, \text{ or } e_{ji} \in \mathcal{E}, \text{ or } i = j, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Given W, we define D as a diagonal matrix with $d_{ii} = \sum_{j=1}^{n} w_{ij}, \forall i \in [1, n]$. So we have $T = D^{-1}W$ as the transition matrix on $\mathcal{G}$. Then $X^1 = TX$ is the propagated embedding through one step. With $\mathcal{N}_i^b$ defined as the neighborhood of v_i with size b ($\forall b > 0$), which includes all nodes that are no more than b steps from v_i on the network, $\mathbf{x}_i^b$ corresponds to the embedding of v_i under the consideration of v_i's network contexts in $\mathcal{N}_i^b$.

As we assume, the formation of clusters requires the propagated embedding to reach a stable state. It corresponds to infinite steps of propagation, or stationary as in random walk terminology. Therefore, we consider a stochastic description about the probability of a node influencing another with a linear approximation of influence propagation. Specifically, we use R to denote the stationary propagation matrix, where we denote the probability that the influence of node v_i is propagated to node v_j as r_{ij}, which satisfies

$$\forall i \neq j \in [1, n]: \; r_{ii} = \beta + \alpha \sum_{e_{lj} \in \mathcal{E}} r_{il} t_{li}, \; r_{ij} = \alpha \sum_{e_{lj} \in \mathcal{E}} r_{il} t_{lj}, \quad (8)$$

where $t_{ij} \in T$ is the transition probability from v_i to v_j, $\beta > 0$ is a constant corresponding to the probability of a node influencing itself, α is the damping coefficient of the propagation process.

According to Eq. 8, we recompute R as

$$R = \beta(I - \alpha T)^{-1} = \beta(I - \alpha D^{-1}W)^{-1}. \quad (9)$$

Subsequently, we have

$$\tilde{X} = X^{inf} = RX = \beta(I - \alpha D^{-1}W)^{-1}X, \quad (10)$$

where β can be simply removed because it is the same for every node and does not affect the clustering results.

Describing network clusters with stable influence propagation, we integrate the leverage of link structures into GRACE by substituting the deep attribute embedding X with the dynamic (*i.e.*, propagated) embedding $\tilde{X}$ before both decoding and clustering computation. By doing this, we require the embeddings of two nodes to be close not only because they share similar important attributes, but also because they send/receive similar influence to/from their neighboring nodes in the network.

C. Jointly Learning Clustering and Embedding

The final challenge of clustering on graphs lies in the lack of labeled data. Most successful deep learning frameworks on image or language processing are usually composed of an unsupervised pre-training step and a supervised fine-tuning

step so that the intrinsic structures of data and supervision can be combined to automatically capture meaningful visual or semantic patterns. However, graph clustering is naturally unsupervised, which means although our deep neural networks can be powerful in exploring various patterns, they can not receive feedback from the environment and can never explicitly know what patterns they find are useful.

To address this challenge, we get inspired by recent progress on self-training neural networks. The idea is to let the model iteratively make predictions and learn from its own high confidence predictions, which in turn helps improve the low confidence predictions [43]. Accordingly, we jointly train clustering and embedding by leveraging the common strategy of minimizing the KL divergence between a soft clustering distribution $\mathcal{Q}$ and an auxiliary target distribution $\mathcal{P}$ [14]–[16], which was originally used to cluster images and documents without interactions.

Specifically, based on the propagated embedding $\tilde{\mathcal{X}}$, besides reconstructing node attributes through GRACE-decoder, we input $\tilde{\mathcal{X}}$ into a GRACE-cluster module, which computes soft clustering assignments $\mathcal{Q}$ to all nodes in $\mathcal{V}$. We compute $\mathcal{Q}$ as a kernel function that measures the similarity between node embeddings and cluster centers according to the equation of Student's t-distribution, which measures the probability of assigning node v_i to the kth cluster [79] as

$$q_{ik} = \frac{(1 + ||\mathbf{x}_i - \mathbf{u}_k||^2)^{-1}}{\sum_j (1 + ||\mathbf{x}_i - \mathbf{u}_j||^2)^{-1}}, \tag{11}$$

To leverage the idea of self-training, we further improve cluster purity and stress on confident assignments by raising q to the second power and then normalizing across all clusters by computing

$$p_{ik} = \frac{q_{ik}^2 / f_k}{\sum_j q_{ij}^2 / f_j}, \tag{12}$$

where $f_k = \sum_i q_{ik}$ is the total number of nodes softly assigned to the kth cluster. Our self-training clustering loss is then defined as the KL divergence between the soft assignment distribution $\mathcal{Q}$ and the auxiliary target distribution $\mathbf{P}$ as following

$$\mathcal{J}_2 = KL(\mathcal{P}||\mathcal{Q}) = \sum_i \sum_k p_{ik} log \frac{p_{ik}}{q_{ik}}. \tag{13}$$

D. Training, Implementation, and Scalability

We achieve the joint learning of clustering and embedding by combining pre-training and co-training. Initialization is known to be crucial for most clustering algorithms. Therefore, before assigning nodes to clusters for the first time, we pre-train our GRACE-encoder and GRACE-decoder without GRACE-cluster for T_0 iterations.Then we run K-means on the pre-trained embedding $\mathcal{X}^0$ to initialize the cluster centers $\mathcal{U}^0$. Next, we plug in GRACE-cluster, and simultaneously optimize the embedding loss $\mathcal{J}_1$ and clustering loss $\mathcal{J}_2$ through co-training for T iterations.

Besides training the neural networks, we need to compute the inversion of a large $n \times n$ matrix for stationary propagation, which could be quite time-consuming. Fortunately, this computation needs to be done only once before training and can be further approximated by multiplying T for multiple times. We implement GRACE using TensorFlow, which runs efficiently on GPU. The codes will be available upon the acceptance of this work.

V. Experiments

In this section, we evaluate GRACE for graph clustering with extensive experiments on six real-world standard public datasets.

A. Experimental Settings

Datasets. We use six real-world attributed network datasets. In social networks, nodes correspond to users and links correspond to observed user connections such as friendships on Facebook and followings on Twitter. Each social network datasets include multiple ego-networks, so all compared algorithms are run on each of them with the final performance averaged over all separate runs. In citation networks, nodes are scientific publications (papers), and links are generated based paper citations. A summary of statistics of these datasets are presented in Table I before.

Compared algorithms. We compare with two groups of algorithms from the state-of-the-art to comprehensively evaluate the performance of GRACE.

Network Community detection algorithms. Some classic algorithms are based on network links alone, while more recent ones also leverage node attributes. We compare with them to show the power of GRACE from deep embedding and influence propagation.

- **MinCut** [12]: a classic community detection algorithm based on modularity.
- **CESNA** [18]: a generative model of links and attributes to detect communities.
- **PCL-DC** [19]: a unification of a conditional model for link analysis and a discriminative model for attribute analysis.
- **CP** [20]: a series of state-of-the-art community detection methods based on content propagation. We compare with the best variant called CPIP-SI.

Network node embedding algorithms. While we find unsupervised node embedding helpful in capturing both link structures and node attributes, we compare with the state-of-the-art embedding algorithms to show that GRACE is advantageous for clustering networks, especially those with sparse noisy attributes. The embedding learned by these algorithms are fed into the same k-means clustering algorithm to produce network clustering results.

- **node2vec** [25]: state-of-the-art non-attributed network embedding algorithm based on biased random walks. It has been shown to be more stable and effective than other competitive algorithms like DeepWalk [23] and LINE [24] in various works.

- **PTE** [33]: a network embedding algorithm that generalizes the popular LINE [24] algorithm by incorporating node attributes through graph augmentation.
- **NRCL** [35]: a network embedding algorithm that bridges the information gap by learning a robust consensus for link-based and attribute-based network representations.
- **ANRL** [34]: a network embedding algorithm that leverages a neighbor enhancement autoencoder to jointly model node attributes and reconstruct target neighbors.

The number of clusters to detect is tuned via standard 5-fold cross validation for all algorithms. The implementations of all compared algorithms are provided by their original authors, except for MinCut, which is provided in the SNAP project[1].

B. Performance On Attributed Network Clustering

We quantitatively evaluate GRACE against all baselines, based on two widely used metrics for evaluating clustering results, *i.e.*, F1 similarity (F1) and Jaccard similarity (JC), as formulated in [20]. We run each non-deterministic algorithm ten times to record the means and standard deviations. For deterministic algorithms, we run one time and treat the standard deviations as 0. Then we conduct paired statistical t-tests by putting GRACE against all baselines.

The parameters of baselines are all tuned to the best through cross-validation. For GRACE, we empirically set the weight of clustering loss λ to 0.1; for the embedding model, we set the number of hidden layers H to 2, with the first layer being half of the size as the original attributes (κ) and following layers halving the sizes (*i.e.*, $\kappa \to \kappa/2 \to \kappa/4$ and $\kappa/4 \to \kappa/2 \to \kappa$ for the encoder and decoder, respectively); dropout rate is set to 0.5; for the influence propagation model, we set the damping factor α to 0.9, stressing on link structures in small local neighborhoods; for joint training embedding and clustering, we set the number of epochs $T_0 = 100$, $T = 30$ and $S = 30$ for pre-training and co-training, respectively. As we will show in the following subsection, GRACE is not very sensitive to the setting of hyper-parameters and no much tuning is needed to achieve satisfactory performance.

Table II shows the mean $F1$ and JC scores evaluated for all compared algorithms. The results all passed the significant t-tests with p-value 0.005. GRACE constantly outperforms all eight compared algorithms by large margins across all six datasets in both $F1$ and JC scores, while baselines have varying performances. This indicates the robustness and general advantages of GRACE.

Taking a closer look, we observe that the advantage of GRACE over baselines is more significant on the social network datasets, where node attributes are known to be noisier. Among the citation networks, GRACE outperforms baselines more on Cora and Citeseer, the node attributes of which are much sparser than PubMed. These two facts indicate the effectiveness of the GRACE deep embedding module in modeling sparse noisy node attributes.

[1]http://snap.stanford.edu/snap/index.html

For both community detection and network embedding baselines, combining node attributes with link structures usually lead to better performance. However, this is not always true (*e.g.*, consider MinCut *vs.* CESNA on the citation networks, node2vec *vs.* ANRL on Twitter and Cora). It implies the correctness and importance of our interpretation of clusters as a consequence of network dynamics and subsequently confirms our appropriate integration of attributes and links via the influence propagation model.

All experiments are done on a server with four GeForce GTX 1080 GPUs and a 12-core 2.2GHz CPU. The runtime of GRACE is comparable to most baselines on a single CPU, while it runs $10+$ times faster on GPUs.

C. Robustness Towards Noisy Node Attributes

As we stress in this work, GRACE is robust to complex attributes with sparse and noisy entries, because it employs deep attribute embedding to capture the essential attributes, and leverages network structures as well as the underlying cluster assignments to refine the attribute embedding space. To fully evaluate such robustness, we conduct a series of controlled experiments by randomly removing and flipping the original attributes in the datasets, and study the behaviors of all compared algorithms.

Figures 2 and 3 show the performance curves of compared algorithms when different amounts of attributes are removed (changed from 1 to 0) or flipped (changed both from 1 to 0 or 0 to 1), respectively. As we can see from Figure 2, while the performances of baseline algorithms clearly drop as more attributes are removed, the performance of GRACE almost does not change. This is especially true on Citeseer, where the original attributes might be quite redundant. As we can also see from Figure 3, the performances of baseline algorithms rapidly drop as more attributes are flipped, but the performance of GRACE only drops slightly. The performance changes are the slightest on Facebook, where the original attributes are already quite noisy.

Such results clearly indicate the robustness of GRACE towards noisy node attributes regarding random missing and inaccurate entries. Due to space limitations, we only show the $F1$ scores on three datasets, but we observe very similar trends of both metrics on all of the six datasets.

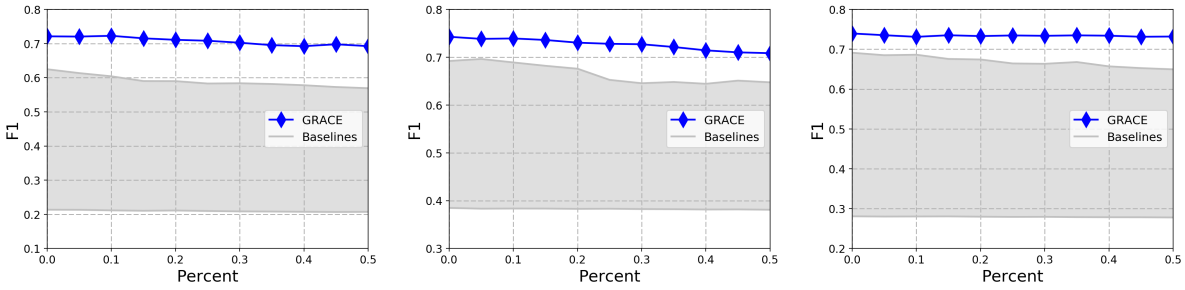
D. Model Selection

We comprehensively analyze the performance of GRACE with different attribute embedding and influence propagation architectures. The results confirm our insights about sparse noisy attributes and network dynamics, while also justify our design of deep embedding and stationary propagation models.

Embedding Size and Depth. As we believe deep embedding is crucial for dealing with sparse noisy attributes, we explicitly study the impact of embedding architectures by varying the size Z and depth H of our encoder and decoder. Specifically, we vary the embedding size Z from 256 for datasets with lower dimensions of original attributes (Facebook and

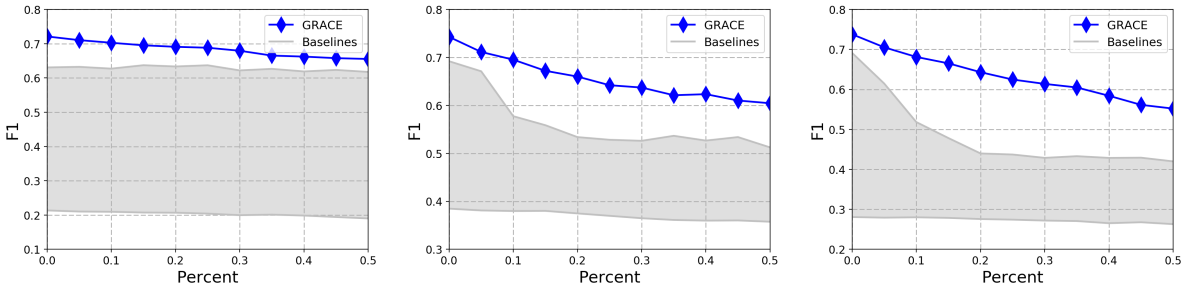
Algorithm	Facebook		Gplus		Twitter		Cora		Citeseer		PubMed	
	F1	JC	F1	JC	F1	JC	F1	JC	F1	JC	F1	JC
MinCut	0.2272	0.0731	0.2212	0.1355	0.2956	0.0970	0.4876	0.2897	0.4189	0.2144	0.5786	0.3419
CESNA	0.4532	0.3019	0.2057	0.1989	0.2976	0.1970	0.4244	0.2115	0.3643	0.1321	0.4612	0.2724
PCL-DC	0.4621	0.2554	0.2373	0.2156	0.3058	0.2062	0.5459	0.3723	0.5041	0.3227	0.5848	0.3978
CP	0.6248	0.3605	0.1450	0.1027	0.2083	0.1580	0.6921	0.5192	0.6912	0.4959	0.7017	0.5212
node2vec	0.4956	0.1565	0.2135	0.2006	0.2742	0.1894	0.6314	0.4638	0.4671	0.2894	0.6604	0.5020
PTE	0.2135	0.1184	0.1697	0.0671	0.1880	0.1016	0.3848	0.2526	0.2807	0.1366	0.4876	0.2410
NRCL	0.4938	0.2135	0.2254	0.2097	0.2988	0.2015	0.5219	0.3283	0.4439	0.2794	0.6280	0.4898
ANRL	0.5362	0.2038	0.2247	0.1928	0.2517	0.1584	0.6486	0.4735	0.5301	0.3374	0.6872	0.5146
GRACE	**0.7212**	**0.4436**	**0.3110**	**0.2559**	**0.4132**	**0.2620**	**0.7428**	**0.5579**	**0.7397**	**0.5362**	**0.7378**	**0.5766**

TABLE II: **Performance comparison with two groups of baselines on six real network datasets.**



(a) Facebook (b) Cora (c) Citeseer

Fig. 2: F1 scores of compared algorithms when node attributes are randomly removed.



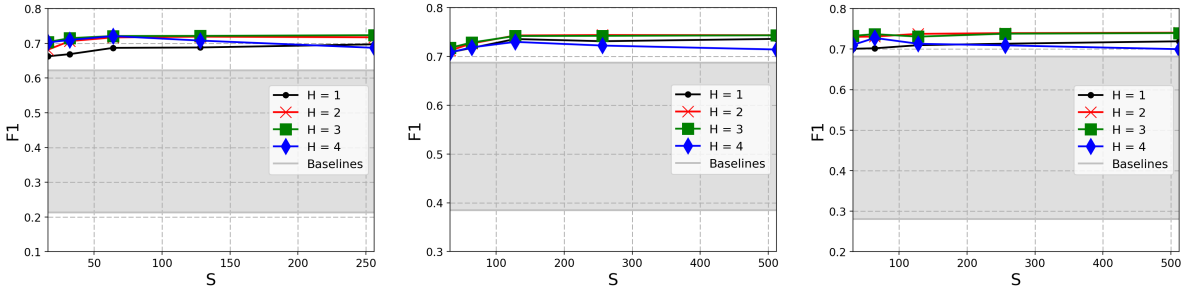
(a) Facebook (b) Cora (c) Citeseer

Fig. 3: F1 scores of compared algorithms when node attributes are randomly flipped.

PubMed) and 512 for the rest, to a small number like 16 or 32, to understand the attribute complexity of each dataset and how compact the useful representations can be. Each time, we halve the embedding dimension by 2. For each dataset, we vary the embedding depth H from 1 to 4, to see how deep neural networks can help to effectively capture the compact representations. For simplicity, we do not use hidden layers of decreasing or increasing sizes as in Section V.2. Instead, we set the sizes of all hidden layers to be the same. For example, with $Z = 256$ and $H = 2$, the encoder and decoder have the architectures of $\kappa \rightarrow 256 \rightarrow 256$ and $256 \rightarrow 256 \rightarrow \kappa$, respectively.

Figure 4 shows the $F1$ scores on three datasets. The JC scores on them and both metrics on other datasets show similar trends and are omitted due to the space limitations. For Facebook, the ranges of x-ticks are smaller, because their original attributes are of lower dimensions. As we can observe, the embedding size Z does not have a large impact on the performance of GRACE, except for the cases where it is too small for the embedding to capture the complex attribute semantics. This happens apparently on Facebook, probably because its attributes are less sparse and more complex, and slightly Cora, indicating its attributes to be also more complex than those of Citeseer. Setting Z to a value like half of the size of the original attributes will usually lead to satisfactory results.

As for the embedding depth H, shallower neural networks with $H = 1$ perform significantly worse than their deeper competitors on most social network datasets, for example, Facebook, which are known to have noisier attributes. This clearly confirms our insight of leveraging the power of deep embedding to effectively capture network node attributes. Since we apply dropout in both encoder and decoder, only slight overfitting is observed on a few datasets that have rel-



(a) Facebook (b) Cora (c) Citeseer

Fig. 4: F1 scores of model selection experiments on embedding size and depth.

atively small networks (*e.g.*, Facebook), when the embedding depth is too large ($H = 4$). Setting H to a value like 2 or 3 tends to produce stable results.

Stable Propagation and Its Approximations. In this work, we understand network clusters as a consequence of influence propagation on networks, and we believe the cluster structures are most significant when such propagation reaches stability, *i.e.*, the stationary distribution. In Eq. 9, we compute the stationary transition matrix R, which involves the inversion of a large matrix, which can be quite time comsuming. Here, we simply use T^b, the multiplications of T, as an approximation of R, to show how well we can capture network clusters without reaching the stationary distribution.

As we can observe from Figure 5, the number of propagation steps has a large impact on the performance of GRACE, especially when the number is small. This demonstrates the utility of our influence propagation model. As the number of steps grows large and an approximated stationary distribution is reached, the performance becomes stable. However, there is still a performance gap between the approximated and the true stationary distributions, because the damping effect cannot be trivially approached through multiplying one-step transition matrices, but cluster structures are often observed within small network neighborhoods.

E. Case Studies with Visualization

To explicitly see how GRACE captures network clusters through the end-to-end combination of deep embedding, influence propagation, and self-training clustering modules, we visualize the original attribute space, together with the embedding spaces computed by ANRL (the state-of-the-art attributed network embedding algorithm) and GRACE on one of the real-world Facebook egonetworks.

Figure 6 visualizes the embedding spaces reduced to 2-dimensional through PCA [80]. We use different colors and markers to draw nodes in different ground-truth clusters. As we can see, nodes from different clusters clutter a lot in the original attribute space, probably due to the attribute sparsity and noise. The situation is better with the state-of-the-art ANRL embedding, which jointly considers node attributes and link structures. But the results still do not demonstrates obvious structures for each of the clusters, thus posing more challenges to the following clustering methods. This might be because the embeddings are not properly optimized towards the clustering objective. The embedding produced by our GRACE model is much better than both of them, where nodes within the same ground-truth clusters distribute tightly in the embedding space, and those within different clusters lie far apart.

Figure 7 visualizes the pair-wise node cosine similarities in different spaces with heat maps. Since nodes on the $x-$ and $y-$ axes are grouped by the ground-truth cluster assignments in the same order, similarities among nodes in the diagonal blocks corresponding to the clusters should have higher values. The embedding generated by our GRACE model is the only one that demonstrates clear diagonal blocks, indicating its power of refining the attribute embedding space through combining deep attribute embedding and network influence propagation to capture the underlying clustering structures.

VI. Conclusion

Recently, there has been a trend in applying deep learning algorithms for network modeling. However, none of them jointly considers the noisy node attributes and link structures together with the underlying clustering structures in an end-to-end unsupervised fashion. While GRACE is designed for network clustering, it provides a promising way of deeply exploring noisy network attributes and links. It is promising to see the both products, *i.e.*, the network representations and clustering results, to be further leveraged for various downstream applications on real-world attributed networks.

Acknowledgement

Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS-16-18481, IIS-17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

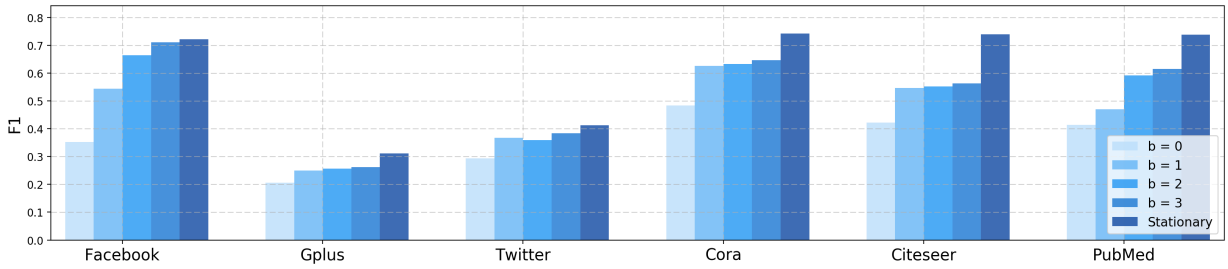
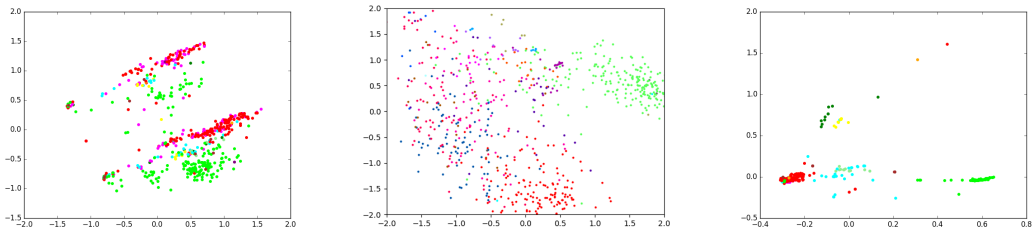
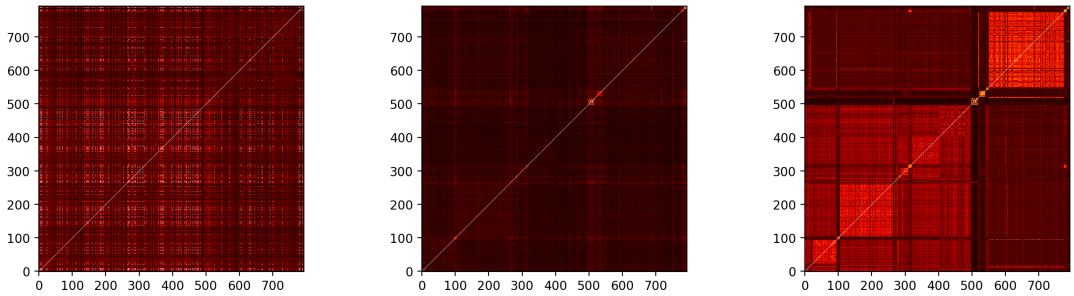


Fig. 5: F1 scores of model selection experiments on stable influence propagation and its approximations.



(a) Original attributes (b) ANRL embedding (c) GRACE embedding

Fig. 6: Visualization of embedding spaces computed on ego-network 1684 from the Facebook dataset.



(a) Original attributes (b) ANRL mbedding (c) GRACE embedding

Fig. 7: Visualization of embedding pair-wise node similarities on ego-network 1684 from the Facebook dataset.

REFERENCES

[1] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *WSDM*. ACM, 2017, pp. 731–739.

[2] ——, "Accelerated attributed network embedding," in *SDM*, 2017.

[3] J. Li, H. Dani, X. Hu, J. Tang, Y. Chang, and H. Liu, "Attributed network embedding for learning in a dynamic environment," in *CIKM*, 2017.

[4] J. J. Pfeiffer III, S. Moreno, T. La Fond, J. Neville, and B. Gallagher, "Attributed graph models: Modeling network structure with correlated attributes," in *WWW*, 2014, pp. 831–842.

[5] G.-J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang, "Exploring context and content links in social media: A latent space method," *TPAMI*, vol. 34, no. 5, pp. 850–862, 2012.

[6] A. P. Streich, M. Frank, D. Basin, and J. M. Buhmann, "Multi-assignment clustering for boolean data," in *ICML*, 2009, pp. 969–976.

[7] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *JMLR*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[8] R. Li, C. Wang, and K. C.-C. Chang, "User profiling in an ego network: co-profiling attributes and relationships," in *WWW*, 2014, pp. 819–830.

[9] C. Yang, L. Zhong, L.-J. Li, and L. Jie, "Bi-directional joint inference for user links and attributes on large social graphs," in *WWW*, 2017.

[10] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation," in *KDD*, 2017.

[11] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[12] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[13] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *WSDM*, 2013.

[14] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016, pp. 478–487.

[15] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *IJCAI*, 2017.

[16] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep clustering with convolutional autoencoders," in *ICNIP*, 2017, pp. 373–382.

[17] W. Zhou, H. Jin, and Y. Liu, "Community discovery and profiling with social messages," in *KDD*, 2012, pp. 388–396.

[18] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *ICDM*, 2013, pp. 1151–1156.

[19] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: a discriminative approach," in *KDD*, 2009.

[20] L. Liu, L. Xu, Z. Wangy, and E. Chen, "Community detection based on structure and content: A content propagation perspective," in *ICDM*, 2015, pp. 271–280.

[21] O. Tsur and A. Rappoport, "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities," in *WSDM*, 2012, pp. 643–652.

[22] J. Leskovec and J. J. Mcauley, "Learning to discover social circles in ego networks," in *NIPS*, 2012, pp. 539–547.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.

[24] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.

[25] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.

[26] S. Cao, W. Lu, and Q. Xu, "Grarep:learning graph representations with global structural information," in *CIKM*, 2015, pp. 891–900.

[27] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *KDD*, 2016, pp. 1225–1234.

[28] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations." in *AAAI*, 2016, pp. 1145–1152.

[29] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*, 2016, pp. 31–35.

[30] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *NIPS*, 2017, pp. 24–33.

[31] C. Song, F. Liu, Y. Huang, L. Wang, and T. Tan, "Auto-encoder based data clustering," in *CIARP*, 2013, pp. 117–124.

[32] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering." in *AAAI*, 2014, pp. 1293–1299.

[33] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *KDD*, 2015, pp. 1165–1174.

[34] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "Anrl: Attributed network representation learning via deep neural networks." in *IJCAI*, 2018, pp. 3155–3161.

[35] X. Wei, L. Xu, B. Cao, and P. S. Yu, "Cross view link prediction by learning noise-resilient representation consensus," in *WWW*, 2017.

[36] Z. Yang, W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in *ICML*, 2016.

[37] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *ICML*, 2016, pp. 2014–2023.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[39] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *ICLR*, 2016.

[40] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008, pp. 1096–1103.

[41] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *ICASSP*, 2013, pp. 8595–8598.

[42] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su, "Understanding retweeting behaviors in social networks," in *CIKM*, 2010.

[43] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *CIKM*, 2000, pp. 86–93.

[44] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[45] G. Namata, B. London, L. Getoor, B. Huang, and U. EDU, "Query-driven active surveying for collective classification," in *MLG*, 2012.

[46] N. Du, B. Wu, X. Pei, B. Wang, and L. Xu, "Community detection in large-scale social networks," in *WebKDD*, 2007, pp. 16–25.

[47] Y. Ruan, D. Fuhry, and S. Parthasarathy, "Efficient community detection in large networks using content and links," in *WWW*, 2013.

[48] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos, "Pics: Parameter-free identification of cohesive subgroups in large attributed graphs," in *SDM*, 2012, pp. 439–450.

[49] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[50] C. Yang, X. Shi, L. Jie, and J. Han, "I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application," in *KDD*, 2018.

[51] C. Yang, C. Zhang, X. Chen, J. Ye, and J. Han, "Did you enjoy the ride? understanding passenger experience via heterogeneous network embedding," in *ICDE*, 2018.

[52] C. Yang, M. Liu, V. W. Zheng, and J. Han, "Node, motif and subgraph: Leveraging network functional blocks through structural convolution," in *ASONAM*, 2018.

[53] C. Yang, J. Zhang, and J. Han, "Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks," in *ICDM*, 2020.

[54] A. G. Duran and M. Niepert, "Learning graph representations with embedding propagation," in *NIPS*, 2017.

[55] X. Pei, C. Chen, and Y. Guan, "Joint sparse representation and embedding propagation learning: A framework for graph-based semisupervised learning," *TNNLS*, vol. 28, no. 12, pp. 2949–2960, 2016.

[56] C. Yang, M. Liu, F. He, X. Zhang, J. Peng, and J. Han, "Similarity modeling on heterogeneous networks via automatic path discovery," in *ECML-PKDD*, 2018.

[57] C. Yang, J. Zhang, and J. Han, "Neural embedding propagation on heterogeneous networks," in *ICDM*, 2019.

[58] F.-Y. Sun, M. Qu, J. Hoffmann, C.-W. Huang, and J. Tang, "vgraph: A generative model for joint community detection and node representation learning," in *NIPS*, 2019.

[59] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, "Gemsec: Graph embedding with self clustering," in *ASONAM*, 2019.

[60] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *IJCAI*, 2015.

[61] C. Yang, A. Pal, A. Zhai, N. Pancha, J. Han, C. Rosenberg, and J. Leskovec, "Multisage: Empowering gcn with contextualized multi-embeddings on web-scale multipartite networks," in *KDD*, 2020.

[62] C. Yang, J. Zhang, H. Wang, S. Li, M. Kim, M. Walker, Y. Xiao, and J. Han, "Relation learning on social networks with multi-modal graph edge variational autoencoders," in *WSDM*, 2020.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[64] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR*, 2011, pp. 3361–3368.

[65] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010.

[66] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," *SSST*, 2014.

[67] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *CoRR*, 2015.

[68] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," *Computer Science*, 2013.

[69] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016.

[70] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *TNNLS*, vol. 20, no. 1, pp. 61–80, 2009.

[71] J. Wang, W. V. Zheng, and K. C.-C. Chang, "Topological recurrent neural network for diffusion prediction," in *ICDM*, 2017.

[72] C. Li, J. Ma, X. Guo, and Q. Mei, "Deepcas: an end-to-end predictor of information cascades," in *WWW*, 2017, pp. 577–586.

[73] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *JLMR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[74] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.

[75] D. Chakrabarti, S. Funiak, J. Chang, and S. A. Macskassy, "Joint inference of multiple label types in large networks," in *ICML*, 2014.

[76] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing letters*, vol. 12, no. 3, pp. 211–223, 2001.

[77] M. Granovetter, "Threshold models of collective behavior," *American journal of sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[78] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "Pagerank with priors: An influence propagation perspective," in *IJCAI*, 2013.

[79] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[80] I. T. Jolliffe, "Principal component analysis and factor analysis," in *Principal component analysis*. Springer, 1986, pp. 115–128.