Attributed Network Embedding based on Mutual Information Estimation

Xiaomin Liang
Sun Yat-Sen University
liangxm23@mail2.sysu.edu.cn

Daifeng Li*
Sun Yat-Sen University
lidaifeng@mail.sysu.edu.cn

Andrew Madden
Sun Yat-sen University
admadden@hotmail.com

ABSTRACT

Attributed network embedding (ANE) attempts to represent a network in short code, while retaining information about node topological structures and node attributes. A node's feature and topological structure information could be divided into different local aspects, while in many cases, not all the information but part of the information contained in several local aspects determine the relations among different nodes. Most of the existing works barely concern and identify the aspect influence from network embedding to our knowledge. We attempt to use local embeddings to represent local aspect information and propose InfomaxANE which encodes both global and local embeddings from the perspective of mutual information. The local aspect embeddings are forced to learn and extract different aspect information from nodes' features and topological structures by using orthogonal constraint. A theoretical analysis is also provided to further confirm its correctness and rationality. Besides, to provide complete and refined information for local encoders, we also optimize feature aggregation in SAGE with different structures: feature similarities are concerned and aggregator is seperated from encoder. InfomaxANE is evaluated on both node clustering and node classification tasks (including both transductive and inductive settings) with several benchmark datasets, the results show the outperformance of InfomaxANE over competitive baselines. We also verify the significance of each module in our proposed InfomaxANE in the additional experiment.

CCS CONCEPTS

• **Mathematics of computing** → **Graph algorithms**; **Approximation algorithms**;

KEYWORDS

Network embedding; Attributed network; Mutual information; Local embedding

ACM Reference Format:
Xiaomin Liang, Daifeng Li, and Andrew Madden. 2020. Attributed Network Embedding based on Mutual Information Estimation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412008

1 INTRODUCTION

Network mining is an important area of academic research, with significant practical applications. The question of how to represent a network is fundamental to network mining. Network embedding (NE) aims to find a low-dimensional representation of nodes in the network, while preserving information about the nodes' proximity. The learned low-dimensional embeddings can be applied as latent features for downstream tasks like node classification, link prediction, and node recommendation.

Network embedding is treated as matrix factorization in the early works [1–3]. Inspired by Skip-Gram model from NLP area, the practice DeepWalk [4] and Node2vec [5] are arouse. Various sampling strategies have been used to encode different types of structural features [6, 7]. Researchers become interested in neural networks following the development of deep learning, and several typical neural network structures are adapted to network embedding, such as GAN [8, 9], RNN [10], LSTM [11], CNN [12, 13]. The aforementioned methods are designed to learn the topological information of the networks, but nodes are usually associated with rich attribute information, which may play an important role in the downstream tasks. Unlike plain network embedding, attribute network embedding (ANE) takes into account, not only topological structure information, but also attribute information. Collaborative training is used to determine the nature of interactions between the two types of information, after each has been embedded separately [14, 15]. Besides, incorporating the node attributes matrix into the factorization processing allows the matrix factorization methods to take advantage of both types of information [16], or utilizing attribute information to make a better random walk [17]. The efficiency and effectiveness of multi-layer feature aggregation helped to make the approach popular, it encodes topological structures and attribute information by aggregating the features of nodes in their K-order neighborhood [13, 18, 19].

In addition to the problem of information incorporation, the question of how to capture high-quality embeddings is still to be explored. An issue addressed here is that of maximizing the mutual information between node features and embeddings to preserve information that is "useful" to the objective. Mutual information calculation is complicated, especially for high-dimensional and continuous variables; however, Belghazi et al's Mutual Information Neural Estimation (MINE) approach [20] makes the effective computation of mutual information possible. In this work, we will use mutual information estimation theory to determine the node

[*] Corresponding author: Daifeng Li, lidaifeng@mail.sysu.edu.cn.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412008

embeddings. Recent representative researches of mutual information based graph embedding includes DGI [21], GMI [22] and etc, which design different embedding structures to obtain high quality graph embeddings based on mutual information maximization. However, most of above researches seldom consider locality and local embeddings.

Locality is important in the task of training. Convolutional neural networks (CNNs) that operate on locally connected regions really make a breakthrough in image area; manifold learning aims to preserve local structure in high-dimensional space while reducing the dimensionality. The locality of node features in graph representation learning is also valuable, in citation networks for example, one article may cite another because of global embedding similarities, but the citation may also be due to partial aspect similarity: the two articles may use a similar method, have common research objects or share application scenarios. However, the locality of node features is rarely concerned in current researches, since the principal problem is how to define the local parts of node features. These features are discrete or continuous attributes, or are vector representations derived from text associated with the nodes. The dimensions of node features are not sequential, so changing the order of dimensions will not change the expression of the node features as a whole. Besides, the correlation between dimensions is unknown, so they cannot be directly divided into parts or put into groups. In the work presented here, we build up several encoders to learn the local embeddings, and constrain the mutual information between local embeddings to encourage them to learn diverse information.

To provide complete and refined information for above-mentioned local encoders, we accept the feature aggregation idea to incorporate topological structure and node features, but we just aggregate node features in each layer without trainable parameters mapping and keep the dimensions of nodes unchanged. Besides, feature similarity is added as aggregation weights considering a more similar neighbor node should have more impact on one node.

In this work, we propose InfomaxANE, an unsupervised attributed network embedding model that encodes node features' global and local information using mutual information estimation theory. The mutual information between local embeddings are constrained to enforce them to learn different local information. InfomaxANE incorporates the topological structure and node features with refined multi-layer feature aggregation, in which feature similarities are concerned and aggregator is seperated from encoder. To sum up, the main contributions of this work are as followed:

- We introduce mutual information estimation theory into attributed network embedding and propose an attributed network embedding model that considers both global and local information of node features.
- We successfully encode the local information of node features by minimizing the mutual information among different local embedding, it works out according to the experiment.
- We demonstrate that both features similarity weights and separation of aggregator and encoder during the feature aggregation period can bring performance gains.
- We demonstrate the superiority of proposed method by comparing it with several advanced baselines on node clustering task and transductive/inductive node classification task. It should be noted that the proposed method achieves absolute gains of 1.03% 8.61% in the node clustering task.

2 RELATED WORK

Feature aggregation strategies. SAGE [18] is the first to bring out the concept of feature aggregation which can learn the topological structure and node features simultaneously by aggragating node features in the defined neighborhood. Feature aggregation in SAGE is inherited from GCN [13] but with more flexibility. In subsequent research, attention mechanism was used to capture better aggregated features [23, 24], SPINE [19] redefine the neighborhood by rooted random walk sampling. There is one thing that we notice, feature aggregation and reduction are processed simultaneously in the mentioned works, it means trainable parameters are introduced in the lay-wise feature aggregation, lower layer aggregated features will affect the follow-up feature aggregate layers, will it have an influence on the final effectiveness and efficiency, it is one of the problems we want to explored in this work. Except for this reason, we want to provide complete node features for the follow-up encoder to encode global/local information, so we choose to separate aggregator and encoder in this work.

Mutual information estimation. Mutual information (MI) is an important theory in representation learning. In supervised learning, the information bottleneck theory [25, 26] considers the representation problem as finding the short codes of inputs but preserving the maximum information of targets/labels. This theory has been introduced to many deep neural networks commonly; it might lighten the black box of deep learning in terms of information [27]. In unsupervised learning, mutual information maximization aims to force embeddings to be distinctive by adding specific information [28], such as manually added labels by negative sampling [29, 30]. Contrastive Predictive Coding (CPC) [29] maximizes the mutual information between the encoded representations. Deep Infomax [30] maximizes the mutual information between input and global/local embedding pairs. Feature locality is taken in to consideration in Deep Infomax, but it is not suitable for network embedding, node features are not "regular", they cannot be directly divided into parts like images can. However, Information Competing Process (ICP) [28] suggests that it may be possible to constrain the encoded local embeddings by minimizing the mutual information between them to force them to learn different information.

Several network embedding approaches based on MI theory are proposed recently. DGI [21] encodes the global and local structure by maximizing the mutual information between patch representations and the corresponding high-level summaries of graphs. GMI [22] measures the mutual information from both node features and topological structure. Both DGI and GMI directly encode nodes' feature and structural information into global embeddings without considering and extracting local aspect information. In InfomaxANE, we aim to consider both global and local aspect information into a unified framework and make full use of them to further improve the embedding quality.

3 PROPOSED METHOD

Let G = {V, A, X} denotes an attributed directed/undirected network, where V is a set of vertices with size |V|=N, A is the adjacency matrix with size N*N, which represents the edges (relationship between nodes), A_{ij}=1 if node v_i is linked with node v_j, otherwise A_{ij}=0. X is the node features matrix with size N*f, where f represents the number of features of each node in V, so the *i*th row in X represents the f-dimension features of node v_i. The target of network embedding is to find low-dimensional continuous embedding for each node, as denoted by Z^{N^*d}, where d is the dimension of node embedding.

3.1 Preliminaries of mutual information estimation

Mutual information is defined as: I (Z, X) = H(Z) - H(Z|X), where H is the entropy, Z and X are two random variables. The mutual information between Z and X can be perceived as the decrement of the uncertainty of Z given X. I (Z, X) is equivalent to the KL divergence between the join distribution P(Z, X) and production of marginal distributions P(Z)P(X) (see Eq. 1).

$$\begin{aligned}
\mathrm{I(Z;X)} &= H(Z) - H(Z|X) \\
&= \sum_{z} p(z) \log \frac{1}{p(z)} - \sum_{x,z} p(x,z) \log \frac{1}{p(z|x)} \\
&= \sum_{x,z} p(x,z) \log \frac{p(x,z)}{p(x)p(z)} \\
&= KL(P(X,Z)||P(X)P(Z))
\end{aligned} \tag{1}$$

KL-divergence is theoretically unbounded, so maximizing KL-divergence is likely to get an infinite result (p(z) is infinitely close to 0 accoring to Eq. 1), which is not convenient for the model to maximize mutual information. Since what we want is to widen the distance between p(Z, X) and $p\ (X)\ p\ (Z)$, we can replace KL-divergence with a bounded divergence, for example, JS-divergence [31]. The mutual information maximization between Z and X turns out to be the maximization of the JS-divergence between p(X, Z) and $p\ (X)\ p\ (Z)$. We refer to the variational estimation of f-divergence in f-GAN [31], where f-divergence is a general form of JS-divergence as follow [31]:

$$\begin{aligned}
\mathrm{JS(p(Z,X)||p(X)p(Z))} = \max_{T} \Big(&\mathbb{E}_{(x,z)\sim p(Z|X)p(X)} [\log \sigma(T(x,z))] \\
&+ \mathbb{E}_{(x,z)\sim p(X)p(Z)} [\log(1 - \sigma(T(x,z)))] \Big)
\end{aligned} \tag{2}$$

T(x, z) is a discriminator constructed by a neural network to calculate the distance between input x and its representation vector z. σ is an activation function. We can observe that the problem for now is transformed from JS-divergence estimation into discriminate network T(x, z) training issue. We can take the input x and its corresponding embedding z as positive sample, x and randomly selected z based on P(X)P(Z) distribution as negative sample, to train the discriminator T(x, z).

3.2 Proposed method

The proposed method, InfomaxANE (see Figure 1), is comprised of three modules: feature aggregator, global/local encoder and discriminator. The feature aggregator is responsible for redefining node feature and structural information by aggregating them from their K-order neighborhood; the global/local encoder is targeted at learning the low-dimensional continuous global/local embedding for each node; and the task of distinguishing positive samples from negative ones falls into discriminator. The general objective of InfomaxANE is to maximize the mutual information between inputs and corresponding global and local embeddings (see Eq. 3):

$$\mathcal{L} = \max \left(\alpha * I_{\text{global}}(z;x) + \beta \frac{1}{c} \sum_{i}^{c} I_{\text{local}}\left(z^{(i)};x\right) \right) \tag{3}$$

x and z denotes global input and global embedding respectively; $z^{(i)}$ denotes *i*th local embedding; c is the number of local embeddings; α and β are hyper-parameters that balance the contributions of global and local mutual information loss.

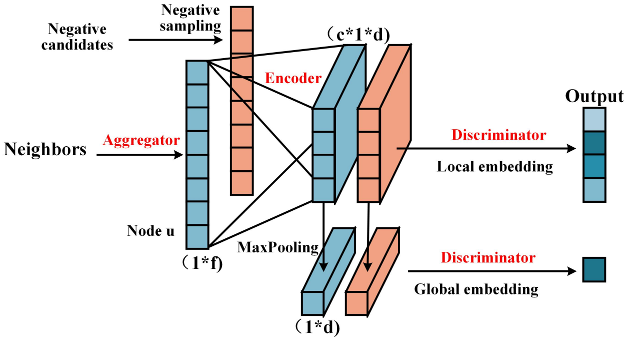


Figure 1: An overview of InfomaxANE.

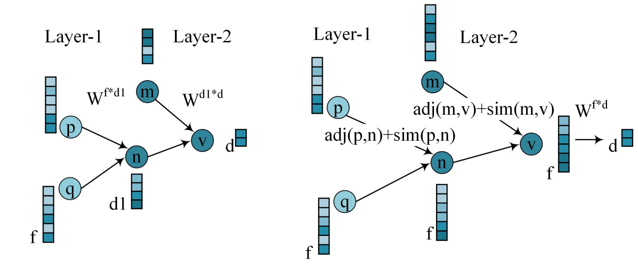
Feature aggregator. In SAGE, node feature aggregation and encoding are processed simultaneously (Figure 2 (a)); but here, we try to separate them. As a result, there are no trainable parameters in aggregator, so feature aggregation can be pre-computed off-line. Neighbors with more similar features should have greater influence on the specific node: InfomaxANE attempts to take this into account. The aggregate computation is shown in Eqs s. 4-5:

$$x_u^{(k)} = \sum_{v \in \mathcal{N}_u} \alpha_{uv} x_v^{(k-1)} \tag{4}$$

$$\alpha_{uv} = \frac{\left(1 + x_u^{(k-1)} x_v^{(k-1)^{\mathrm{T}}}\right)}{\sum_{v \in \mathcal{N}_u} \left(1 + x_u^{(k-1)} x_v^{(k-1)^{\mathrm{T}}}\right)} \tag{5}$$

$x^{(k)}{}_u$ denotes the features of node u in *k*th layer aggregation, a_{uv} is the influence weight of node v on node u, $\mathcal{N}(u)$ denotes the neighbors of node u (including node u itself). InfomaxANE can carry out multi-layer aggregation, in which the features in *k*th comes from the aggregated features in *(k-1)*th.

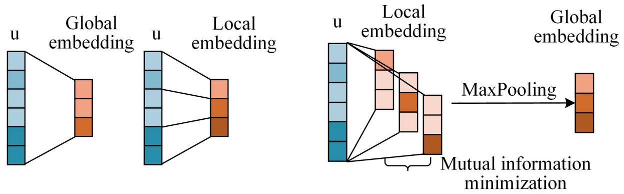
Global/local encoder. Feeding only one part of the input into the encoder at one time is an intuitive way to encode local information (see 3(a)). However, the dimensions of node features are not sequential, the correlation between different dimensions is unknown, so we can't just cut the input features into blocks or divide them into groups. In InfomaxANE (see Figure 3(b), the complete f-dimensional input features are fed into encoders and encoded as c d-dimensional embeddings with a stacking parameter matrix $W^{c^*f^*d}$ as follow:



(a) Feature aggregation in SAGE

(b) Feature aggregation in InfomaxANE

Figure 2: Comparison of feature aggregation between SAGE and InfomaxANE. (a) SAGE reduce the node features from f-dim to d-dim with parameters matrix W in the *i*th layer after aggregate node features from neighbors, and the operation repeat in *(i+1)*th layer using the aggregated node features obtained in *i*th layer. (b) InfomaxANE keeps the dimensions of node features unchanged during the aggregation, and feature similarity is concerned in the aggregate operation.



(a) Intuitive implement of global and local embedding

(b) Global and local embedding in InfomaxANE

Figure 3: Global and local embedding in InfomaxANE.

$$Z^{(\mathrm{i})} = \sigma\left(X^{(k)} W^{(i)}\right) \tag{6}$$

X$^{(k)}$ is the matrix of all nodes $x_u^{(k)}$ in Eq. 4. W$^{(i)}$ is the *i*th matrix in W^{c*f*d} with size f*d, RELU is used as activate function $\sigma(.)$. We take the maximum value of each dimension in c local embeddings as the global embedding, in other words, the global embedding is the max-pooling result of c stacking local embeddings (see Figure 3(b)). The global embedding is the final embedding for each node.

Discriminator. The general objective of InfomaxANE is to maximize the mutual information between inputs and global/local embeddings, namely, max $I_{\text{global}}\,(Z, X)$ and max $I_{\text{local}}\left(Z^{(i)}, X\right)$. All dimensions of node features are fed into each encoder, so the objectives can be represented as follow:

$$\begin{aligned} \max I_{\text{global}}(Z, X) = \max_T \big(& [\log \sigma(\mathrm{T}(x, z))] \\ & + \mathbb{E}_{(x,z)\sim p(x)p(z)}\left[\log(1 - \sigma(T(x, z)))\right]\big) \end{aligned} \tag{7}$$

$$\begin{aligned} \max I_{\text{local}}\left(Z^{(i)}, X\right) = \max_T \Big(& \left[\log \sigma\left(T\left(x, z^{(i)}\right)\right)\right] \\ & + \mathbb{E}_{(x,z)\sim p(x)p(z)}\left[\log\left(1 - \sigma\left(T\left(x, z^{(i)}\right)\right)\right)\right]\Big) \end{aligned} \tag{8}$$

In the network embedding scenario, nodes should be close to their neighboring nodes, so as a positive sample, we take input x and its neighbor's node embedding z pair, and as a negative sample, we take x and sampled z pair. Distinguishing between positive and negative samples is the task of discriminator. Similar with GMI [22], the element-wise production of input pairs is used as input for the discriminator. Consequently, the maximization of mutual information can be expressed as Eq. 9, where u and v are neighbor nodes, N_u is the neighbor set of u and v $_n$ is a negative sample corresponding to u and v. We used a simple negative sampling strategy in this work: nodes randomly selected outside the h-order of input nodes are taken as negative samples. The negative sampling strategy is mainly based on the idea of graph embedding that the target node has a higher probability to have strong correlations with its 1-hop neighbors, while its correlations with h-hop nodes could be neglected in most of cases. I(z $_u$; x $_v$) is mutual information between z $_u$ and x $_v$, which could be obtained by a discriminator constructed by a neural network.

$$\begin{aligned} \max I(z_u, x_u) = \max \Bigg(& \left[\log \sigma\left(\sum_{v\in\mathcal{N}_u} z_u^T z_v I(z_u, x_v)\right)\right] \\ & + \mathbb{E}_{(v_n, z_{v_n})\sim p(x)p(z)}\left[\log\left(1 - \sigma\left(\sum_{v_n} z_u^T z_{v_n} I(z_u, x_{v_n})\right)\right)\right]\Bigg) \end{aligned} \tag{9}$$

Let's retrospect to the mutual information minimization among local embeddings, which encouraging local embeddings to learn different local aspect information. If it is assumed that the local embeddings of one node are represented as {z $^{(1)}$, z$^{(2)}$, …, z$^{(c)}$}, the mutual information between every two of them is defined as I(z$^{(i)}$, z$^{(j)}$), then I(z$^{(i)}$, z$^{(j)}$) reaches its minimum value zero when $p\left(z^{(i)}, z^{(j)}\right) = p\left(z^{(i)}\right) p\left(z^{(j)}\right)$, which indicates that z$^{(i)}$ and z$^{(j)}$ are independent to each other, but it's hard to compute the dependency for such high-dimensional (compared to 2-dims or 3-dims) and continuous vectors. Information Competing Process (ICP) [28] builds two predictors for two parts of embeddings, so the two parts can't be transformed to each other. We took a similar approach, but used c local embeddings. This required c(c-1), resulting in far more costs. To alleviate this problem, we loosen the constraint of independency to linear independency. Orthogonality is a special case of linear independence of vector groups: mutually orthogonal vector groups must be linear independent. So the constraint of orthogonality between different local embeddings is added into the main objective of enforcing the representation of different information. This constraint is fulfilled by minimizing the distance between Z$^{\mathrm{T}}$Z and the identity matrix. In this study, L1 norm of matrix difference is used as the matrix distance measure in this work, then the constraint is defined as $\left|Z^T Z - I\right|$ at the last.

A theorem is introduced to further confirm the correctness and rationality of the proposed locality embedding, which could be seen in Theorem 1.

Theorem 1. Assume the input of node v $_i$ is x $_i$, which satisfies probability distribution P(X), where X is the set of all N nodes in graph G. Assume the locality embedding of x $_i$ is Z $_i$ = $\{z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(c)}\}$, where c is the number assignment of local embedding, and the probability of Z $_{\rm i}$ is P(Z $_{\rm i}$), Z = $\{Z_1, Z_2, \ldots, Z_{\rm N}\}$. Then the optimal classifier between the joint distribution p(X, Z) and the product of marginals p(X)p(Z), has an error rate upper bounded by $\sum_{i=1}^{N} p(X_{\rm i}|Z_{\rm i})p(Z_{\rm i})^2/2$.

Proof. For each node i, the product of marginals is: p(Z $_{\rm i}$)p(X $_{\rm i}$) = p(X $_{\rm i}$ | Z $_{\rm i}$)p(Z $_{\rm i}$) ^{2}, while for all the N nodes, the total probability of drawing a sample from product of marginal is $\sum_{i=1}^{N} p(X_{\rm i}|Z_{\rm i})p(Z_{\rm i})^2$. As the count of error samples should be drawn from both joint and product of marginal probability distributions, and samples from joint distribution have lower error rate than those from marginal probability [21], for an extreme situation of a binary classifier, all samples drawn from marginal probability could not be identified (50% error samples), if all the samples are also drawn by joint probability distribution, the total percentage of error samples should be smaller than 50%, and an upper error rate bound should be $\sum_{i=1}^{N} p(X_{\rm i}|Z_{\rm i})p(Z_{\rm i})^2/2$.

For a mutual information based network embedding model without considering locality embedding, such as GMI [22], assume the node embedding of x $_{\rm i}$ is $Z_i^{'}$, then by applying similar proof process, its upper error rate bound should be $\sum_{i=1}^{N} p(X_{\rm i}|Z_i^{'})p(Z_i^{'})^2/2$. Then we could obtain a lemma as below:

Lemma 1. The error rate bound of { X, Z } is smaller than that of { X, Z $^{'}$ }, which means locality embedding could obtain higher quality of embeddings compare with methods without considering locality.

Proof. Because $\sum_{i=1}^{N} p(X_{\rm i}|Z_{\rm i})p(Z_{\rm i})^2/2 = \sum_{i=1}^{N} p(X_{\rm i})p(Z_{\rm i})/2$, so proving p(Z $_{\rm i}$) < p($Z_i^{'}$) is equal to providing evidence that Z has bigger solution space than that of Z $^{'}$. Firstly, Z with orthogonal constraint $\left|Z^T Z - I\right|$ could be seen as a set of orthonormal basis of the space, which could represent all the allowable states in the space of Z $^{'}$. Secondly, by taken $\left|Z^T Z - I\right|$ as a regularization term, all allowable states in the space of Z $^{'}$ could be generated as new states in the space of Z by adding Hessian matrix based offset value on each dimension. Thus the parameter space of Z $^{'}$ is almost included by Z's parameter space and we can conclude that Z is a generalization of Z $^{'}$, and Lemma 1 has been proved.

The final objective function in InfomaxANE is shown in Eq. 10 after putting all parts together. α, β, γ are hyper-parameters that balance the three items in the objective. We used Stochastic Gradient Descent (SGD) as optimizer to update trainable parameters.

$$\begin{aligned}\mathcal{L} = \min\Bigg(&-\alpha * \Bigg(\Bigg[\log\sigma\Bigg(\sum_{v\in\mathcal{N}_u} z_u^T z_v I(z_u^T; z_v)\Bigg)\Bigg]\\ &+\mathbb{E}_{(x,z)\sim p(x)p(z)}\Bigg[\log\Bigg(1-\sigma\Bigg(\sum_{v_n} z_u^T z_{v_n} I(z_u^T; z_{v_n})\Bigg)\Bigg)\Bigg]\Bigg)\\ &-\beta * \frac{1}{c}\sum_{i}^{c}\Bigg(\Bigg[\log\sigma\Bigg(\sum_{v\in\mathcal{N}_u} z_u^T z_v^{(i)} I(z_u^T; z_v^{(i)})\Bigg)\Bigg]\\ &+\mathbb{E}_{(x,z)\sim p(x)p(z)}\Bigg[\log\Bigg(1-\sigma\Bigg(\sum_{v_n} z_u^{(i)T} z_{v_n}^{(i)} I(z_u^{(i)T}; z_{v_n}^{(i)})\Bigg)\Bigg)\Bigg]\Bigg)\\ &+\gamma * \left|Z^T Z - I\right|\Bigg)\end{aligned} \tag{10}$$

Table 1: Statistics of datasets used in our experiment. * The task on PPI is a multilabel classification problem.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,278	1,433	7
Citeseer	3,312	4,660	3,703	6
Wiki	2,405	12,761	4,973	17
PubMed	19,717	44,327	500	3
Reddit	232,965	11,606,919	602	41
PPI	56,944	806,174	50	121*

4 EXPERIMENTS

4.1 Settings

Dataset. To evaluate the proposed method, we used four established benchmark datasets (Cora, Citeseer, Wiki and PubMed) in the node clustering task and transductive node classification task, and two benchmark datasets (Reddit and PPI) in inductive node classification task. Table 1 gives a summary of these datasets.

Cora, Citeseer and PubMed are academic citation networks in which each node of network is an article and each edge of network is a citation. Features of each node are expressed in a bag-of-words representation extracted from the corresponding article. Wiki is a web-page linking network in which each node is a web page. The edges of Wiki denote the hyperlink between web pages, and the feature of each node is also a bag-of-words representation. The node label in these four datasets refers to the topics that a node belongs to. We use the public datasets provided by DANE [15].

Reddit is a large-scale social network in which each node represents a different post. The node label in Reddit is the topical community that the node belongs to [18]. PPI is a protein-protein interactions dataset consists of multiple graphs, each graph in PPI is related to a different human tissue [32], the node labels in PPI reveal the protein functions that the node have. Reddit and PPI are available in the website: http://snap.stanford.edu/graphsage/, provided by SAGE [18].

Baselines. For node clustering task and transductive classification task, we compare InfomaxANE against the following baselines:

SVD-att, SVD-adj, SVD-mix: node embeddings are reduced vectors coming from features, adjacency and the concatenation of both using SVD.

LINE [33]: LINE employs Skip-Gram with negative sampling to learn the node embedding, the node pairs come from neighbors or two-hop away nodes.

Node2vec [5]: Node2vec uses Skip-Gram with negative sampling to learn the node embedding, the node pairs are generated by a flexible random walk balancing the BFS and DFS.

GAE/VGAE [34]: GAE/VGAE is a combination of GCN and auto-encoder/variational auto-encoder.

SAGE [18]: SAGE learns the node embedding by sampling and aggregating the features of a node's K-order neighborhood. We use the same negative sampling strategy with InfomaxANE in SAGE for better comparison.

DANE [15]: DANE captures the underlying high non-linearity in both topological structure and attributes with deep auto-encoders.

Table 2: Parameters setting in InfomaxANE.

Parameter	Explanation	Cora	Citeseer	Wiki	PubMed	Reddit	PPI
K	Layers of aggregator	2	2	1	2	2	2
c	Number of local embeddings	4	8	8	4	8	8
h	Negative sampling range	5	5	4	3	-	-
lr	Learning rate	0.7	0.7	0.1	0.7	0.01	0.00001
α	Global mutual information loss	2	1	1	2	1	0.1
β	Local mutual information loss	0.5	0.1	0.1	0.5	1	0.1
γ	Constraint loss	1	0.1	1	1	1	0.1

For inductive node classification task, we adopt the following advanced baselines:

DGI [21]: maximize the mutual information between patch representations and summaries of corresponding graphs.

GMI [22]: maximize mutual information from two aspects (node features and topological structures).

Parameters settings. The key parameters settings of baselines were as follows: For SVDs (including SVD-feat, SVD-adj, SVD-mix), we selected the optimal result within the embedding dimension range of [16, 32, 64, 128]. For LINE [33], the number of negative samples was set as 5, the final embedding was the concatenation of first-order embedding and second-order embedding. For Node2vec [5], the embedding dimension was set as 128, the window size was 10, the walk length was 80 and the number of walks for each node was 10. For GAE/VGAE [34], the embedding dimension was 16, the hidden size was 32. For SAGE [18], the embedding dimension was set as 128, the layers of aggregation was set as 2, the number of negative samples was 5, and batch-size was 256. For DANE [15], we use the default parameters provided by source codes on all experiment datasets. In order to obtain a more impartial evaluation results, some parameters assignment of baseline models are optimized, for example, by adopting the same negative sampling with InfomaxANE, the performance of SAGE is significantly improved compared with its original version.

For proposed InfomaxANE, we fixed the output dimensions of encoder as 128, batch size as 256, the number of negative samples as 5. Table 2 gives the other parameter settings in InfomaxANE on different datasets. To make full use of aggregated features and achieve optimal results, for Reddit and PPI, we put each-layer aggregated features including original features into the same encoder, and concatenated the output embeddings, so that the final embedding dimension for Reddit was 128*3=384. We found it beneficial to take the concatenation of global embeddings and mean vectors of local embeddings as final node embeddings for PPI, so the final embedding dimensions for PPI was 128*3*2=768. [1]

4.2 Node clustering

For the node clustering task, we used K-means to cluster the learned embeddings of nodes, and we used clustering accuracy as the evaluation metric. Results of the clustering task are shown in Table 3. InfomaxANE was found to outperform the baselines (by 1.03% to 8.61%) on all four experiment datasets.

Table 3: Node clustering results. Elements in bold are the best results, those with underline are second-best. The "improvements" row shows the difference between the best and second best results.

Baselines	Cora	Citeseer	Wiki	PubMed
SVD-feat	50.41	64.49	33.22	36.26
SVD-adj	35.01	25.18	25.32	39.57
LINE	45.24	28.99	39.46	64.18
Node2vec	62.78	40.61	41.95	63.74
SVD-mix	35.04	24.28	31.48	39.59
GAE	66.03	55.01	43.24	67.05
VGAE	64.55	57.10	44.03	68.81
SAGE	63.44	61.47	41.58	71.35
DANE	71.53	45.53	41.29	64.58
InfomaxANE	**74.19**	**69.20**	**52.64**	**72.32**
improvements	**+2.66**	**+4.71**	**+8.61**	**+1.03**

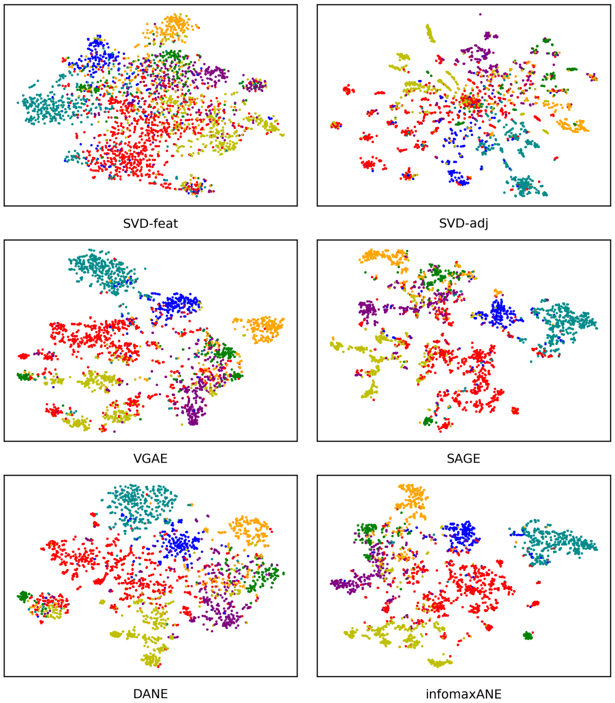
Figure 4 shows a visualization of the learned embeddings with t-SNE [35]. Due to limited space, only the visualization of partial baselines on dataset Cora are shown. Our aim was to group learned embeddings, so that those in the same category are close to each other, while those in different categories are further apart. We can observe that our approach achieves better clustering performance than the others for most cases.

4.3 Node classification

Transductive learning. For the transductive classification task, as with the evaluation setting with DANE [15], we employ one versus rest strategy to train a multi-class classifier with Logistic Regression as base classifier, different ratios of node labels (1%, 5%, 10%, 30%, 50%) are randomly selected to act as training set, and the rest nodes compose testing set. Micro-F1 and Macro-F1 are the metrics of node classification task. Table 4-7 are the summaries of transductive classification results. We can find that our proposed method InfomaxANE is significantly superior to the feature-only and structure-only approaches, and outperforms the other feature + structure approaches in most cases.

InfomaxANE clearly outperforms Cora (Table 4) and Citeseer (Table 5). For Wiki (Table 6) and PubMed (Table 7) the picture

[1] Codes and links to the datasets: https://github.com/lxm36/InfomaxANE


SVD-feat

SVD-adj

VGAE

SAGE

DANE

infomaxANE

Figure 4: Visualization results of different methods for Cora.

is more mixed. For Wiki, the best performing baseline was SVD-mix, a simple reduction on the concatenation of raw features and adjacency. It should be noted that the raw features in Wiki are discriminative. Raw features with SVD already manage to perform better than the other baselines in most cases. This is particularly noticeable when they are compared to feature-aggregated methods such as GAE/VGAE and SAGE. However, this is not true of the proposed InfomaxANE, which outperforms SVD-feat and performs almost as well as SVD-mix. We can infer from this finding that complicate operation may sometimes ruin the distinguishability of raw features, making it necessary to take more considerations in such cases. For PubMed, InfomaxANE comes a close to DANE, the best baseline. InfomaxANE performs better with the lower labeled ratios, while DANE outperforms with higher labeled ratios, and even with the higher labeled ratios, the differences are no more than 1.5%. Finally, compared to SAGE, InfomaxANE performs better on all the experiment datasets.

Inductive learning. For inductive node classification task, we feed the learned embeddings of nodes in training set into a logistic regression classifier and calculate the micro-F1 on test nodes, we repeat the classifier training for 50 times and take the average micro-F1 score as the final evaluation metric [21, 22]. Suggested by DGI [21], for PPI, we standardize the learned embeddings across training set before providing them to logistic regression classifier.

Table 8 reports the average micro-F1 scores of InfomaxANE and baselines. As observed, our proposed InfomaxANE manage to outperform all other competitive baselines, which substantiates the effectiveness of InfomaxANE. It should be mentioned that the node features in PPI is extreme sparse, around 1.9% of elements in feature matrix are nonzero values, besides, 42% of nodes have zero feature values [20], [32]. We consider the strong performance of InfomaxANE partially benefits from fully usage of different level aggregated features.

4.4 Module analysis

To establish the significance of each module in InfomaxANE, we trained several variants of the models derived from InfomaxANE, but with one module removed. The importance of different modules was then assessed by evaluating the node clustering task on the experimental datasets. The variant models' settings and node clustering results are shown in Table 9. Sage and DANE are taken as baselines, while Sage is optimized by tuning parameters and adopting new negative sampling strategy, so the performance of Sage is better than its original version. The proposed InfomaxANE was better than the other variant models in most cases on Cora, Wiki and Pubmed, but for Citeseer, InfomaxANE-global gave the best results.

The proposed InfomaxANE benefits from feature similarities added in aggregation compared with InfomaxANE-no_sims, especially for Wiki, while for Citeseer and PubMed, the contribution of feature similarity is small. To further illustrate this phenomenon, average degree, average clustering coefficient, average feature density are calculated in Table 10, Wiki has the highest score on both structure and feature attributes, which could have positive contribution on the effectiveness of similarity matrix. Besides, compared with Pubmed, connections among each node and its neighbors are stronger in Cora. The variant model which only considers global mutual information loss (InfomaxANE-global), compared poorly with InfomaxANE, with decrements ranging from 0.42% to 2.03%. Infomax-local, which only considers local embeddings, performed poorly overall, the infomax-unconstained, which take both global mutual information loss and local mutual information loss into consideration but not constrain the mutual information between local embeddings, performs worse than infomaANE and close to InfomaxANE-global. Besides, compared with traditional trainable aggregator, the proposed aggregator in InfomaxANE could improve training speed by about 2 times on average, while the performance has no significant difference.

Three interesting observations emerge from these results: 1) Global embedding is the key component of the proposed method. Without it, performance would be poor. 2) local embedding is helpful, but it won't work without constraints. 3) Taken as a whole, InfomaxANE is the best of all the variants. Every module in InfomaxANE have its contributions to the final performance.

4.5 Running efficiency

Running efficiency is an important factor of the model evaluation except for task performance. SAGE is comparable to InfomaxANE since they have similar structure. Therefore, we compared the running efficiency of InfomaxANE with that of SAGE. We ran InfomaxANE and SAGE on dataset Cora three times (100 iterations each time) in the same running environment and calculated the average running time for a single iteration. The results are 7.75 seconds and 6.71 seconds respectively. InfomaxANE is around one second

Table 4: Node classification result of Cora.

Labeled ratio	1%		5%		10%		30%		50%	
	micro	macro	micro	macro	micro	macro	micro	macro	micro	macro
SVD-feat	49.38	44.57	66.34	63.22	68.50	64.95	70.73	67.05	71.20	67.87
SVD-adj	40.69	39.35	60.05	59.34	66.78	64.70	74.00	72.19	76.37	74.64
LINE	47.22	44.93	65.92	63.09	72.15	70.85	79.11	77.58	81.02	79.89
Node2vec	62.40	58.15	72.21	70.87	75.39	73.47	79.43	77.76	80.65	79.63
SVD-mix	55.46	50.40	73.53	70.03	76.83	74.07	82.07	80.26	82.13	80.60
GAE	67.47	67.50	79.13	77.26	79.61	78.18	82.86	81.13	83.09	81.45
VGAE	63.33	63.95	74.93	71.57	76.83	74.90	79.43	77.61	81.09	79.28
SAGE	67.36	56.24	78.90	76.60	80.15	77.37	84.18	82.29	85.75	84.19
DANE	52.14	44.73	75.71	73.63	78.92	77.43	83.70	82.29	83.83	82.28
InfomaxANE	**75.42**	**72.54**	**81.15**	**79.60**	**83.92**	**82.49**	**87.34**	**85.60**	**88.63**	**87.12**
improvements	**+7.95**	**+5.04**	**+2.02**	**+2.34**	**+3.77**	**+4.31**	**+3.16**	**+3.31**	**+2.88**	**+2.93**

Table 5: Node classification result of Citeseer.

Labeled ratio	1%		5%		10%		30%		50%	
	micro	macro	micro	macro	micro	macro	micro	macro	micro	macro
SVD-feat	50.11	45.72	64.22	58.90	67.06	62.06	71.37	66.39	70.77	66.22
SVD-adj	24.70	19.27	42.10	37.66	46.13	41.96	49.98	45.14	52.60	46.77
LINE	29.06	23.69	43.76	39.48	48.81	45.05	58.52	55.27	63.41	59.54
Node2vec	40.35	35.22	45.22	41.21	50.59	46.76	55.07	51.60	57.19	53.18
SVD-mix	55.96	49.08	66.76	59.92	66.02	60.14	71.19	65.82	71.98	67.11
GAE	49.62	44.08	64.38	55.67	65.58	57.24	69.00	61.21	68.24	60.54
VGAE	48.58	43.22	61.39	54.75	64.81	56.22	66.15	57.63	65.04	56.33
SAGE	67.34	59.53	69.88	61.37	70.78	63.37	73.48	65.89	73.91	69.00
DANE	40.96	35.01	58.25	53.09	60.68	56.27	69.81	65.88	71.44	67.05
InfomaxANE	**70.81**	**64.09**	**71.78**	**65.21**	**73.06**	**66.27**	**75.59**	**70.47**	**75.79**	**70.76**
improvements	**+3.47**	**+4.55**	**+1.90**	**+3.84**	**+2.28**	**+2.90**	**+2.11**	**+4.08**	**+3.81**	**+1.76**

Table 6: Node classification result of Wiki.

Labeled ratio	1%		5%		10%		30%		50%	
	micro	macro	micro	macro	micro	macro	micro	macro	micro	macro
SVD-feat	47.88	27.02	64.73	53.68	69.10	57.44	75.59	64.60	76.64	66.21
SVD-adj	26.29	14.88	45.51	37.53	51.93	42.58	59.80	47.31	63.26	50.42
LINE	26.08	14.66	49.76	40.52	57.32	47.42	62.89	49.34	66.83	54.41
Node2vec	26.46	14.47	51.74	43.13	57.04	44.62	63.02	46.18	65.09	52.83
SVD-mix	**48.09**	**27.28**	69.06	56.75	73.44	61.59	**79.22**	**69.34**	**79.88**	67.54
GAE	29.53	15.34	57.46	44.07	60.60	46.49	63.66	49.05	65.42	49.86
VGAE	29.90	14.35	52.65	38.88	58.01	41.96	62.00	43.51	63.84	46.89
SAGE	27.59	14.29	59.17	43.31	66.05	52.48	68.88	53.27	71.07	58.16
DANE	41.23	21.34	66.52	52.69	72.89	60.07	75.30	63.18	78.89	**70.04**
InfomaxANE	45.44	23.47	**70.81**	**57.34**	**74.69**	**62.80**	77.26	66.14	79.55	69.23
improvements	**-2.65**	**-3.81**	**+1.75**	**+0.59**	**+1.25**	**+1.21**	**-1.96**	**-3.20**	**-0.33**	**-0.81**

Table 7: Node classification result of PubMed.(%)

Labeled ratio	1%		5%		10%		30%		50%	
	micro	macro	micro	macro	micro	macro	micro	macro	micro	macro
SVD-feat	75.66	76.21	82.19	82.26	83.48	83.54	84.39	84.44	85.06	85.18
SVD-adj	68.55	66.49	75.39	73.29	76.77	75.02	77.97	76.46	78.39	76.94
LINE	71.06	68.66	78.50	76.65	79.44	77.85	80.32	79.02	80.53	79.22
Node2vec	72.11	69.86	74.95	72.92	75.92	74.24	77.02	75.46	78.02	76.81
SVD-mix	79.89	79.59	83.55	83.17	84.38	84.04	85.12	84.82	85.42	85.21
GAE	81.02	80.34	81.99	81.29	82.12	81.50	82.17	81.54	82.27	81.71
VGAE	81.12	80.66	81.88	81.29	81.92	81.38	82.38	81.88	82.64	82.19
SAGE	82.27	81.44	82.85	82.02	83.26	82.58	83.75	83.15	84.01	83.46
DANE	80.61	80.17	84.88	84.47	86.22	**85.93**	**87.49**	**87.23**	**88.44**	**88.26**
InfomaxANE	**85.29**	**84.64**	**86.17**	**85.58**	**86.47**	85.87	86.83	86.28	87.35	86.85
improvements	**+3.02**	**+3.20**	**+1.29**	**+1.11**	**+0.25**	**-0.06**	**-0.66**	**-0.95**	**-1.09**	**-1.41**

Table 8: Average micro-F1 scores on inductive tasks (%).

Method	Reddit	PPI
Raw features	58.5	42.2
DeepWalk	32.4	-
DeepWalk+Features	69.1	-
SAGE-GCN	90.8	46.5
SAGE-mean	89.7	48.6
SAGE-LSTM	90.7	48.2
SAGE-pool	89.2	50.2
DGI	94.0 ±0.10	63.8 ±0.20
GMI-mean	95.0 ±0.02	65.0 ±0.02
InfomaxANE	**95.6 ± 0.01**	**67.9 ± 0.03**

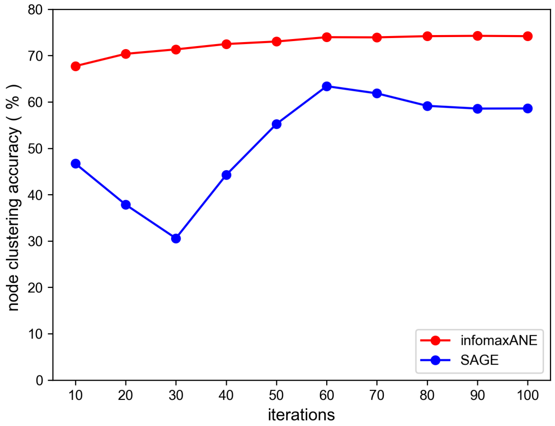


Figure 5: Running process of InfomaxANE and SAGE for dataset Cora

slower than SAGE in every iteration on average due to its larger number of parameters and more calculations in feature aggregation. Figure 5 is drawn to show the changing of node clustering accuracy along with the iterations. We can observe that InfomaxANE has a higher starting point and more stable rising curve, which show the efficiency and stability of InfomaxANE.

5 CONCLUSION

We propose InfomaxANE, an unsupervised attributed network embedding approach. We treat the node embedding problem as mutual information maximization between inputs and embeddings, not only global embedding but also local embedding of node features is concerned in InfomaxANE to better utilize the node information. We encourage local embeddings to learn different local information by constraining the mutual information between every two of them. We leverage feature aggregation to complete the combination of topological structure and node attributes but separate aggregation from encoding. We also add node feature similarity into feature aggregation for better service to the subsequent encoding work. Our proposed InfomaxANE successfully obtains higher quality embeddings according to the significant superior on node clustering task, transductive/inductive node classification task. We also verify the contributions of each module in InfomaxANE.

ACKNOWLEDGMENTS

This research was supported by Chinese National Nature Science Youth Foundation Research (No: 61702564); Soft Science Foundation of Guangdong Province (No: 2019A101002020); Young Teacher Training Project in Sun Yat-sen University (No: 20000-31610507).

REFERENCES

[1] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, 2018.
[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
[3] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.
[4] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

Table 9: Node clustering accuracy of variant models. α-global mutual information loss, β=local mutual information loss, γ=constraint loss, c=number of local embeddings, feat_sims=feature similarity is considered or not during aggregation.

Variant models	Settings	Cora	Citeseer	Wiki	PubMed
SAGE	dim=128, h=5	63.44	61.47	41.58	71.35
DANE	dim=100	71.53	45.53	41.29	64.58
InfomaxANE-no_sims	feat_sims=False	73.74	69.32	48.94	**72.62**
InfomaxANE-global	$\beta = 0, \gamma = 0$, c=1	72.16	**70.23**	50.98	71.90
InfomaxANE-local	$\alpha = 0$	58.38	32.82	43.70	67.14
InfomaxANE-unconstrained	$\gamma = 0$	72.01	69.84	50.44	71.49
InfomaxANE	-	**74.19**	69.20	**52.64**	72.32

Table 10: Statistical analysis of four datasets based on structure feature and attribute feature.

Statistical Attribute	Indicator	Cora	Citeseer	Wiki	PubMed
Structure Attribute	Average degree	3.90	2.81	**10.61**	4.50
	Average clustering coefficient	0.2407	0.1426	**0.3758**	0.06
Feature Attribute	Average feature density	0.0134	0.0088	**0.1303**	0.1020

[5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[6] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 40–48, 2016.

[7] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394, 2017.

[8] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zügner, and Stephan Günnemann. NetGAN: Generating Graphs via Random Walks. In *International Conference on Machine Learning*, pages 609–618, 2018.

[9] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *International Conference on Machine Learning*, pages 5708–5717, 2018.

[11] Aynaz Taheri, Kevin Gimpel, and Tanya Berger-Wolf. Learning graph representations with recurrent neural network autoencoders. *KDD Deep Learning Day*, 2018.

[12] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[13] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[14] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2257–2270, 2018.

[15] Hongchang Gao and Heng Huang. Deep attributed network embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3364–3370. AAAI Press, 2018.

[16] Yang Cheng, Deli Zhao, Deli Zhao, Edward Y. Chang, and Edward Y. Chang. Network representation learning with rich text information. In *International Conference on Artificial Intelligence*, 2015.

[17] Xiao Huang, Qingquan Song, Yuening Li, and Xia Hu. Graph recurrent networks with attributed random walks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 732–740, 2019.

[18] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NIPS*, 2017.

[19] Junliang Guo, Linli Xu, and Jingchang Liu. SPINE: structural identity preserved inductive network embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2399–2405. AAAI Press, 2019.

[20] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Devon Hjelm, and Aaron Courville. Mutual Information Neural Estimation. In *International Conference on Machine Learning*, pages 530–539, 2018.

[21] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[22] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, pages 259–270, 2020.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[24] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294*, 2018.

[25] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[26] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[27] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[28] Jie Hu, Rongrong Ji, ShengChuan Zhang, Xiaoshuai Sun, Qixiang Ye, Chia-Wen Lin, and Qi Tian. Information Competing Process for Learning Diversified Representations. In *Advances in Neural Information Processing Systems*, pages 2175–2186, 2019.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[30] Samuel Lavoie-Marchildon Karan Grewal Phil Bachman Adam Trischler Yoshua Bengio R Devon Hjelm, Alex Fedorov. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[31] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[32] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.

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

[34] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

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