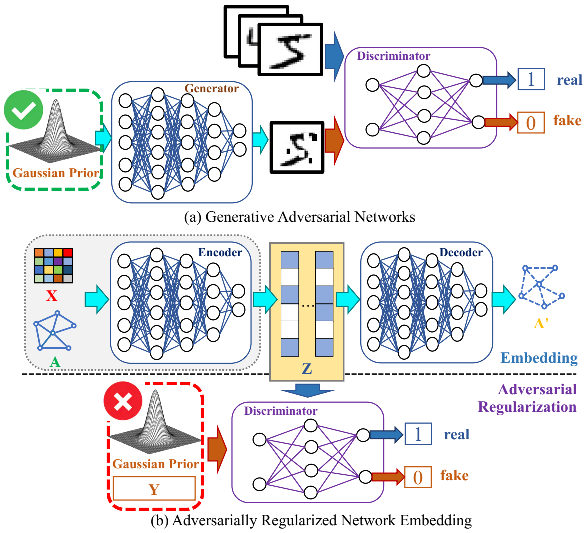
JANE: Jointly Adversarial Network Embedding

Liang Yang[1,2] , **Yuexue Wang**[1] , **Junhua Gu**[1*] , **Chuan Wang**[2] , **Xiaochun Cao**[2] , **Yuanfang Guo**[3]
[1]School of Artificial Intelligence, Hebei University of Technology, China
[2]State Key Laboratory of Information Security, Institute of Information Engineering, CAS, China
[3]School of Computer Science and Engineering, Beihang University, China
yangliang@vip.qq.com, wwwwwww1107@outlook.com jhgu@hebut.edu.cn, wangchuan@iie.ac.cn,
caoxiaochun@iie.ac.cn, andyguo@buaa.edu.cn

Abstract

Motivated by the capability of Generative Adversarial Network on exploring the latent semantic space and capturing semantic variations in the data distribution, adversarial learning has been adopted in network embedding to improve the robustness. However, this important ability is lost in existing adversarially regularized network embedding methods, because their embedding results are directly compared to the samples drawn from perturbation (Gaussian) distribution without any rectification from real data. To overcome this vital issue, a novel Joint Adversarial Network Embedding (JANE) framework is proposed to jointly distinguish the real and fake combinations of the embeddings, topology information and node features. JANE contains three pluggable components, Embedding module, Generator module and Discriminator module. The overall objective function of JANE is defined in a min-max form, which can be optimized via alternating stochastic gradient. Extensive experiments demonstrate the remarkable superiority of the proposed JANE on link prediction (3% gains in both AUC and AP) and node clustering (5% gain in F1 score).



(a) Generative Adversarial Networks

(b) Adversarially Regularized Network Embedding

Figure 1: The difference between the Generative Adversarial Networks and the existing Adversarially Regularized Network Embeddings. (a) In the former one, Gaussian prior possesses the capability of exploring the latent semantic space and capturing semantic variations by comparing the generated data with the real ones. (b) The latter one only regularizes the embedding results by discriminating the generated drawn from a Gaussian distribution.

1 Introduction

Traditional network analysis designs exclusive end-to-end approaches for different tasks, such as node classification, community detection and link prediction, etc. Motivated by representation learning, network embedding unifies many network analysis tasks as a node representation learning framework. Thus, the node representation learning plays a vital role in the latter development stage of network analysis. In the past five years, many network embedding approaches [Cai *et al.*, 2018; Shen *et al.*, 2018] have been proposed. Some of them are motivated from language model, i.e., word2vec [Mikolov *et al.*, 2013], such as DeepWalk [Perozzi *et al.*, 2014] and node2vec [Grover and Leskovec, 2016]. Others adopt the mechanisms of topology reconstruction with matrix factorization or Auto-Encoder, such as SDNE [Wang *et al.*, 2016], NetMF [Qiu *et al.*, 2018] and TADW [Yang *et al.*, 2015]. Recently proposed Graph Neural Networks (GNNs) [Wu *et al.*, 2019b; Yang *et al.*, 2019c; Yang *et al.*, 2019a; Yang *et al.*, 2019b], such as Graph Convolutional Network (GCN) [Kipf and Welling, 2017] and Graph Attention Network (GAT) [Velickovic *et al.*, 2018], achieve remarkable performance in semi-supervised node representation classification.

Inspired by Generative Adversarial Network (GAN) [Goodfellow *et al.*, 2014], which is capable of exploring the latent semantic space and capturing semantic variation in the data distribution via a perturbation distribution [Donahue *et al.*, 2017], adversarial learning is progressively adopted into network embedding to improve robustness [Dai *et al.*, 2018; Pan *et al.*, 2018; Pan *et al.*, 2019]. As shown in Figure 1(a), this capability in GAN is obtained by comparing the generated fake data, which are produced based on the sam-

[*]Corresponding author.

ples drawn from a perturbation (usually Gaussian) distribution, with the real data. Unfortunately, this capability tends to be lost in adversarial learning based network embedding approaches, because the perturbation distribution cannot be rectified with real data. In fact, the samples generated from a perturbation distribution are directly compared to the embedding results, which are obtained from an auto-encoder, in adversarial learning based network embedding, as shown in Figure 1(b). For example, Pan et al. utilize adversarial learning to regularize the embeddings obtained from Graph Auto-Encoder [Kipf and Welling, 2016] by comparing the embeddings with the samples drawn from a Gaussian distribution [Pan *et al.*, 2018]. This naive regularization only force the learned embeddings to be more consistent with the Gaussian distribution, rather than capturing the semantic variations. Therefore, existing network embedding approaches cannot effectively benefit from adversarial learning.

To overcome this deficit in existing adversarial learning based network embedding approaches, a novel Joint Adversarial Network Embedding (JANE) framework is proposed in this paper. Instead of directly comparing the embeddings with the samples drawn from a Gaussian distribution, JANE jointly distinguishes the real and fake combinations of the embeddings, topology information and node features. The combined topology information and node features rectifies the Gaussian distribution to capture the semantic variations in latent space, as in GAN. Specifically, our JANE framework consists of three pluggable components, Embedding module and Generator module and Discriminator module as shown in Figure 2. In the Embedding module, attention-based layer-wise propagation is adopted to seamlessly and flexibly combine the topology information and node features. The Generator module creates the fake topology information and node features from fake embeddings. To improve the efficiency, the fake embeddings are constructed by adding Gaussian noises to the embeddings obtained in the Embedding module, instead of directly sampling from a Gaussian distribution. The overall objective function of JANE is defined in a min-max form, which can be optimized via alternating stochastic gradient. In each iteration, the Discriminator module is optimized by maximizing the objective function w.r.t. its parameters, then the Generator and Embedding modules are optimized by minimizing it w.r.t. their parameters. The main contributions of this paper are summarized as follows:

- We analyze the adversarial mechanism in existing adversarially regularized network embedding methods, and reveal their inabilities to capture semantic variations.
- We propose a novel Jointly Adversarial Network Embedding (JANE) framework with pluggable components to benefit the embedding methods from the adversarial mechanism.
- Extensive experiments on link prediction and node clustering demonstrate the remarkable superiority of the proposed JANE over 12 state-of-the-art methods.

2 Notations

Define an attribute network as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ with vertices $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ and edges $\mathcal{E} = \{e_1, e_2, ..., e_M\}$. The attributes of all the vertices are represented via an attribute matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ where the nth row of which, $\mathbf{x_n} \in \mathbb{R}^{1 \times F}$, corresponds to the attributes of vertex v_n in the form of a F-dimensional vector. The network topology is represented by an adjacency matrix $\mathbf{A} = [a_{ij}] \in \{0, 1\}^{N \times N}$, where $a_{ij} = 1$ if an edge exists between the vertices v_i and v_j and vice versa. Typical attribute network embedding problem seeks a low-dimensional representation $\mathbf{Z} \in \mathbb{R}^{N \times P}$ for all the vertices in $\mathcal{V}$, where P is the dimension of the embedding. For convenience, $\mathbf{Y} \in \mathbb{R}^{N \times P}$, which possesses the same dimension as the embedding matrix $\mathbf{Z}$, is employed to denote the N samples drawn from a specific prior distribution, e.g, Gaussian.

3 Motivations

The remarkable ability of Generative Adversarial Networks(GANs)[Goodfellow *et al.*, 2014], which learns to generate complex data distribution, such as images, from simple latent (Gaussian) distribution, demonstrates that GANs can effectively explore the latent semantic space and capture semantic variations in the data distribution [Donahue *et al.*, 2017]. Motivated by this, adversarial learning is adopted by network embedding [Dai *et al.*, 2018; Pan *et al.*, 2018; Pan *et al.*, 2019]. Pan et al. leverages GAN to regularize the embedding results of Graph Auto-Encoder [Kipf and Welling, 2016], which adopts Graph Convolutional Network (GCN) [Kipf and Welling, 2017] as the encoder and reconstructs the topology information (adjacency matrix) [Pan *et al.*, 2018]. They further extend this framework to reconstruct both the topology and node attributes [Pan *et al.*, 2019]. On the other hand, Dai et al. propose Inductive DeepWalk, which is the variant of DeepWalk [Perozzi *et al.*, 2014], and utilize GAN to regularize the embedding results of Inductive DeepWalk. Since DeepWalk is equivalent to factorizing the multi-hop normalized adjacency matrix [Qiu *et al.*, 2018], i.e, reconstructing the topology information, it can also be considered as adversarially regularizing the embedding results of an auto-encoder. Therefore, most existing network embedding approaches, which are developed based on adversarial learning, can be unified into a framework of adversarially regularizing the embedding results of an auto-encoder, as shown in Figure 1(b).

Typically, the existing methods directly regularize the embedding results against the samples generated from a Gaussian distribution. They iteratively optimize the encoder and the discriminator to constrain the generated (encoded) embedding results to be indistinguishable from the samples, which are generated from a Gaussian distribution. However, this regularization approach cannot guarantee the embedding results to be robust. It only indicates that the embedding results obey Gaussian distribution. This phenomenon is originated from the inappropriate usage of the adversarial learning mechanism. In GAN [Goodfellow *et al.*, 2014], the Gaussian prior can effectively capture the semantic variations in the data distribution, because the generated data are compared to the real ones, as shown in Figure 1(a). Unfortunately, the samples drawn from the Gaussian distribution are directly treated as fake data and compared to the real network embeddings in network embedding approaches. Without the

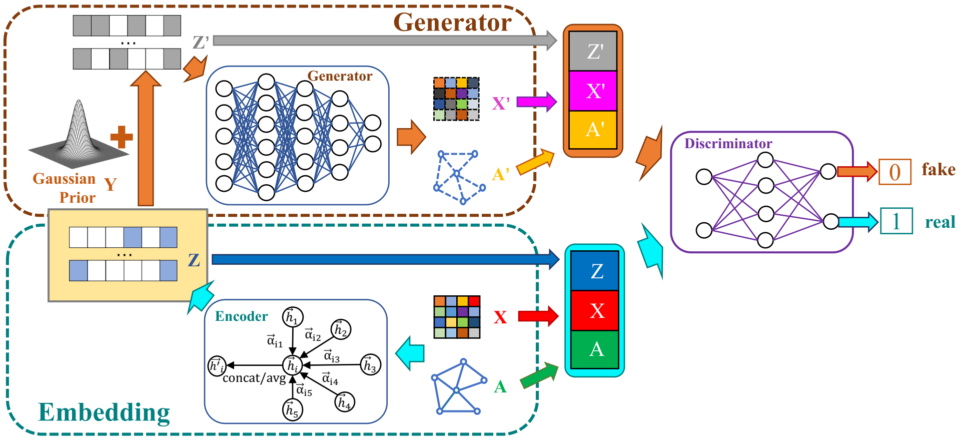


Figure 2: The proposed Joint Adversarial Network Embedding (JANE) framework. It consists of three pluggable components: Embedding module, Generator module and Discriminator module. All of them are replaceable. Different form the existing Adversarially Regularized Network Embedding, JANE discriminates the real (in cyan box) and fake (in orange box) combinations of embeddings, topology information and node features to constrain the Gaussian Prior to capture the semantic variations in latent space, as in GAN.

discrimination between the generated data and the real data, the Gaussian prior can hardly possess the ability of exploring the latent semantic space and capturing semantic variations. Since the existing network embeddings with adversarial learning cannot fully leverage the effectiveness of adversarial learning on semantic space exploration, they cannot significantly improve the performance of original embedding methods based on GAE [Kipf and Welling, 2016], which is demonstrated in Section 5.

4 Framework

In this section, a novel Joint Adversarial Network Embedding (JANE) framework is proposed. The overview is firstly provided, followed by its three specific components: discriminator, embedding and generator modules. Finally, our objective function and optimization details are given.

4.1 Overview

As summarized in Section 3, the main drawback in Adversarially Regularized Network Embedding is the inability to capture semantic variations due to the direct comparison between the embeddings and samples generated from a Gaussian prior. To overcome this vital issue, a novel Joint Adversarial Network Embedding (JANE) framework is proposed to jointly discriminate the real and fake combinations of topology, node features and embeddings, instead of directly comparing the embeddings with the samples drawn from Gaussian distribution. As illustrated in Figure 2, JANE consists of three pluggable components, Embedding module (E), Generator module (G), and Discriminator module (D). Although the philosophy of JANE is similar to that of GAN, as given in Figure 1(a), they still possess two major differences. Firstly, an Embedding module is introduced in JANE, because the intention of JANE is to produce network embeddings instead of generating new data(attribute network). Secondly, instead of generating fake attribute network completely based on the samples obtained from Gaussian distribution, as in GAN, the embedding results of real attribute network (as shown in yellow rectangle) are combined with the samples generated from Gaussian distribution to produce the fake embeddings, and then generate the fake attribute network. In the following, the adopted discriminator, embedding and generator methods are introduced. Note that the proposed JANE is an adversarial network embedding framework, thus its three components are all replaceable.

4.2 Discriminator Module

The most remarkable difference of JANE from the existing adversarial learning based network embedding approaches is the input to the discriminator. To rectify the Gaussian prior with real data, the discriminator of our proposed JANE distinguishes the real and fake combinations of topology, node features and embedding results, as

$$\mathbf{R} = (\mathbf{Z}||\mathbf{A}||\mathbf{X}), \quad \mathbf{R}' = (\mathbf{Z}'||\mathbf{A}'||\mathbf{X}'), \tag{1}$$

where $\mathbf{A}$ and $\mathbf{X}$ are the given real adjacency matrix and node features, $\mathbf{Z}$ and $\mathbf{Z}'$ are the real and fake embeddings, $\mathbf{A}'$ and $\mathbf{X}'$ are the generated fake adjacency matrix and node features. $||$ denotes the concatenation operation. This kind of input possesses two characteristics. Firstly, topology and node features are both included into the input of our discriminator. This is motivated by GAN [Goodfellow *et al.*, 2014; Donahue *et al.*, 2017] that only the comparison between the real and fake data can constrain the Gaussian prior to capture semantic variations in latent embedding space. Secondly, instead of directly comparing embedding results with the samples drawn from Gaussian distribution as in Figure 1(b), JANE compares real and fake embeddings, because we intend to seek network embeddings instead of generating new attribute network. To distinguish whether the input combination is real, the discriminator is built on a multi-layer perceptron (MLP) parameterized by $\mathbf{W}_D$ and the output layer only has one dimension based on a sigmoid function.

4.3 Embedding Module

The embedding module encodes the attribute network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ as a collection $\mathbf{Z}$ of real value vectors, each row of which corresponds to a vertex. Recently, graph neural networks (GNNs) achieve state-of-the-art in semi-supervised node classification on attribute networks [Wu *et al.*, 2019b]. Motivated from a first-order approximation of the spectral graph convolution, Graph Convolutional Network (GCN) [Kipf and Welling, 2017] adopts layer-wise attribute propagations to augment the node features. Although GCN significantly improves the performance of node classification, its main drawback is the fixed propagation weights, which are completely determined by the degrees of the two connected nodes [Wu *et al.*, 2019a; Li *et al.*, 2018]. To make the embedding module flexible, the propagation weights are assumed to be learnable according to the features in the two connected nodes. Here, self-attention [Bahdanau *et al.*, 2015; Wang *et al.*, 2019] is adopted to estimate the weights, similar to Graph Attention Network (GAT) [Velickovic *et al.*, 2018],

$$\begin{aligned}\alpha_{ij} &= \mathrm{softmax}(e_{ij}) \qquad (2)\\ &= \frac{\exp(\mathrm{LeakyReLU}(\mathbf{b}^T[\boldsymbol{\Theta}\mathbf{x}_i^T || \boldsymbol{\Theta}\mathbf{x}_j^T]))}{\sum_{k\in\mathcal{N}(i)} \exp(\mathrm{LeakyReLU}(\mathbf{b}^T[\boldsymbol{\Theta}\mathbf{x}_i^T || \boldsymbol{\Theta}\mathbf{x}_k^T]))},\end{aligned}$$

where $\mathbf{x}_i$ is the feature of vertex v_i, i.e., the i^{th} row of $\mathbf{X}$, $\boldsymbol{\Theta} \in \mathbb{R}^{F\times P}$ is the mapping from features to embeddings, and $\mathbf{b} \in \mathbb{R}^{2P}$ is the attention weights to be learned. Then, the node embeddings can be obtained by averaging its weighted neighbourhoods as

$$\mathbf{z}_i = \mathrm{ReLU}\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\boldsymbol{\Theta}\mathbf{x}_j\Big). \qquad (3)$$

For stability, multi-head self-attention [Vaswani *et al.*, 2017] is adopted to boost the performance. Since JANE aims to construct node embeddings instead of node classification, the parameters $\mathbf{b}$ and $\boldsymbol{\Theta}$ cannot be directly obtained with the supervision from the given node label information. In JANE, they are obtained with the weak supervision from the real and fake labels of the combinations. Note that the current embedding module can be replaced with other approaches as long as they possess similar functionalities.

4.4 Generator Module

Since the inputs to the discriminator are the real and fake combinations of the topology, node features and the embeddings, the task of our Generator module is to generate the fake combination based on the samples drawn from a Gaussian distribution. Inspired by GAN, the most straightforward approach is treating the samples generated from a Gaussian distribution $\mathbf{Y} \in \mathbb{R}^{N\times P}$, which possesses the same dimension as the embedding matrix $\mathbf{Z}$, as the fake embeddings, and then generate fake topology and node features. However, it is ineffective for the high-dimensional output, such as adjacency matrix of large network. Since the samples drawn from Gaussian distribution tend to capture the semantic variations of the latent space, the fake embeddings can be constructed from the obtained real embeddings as

$$\mathbf{Z}' = \mathbf{Z} + \mathbf{Y}. \qquad (4)$$

Then, the fake node features and adjacency matrix can be respectively generated from $\mathbf{Z}'$. Here, the simplest generators are adopted for demonstration. The fake node features are generated by feeding the fake embeddings into a fully-connected layer parameterized by $\mathbf{W}_G \in \mathbb{R}^{P\times F}$, as

$$\mathbf{X}' = \mathrm{LeakyReLU}(\mathbf{Z}'\mathbf{W}_G), \qquad (5)$$

where the nonlinear activation function $\mathrm{LeakyReLU}(.)$ is adopted with a negative input slope $\alpha = 0.2$. The fake adjacency matrix is generated by multiplying the fake embedding $\mathbf{Z}'$ with its transposition as

$$\mathbf{A}' = \mathrm{sigmoid}(\mathbf{Z}'\mathbf{Z}'^{\mathbf{T}}), \qquad (6)$$

because previous work indicates that many embedding methods are equivalent to factorizing the topology information matrix, such as multi-hop adjacency matrix [Qiu *et al.*, 2018]. To improve the nonlinearity, the $\mathrm{sigmoid}(.)$ activation function is adopted here. Note that the above generators can also be replaced with others possessing similar functionalities.

4.5 Objective Function and Optimization

The JANE framework consists of three pluggable components, Embedding module (E), Generator module (G), and Discriminator module (D). Let $p_{\mathbf{AX}}$ be the joint distribution of the topology and node features for $(\mathbf{a}, \mathbf{x}) \in \Omega_{\mathbf{A}\times\mathbf{X}}$. The encoder $E : \Omega_{\mathbf{A}\times\mathbf{X}} \rightarrow \Omega_{\mathbf{Z}}$ induces a distribution $p_E(\mathbf{z}|\mathbf{a}, \mathbf{x}) = \delta(\mathbf{z} - E(\mathbf{a}, \mathbf{x}))$, which maps topology $\mathbf{a}$ and node features $\mathbf{x}$ into the latent space $\mathbf{z}$. The generator $G : \Omega_{\mathbf{Z}} \rightarrow \Omega_{\mathbf{A}\times\mathbf{X}}$ generates topology and node attributes from the embeddings with $p_G(\mathbf{a}, \mathbf{x}|\mathbf{z}) = \delta((\mathbf{a}, \mathbf{x}) - G(\mathbf{z}))$ and $p_G(\mathbf{a}, \mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}[p_G(\mathbf{a}, \mathbf{x}|\mathbf{z})]$. The discriminator $D : \Omega_{\mathbf{Z}\times\mathbf{A}\times\mathbf{X}} \rightarrow \{0, 1\}$ takes the inputs from the combinations of embeddings, topology information and node features, and then predicts $P_D(L|\mathbf{z}, \mathbf{a}, \mathbf{x})$, where $L = 1$ if topology $\mathbf{a}$ and node features $\mathbf{x}$ are real, i.e., sampled from the real data distribution $p_{\mathbf{AX}}$, and $L = 0$ if topology $\mathbf{a}$ and node features $\mathbf{x}$ are generated, i.e., the output of $G(\mathbf{z})$, with $\mathbf{z} \sim p_{\mathbf{Z}}$. The objective function of JANE is defined in a min-max form as

$$\min_{G,E}\max_{D} V(D, E, G), \qquad (7)$$

where $V(D, E, G)$ is defined as

$$\begin{aligned}V(D,E,G) := {}& \mathbb{E}_{(\mathbf{a},\mathbf{x})\sim p_{\mathbf{AX}}}\Big[\underbrace{\mathbb{E}_{\mathbf{z}\sim p_E(.|\mathbf{a},\mathbf{x})}[\log D(\mathbf{z}, \mathbf{a}, \mathbf{x})]}_{\log D(E(\mathbf{a},\mathbf{x}),\mathbf{a},\mathbf{x})}\Big]\\ &+\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\Big[\underbrace{\mathbb{E}_{(\mathbf{a},\mathbf{x})\sim p_G(.|\mathbf{z})}[\log(1 - D(\mathbf{z}, \mathbf{a}, \mathbf{x}))]}_{\log(1-D(\mathbf{z},G(\mathbf{z})))}\Big].\end{aligned}$$

Note that our Embedding module (E), Generator module (G), and Discriminator module (D) are parameterized by $(\boldsymbol{\Theta}, \mathbf{b})$, $\mathbf{W}_G$ and $\mathbf{W}_D$, respectively. This min-max objective function can then be optimized via the same alternating stochastic gradient as that in GAN [Goodfellow *et al.*, 2014]. In each iteration, the discriminator parameter $\mathbf{W}_D$ is updated by taking one or more steps in the positive gradient direction, $\nabla_{\mathbf{W}_D} V(D, E, G)$, then the embedding parameters $(\boldsymbol{\Theta}, \mathbf{b})$ and generator parameters $\mathbf{W}_G$ are together updated by taking a step in the negative gradient direction, $-\nabla_{\boldsymbol{\Theta},\mathbf{b},\mathbf{W}_G} V(D, E, G)$.

Dataset	#Nodes	#Edges	#Classes	#Features
CiteSeer	3,327	4,732	6	3,703
Cora	2,708	5,429	7	1,433
PubMed	19,717	44,338	3	500

Table 1: Datasets.

Methods	Cora		Citeseer		PubMed	
	AUC	AP	AUC	AP	AUC	AP
Spectral	84.61	88.50	80.51	85.01	84.22	87.81
DeepWalk	83.11	85.00	80.52	83.61	84.40	84.10
GAE	91.02	92.03	89.54	89.95	96.40	96.50
VGAE	91.41	92.61	90.82	92.02	94.42	94.72
ARGA	92.43	93.23	91.93	93.03	96.81	97.11
ARVGA	92.44	92.64	92.43	93.03	96.51	96.81
JANE	**96.62**	**96.21**	**97.42**	**96.99**	**97.34**	**97.22**

Table 2: Link prediction results.

5 Evaluations

To demonstrate the superiority of the proposed JANE framework, link prediction and node clustering tasks on three widely used citation networks (Cora, Citeseer and Pubmed as shown in Table 1) are adopted. In each network, the nodes represent the scientific publications, while the edges denote the citations. Each node feature is a bag-of-words representation of a publication. Note that the publications are divided into groups according to their research fields.

Experimental settings. For all the experiments, two attention layers with 8 and 1 attention heads are adopted for attention-based embedding module, and three fully-connected layers are employed in both the discriminator and generator. For fair comparison, the dimension of each embedding, i.e. P, is set to 16 for all the methods. Adam optimizer is adopted with the initial learning rates for the discriminator and other two components as 0.001 and 0.008, respectively. Both L2 regularization and dropout are exploited.

5.1 Link Prediction

For the link prediction task, the proposed JANE is compared to 6 state-of-the-art baselines, including Spectral Clustering method [Tang and Liu, 2011], standard DeepWalk [Perozzi *et al.*, 2014], Graph Auto-Encoder (GAE) and its variational extension VGAE [Kipf and Welling, 2016], Adversarially Regularized Graph Auto-Encoder (ARGA) and its variational extension ARVGAE [Pan *et al.*, 2018]. Two metrics, i.e., Area Under Curve (AUC) and Average Precision (AP), are adopted to quantify the performances according to the evaluation protocol in GAE [Kipf and Welling, 2016]. For each citation network, the edges are randomly divided into three groups. 85%, 5% and 10% of the edges are utilized in training, validation (hyper-parameters tuning) and performance testing, respectively. For each network, experiments are repeated 10 times on 10 different random edge partitions, and the average performances are reported in Table 2.

Methods	ACC	Precision	F1	NMI	ARI
K-Means	0.492	0.369	0.368	0.321	0.230
Spectral	0.367	0.193	0.318	0.127	0.031
GraphEncoder	0.325	0.182	0.298	0.109	0.006
DeepWalk	0.484	0.361	0.392	0.327	0.243
DNGR	0.419	0.266	0.340	0.318	0.142
RTM	0.440	0.332	0.307	0.230	0.169
RMSC	0.407	0.227	0.331	0.255	0.090
TADW	0.560	0.396	0.481	0.441	0.332
GAE	0.596	0.596	0.595	0.429	0.347
VGAE	0.609	0.609	0.609	0.436	0.346
ARGA	0.640	0.646	0.619	0.449	0.352
ARVGA	0.638	0.624	0.627	0.450	0.374
JANE	**0.726**	**0.715**	**0.715**	**0.532**	**0.517**

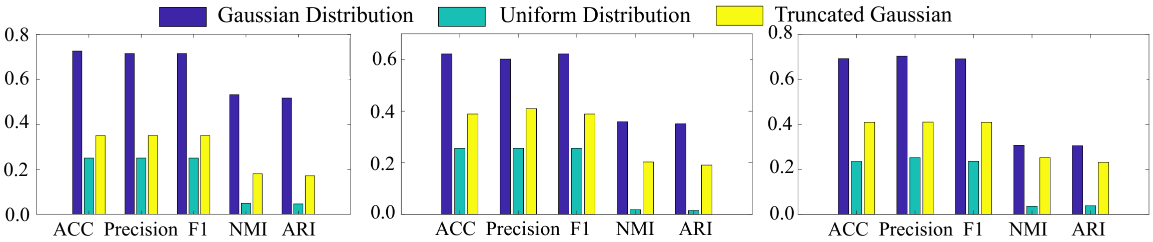
Table 3: Node clustering results on Cora.

As can be observed, the performance improvement of ARGA over GAE, which is the basis of ARGA, is only about 1%, because of the ineffective adoption of the adversarial mechanism. The proposed JANE consistently and significantly outperforms the state-of-the-arts. It achieves about 3% performance gains (both in AUC and AP) in average, compared to ARGA, where an adversarial learning between the embeddings and input samples is exploited. These gains are attributed to the novel adversarial learning between the real and fake combinations of embeddings, topology and node features, and the adopted attention-based embedding method.

5.2 Node Clustering

For the node clustering task, another 6 state-of-the-art methods are adopted besides of the 6 baselines mentioned in Section 5.1, including K-Means, Graph Encoder [Tian *et al.*, 2014], Deep Representations for Graph Clustering (DNGR) [Cao *et al.*, 2016], Relational Topic Models (RTM) [Chang and Blei, 2009], Robust Multi-View Spectral Clustering (RMSC) [Xia *et al.*, 2014] and Text-Associated DeepWalk (TADW) [Yang *et al.*, 2015]. All the baselines can be categorized into 4 classes, methods based on topology, methods based on both the topology and node features, methods based on GNNs and methods based on adversarial learning. To characterize the clustering performance, 6 metrics are utilized, including Accuracy (ACC), Precision, Normalized Mutual Information (NMI), Average Rand Index (ARI) and F1 score, which combines both the precision and recall rates.

The results on the Cora, Pubmed and Citeseer networks are shown in Tables 3, 4 and 5, respectively. The specific clustering performances of our proposed JANE remarkably surpass other baselines in almost all the metrics. For example, JANE achieves 9%, 8% and 2% gains on Cora, Citeseer and Pubmed, respectively, according to the F1 score. JANE's superior performances demonstrate that it can effectively and efficiently explore the latent space and capture the variations in semantic space. Note that the performance gain of JANE is less significant in NMI, which is more sensitive to the results of small cluster, compared to the results in other metrics. It indicates that most of our gains come from the large clusters.



(a) Cora (b) Citeseer (c) Pubmed

Figure 3: The impacts of different perturbation distributions on the clustering performances on three citation networks.

Methods	ACC	Precision	F1	NMI	ARI
K-Means	0.398	0.579	0.195	0.001	0.002
Spectral	0.403	0.498	0.271	0.042	0.002
GraphEncoder	0.531	0.456	0.506	0.209	0.184
DeepWalk	0.684	0.686	0.670	0.279	0.299
DNGR	0.458	0.629	0.467	0.155	0.054
RTM	0.574	0.455	0.444	0.194	0.148
RMSC	0.576	0.482	0.521	0.255	0.222
TADW	0.354	0.336	0.335	0.001	0.001
GAE	0.672	0.684	0.660	0.277	0.279
VGAE	0.630	0.630	0.634	0.229	0.213
ARGA	0.656	0.672	0.646	0.297	0.290
ARVGA	0.671	0.685	0.670	0.290	**0.305**
JANE	**0.692**	**0.703**	**0.691**	**0.307**	**0.305**

Table 4: Node clustering results on PubMed.

Methods	ACC	Precision	F1	NMI	ARI
K-Means	0.540	0.405	0.409	0.305	0.279
Spectral	0.239	0.179	0.299	0.056	0.010
GraphEncoder	0.225	0.179	0.301	0.033	0.010
DeepWalk	0.337	0.248	0.270	0.088	0.092
DNGR	0.326	0.200	0.300	0.180	0.044
RTM	0.451	0.349	0.342	0.239	0.203
RMSC	0.295	0.204	0.320	0.139	0.049
TADW	0.455	0.312	0.414	0.291	0.228
GAE	0.408	0.418	0.327	0.176	0.124
VGAE	0.344	0.349	0.308	0.156	0.093
ARGA	0.573	0.573	0.546	0.350	0.341
ARVGA	0.544	0.549	0.529	0.261	0.245
JANE	**0.622**	**0.602**	**0.622**	**0.359**	**0.351**

Table 5: Node clustering results on Citeseer.

5.3 Impacts of the Perturbation Distributions

By distinguishing the real and fake combinations instead of only the embeddings, JANE constrains the perturbation distribution to capture the semantic variations of the latent space. Thus, the impacts of different perturbation distributions, including Gaussian, Truncated Gaussian and Uniform distributions, on the node clustering task is verified. In Truncated Gaussian distribution, the sampled values, whose magnitude is more than 2 standard deviations from the mean, are dropped. The results, which are shown in Figure 3, demonstrate that the original Gaussian perturbation distribution is more suitable for semantic variation exploration than the others. These phenomenons satisfy the law of large numbers. The worst performance of Uniform distribution can be attributed to its inability of capture the latent space structure due to its uniform property. The performance of Truncated Gaussian distribution is between those of Gaussian and Uniform distributions, because the truncation tends to cause improper fitting to the latent space structure. This experiment illustrates that the semantic variations of latent space can be captured by distinguishing the real and fake combinations of the embeddings, topology and features.

6 Conclusions and Discussion

In this paper, the widely used adversarial mechanism in existing network embedding methods is analyzed and questioned, because they are incapable to capture semantic variations in latent space, which can be attributed to the direct comparison of the embedding results and the samples drawn from Gaussian prior without any rectifications from real data. To overcome this vital issue, a novel Jointly Adversarial Network Embedding (JANE) framework with pluggable components is proposed to benefit the network embedding methods from the adversarial mechanism. Remarkable performance improvements on link prediction and node clustering tasks have been achieved by the proposed JANE, which verifies that the proposed JANE can effectively explore the latent space and capture the semantic variations The experiments for studying the impacts of different perturbation distributions demonstrate the importance of capturing the semantic variations with proper prior distribution.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2017YFC0820106, in part by the National Natural Science Foundation of China under Grant 61972442, Grant 61802282, Grant 61802391 and Grant U1936210, in part by the Key Program of the Chinese Academy of Sciences under Grant QYZDB-SSW-JSC003, in part by the Hebei Province Innovation Capacity Enhancement Project under Grant 199676146H, and in part by the Fundamental Research Funds for Central Universities.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Cai *et al.*, 2018] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE TKDE*, 30(9):1616–1637, 2018.

[Cao *et al.*, 2016] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *AAAI*, pages 1145–1152, 2016.

[Chang and Blei, 2009] Jonathan Chang and David M. Blei. Relational topic models for document networks. In *AISTATS*, pages 81–88, 2009.

[Dai *et al.*, 2018] Quanyu Dai, Qiang Li, Jian Tang, and Dan Wang. Adversarial network embedding. In *AAAI*, pages 2167–2174, 2018.

[Donahue *et al.*, 2017] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *ICLR*, 2017.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.

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

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, pages 3538–3545, 2018.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Pan *et al.*, 2018] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *IJCAI*, pages 2609–2615, 2018.

[Pan *et al.*, 2019] Shirui Pan, Ruiqi Hu, Sai-Fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics*, pages 1–13, sep 2019.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *SIGKDD*, pages 701–710, 2014.

[Qiu *et al.*, 2018] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*, pages 459–467, 2018.

[Shen *et al.*, 2018] Xiaobo Shen, Shirui Pan, Weiwei Liu, Yew-Soon Ong, and Quan-Sen Sun. Discrete network embedding. In *IJCAI*, pages 3549–3555, 2018.

[Tang and Liu, 2011] Lei Tang and Huan Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery*, 23(3):447–478, 2011.

[Tian *et al.*, 2014] Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *AAAI*, pages 1293–1299, 2014.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang *et al.*, 2016] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *SIGKDD*, pages 1225–1234, 2016.

[Wang *et al.*, 2019] Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: A deep attentional embedding approach. In *IJCAI*, pages 3670–3676, 2019.

[Wu *et al.*, 2019a] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019.

[Wu *et al.*, 2019b] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[Xia *et al.*, 2014] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, pages 2149–2155, 2014.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In *IJCAI*, pages 2111–2117, 2015.

[Yang *et al.*, 2019a] Liang Yang, Zhiyang Chen, Junhua Gu, and Yuanfang Guo. Dual self-paced graph convolutional network: Towards reducing attribute distortions induced by topology. In *IJCAI*, pages 4062–4069, 2019.

[Yang *et al.*, 2019b] Liang Yang, Zesheng Kang, Xiaochun Cao, Di Jin, Bo Yang, and Yuanfang Guo. Topology optimization based graph convolutional network. In *IJCAI*, pages 4054–4061, 2019.

[Yang *et al.*, 2019c] Liang Yang, Fan Wu, Yingkui Wang, Junhua Gu, and Yuanfang Guo. Masked graph convolutional network. In *IJCAI*, pages 4070–4077, 2019.