SPINE: Structural Identity Preserved Inductive Network Embedding

Junliang Guo  
University of Science and Technology of China  
guojunjll@mail.ustc.edu.cn

Linli Xu  
University of Science and Technology of China  
linlixu@ustc.edu.cn

Enhong Chen  
University of Science and Technology of China  
cheneh@ustc.edu.cn

ABSTRACT

Recent advances in the field of network embedding have shown that low-dimensional network representation is playing a critical role in network analysis. Most existing network embedding methods encode the local proximity of a node, such as the first- and second-order proximities. While being efficient, these methods are short of leveraging the global structural information between nodes distant from each other. In addition, most existing methods learn embeddings on one single fixed network, and thus cannot be generalized to unseen nodes or networks without retraining. In this paper we present SPINE, a method that can jointly capture the local proximity and proximities at any distance, while being inductive to efficiently deal with unseen nodes or networks. Specifically, we identify the structural identity of each node based on a high-order proximity of the network named Rooted PageRank, followed by a novel biased Skip-Gram Negative Sampling algorithm and a carefully designed objective function to encode local proximities and global structural identities of nodes simultaneously. Extensive experimental results on benchmark datasets demonstrate the superiority of the proposed framework over the state of the art.

CCS CONCEPTS
• Computing methodologies → Learning latent representations; Unsupervised learning;

KEYWORDS
network embedding, inductive representation learning, structural identity

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference Format:
Junliang Guo, Linli Xu, and Enhong Chen. 2018. SPINE: Structural Identity Preserved Inductive Network Embedding. In Proceedings of ACM Conference (Conference’18). ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Network embedding has been successfully applied in a wide variety of network-based machine learning tasks, such as node classification, link prediction, and community detection, etc [4, 12]. Different to the primitive network representation, which suffers from overwhelming high dimensionality and sparsity, network embedding aims to learn low-dimensional continuous latent representations of nodes on a network while preserving the structure and the inherent properties of the network, which can then be exploited effectively in downstream tasks.

Most existing network embedding methods approximate local proximities via random walks or specific objective functions, followed by various machine learning algorithms with specific objective functions to learn embeddings [5, 7, 23, 29]. Specifically, the local proximity of a node is approximated with the routine of learning the embedding vector of a node by predicting its neighborhood, inspired by the word embedding principle [21, 22] which learns the embedding vector of a word by predicting its context.

However, there still exist some potential issues that need further concerns. On one hand, local proximity preserved methods generally do not model nodes far from each other in practice. Meanwhile, in real-world network mining tasks, nodes that are far apart but close in structural identity, or in other words, take similar roles, should perform similarly on specific tasks. Figure 1 shows an example of how nodes with different roles perform in the information diffusion processes on social networks. Nodes with different colors indicate different roles in social networks, i.e., structural hole spanners (red nodes), opinion leaders (yellow nodes) and ordinary users (blue nodes) respectively [19, 33]. Intuitively, nodes with same roles behave similarly even with a large distance (yellow nodes in Figure 1), which is the property that should be preserved in the embedding space. In the meantime, the local proximity of a node is also crucial in network embedding. For example in Figure 1, nodes in the same community should be clustered tightly in the embedding space. Therefore, a desirable network embedding method should preserve the local proximity and the global structural identity of a
node simultaneously to represent the node precisely. Unfortunately, most existing methods fail to consider the local and global structural information at the same time. In principle, it is challenging to interactively integrate the two kinds of information to obtain comprehensive embeddings rather than a trivial linear combination.

On the other hand, most existing network embedding approaches are transductive. To be specific, embeddings are learned on a fixed network, and cannot be directly applied to new joined nodes or other networks. In contrast, inductive methods which are able to generalize to unseen nodes or totally new networks are extensively required in real-world applications, e.g., social recommendation for new users, classification of protein functions in various protein-protein interaction graphs, etc. Unfortunately, traditional network embedding principles such as random walk and matrix factorization are impracticable for unseen nodes or networks, which makes inductive network embedding much more challenging than transductive problems.

In this paper, we propose SPINE, an inductive network embedding framework which jointly preserves local proximities and structural identities of nodes. We show that structural similarities between node pairs can be represented by a high-order proximity of the network known as Rooted PageRank [16], and by assigning each node a structural feature vector based on Rooted PageRank, we can encode structural proximities between nodes by measuring the similarities of their structural features. To construct an inductive framework, we learn an embedding generator rather than directly optimizing a unique embedding for each node, through which local proximities are integrated. To further encode structural identities, we propose a biased Skip-Gram Negative Sampling (SGNS) approach with a novel positive sampling strategy guided by structural similarities between nodes. Furthermore, the objective function of SPINE is carefully designed to enhance the structural information contained in the embedding generator.

Our main contributions can be summarized as follows:

- We propose SPINE to jointly preserve local proximities and structural identities of nodes, which can be generalized to unseen nodes without retraining.
- We show that structural identities can be encoded with a well-known high-order proximity named Rooted PageRank.
- We augment the original SGNS by proposing a novel positive sampling strategy based on structural similarities between structural features of nodes. The adapted SGNS resolves the limitation of distance while preserving the basic local proximities between nodes.
- We empirically evaluate SPINE on real-world network mining tasks in both transductive and inductive settings on various datasets to show its superiority. We furthermore validate that the structural identities are perfectly preserved in the learned embeddings.

2 RELATED WORK

2.1 Structural Identity

Structural identity has been extensively studied for decades [1, 18, 25] since being proposed by sociologists. It describes the similarity between nodes according to their local network structures. At the early stage, structural identity is restricted to structural equivalence [1, 18], which identifies equivalent nodes with the same connection patterns to exactly the same neighbors. Obviously, this definition is too strict thus impractical for real-world networks. Regular equivalence is a relaxation of structural equivalence, which defines roles to represent the structural information of nodes. Specifically, nodes play the same role if they have similar connections to nodes of other roles [25]. With this definition, how to represent the roles of nodes becomes particularly important. The most common strategy is to assign a feature vector (e.g., node degree, neighbors’ degrees, etc.) for each node to represent its role, known as feature-based roles [25]. Therefore, it is important that the designed structural features should capture the structural identity of a node, such that nodes with the same roles should share similar structural feature vectors.

In this paper, we leverage Rooted PageRank to construct structural feature vectors for nodes, and show that our definition can correctly capture the structural similarities between nodes.

2.2 Network Embedding

The rapid growth of applications based on networks has brought significant attention to network embedding. Motivated by the advances of word embedding, a number of random-walk based network embedding methods have been proposed [7, 23]. Similarly in LINE [29] and GraRep [5], network embedding is learned by directly optimizing the objective function inspired from the Skip-Gram model. While the above methods only exploit the structural information of the network, many efforts have been made to incorporate heterogeneous information into network embedding [6, 10, 30, 32]. We refer readers to [4] for a comprehensive survey.

However, it is worth noting that few of the existing works above consider the structural identity between nodes, and fail to handle proximities between nodes at distances. To address that, struc2vec [24] approximates the structural identity by constructing a multi-layer weighted complete graph, on which a random walk is executed, followed by the Skip-Gram model to learn network embeddings. While preserving the structural identity, the basic local proximities of nodes are ignored in struc2vec, which limits its applicability on real-world network mining tasks. Up to now, the interactive integration of local proximity and global structural identity is still an open problem.

Furthermore, most of the existing methods, including struc2vec, are transductive. When dealing with newly arrived nodes or networks, the whole process has to be rerun to obtain new embeddings, which surely damages the efficiency of these methods. To address that, several dynamic network embedding methods [15, 20] are proposed to deal with new nodes. Nevertheless, these approaches cannot be generalized to totally unseen networks. Recently, Graphsage [8] tackles this challenge by recursively training a set of aggregators for each node to integrate its neighbors’ content as the embedding of the current node in every iteration, while Graph2Gauss [2] first represents nodes with Gaussian distributions, which are then optimized with ranking based loss to incorporate structural information. Although being efficient, Graph2Gauss only considers $k$-hop local proximities in their loss function, while
Graphsage and struc2vec both consider $k$-hop local structural prox- 

inities in a hierarchical manner, i.e., nodes at the $k$-th iteration or layer contain the structural information from their neighbors 

within $k$ hops. In practice, these methods cannot deal with nodes 

at arbitrary distances unless with sufficient iterations, which is 

costly for real-world tasks. Most recently, GAT [31] automatically 

learns the content aggregation weights between neighbors and the 

current node to achieve a supervised inductive node classification 

framework, which only considers local proximities as well. As a 

comparison, SPINE, as an unsupervised method, can achieve com-

petitive performance with a smaller scale of parameters, which will 

be shown empirically in Section 4.

In this paper, we propose an inductive network embedding 

framework from the point of jointly approximating local proximities and 

structural feature generation, embedding generation and biased 

SGNS optimization. Given an undirected network $G = (V, E, F)$, in 

which a set of nodes $V$ are connected by a set of edges $E$, and 

$F \in \mathbb{R}^{|V| \times |E|}$ is the content matrix of nodes. The adjacency matrix 

is $A$ where $A_{i,j} = w_{i,j}$ is the edge weight between node $v_i$ and 

$v_j$, and we denote the corresponding transition matrix as $P$, where 

$P_{i,j} = \frac{w_{i,j}}{\sum_{k=1}^{|V|} w_{i,k}}$ represents the transition probability between node 

$v_i$ and $v_j$.

3 SPINE

In this section, we propose Structural Identity Preserved Inductive 

Network Embedding (SPINE), a novel inductive approach for unsu-

ervised network embedding. SPINE consists of three components: 

structural feature generation, embedding generation and biased 

SGNS optimization. Following the examples in Definition 3.1, the centrality of 

$v_i \equiv v_j$ can be regarded as a complete node property of $v_i$. There are multiple 

ways to measure the centrality of a node, while the original 

PageRank [3] is exactly a variant of eigenvector centrality. At each iteration, the PageRank value $\pi_i$ of node $v_i$ is updated as in [13]:

$$\pi_i^T = \pi_i^T M.$$  

and $M$ is the Google matrix defined as:

$$M = \beta P + (1 - \beta) g_i I^T$$

where $g_i = (\frac{1}{|V|}, \cdots, \frac{1}{|V|})^T$ and $I = (1, \cdots, 1)^T$. According to 

Equation (1), as a variant of PageRank, the only difference between $S_i^{\text{RPR}}$ and the original PageRank is the choice of $g_i$. Therefore, the 
target matrix of Rooted PageRank can be written as:

$$M_i^{\text{RPR}} = \beta P + (1 - \beta) I$$

where $I$ is the identity matrix. Thus $S_i^{\text{RPR}}$ is the leading left hand eigenvector of $M_i^{\text{RPR}}$, i.e., $S_i^{\text{RPR}}$ satisfies: 

$$S_i^{\text{RPR}} = \frac{S_i^{\text{RPR}}}{M_i^{\text{RPR}}}.$$ 

As a consequence, $S_i^{\text{RPR}}$ is also a variant of eigenvector centrality, thus can be further regarded as a complete structural property to replace the structural identity of $v_i$, i.e., $T_i = S_i^{\text{RPR}}$.

3.2 Structural Feature Generation

So far we have verified that RPR preserves structural identities of 

nodes. In this section we first state our motivation of choosing $S_i^{\text{RPR}}$ as the structural identity instead of others, and proceed to introduce the structural feature generation method based on it.

To construct an inductive method, we expect the length of the structural description of a node is independent of the total number of nodes $|V|$ and fixed at $k$. In this paper, we use the top $k$ values of $S_i^{\text{RPR}}$ as the structural description of $v_i$. Compared with other high-order proximities or node properties (e.g., the original PageRank), Rooted PageRank captures the global structural information of the network, while being tailored to encode the local structural information of root nodes according to the definition, and thus can better represent the importance of neighbors at various distances to the present node [9]. Moreover, it has been theoretically and empirically proven [17] that in a network with power-law degrees
with a suitably chosen \(\text{RootedRandomWalk}\) and sample the most important nodes to encode the proximity of this paper. Different to struc2vec and Graphsage which leverage \(v\) of the structural feature vector. The largest \(k\) the length per random walk respectively, and as shown in Algorithm 2. The complete procedure is summarized in Algorithm 1. The complete procedure is illustrated in Algorithm 3. The similarity metric in line 4 is minimizing the objective function as:

\[
J(e_i \mid e_p) = -\log(\sigma(e_i^T e_p)) - K \cdot \mathbb{E}_{e_n} \mathbb{P}(e_p | e_n) \log(\sigma(-e_i^T e_n))
\]

where \(\sigma(\cdot)\) is the sigmoid function and \(K\) controls the negative sampling number. \(e_p\) and \(e_n\) are positive and negative nodes of \(v_i\) respectively, while \(e_p\) and \(e_n\) are the corresponding embedding vectors. Technically, negative nodes are sampled from a distribution \(P_n\), and for most network embedding methods [7, 8, 23], positive nodes are defined as nodes that co-occur with \(v_i\) in a fixed-size window in random walk sequences.

In SPINE, to encourage the similarity of embeddings to jointly encode the similarity in terms of structural identities and local proximities simultaneously, we design a novel biased positive sampling strategy based on the structural features generated from Algorithm 1. The complete procedure is illustrated in Algorithm 3. Specifically, we define a structural rate \(\alpha \in (0, 1)\) to control the ratio of structural sampling and local proximity sampling. With probability \(\alpha\), a positive sample of \(v_i\) is sampled according to the similarities between their structural features (starting from line 2). Otherwise, the positive node is sampled from nodes that co-occur near \(v_i\) on trivial random walks (starting from line 9), which is preprocessed and stored in \(L_i\). The similarity metric in line 4 can be chosen from Euclidean distance, cosine similarity, Dynamic Time Difference (DTD) metric, and other options.

### 3.3 Embedding Generation

To construct an inductive network embedding framework, we generate embeddings instead of directly optimizing the embedding matrix, through which the structural information and the content information of networks can be jointly incorporated.

As stated above, the \(k\) values in \(T_i\) indicate \(k\) largest structural proximities between \(v_i\) and nodes co-occurring with \(v_i\) in rooted random walks. We denote the content matrix of the corresponding \(k\) nodes as \(F^k_i \in \mathbb{R}^{k \times f}\), which is constructed row by row according to the same order of RPR values in \(T_i\). Given the structural features \(T_i\) and node content \(F^k_i\), we propose an embedding generation method for \(v_i\).

Specifically, we first employ a single-layer multilayer perceptron (MLP) to map nodes from the content space to the embedding space, then compute a linear combination of the \(k\) vectors with respect to the corresponding weights in \(T_i\). Formally, denote the dimensionality of embeddings as \(d\), the weight matrix of the MLP is \(W_M \in \mathbb{R}^{f \times d}\), then the embedding generation process can be written as:

\[
e_i = \sigma(\sum_{j=1}^{k} T_{ij} F^k_{ij} W_M)
\]

where \(F^k_{ij} \in \mathbb{R}^f\) is the \(j\)-th row of \(F^k_i\), \(\sigma\) is the non-linear activation function, and \(e_i \in \mathbb{R}^d\) is the embedding vector of node \(v_i\).

### 3.4 Biased SGNS

The Skip-Gram Negative Sampling (SGNS) model is widely used in representation learning, which is based on the principle of learning the embedding vector of a word/node by predicting its neighbors. More formally, given an embedding vector \(e_i\), SGNS is minimizing the objective function as:

\[
J_i = -\log(\sigma(\langle e_i, e_p \rangle)) - K \cdot \mathbb{E}_{e_n} \mathbb{P}(e_p) \log(\sigma(\langle e_i, e_n \rangle))
\]

where \(\sigma(\cdot)\) is the sigmoid function and \(K\) controls the negative sampling number. \(e_p\) and \(e_n\) are positive and negative nodes of \(v_i\) respectively, while \(e_p\) and \(e_n\) are the corresponding embedding vectors.

When calculating \(S^{RPR}_i\), considering the inductive prerequisite, the transition matrix \(P\) which encodes the structure of the network may be unavailable. Alternatively, we approximate \(S^{RPR}_i\) from the local structure around \(v_i\) through a Monte Carlo approximation, as shown in Algorithm 2. The complete procedure is summarized in Algorithm 1, where \(m\) and \(l\) indicate the number of repeats and the length per random walk respectively, and \(k\) controls the length of the structural feature vector. The largest \(k\) values of \(S^{RPR}_i\) are taken as the structural description of \(v_i\), denoted as \(T_i\) in the rest of this paper. Different to struc2vec and GraphSage which leverage \(k\)-hop neighborhood structures in a recursive manner, we evaluate and sample the most important nodes to encode the proximity of the current node from multiple distance, which is more efficient in time complexity and parameter magnitude.

### Algorithm 1 Rooted random walk sampling

**Input:** the graph \(G\), the present node \(v_i\), the continuation probability \(\beta_RPR \in (0, 1)\), hyper-parameters \(k, m, l\)

**Output:** the structural feature vector \(T_i\) of \(v_i\)

1. Initialize a counter \(C_i \in \mathbb{R}^{|V|}\)
2. repeat
3. \(P_s = \text{RootedRandomWalk}(G, v_i, l, \beta_RPR)\)
4. for \(v_j \in P_s\) do
5. \(C_i[j] \leftarrow C_i[j] + 1\)
6. end for
7. until \(m\) times
8. \(C_i \leftarrow \text{Sort } C_i\) in a descending order
9. \(T_i \leftarrow \text{The first } k\text{ elements of } C_i\)
10. return \(T_i \leftarrow T_i / \text{sum}(T_i)\)

### Algorithm 2 RootedRandomWalk

**Input:** the graph \(G\), the present node \(v_i\), the continuation probability \(\beta_RPR \in (0, 1)\), the walk length \(l\)

**Output:** a rooted random walk sequence \(P_s\)

1. Initialize a list \(P_s = \{v_i\}\)
2. for \(j = 1\) to \(l - 1\) do
3. \(v_{cur} \leftarrow \text{the last element of } P_s\)
4. if \(\text{random}(0, 1) < \beta_RPR\) then
5. Randomly select a node \(v_j\) from the neighbors of \(v_{cur}\)
6. Append \(v_j\) to the end of \(P_s\)
7. else
8. Append \(v_i\) to the end of \(P_s\)
9. end if
10. return \(P_s\)

(e.g., Twitter), the RPR values also follow a power-law distribution. Therefore, the largest \(k\) RPR values are able to effectively and differentially represent the structural information of a node with a suitably chosen \(k\) (usually related to the average degree of the network). As a comparison, methods including struc2vec and GraphSage utilize the degree sequence of \(k\)-hop neighbors as the structural description of \(v_i\) at the \(k\)-th iteration. Since the importance of different neighbors to \(v_i\) cannot be distinguished only from their degrees, they have to consider all neighbors in the subsequent procedures, which may cause huge redundancies in time and space complexity.
Warping (DTW) [26], etc., and the normalization step in line 7 can be implemented with average, softmax, etc. In our experiments, we use DTW which is designed to compare ordered sequences as the similarity metric. Similarities are then normalized as $\frac{e_{ij}}{\sum_{i,j} e_{ij}}$.

The structural sampling paradigm alleviates the limitation of distance. In practice, it is redundant to compute the structural similarity between $v_i$ and all the other nodes, since nodes with completely different local structures are nearly impossible to be sampled as positive pairs through structural sampling (for example, the yellow and blue nodes in Figure 1). Intuitively, nodes with similar degrees are likely to have similar local structures. Based on this intuition, we reduce the redundancy by only considering nodes which have similar degrees with the present node $v_i$. Specifically, given an ordered list of node degrees, we choose the candidates for structural sampling by taking $O(\log |V|)$ nodes in each side from the location of $v_i$. As a consequence, the time complexity of structural sampling for each node is reduced from $O(|V|)$ to $O(\log |V|)$.

In addition, instead of utilizing pure structural sampling or local sampling (i.e., $\alpha = 1$ or $\alpha = 0$), the ratio $\alpha$ controls the tradeoff between preserving structural identities and local proximities of nodes. As discussed in Section 1, the two kinds of information should be jointly leveraged to comprehensively represent real-world networks instead of singly preserved. Therefore, we enforce and control their interaction through ratio $\alpha$.

Algorithm 3 Biased positive sampling

| Input: | the structural feature matrix $T$, the present node $v_i$, a node list $L_i$, the structural rate $\alpha$ |
| Output: | $v_p$, which is a positive sample of $v_i$ |

1: Initialize an empty list $P_s = []$
2: if random(0, 1) $< \alpha$ then
3: for $j = 1$ to $|V|$, $j \neq i$ do
4: $t \leftarrow$ Compute the similarity between $T_i$ and $T_j$
5: $P_s \leftarrow$ Append $t$ to $P_s$
6: end for
7: $P_s \leftarrow$ Normalize $P_s$ to $[0, 1]$
8: $v_p \leftarrow$ Sample a node according to $P_s$
9: else
10: $v_p \leftarrow$ Randomly choose a node from $L_i$
11: end if
12: return $v_p$

3.5 Learning and Optimization

We introduce the biased SGNS based objective function of our framework in this section. We propose two types of embeddings for each node $v_i \in V$, i.e., a content generated embedding $e_i$ as defined in Equation (4), and a structure based embedding $s_i$ which is the $i$-th row of an auxiliary embedding matrix $W_S \in \mathbb{R}^{V \times d}$. In real-world network-based datasets, the content of nodes is likely to be extremely sparse and weaken the structural information incorporated during the generation process of $e_i$. Therefore we employ a direct interaction between $e_i$ and $s_i$ to strengthen the structural information contained in the learned embeddings.

Formally, given the present node $v_i$ and its positive sample $v_j$, which is sampled according to Algorithm 3, the pairwise objective function of SPINE can be written as:

$$F(v_i, v_j) = \lambda_1 \cdot J(e_i | e_j) + \lambda_2 \cdot J(s_i | s_j) + (1 - \lambda_1 - \lambda_2) \cdot [J(e_i | s_j) + J(s_i | e_j)]$$

where $\lambda_1$ and $\lambda_2$ control weights of different parts, and $J(\cdot | \cdot)$ denotes the pairwise SGNS between two embeddings defined in Equation (5). Intuitively, the generator and the auxiliary embedding matrix should be well trained through their single loss as well as obtaining each other’s information through the interaction loss, where $\lambda_1$ and $\lambda_2$ determine which one is the primary part.

The structure based embeddings $W_S$ are regarded as parameters, which, along with the parameters $W_M$ of the MLP in Equation (4), constitute all the parameters to be learned in our framework. Eventually, the objective of our framework is:

$$\min_{W_S, W_M} \sum_{i,j \neq i} |V| F(v_i, v_j)$$

Afterwards, various optimizers can be utilized to solve the objective function, including SGD, Adam [11], etc. In practice, $W_S$ can be randomly initialized or pretrained with local proximity preserved methods. It is worth noting that the interaction paradigm in Equation (6) can be generalized to much broader scenarios where one can integrate two arbitrary types of information.

By optimizing the objective function defined in Equation (7), the embedding generator proposed in Section 3.3, which is parameterized by $W_M$, can be learned. The embedding generator is supposed to contain the content information and the structural information as well as local proximities and structural identities simultaneously. Therefore, during inference, we drop the auxiliary embedding matrix $W_S$ and only keep the trained embedding generator for prediction. In the sequel, embeddings of unseen nodes can be generated by first constructing structural features via Algorithm 1 and then following the paradigm described in Section 3.3.

4 EXPERIMENTS

We evaluate SPINE in different scenarios to show its potential of simultaneously preserving local proximity and structural identity, in transductive and inductive settings, as well as on general network mining tasks and particularly designed structural similarity tasks. We start from introducing the experimental setup, including the datasets we use and the baselines we compare to, followed by experimental settings.

4.1 Experimental Setup

We test the proposed model on four benchmark datasets to measure its performance on real-world tasks, and one small scale social network to validate the structural identity preserved in the learned
Table 2: Accuracy of transductive node classification (in percentage)

| Method   | Citeseer | Cora | Pubmed |
|----------|----------|------|--------|
| node2vec | 47.2     | 69.8 | 70.3   |
| struc2vec| 21.1     | 24.2 | 40.7   |
| n+s      | 42.7     | 68.3 | 67.1   |
| n+f      | 58.6     | 76.0 | 70.5   |
| s+f      | 56.2     | 55.1 | 42.8   |
| n+s+f    | 57.0     | 73.7 | 67.3   |
| Graphsage| 52.6     | 66.7 | 75.2   |
| GAT      | 72.5 ± 0.7 | 83.0 ± 0.7 | 79.0 ± 0.3 |
| SPINE    | 68.4 ± 0.3 | 83.7 ± 0.4 | 78.5 ± 0.3 |
| SPINE-p  | 69.2 ± 0.2 | 82.2 ± 0.7 | 82.2 ± 0.3 |

embeddings. The statistics of datasets are summarized in Table 1. Note that while we focus on content-rich networks in experiments, our method can be flexibly generalized to networks without content by extracting structural features such as node degrees as content, following the similar strategy in [8].

For the node classification task, we test our method on Citation Networks [27], where nodes and edges represent papers and citations respectively, and each paper is described by a binary vector or a TFIDF word vector. To test the performance of SPINE while generalizing across networks, we further include PPI [28], which consists of multiple networks corresponding to different human tissues. To measure the structural identity preserved in embeddings, we test SPINE on a subset of Facebook dataset [14], denoted as FB-686, in which nodes and links represent users and their connections, and each user is described by a binary vector.

As for the baselines, we consider unsupervised network embedding methods including node2vec [7], struc2vec [24] and their variants. As node2vec and struc2vec only leverage structural information of networks, for a fair comparison, we concatenate raw content vectors to the learned embedding vectors as new baselines, denoted as n+s (node2vec with features) and s+f (struc2vec with features). In addition, considering that node2vec and struc2vec are designed to preserve the local proximity and the structural identity respectively, we concatenate the corresponding embeddings learned by these two methods to form a new baseline, denoted as n+s, to illustrate the superiority of SPINE over the linear combination of the two proximities. Similarly, n+s+f denotes the content incorporated variant of n+s. On the other hand, in addition to transductive methods, we also consider Graphsage, a recent inductive network embedding method which jointly leverages structural and content information. We compare with the unsupervised variant of Graphsage, and we report the average performance of various aggregators proposed in the paper as its final results, since different aggregators perform inconsistently over different datasets and tasks. We also report the performance of the most recent supervised inductive node classification method GAT [31]. Random and raw feature results are also included as baselines in this setting.

For our method, we use SPINE and SPINE-p to indicate the variants with W5 randomly initialized and pretrained with node2vec respectively. To make predictions based on the embeddings learned by unsupervised models, we use one-vs-rest logistic regression as the downstream classifier. For all the methods, the dimensionality of embeddings is set to 200. As for the hyper-parameters, we keep the following settings for all tasks and datasets: ratios λ1 and λ2 in Equation (7) are set to 0.4 and 0.2 respectively, while the structural rate α in Algorithm 3 and the restart rate βRPR of rooted random walk are both set to 0.5. The learning rate of the Adam optimizer is set to 0.001. For methods that leverage random walks, we set the number of repeats for each node to 10, the length of random walks to 40 and the window size to 5 for a fair comparison.

4.2 Node Classification

We first evaluate the performance of SPINE on node classification, a common network mining task. Specifically, we conduct the experiments in both transductive and inductive settings. For the transductive setting, we use the same scheme of training/test partition as in [8]. To be concrete, we randomly sample 20 instances from each class as the training data, and 1000 instances from the rest of the dataset as the test data. As for the inductive setting, on citation networks, we randomly remove 20%, 40%, 60% and 80% nodes and the corresponding edges, these nodes are then treated as test nodes with the remaining network as the training data. Meanwhile on the PPI network, we follow the same dataset splitting strategy as in [8, 31], i.e., 20 networks for training, 2 for validation and 2 for testing, where the validation and testing networks remain unseen during training. For both settings we repeat the process 10 times and report the mean score.

4.2.1 Transductive Learning. Results in the transductive setting are reported in Table 2. We can observe that SPINE outperforms all unsupervised embedding methods, and performs comparably with the state-of-the-art supervised framework GAT. In addition, n+s performs worse than node2vec, which implies that a simple linear combination of local proximity preserved and structural identity preserved embeddings is incapable of generating a meaningful representation that effectively integrates the two components. Meanwhile, the performance of all the content-augmented variants are better than the original structure-only models but still inferior to SPINE, which shows that concatenation of content features is useful, however the content aggregation method we propose can better consolidate the content and structure information. Furthermore, the comparison between the two variants of SPINE indicates that while the basic model of SPINE already achieves a competitive performance, we can further enhance the model with initializations.

Figure 2: Comparison of running time on Cora, with training batches of size 512 and inference on the full test set (1000 nodes).

Code is available at https://github.com/lemmonation/spine
Table 3: Accuracy of inductive node classification w.r.t node removal rate (in percentage)

| Methods      | 20% | 40% | 60% | 80% |
|--------------|-----|-----|-----|-----|
| Citeese       |     |     |     |     |
| Random       | 19.5| 20.4| 16.7| 17.7|
| RawFeats     | 63.9| 62.2| 60.3| 57.7|
| Graphsage    | 58.5| 53.9| 47.8| 41.4|
| SPINE        | 75.4| 72.1| 71.5| 68.7|
| Cora         |     |     |     |     |
| Random       | 18.8| 22.0| 19.1| 20.1|
| RawFeats     | 66.6| 64.7| 64.6| 59.6|
| Graphsage    | 73.1| 66.4| 58.8| 48.6|
| SPINE        | 86.7| 84.1| 82.1| 77.9|
| Pubmed       |     |     |     |     |
| Random       | 38.5| 39.8| 39.3| 38.9|
| RawFeats     | 75.7| 75.4| 74.6| 72.9|
| Graphsage    | 79.9| 79.4| 78.2| 76.4|
| SPINE        | 85.7| 83.7| 83.0| 78.8|

Figure 3: Node classification results on PPI. The left vertical axis indicates the micro-F1 score while the right indicates the macro-F1 score. Both are in percentage.

Figure 4: Euclidean distance distribution between mirrored node pairs and connected node pairs on the FB-686 dataset.

4.3 Structural Identity

So far we have verified the performance of SPINE on real-world datasets and network mining tasks in both transductive and inductive settings to justify the improvement brought by structural identities during representation learning. We proceed to explicitly validate that the learned embeddings of SPINE successfully preserve structural identities. Here we investigate the structural identity on the FB-686 dataset mainly following the settings in [24].

4.3.1 Transductive Learning. We first consider the transductive setting. Specifically, for the original network, we construct a mirror network and relabel the nodes, and consider the union of two networks as the input. As a consequence, node pairs between original nodes and their mirror nodes are obviously structurally equivalent, thus should be projected close in the embeddings space. We then evaluate the Euclidean distance distribution between embeddings of the mirror node pairs and all the node pairs connected by edges, denoted as \( P_m \) and \( P_a \) respectively. Intuitively, if embeddings successfully preserve structural identities of nodes, \( \mathbb{E}[P_m] \) should be much smaller than \( \mathbb{E}[P_a] \). Meanwhile, \( \mathbb{E}[P_a] \) reflects the distance of local proximities in the network. As a consequence, embeddings that perfectly preserve local proximity and structural identity should have small values of \( \mathbb{E}[P_a] \) and \( \mathbb{E}[P_m] \), along with a big ratio of \( \mathbb{E}[P_a]/\mathbb{E}[P_m] \).

Results of SPINE and struc2vec with respect to the two distributions are shown in Figure 4. Obviously, compared to struc2vec, embeddings learned by SPINE yield smaller distances between both mirrored node pairs and connected node pairs, indicating the structural identity and local proximity are jointly preserved better. In addition, the ratio between \( \mathbb{E}[P_a] \) and \( \mathbb{E}[P_m] \) is 13.40 and 5.72 for SPINE and struc2vec respectively, which means SPINE distinguishes the two proximities more clearly.

Further to test the robustness of SPINE to edge removal and changing content, we randomly sample two new networks from the original FB-686 network. Specifically, we preserve each edge in the original network with probability \( s \), and randomly exchange a 1’s location with another 0’s location in each node’s content vector. Consequently, from the view of structure, the probability for an original edge contained both in the two generated networks is \( s^2 \),
Table 4: Pearson and Spearman coefficients between structural distance and Euclidean distance for connected node pairs on citation networks.

| Dataset | Pearson (p-value) | Spearman (p-value) |
|---------|------------------|--------------------|
| Citeseer| 0.72 (0.0)       | 0.74 (0.0)         |
| Cora    | 0.77 (0.0)       | 0.79 (0.0)         |
| Pubmed  | 0.78 (0.0)       | 0.84 (0.0)         |

and smaller $s$ indicates less structure correlation between the two generated networks. From the view of content, mirrored nodes are nearly impossible to have identical content due to the sparsity of content vectors.

Results are illustrated in Figure 5. As edges of two generated networks change with different $s$, the comparison regarding distance distributions of connected node pairs ($P_d$) becomes meaningless, therefore we focus on structural identities. As can be observed in Figure 5, the ratio between $E[P_d]$ and $E[P_m]$ is not significantly affected by the degree of structure perturbation $s$, which indicates that SPINE can robustly distinguish and preserve structural identity and local proximity even with perturbation in structure and content.

We proceed to verify that embeddings learned by SPINE also preserve structural identities in real-world tasks. We compute the correlation between the structural distance (or similarity) defined in line 4, Algorithm 3 and the Euclidean distance in the embedding space for all the connected node pairs. The values of correlation measured by Pearson and Spearman coefficients are listed in Table 4, which indicates that there indeed exists a strong correlation between the two distances, validating that SPINE successfully preserves the defined structural similarity in the embedding space.

4.3.2 Inductive Learning. To verify whether SPINE can capture the structural identity across networks, we generate four new networks $G_1$, $G_2$, $G_3$ and $G_4$, from the original FB-686 network with $s = 0.2$. The combination of $G_1$ and $G_2$ is considered as training networks, while $G_3$ and $G_4$ constitute the test data. After training, the embeddings of nodes in $G_3$ and $G_4$ are inferred, on which we compute the distance distributions between embeddings of connected node pairs and mirrored node pairs. Intuitively, as $G_1$ and $G_2$ are separated, baselines which only consider local proximities are not able to capture the structural similarity between networks. Results are shown in Figure 6. Although the structural correlation between training and testing networks is small given $s = 0.2$, the two distance distributions learned by SPINE are still strikingly different, indicating that SPINE can learn high-level representations of structural identities from training networks rather than just storing them, which leads to the generalization ability to identify similar structural identities in unseen networks. The two distributions learned from Graphsage are practically identical, justifying our intuition and the necessity of preserving structural identities.

4.4 Parameter Study

In order to evaluate the influence of parameters on the performance of SPINE, we conduct experiments on a subset of PPI datasets, denoted as subPPI, which contains 3 training networks and 1 test network. We first investigate the effect of the structural rate $\alpha$ in Algorithm 3, then test the robustness of SPINE with respect to varying $\lambda_1$ and $\lambda_2$ values in Equ (7), both on node classification.

4.4.1 Effect of $\alpha$. We vary the value of the structural rate $\alpha$ from 0 to 1 with an interval of 0.2, and report the results in Figure 7. As
expected, with increasing \( \alpha \) values, the classification performance increases first and then decreases when \( \alpha \) gets too large. The results show that only considering the structural identity or local proximity will impair the quality of learned embeddings in real-world tasks. Therefore, the structural rate \( \alpha \) in Algorithm 3 is crucial for enhancing the quality of learned embeddings.

### 4.4.2 Effects of \( \lambda_1 \) and \( \lambda_2 \)

To investigate the influence of ratios \( \lambda_1 \) and \( \lambda_2 \) in Equation (7), we vary the value of \( \lambda_1 \) from 0.2 to 1.0 and the value of \( \lambda_2 \) from 0.2 to 0.6, both with an interval of 0.2. Results of node classification on subPPI are reported in Table 5, from which we can conclude that the interaction between the generator and \( W_S \) successfully enhances the structure information carried in the generator as expected, as \( \lambda_1 = 1.0 \) (no interaction) achieves the worst performance.

### 5 CONCLUSION

In this paper, we propose SPINE, an inductive network embedding approach which is able to jointly preserve structural identities and local proximities of nodes while being generalized to unseen nodes or networks. We assign a structural feature vector to each node based on Rooted PageRank, which is verified to be a measure of structural identities of nodes. Instead of directly optimizing embeddings, we learn an embedding generator leveraging the structural features of each node to incorporate the structural and content information of nearby nodes. In addition, to jointly preserve structural identities and basic local proximities of nodes, we propose a biased Skip-Gram Negative Sampling algorithm with a novel positive sampling procedure, based on which a carefully designed objective function is proposed to enhance the structural information contained in the embedding generator. Extensive experiments demonstrate the superiority of SPINE over the state-of-art baselines on both transductive and inductive tasks. An interesting direction of future work is to introduce structural identity in other network-based data mining tasks such as social recommendation.

### REFERENCES

[1] Vladimir Batagelj, Ana Ilic, and Patrick Doreian. 1992. Direct and indirect methods for structural equivalence. *Social networks* 14, 1 (1992), 63–90.

[2] Aleksandar Bojchevski and Stephan Günnemann. 2017. Deep gaussian embedding of attributed graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815* (2017).

[3] Sergey Brin and Lawrence Page. 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks* 56, 18 (2012), 3825–3833.

[4] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2017. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. *arXiv preprint arXiv:1709.07604* (2017).

[5] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM.

[6] Rui Feng, Yang Yang, Wenjie Hu, Fei Wu, and Yueling Zhuang. 2018. Representation Learning for Scale-free Networks. In *AAAI*.

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

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

[9] Taher H Havewala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web* ACM, 517–526.

[10] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 731–739.

[11] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[13] Amy N Langville and Carl D Meyer. 2011. Google’s PageRank and beyond: The science of search engine rankings. Princeton University Press.

[14] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data (2014).

[15] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. 2017. Attributed Network Embedding for Learning in a Dynamic Environment. *arXiv preprint arXiv:1706.01860* (2017).

[16] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *journal of the American society for Information Science and Technology* 58, 7 (2007), 1019–1031.

[17] Nelly Litvak, Werner RW Scheunhardt, and Yana Volkovich. 2007. In-degree and PageRank: why do they follow similar power laws? *Internet mathematics* (2007).

[18] Francois Lorrain and Harrison C White. 1971. Structural equivalence of individuals in social networks. *The journal of mathematical sociology* (1971), 49–80.

[19] Tiancheng Lou and Jie Tang. 2013. Mining structural hole spanners through information diffusion in social networks. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 825–836.

[20] Jianxin Ma, Peng Cui, and Wenzhu Zhu. 2018. DepthLGP: Learning Embeddings of Out-of-Sample Nodes in Dynamic Networks. In *AAAI*.

[21] Tomasz Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *preprint arXiv:1301.3781* (2013).

[22] Tomasz Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[23] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.

[24] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. struc2vec: Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 385–394.

[25] Ryan A Rossi and Nesreen K Ahmed. 2015. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering* 27, 4 (2015), 1112–1131.

[26] Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–586.

[27] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93.

[28] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. 2006. BioGRID: a general repository for interaction datasets. *Nucleic acids research* 34, suppl 1 (2006), D535–D539.

[29] Jian Tang, Meng Qu, Mingjie Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2017. Deep clinical symptom embedding. *arXiv preprint arXiv:1710.10993* (2017).

[30] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vol. 1. 1722–1731.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv preprint arXiv:1710.10903* (2017).

[32] Zheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network Representation Learning with Rich Text Information. In *IJCAI*.

[33] Yang Yang, Jie Tang, Cane Wing-It Leung, Ziyu Sun, Qiong Chen, Junzhi Li, and Qiang Yang. 2015. RAIN: Social Role-Aware Information Diffusion. In *AAAI*. 367–373.

[34] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. *preprint arXiv:1603.06861* (2016).