SPDGI: Meta-Structure and Meta-Path based Deep Graph Infomax

1st Fujiao Ji
College of Computer Science and Engineering,
Shandong University of Science and Technology
Qingdao, China
fujiaoji@sdust.edu.cn

2nd Zhongying Zhao
College of Computer Science and Engineering,
Shandong University of Science and Technology
Qingdao, China
zyzhao@sdust.edu.cn

3rd Chao Li
College of Electronic and Information Engineering,
Shandong University of Science and Technology
Qingdao, China
lichao@sdust.edu.cn

***Abstract*—Graph Neural Network, as a powerful graph representation technique based on deep learning, has achieved great success in various applications. Numerous heterogenous network embedding methods have been proposed for effectively learning node representations that preserve both structural and attributed information. However, most of the existing works exploit either local or global information, thus are not adequate to capture enough information for heterogeneous networks. In this paper, we propose a meta-Structure and meta-Path based Deep Graph Infomax (SPDGI) method for heterogeneous information networks. Specifically, we first capture rich semantic information via meta-structure and meta-path. Then, we design two methods by considering the different ways in handling meta-structure. We further utilize the graph convolution module and semantic level attention mechanism to capture local representations of nodes. Finally, we get the global representation for the graph through an averaging operator and learn the final node representations by maximizing the local-global mutual information. The experimental results on three real-world datasets demonstrate that the proposed SPDGI can achieve performance competitive with state-of-the-art unsupervised models.**

***Index Terms*—network representation learning, heterogeneous network, mutual information**

I. Introduction

Network representation learning is to learn the latent and low-dimensional node representations, which preserve the network's topology (e.g., [1–3]), vertex content, and other side information (e.g., [4–7]). After obtaining representations of nodes, the following tasks (e.g., recommender systems [8–10], link prediction [11–13], and node classification [14–16]) can be easily and efficiently carried out by applying conventional machine learning algorithms.

Since random walk-based objectives over-emphasize proximity information at the expense of structural information [17], Belghazi *et al.* [18] offer a general-purpose parametric neural estimator of mutual information based on dual representations of the KL-divergence [19]. It is scalable, flexible, and completely can be trainable via back-propagation. Their work attracts significant attention to mutual information-based methods. For example, Deep InfoMax (DIM) [20], Deep Graph Infomax (DGI) [21], Heterogeneous Deep Graph Infomax (HDGI) [22], and DMGI [23]. Although the above mutual information-based algorithms have made great progress, there are still some limitations to be explored. For instance, DIM only focuses on image data; DGI is designed to embed a single network in which only one type of node and edge appear; although HDGI leverages meta-paths to represent the composite relations with different semantics, they still lose some important information, such as when nodes satisfy multiple paths simultaneously; DMGI uses a consensus regularization framework to solve diverse relationships in multiplex networks. However, the relation type they used is similar to meta-paths and thus has the same disadvantages as HDGI. The above representative methods are compared in Table I.

TABLE I: The comparison of representative methods (The symbol ✓ indicates the algorithm exploits the corresponding information, while × means not).

Methods		DIM	DGI	HDGI	DMGI	SPDGI
Applicable Condition	Image	✓	×	×	×	×
	Homogeneous Network	×	✓	×	×	×
	Heterogeneous Network	×	×	✓	×	✓
	Multiplex Network	×	×	×	✓	×
Used Modules	Meta-path/Relation	×	×	✓	✓	✓
	Meta-structure	×	×	×	×	✓
	Attention	×	×	✓	✓	✓
	Mutual Information	✓	✓	✓	✓	✓

In this paper, we propose a meta-Structure and meta-Path based Deep Graph Infomax (SPDGI) method for heterogeneous information networks (HIN). First, we utilize both meta-structure and meta-path to capture graph heterogeneity rather than meta-paths alone, which allows us to incorporate more complex semantic information. Second, we obtain negative graphs by shuffling node features. Then, we acquire the node's local representations through an attention mechanism on the embedding learned from various meta-structures and meta-paths. Combined nodes' embedding serves as a global representation. Finally, we maximize the mutual information between local node embedding and global graph embedding to obtain the final node representations. According to the different treatment of the meta-structures, we further design SPDGI-A and SPDGI-P. Both of them tend to utilize the nodes that satisfy diverse paths in meta-structures. SPDGI-A tends to combine these paths, while SPDGI-P is likely to choose nodes that meet all paths. The contributions of this work are summarized as follows:

- We propose a heterogeneous network representation learning model called SPDGI, which integrates meta-structures, meta-paths, and mutual information in an appropriate way.
- We further design SPDGI-A and SPDGI-P, inspired by the series connection and parallel connection: confining nodes that satisfy all paths at the same time or any path in meta-structures.
- Extensive experiments are conducted on real-world datasets to evaluate the performance of the proposed model. The experimental results demonstrate that the representations learned by SPDGI are effective for both node classification and clustering tasks.

The remainder of this paper is organized as follows. Section II briefly reviews the related works. The problem to be solved and preliminary knowledge are formulated in Section III. In Section IV, we present the SPDGI methodology. Section V proves the effectiveness of the proposed model with experimental results and analyses. Finally, Section VI concludes the study and discusses our future work.

II. Related Work

Mutual information is based on Shannon entropy to measure dependence between random variables. However, it is difficult to get mutual information when the probability distributions are unknown [18, 24]. In Belghazi *et al.*'s work [18], they argue that the estimation of mutual information between continuous random variables can be achieved by gradient descent over neural networks. Besides, they propose a general-purpose mutual information neural estimator based on dual representations of the KL-divergence [19]. Inspired by this work, Hjelm *et al.* [20] find that it is insufficient to learn effective representations by downstream tasks and maximizing mutual information between the complete input and the encoder output. To address this problem, they introduce DIM to learn representations in the image area, which trains a model to maximize the mutual information between global representations and patches.

Although these mutual information-based methods are useful, they cannot be applied directly to graphs. How to utilize mutual information in graphs becomes a difficult but interesting problem. Velickovic *et al.* [21] successfully utilize it into graphs by maximizing mutual information between patch representations and the corresponding high-level summary of the graph. But the disadvantage is that the proposed model is not suitable for heterogeneous networks, while they are common in the real world. To solve this problem, Ren *et al.* [22] further exploit it to heterogeneous networks and propose HDGI. To be specific, they use meta-paths, graph convolution module, and semantic-level attention mechanism to capture individual node's local representations. Considering the deficiency that the above strategies only contain relevant information regarding each relation type, and therefore fail to take advantage of the diversity of networks, Park *et al.* [23] present an unsupervised method for embedding attributed multiplex network, which utilizes a consensus regularization framework and a universal discriminator to jointly integrate the embedding from multiple types of relations between nodes. Detailed differences refer to Table I.

TABLE II: The notations used in this paper

Notations	Descriptions
G	The original graph as the input of network embedding.
V, E	The set of nodes/edges in network G.
ϕ, φ	The node/edge mapping function.
$\mathcal{A}, \mathcal{R}$	The type of nodes/edges.
P	The meta-path.
S	The meta-structure.
M	The meta-path or meta-structure, $M \subset \{P, S\}$.
A	The commuting matrix.
$\tilde{A}$	The commuting matrix of negative examples.
$\hat{A}$	The added self-connections commuting matrix.
K	The number of M.
X	The features of nodes.
$\tilde{X}$	The shuffled features of negative nodes.
H	The node representations.
$\tilde{H}$	The node representations of negative examples.
$\mathcal{E}$	The encoder.
$\tilde{s}$	The summary vector of the graph.
$\mathcal{D}$	The discriminator.
I_N	The identity matrix.
W	The adjacency matrix.
W^{M_k}	The layer-specific trainable weight matrix.

III. Problem Definition

In this section, we first introduce preliminary knowledge. Then, we give the problem definition and summarize the symbols used in this paper in Table II. Finally, we give an example to illustrate them.

Definition 1. Heterogeneous Information Network (HIN) [25]. A heterogeneous information network is a directed graph $G = (V, E)$ with a node mapping function $\phi : V \rightarrow \mathcal{A}$ and an edge mapping function $\varphi : E \rightarrow \mathcal{R}$, where each node $v \in V$ belongs to one node type $\phi(v) \in \mathcal{A}$, and each edge $e \in E$ belongs to a particular relation $\phi(e) \in \mathcal{R}$, respectively. Besides, the type of nodes $|\mathcal{A}|$ and the type of edges $|\mathcal{R}|$ satisfy that $|\mathcal{A}| + |\mathcal{R}| > 2$.

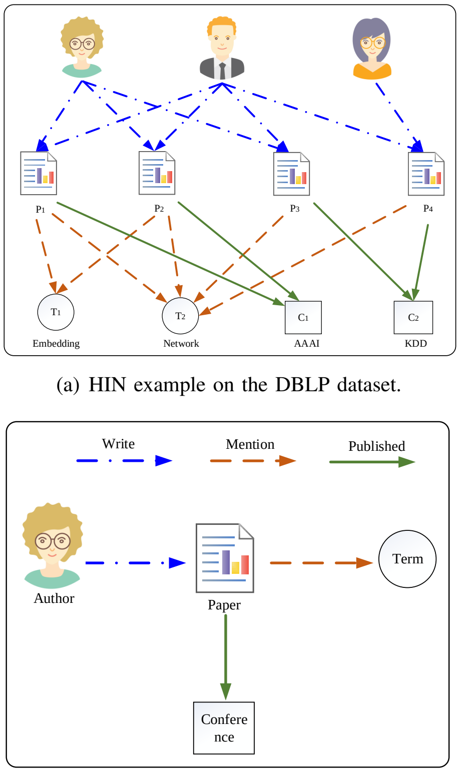
Definition 2. Network Schema [25]. Given a HIN $G = (V, E)$ with a node mapping function $\phi : V \rightarrow \mathcal{A}$ and an edge mapping function $\varphi : E \to \mathcal{R}$, its schema T_G is a directed graph defined over node types $\mathcal{A}$ and edge types $\mathcal{R}$, denoted as $T_G = (\mathcal{A}, \mathcal{R})$.

Definition 3. Meta-path [25]. Meta-path P is defined on the graph of network schema $T_G = (\mathcal{A}, \mathcal{R})$, which is denoted in the form of $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_l} \mathcal{A}_{l+1}$. It can be defined with a composite edge $\mathcal{R}_1 \circ \mathcal{R}_2 \circ \cdots \circ \mathcal{R}_{l+1}$ between type $\mathcal{A}_1$ and $\mathcal{A}_{l+1}$, where $\circ$ denotes the composition operator on edges.

Definition 4. Meta-structure [26]. Given a HIN $G = (V, E)$ with network schema $T_G = (\mathcal{A}, \mathcal{R})$, a meta-structure S is defined as $S = (\mathcal{A}, \mathcal{R}, v_s, v_t)$, where v_s is the source node, v_t is the target node.

Problem Definition. Meta-structure and meta-path based HIN embedding. Given a HIN $G = (V, E)$ with meta-structures $S = (\mathcal{A}, \mathcal{R}, v_s, v_t)$ and meta-paths P as input, the task is to learn the d-dimensional latent representations H for nodes, which not only contains structural and attribute information, but also includes additional but not redundant semantic information.

Example. Given a HIN G in Fig. 1(a) from the DBLP dataset, the abstracted network schema is shown as Fig. 1(b). From the network schema, we select appropriate meta-paths and meta-structures.



(a) HIN example on the DBLP dataset.

(b) HIN schema on the DBLP dataset.

Fig. 1: HIN example and abstracted schema on the DBLP dataset.

IV. SPDGI Methodology

In this paper, we aim to preserve topological information and attributes efficiently in heterogeneous information networks. As is pointed out in [18], mutual information can be introduced to alleviate the mode-dropping issue in generative adversarial networks and can be used to improve inference and reconstructions. However, there are several challenging problems that need to be handled. First, modeling semantic relationship between nodes is capable of producing more effective representations. Thus, we leverage meta-path and meta-structure to capture these semantics. Moreover, to integrate the impacts of different parts, we can directly concatenate the learned embeddings. However, it leads to the results that different types of representations are trained separately, and the impacts among various information cannot be balanced. Therefore, we further introduce the attention mechanism to balance different parts. To obtain node representations that capture the global information content of the entire graph, we utilize the mutual information between the global representation and the local nodes' representations, inspired by [20–22].

With the above ideas, we take the DBLP dataset as an example to illustrate the whole SPDGI framework. Specifically, the HIN G and network schema are shown as Fig. 1(a) and Fig. 1(b). We focus on authors and select P_1, P_2, P_3, and S. After shuffling the network, we compute commuting matrices for the real and negative networks. Nodes' local representations are computed through a GCN modeler and combined by an attention mechanism. Then, they are concatenated as the summary representation of the graph. We further maximize the mutual information between the local node's representation and the global network's representation to obtain the final embedding. The whole process is shown as Fig. 2. According to the different treatment of meta-structures and meta-paths, we design SPDGI-A and SPDGI-P for handling semantic information. We will introduce them in detail in the following subsections.

A. SPDGI-A

SPDGI-A allows nodes at intersections to satisfy any path. It is equivalent to separate the meta-structure into meta-paths, and then combine them into one merged graph. We implement it by adding the commuting matrix. Considering redundancy, we leverage meta-paths that are not included in meta-structures and the merged graph. For example, in Fig. 1, there are four types of nodes (Author, Paper, Conference, and Term) and three kinds of edges (Write, Mention, and Publish). From the HIN schema, we select several meta-paths and meta-structures. They are at the left side of Fig. 2. Then, We employ P_1 and S in SPDGI-A and the calculation of the commuting matrix is shown in Table IV.

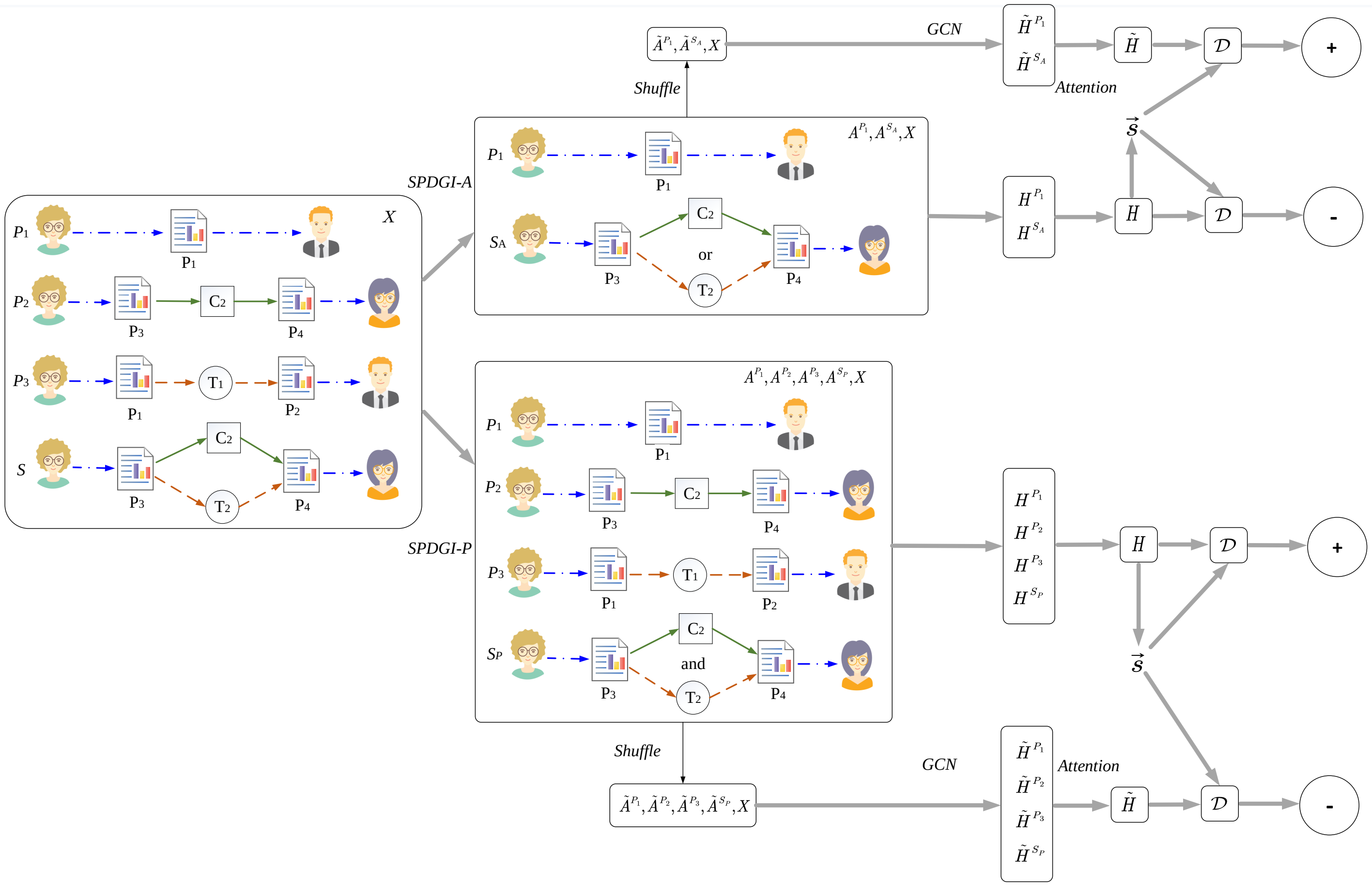


Fig. 2: The overall framework of the proposed SPDGI (Taking the DBLP dataset as an example)

TABLE III: Algorithm of SPDGI

```
Algorithm 1 The overall process of SPDGI.
Input: A HIN graph $G = (V, E)$ with selected meta-structures $S = (\mathcal{A}, \mathcal{R}, v_s, v_t)$ and meta-paths $P$, initial features $X$ and commuting matrix $A$.
Output: node representations.
Calculate meta-path and meta-structure based commuting matrices $A^{M_k}$ through SPDGI-A and SPDGI-P;
Obtain shuffled features $\tilde{X} = Shuffle(X)$;
while not converged do
    for $k = 1...K$ do
        Obtain $H^{M_k}$ and $\tilde{H}^{M_k}$ through Eq. (1);
    Generate nodes' representations $H$ by Eq. (2-5);
    Get a global vector $\vec{s}$ for the whole graph via Eq. (6);
    Maximize the mutual information with the binary cross-entropy loss of the discriminator through Eq. (7);
end while
```

TABLE IV: Computing commuting matrix of SPDGI-A

SPDGI-A: Commuting matrix
$A^{P_1} = W_{AP} \cdot W_{AP}^{\top}$
$A^{S_A} = W_{AP} \cdot \left(W_{PC} \cdot W_{PC}^{\top} + W_{PT} \cdot W_{PT}^{\top}\right) \cdot W_{AP}^{\top}$

TABLE V: Computing commuting matrix of SPDGI-P

SPDGI-P: Commuting matrix
$A^{P_1} = W_{AP} \cdot W_{AP}^{\top}$
$A^{P_2} = W_{AP} \cdot W_{PC} \cdot W_{PC}^{\top} \cdot W_{AP}^{\top}$
$A^{P_3} = W_{AP} \cdot W_{PT} \cdot W_{PT}^{\top} \cdot W_{AP}^{\top}$
$A^{S_P} = W_{AP} \cdot \left[\left(W_{PC} \cdot W_{PC}^{\top}\right) \odot \left(W_{PT} \cdot W_{PT}^{\top}\right)\right] \cdot W_{AP}^{\top}$

B. SPDGI-P

For SPDGI-P, we constrain nodes to meet all paths. It means that the intersected nodes can be reserved when they satisfy all paths in the meta-structure. We carry out it via adopting element-wise product between their commuting matrices. The final matrix is combined with meta-path based commuting matrices to obtain the node's local representations. Same as the example mentioned in SPDGI-A, we also choose several meta-paths and meta-structures. The difference is we leverages P_1, P_2, P_3 and S for SPDGI-P considering redundancy. Therefore, the calculation of the commuting matrix is shown in Table V.

C. Local Representation

SPDGI shuffles the rows of node's feature matrix and keep the commuting matrix unchanged to obtain the negative graph, which is in line with previous works, like [22]. For each M_k ($M \subset \{P, S\}$, k in $[1, K]$), we utilize Graph Convolution Network (GCN) as the encoder $\mathcal{E}$. Therefore, the node representation for each M_k can be obtained by GCN through Eq. (1).

$$H^{M_k} = \left(\hat{D}^{M_k \ -\frac{1}{2}} \hat{A}^{M_k} \hat{D}^{M_k \ -\frac{1}{2}}\right) X W^{M_k}, \tag{1}$$

where $\hat{A}^{M_k} = A^{M_k} + I_N$, I_N is the identity matrix, $\hat{D}^{M_k}$ is the diagonal node degree matrix of A^{M_k} and W^{M^k} is a layer-specific trainable weight matrix.

Then, following Ren *et al.* and Wang *et al.*'s work [22, 27], we also use a semantic attention layer to explore how much each part should contribute to the final representations. However, they only utilize meta-paths, while we use semantic attention to automatically learn the importance of different meta-paths and meta-structures based embeddings. Specifically, taking K groups of semantic-specific node embeddings H^{M_k} as input, the learned weights $\{\beta^{M_1}, \beta^{M_2}, ..., \beta^{M_K}\}$ of each meta-path and meta-structure can be shown as follows:

$$\{\beta^{M_1}, \beta^{M_2}, ..., \beta^{M_K}\} = att_{sem}(H^{M_1}, H^{M_2}, ..., H^{M_K}), \tag{2}$$

where

$$\beta^{M_k} = \frac{\exp\left(e^{M_k}\right)}{\sum\limits_{i=1}^{K} \exp\left(e^{M_i}\right)}, \tag{3}$$

$$e^{M_k} = \frac{1}{N} \sum_{n=1}^{N} \tan\mathrm{h}\left(\boldsymbol{q^T} \cdot \left[W_{sem} \cdot h_n^{M_k} + \boldsymbol{b}\right]\right). \tag{4}$$

In Eq. (4), e^{M_k} denotes the importance of each part; $\boldsymbol{q}$ is the semantic-level attention vector; and W_{sem} is a linear transformation parameter matrix.

Then, with the learned weights, the heterogeneous graph node representation H is obtained through Eq. (5).

$$H = \sum_{k=1}^{K} \beta^{M_k} \cdot H^{M_k}. \tag{5}$$

D. Global Representation

The objective of SPDGI is to maximize the mutual information between local representations and the global representation. The local node representations are obtained in Section. IV-C, and we need the summary vector to represent the global information of the entire heterogeneous graph. In this paper, we use the mean of node representations to get the global summary vector:

$$\vec{s} = Readout\left(H\right) = \sigma\left(\frac{1}{N} \sum_{i=1}^{N} \vec{h}_i\right). \tag{6}$$

E. Mutual information based discriminator

Inspired by previous works (e.g. [20–23]), we maximize the mutual information based on the Jensen Shannon divergence between the joint and the product of marginals and use the following objectives:

$$\mathcal{L} = \tfrac{1}{N+M}\left(\textstyle\sum_{i=1}^{N} \mathbb{E}_{(X,A)}\left[\log \mathcal{D}\left(\vec{h}_i, \vec{s}\right)\right] + \textstyle\sum_{j=1}^{M} \mathbb{E}_{(\tilde{X},\tilde{A})}\left[1 - \log \mathcal{D}\left(\vec{\tilde{h}}_j, \vec{\tilde{s}}\right)\right]\right), \tag{7}$$

where N and M denote the number of nodes and negative examples, respectively; $\mathcal{D}$ is a simple bilinear scoring function (similar to [21, 22, 28]), shown as Eq. (8):

$$\mathcal{D}\left(\vec{h}_i, \vec{s}\right) = \sigma\left(\vec{h}_i^T \boldsymbol{W} \vec{s}\right). \tag{8}$$

V. Experiment

A. Datasets

To make fair comparisons with HDGI [22], which is the most relevant baseline method, we conduct experiments on the datasets used in their original paper in terms of node classification and node clustering tasks.

- **IMDB** [29]. It contains 4275 movies (M), 5431 actors (A), 2082 directors (D), and 7313 keywords (K). We set movies as the target nodes. For the IMDB dataset, the classification task is to classify movies into three classes (Action, Comedy, and Drama) according to their genre.
- **DBLP** [30]. This is a research paper set, which contains 4057 authors (A), 14328 papers (P), 20 conferences (C), and 8789 terms (T). We set authors as the target nodes. For the DBLP dataset, the classification task is to classify authors into four areas (Database, Data Mining, Information Retrieval, and Machine Learning) according to the research topic.
- **ACM** [27]. It is a research paper set, which contains 3025 papers (P), 5835 authors (A), and 56 subjects (S). For the ACM dataset, the classification task is to classify the papers into three classes (Database, Wireless Communication, and Data Mining).

B. Baselines

We compare the proposed model with some state-of-the-art baselines to evaluate its performance.

- **DGI [21]:** It is an unsupervised manner for homogeneous graph, which relies on maximizing mutual information between patch representations and corresponding high-level summaries of the graph. In this paper, we apply

TABLE VI: Statistics of experimental datasets.

Dataset	Node	Edge	Meta-path Meta-structure	Average Degree (target node)	Class
IMDB	M [4275] A [5431] D [2082] K [7313]	M-A [12838] M-D [4280] M-K [20529]	MAM MDM MKM M(ADK)M	5.15 18.21 78.00 \	3
DBLP	A [4057] P [14328] C [20] T [8789]	A-P [19645] P-C [14328] P-T [88420]	APA APCPA APTPA AP(CT)PA	2.74 1232.56 1669.28 \	4
ACM	P [3025] A [5835] S [56]	P-A [9744] P-S [3025]	PAP PSP P(AS)P	9.68 730.83 \	3

DGI to meta-path based homogeneous graph and report the average performance.

- **DMGI [23]:** It is an unsupervised network embedding method for the attributed multiplex network, which jointly integrates the node embeddings from multiple graphs by introducing the consensus regularization framework and the universal discriminator.
- **HDGI-C [22]:** It employs meta-paths and graph convolution module with a semantic-level attention mechanism to capture local representations of nodes in heterogeneous information networks. Then, HDGI learns high-level node representations by maximizing the local-global mutual information.
- **SPDGI-A:** The proposed meta-structure and meta-path based deep graph infomax method, which allows nodes to satisfy any path in meta-structures.
- **SPDGI-P:** The proposed meta-structure and meta-path based deep graph infomax method, which constrains nodes to satisfy all paths in meta-structures.

C. Node Classification

We conduct experiments with different training ratios on these three datasets to achieve more general and convincing comparison. We take fixed 10 percent of data as the validation set. Except for the training data and validation data, the rest data are set as test data. All data are chosen randomly. We use early stopping with the patience of 20, i.e. we stop training if the validation loss does not decrease for 20 consecutive epochs. We tune learning rate from $\{0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01\}$. The final dimensions refer to Section V-E. To keep the results stable, we repeat the classification process 10 times and report the average Macro-F1 and Micro-F1 in Table VII.

TABLE VII: Quantitative results on the node classification task.

Dataset	Metrics	Training	DGI	DMGI	HDGI-C	SPDGI-A	SPDGI-P
IMDB	Macro-F1	20%	0.4830	0.6015	0.6068	**0.6501**	0.5899
		40%	0.5028	0.6203	0.6152	**0.6944**	0.6178
		60%	0.4894	0.6395	0.6234	**0.7010**	0.6378
		80%	0.5112	0.6495	0.6228	**0.7186**	0.6665
	Micro-F1	20%	0.5021	0.6010	0.6046	**0.6517**	0.5871
		40%	0.5199	0.6183	0.6127	**0.6950**	0.6131
		60%	0.5048	0.6353	0.6203	**0.7022**	0.6339
		80%	0.5165	0.6502	0.6239	**0.7218**	0.6663
DBLP	Macro-F1	20%	0.7379	0.8225	0.9092	0.8985	**0.9231**
		40%	0.7400	0.8243	**0.9298**	0.9162	0.9233
		60%	0.7361	0.8432	0.9191	0.9107	**0.9335**
		80%	0.7374	0.8450	0.9206	0.9098	**0.9373**
	Micro-F1	20%	0.7478	0.8303	0.9150	0.9040	**0.9272**
		40%	0.7556	0.8307	**0.9336**	0.9198	0.9265
		60%	0.7508	0.8478	0.9269	0.9182	**0.9386**
		80%	0.7475	0.8646	0.9235	0.9126	**0.9412**
ACM	Macro-F1	20%	0.7352	0.9294	0.9167	**0.9322**	0.9044
		40%	0.7220	0.9280	0.9063	**0.9327**	0.9104
		60%	0.7322	0.9223	0.9398	**0.9515**	0.8869
		80%	0.7392	0.9162	0.9236	**0.9583**	0.9138
	Micro-F1	20%	0.7670	0.9298	0.9163	**0.9310**	0.9039
		40%	0.7534	0.9278	0.9050	**0.9326**	0.9089
		60%	0.7648	0.9221	0.9405	**0.9499**	0.8868
		80%	0.7624	0.9109	0.9205	**0.9570**	0.9139

From Table VII, we can see that SPDGI has a good performance. For homogeneous graph embedding methods, we apply them to meta-path based homogeneous graphs and report the average values. Compared with DGI, the proposed SPDGI has a better result in these three datasets because of its effectiveness in capturing more semantic information. Besides, SPDGI gives different representations of different weights, while DGI averages the result. For heterogeneous graph embedding methods, we can observe that DMGI and HDGI-C have a good performance on the ACM and DBLP datasets. This is because both of them seize heterogeneous information and leverage the strength of mutual information. The proposed SPDGI-A works effectively in IMDB and ACM datasets, while SPDGI-P works well on the DBLP dataset. This is due to the properties of datasets and models. SPDGI tends to preserve additional information when nodes satisfy a meta-structure. Particularly, SPDGI-P tends to constrain nodes satisfying each channel in meta-structure, while SPDGI-A is inclined to allow nodes to satisfy any path. Therefore, SPDGI-A is to combine paths in meta-structure, while SPDGI-P is to increase additional information. On the IMDB dataset, the average degree of nodes is small, thus extra information is able to improve results. On the DBLP dataset, the meta-path contains a great deal of information, superfluous information will decrease the performance. Under these circumstances, SPDGI-A performs worse than SPDGI-P and HDGI-C. On the ACM dataset, the degree distribution of nodes in 'PAP' and 'PSP' meta-paths varies greatly. Thus, it is important to capture nodes that satisfy different paths at the same time and give them different weights

With the above analysis, we can find that the proposed SPDGI-A and SPDGI-P can achieve good performances when networks contain little information. Given enough information, SPDGI-P is appropriate to provide additional but not redundant information. When the nodes' distribution varies greatly in meta-structure, SPDGI-A can make a good balance by merging them into one graph.

D. Node Clustering

We also evaluate the embeddings learned from the previously mentioned algorithms on the task of node clustering. Once the proposed SPDGI has been trained, we can get the node embedding. Here we utilize the KMeans to conduct node clustering. The number of clusters on IMDB, DBLP, and ACM datasets is set to their true label classes. We adopt NMI

TABLE VIII: Quantitative results on the node clustering task.

Dataset	Metrics	DGI	DMGI	HDGI-C	SPDGI-A	SPDGI-P
IMDB	NMI	0.0275	0.019	0.0387	**0.1367**	0.0117
	ARI	0.0154	0.0181	0.0085	**0.1483**	0.0029
DBLP	NMI	0.3063	0.4339	0.4014	0.6309	**0.7180**
	ARI	0.3013	0.4638	0.3635	0.6955	**0.7798**
ACM	NMI	0.6129	0.5767	0.5116	**0.6525**	0.4909
	ARI	0.5971	0.6074	0.4718	**0.6466**	0.4511

and ARI to assess the quality of clustering results. Since the performance of KMeans is affected by initial centroids, we repeat the process 100 times. Other parameters are same to Section V-C. The average results are reported in Table VIII

From Table VIII, we can see that DGI cannot perform well on both IMDB and DBLP datasets because it is not able to balance weights from various meta-path based homogeneous graphs. However, on the ACM dataset, the DGI model has good results, even better than DMGI, HDGI-C, and SPDGI-P, while it has a bad performance in the classification task. This probably also because of the distribution of the ACM dataset. Papers with the same subjects are likely to be in the same group. Although HDGI-C and DMGI are designed for heterogeneous networks, they still miss some information between nodes that satisfy several meta-paths simultaneously, making the representations not effective enough. The verification based on node clustering tasks also demonstrates that SPDGI can learn effective representations by considering the additional information. Similar to the classification task, SPDGI-A performs better on the IMDB and ACM datasets, SPDGI-P can achieve a good performance on the DBLP dataset.

E. The Effect of Embedding Dimensions.

In this section, we investigate the effect of dimensions for the final embedding in SPDGI. The result on the three datasets is shown as Fig. 3. We can observe that with the growth of the embedding dimension, the performance raises first and then starts to decrease or be stable. The reason is that the proposed models need suitable dimensions to represent information. A smaller or larger dimension may cause deficient representations or additional redundancies. Therefore, considering the performance of result and operating efficiency, we choose 512, 256, 128 for SPDGI-A and 128, 256, 128 for SPDGI-P on the IMDB, DBLP, and ACM datasets, respectively.

VI. Conclusion and Future Work

In this paper, we propose a simple yet effective unsupervised method for heterogeneous information network representation learning, named SPDGI. It integrates several meta-paths and meta-structures through an attention mechanism to obtain local representations of nodes and get the global vector by summarizing them. Finally, through maximizing the local-global mutual information, SPDGI learns high-level representations of nodes. We demonstrate the effectiveness of learned representations for both node classification and clustering tasks on three datasets. We are also optimistic that mutual information maximization is a promising future direction for unsupervised representation learning. In our future work, we will study transfer learning and try to extend mutual information to this area.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant No. 62072288, 61702306, 61433012, 71772107), the Taishan Scholar Program of Shandong Province (Grant No. ts20190936), the Natural Science Foundation of Shandong Province (Grant No. ZR2018BF013), the Key Project of Industrial Transformation and Upgrading in China (No. TC170A5SW), the SDUST Research Found for Innovative Team (Grant No. 2015TDJH102), Open Project of Guangxi Key Laboratory of Trusted Software (Grant No. KX201535), Open Project from CAS Key Lab of Network Data Science and Technology (Grant No. CASNDST202007), and Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province (Grant No. CICIP2020001).

References

[1] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 701–710.

[2] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” in *Proceedings of the 24th International Conference on World Wide Web*, 2015, pp. 1067–1077.

[3] Y. Dong, N. V. Chawla, and A. Swami, “metapath2vec: Scalable representation learning for heterogeneous networks,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 135–144.

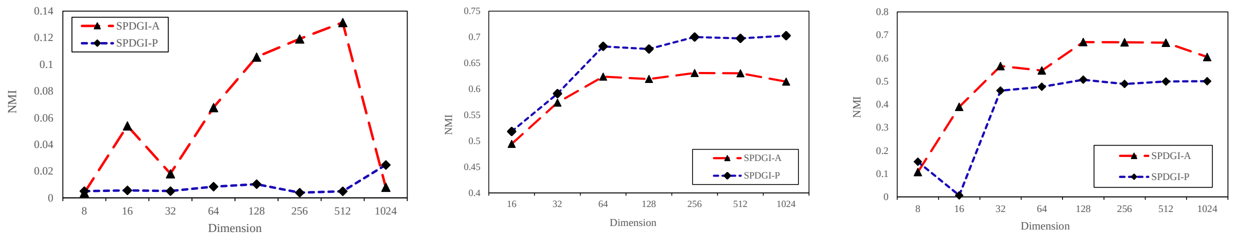
[4] Z. Zhao, H. Zhou, L. Qi, L. Chang, and M. Zhou, “Inductive representation learning via cnn for partially-unseen attributed networks,” *IEEE Transactions on Network Science and Engineering*, DOI https://doi.org/10.1109/TNSE.2020.3048902.

[5] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “Network representation learning: A survey,” *IEEE Transaction on Big Data*, vol. 6, pp. 3–28, 2020.

[6] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, “ANRL: Attributed network representation learning via deep neural networks,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3155–3161.

[7] Z. Zhao, H. Zhou, C. Li, J. Tang, and Q. Zeng, “DeepEmLAN: Deep embedding learning for attributed networks,” *Information Science*, vol. 543, pp. 382–397, 2021.

[8] Z. Zhao, X. Zhang, H. Zhou, C. Li, M. Gong, and Y. Wang, “HetNERec: Heterogeneous network embed-



(a) IMDB.

(b) DBLP.

(c) ACM.

Fig. 3: Dimension of the final embedding.

ding based recommendation," *Knowledge-Based Systems*, vol. 204, p. 106218, 2020.

[9] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, pp. 357–370, 2019.

[10] Z. Zhao, Y. Yang, C. Li, and L. Nie, "GuessUNeed: Recommending courses via neural attention network and course prerequisite relation embeddings," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, pp. 1–17, 2020.

[11] T. Li, J. Zhang, P. S. Yu, Y. Zhang, and Y. Yan, "Deep dynamic network embedding for link prediction," *IEEE Access*, vol. 6, pp. 29 219–29 230, 2018.

[12] S. Abu-El-Haija, B. Perozzi, and R. Al-Rfou, "Learning edge representations via low-rank asymmetric projections," in *Proceedings of the ACM on Conference on Information and Knowledge Management*, 2017, pp. 1787–1796.

[13] W. Liu, P. Chen, S. Yeung, T. Suzumura, and L. Chen, "Principled multilayer network embedding," in *Proceedings of the IEEE International Conference on Data Mining Workshops*, 2017, pp. 134–141.

[14] N. Sheikh, Z. T. Kefato, and A. Montresor, "Semi-supervised heterogeneous information network embedding for node classification using 1d-cnn," in *Proceedings of the 5th International Conference on Social Networks Analysis, Management and Security*, 2018, pp. 177–181.

[15] N. Liu, Q. Tan, Y. Li, H. Yang, J. Zhou, and X. Hu, "Is a single vector enough?: Exploring node polysemy for network embedding," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 932–940.

[16] R. Hussein, D. Yang, and P. Cudré-Mauroux, "Are meta-paths necessary?: Revisiting heterogeneous graph embeddings," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 437–446.

[17] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 385–394.

[18] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, R. D. Hjelm, and A. C. Courville, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 530–539.

[19] A. Ruderman, M. D. Reid, D. García-García, and J. Petterson, "Tighter variational representations of f-divergences via restriction to probability measures," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 1155–1162.

[20] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *Proceedings of the 7th International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bklr3j0cKX.

[21] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proceeding of the 7th International Conference on Learning Representations*, 2019, DOI https://doi.org/10.17863/CAM.40744.

[22] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "Heterogeneous deep graph infomax," in *Workshop of Deep Learning on Graphs: Methodologies and Applications co-located with the 34th AAAI Conference on Artificial Intelligence*, 2020. [Online]. Available: http://arxiv.org/abs/1911.08538.

[23] C. Park, D. Kim, J. Han, and H. Yu, "Unsupervised attributed multiplex network embedding," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020, pp. 5371–5378.

[24] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, pp. 1191–1253, 2003.

[25] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, pp. 992–1003, 2011.

[26] Z. Huang, Y. Zheng, R. Cheng, Y. Sun, N. Mamoulis, and X. Li, "Meta structure: Computing relevance in large heterogeneous information networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, 2016, pp. 1595–1604.

[27] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proceedings of the World Wide Web Conference*, 2019, pp. 2022–2032.

[28] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: http://arxiv.org/abs/1807.03748

[29] C. Wu, A. Beutel, A. Ahmed, and A. J. Smola, "Explaining reviews and ratings with PACO: poisson additive co-clustering," in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 127–128.

[30] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han, "Graph-based consensus maximization among multiple supervised and unsupervised models," in *Processing of the 23rd Annual Conference on Neural Information Processing Systems*, 2009, pp. 585–593.