Heterogeneous Deep Graph Infomax

Yuxiang Ren,[1] Bo Liu,[2] Chao Huang,[2] Peng Dai,[2] Liefeng Bo,[2] Jiawei Zhang[1]
[1]IFM lab, Florida State University
[2]JD Finance America Corporation, AI lab
yuxiang@ifmlab.org, kfliubo@gmail.com, chuang7@nd.edu, peng.dai@jd.com, liefeng.bo@jd.com, jiawei@ifmlab.org

Abstract

Graph representation learning is to learn universal node representations that preserve both node attributes and structural information. The derived node representations can be used to serve various downstream tasks, such as node classification and node clustering. Inspired by the emerging mutual information-based learning algorithm, in this paper we propose an unsupervised graph neural network **H**eterogeneous **D**eep **G**raph **I**nfomax (*HDGI*) for heterogeneous graph representation learning. We use the meta-path to model the structure involving semantics in heterogeneous graphs and utilize graph convolution module and semantic-level attention mechanism to capture individual node local representations. By maximizing the local-global mutual information, *HDGI* effectively learns high-level node representations. Experiments show that *HDGI* remarkably outperforms state-of-the-art unsupervised graph representation learning methods. We even achieve comparable performance in node classification tasks when comparing with state-of-the-art supervised end-to-end GNN models.

1 Introduction

Numerous real-world applications, such as social networks [Zhang, 2018] and knowledge graphs [Wang *et al.*, 2017a] exhibit the favorable property of graph data structure. Meanwhile, handling graph data is very challenging. Because each node has its unique attributes, and the connections between nodes convey important information. When learning from the attributes of individual nodes and the connection information among them simultaneously, the task becomes more challenging.

Traditional machine learning methods focus on the features of individual nodes, which obstructs their ability to process graph data. Graph neural networks (GNNs) for representation learning of graphs learn nodes' new feature vectors through a recursive neighborhood aggregation scheme [Xu *et al.*, 2018]. With the support of sufficient training samples, a rich body of supervised graph neural network models have been developed [Kipf and Welling, 2017a; Veličković *et al.*, 2017; You *et al.*, 2018]. However, labeled data is not always available in graph representation learning tasks, and those algorithms are not applicable to the unsupervised learning settings. To alleviate the training sample scarcity problem, unsupervised graph representation learning has aroused extensive research interest. The goal of this task is to learn a low-dimensional representation of each graph node. The representation preserves graph topological structure and node content. Meanwhile, the learned representations can be applied to conventional sample-based machine learning algorithms as well.

Most of the existing unsupervised graph representation learning models can be roughly grouped into factorization-based models and edge-based models. Specifically, factorization-based models capture the global graph information by factorizing the sample affinity matrix [Zhang *et al.*, 2016; Yang *et al.*, 2015; Zhang *et al.*, 2016]. Those methods tend to ignore the node attributes and local neighborhood relationships, which usually contain important information. Edge-based models exploit the local and higher-order neighborhood information by edge connections or random-walk paths. Nodes tend to have similar representations if they are connected or co-occur in the same path [Kipf and Welling, 2017b; Duran and Niepert, 2017; Hamilton *et al.*, 2017a; Perozzi *et al.*, 2014]. Edge-based models are prone to preserve limited order node proximity and lack a mechanism to preserve the global graph structure. The recently proposed deep graph infomax (DGI) [Veličković *et al.*, 2019] model provides a novel direction that considers both global and local graph structure. DGI maximizes the mutual information between graph patch representations and the corresponding high-level summaries of graphs. It has shown competitive performance even compared with supervised graph neural networks in benchmark homogeneous graphs.

In this paper, we explore the mutual information-based learning framework in heterogeneous graph representation problems. The networked data in the real world usually contain complex structures (involving multiple types of nodes and edges), which can be formally modeled as heterogeneous graphs (HG). Compared with homogeneous graphs, heterogeneous graphs contain more detailed information and rich semantics among multi-typed nodes. Taking the bibliographic network in Figure 1 as an example, it contains three types of nodes (Author, Paper and Subject) as well as two types of edges (Write and Belong-to). Besides, the individual nodes

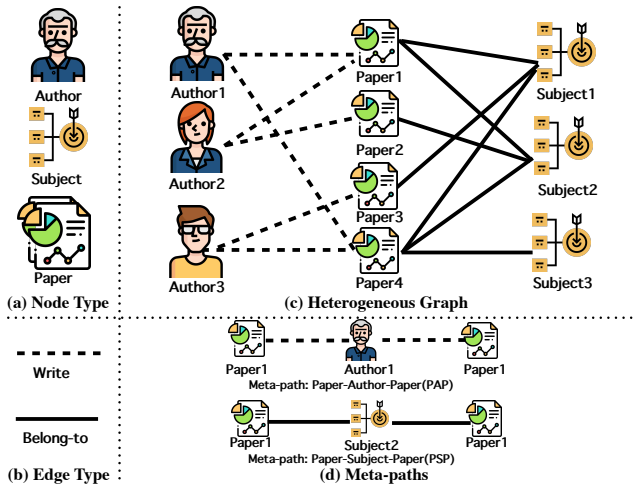


Figure 1: An example of heterogeneous bibliographic graph.

themselves also carry abundant attribute information (e.g., paper textual contents). Due to the diversity of node and edge types, the heterogeneous graph itself becomes more complex, and the diverse (direct or indirect) connections between nodes also convey more semantic information. In heterogeneous graph studies, meta-path [Sun *et al.*, 2011] has been widely used to represent the composite relations with different semantics. As illustrated in Figure 1(d), the relations between paper can be expressed by PAP and PSP, which represent papers written by the same author and papers belonging to the same subject, respectively. GNNs initially proposed for homogeneous graphs may encounter great challenges to handle relations with different semantics in heterogeneous graphs.

To address the above challenges, we propose a novel meta-path based unsupervised graph neural network model for heterogeneous graphs, namely **H**eterogeneous **D**eep **G**raph **I**nfomax (*HDGI*). In summary, our contributions in this paper can be summarized as follows:

- This paper presents the first model to apply mutual information maximization to representation learning in heterogeneous graphs.
- Our proposed method, *HDGI*, is a novel unsupervised graph neural network. It handles graph heterogeneity by utilizing an attention mechanism on meta-paths and deals with the unsupervised settings by applying mutual information maximization.
- Our experiments demonstrate that the representations learned by *HDGI* are effective for both node classification and clustering tasks. Moreover, its performance can also beat state-of-the-art GNN models, where they have the additional supervised label information.

2 Related Work

Graph representation learning. Graph representation learning has become a non-trivial topic [Cui *et al.*, 2018] because of the ubiquity of graphs in the real world. As a data type containing rich structural information, many models[Grover and Leskovec, 2016; Tang *et al.*, 2015] acting on graphs learn the representations of nodes based on the structure of the graph. DeepWalk [Perozzi *et al.*, 2014] uses the set of random walks over the graph in SkipGram to learn node embeddings. Several methods [Ou *et al.*, 2016; Wang *et al.*, 2017b] attempt to retrieve structural information through the matrix factorization. However, all the above methods are proposed for homogeneous graphs.

Heterogeneous graph learning. In order to handle the heterogeneity of graphs, metapath2vec [Dong *et al.*, 2017] samples random walks under the guidance of meta-paths and learns node embeddings through the skip-gram in heterogeneous graphs. HIN2Vec [Fu *et al.*, 2017] learns the embedding vectors of nodes and meta-paths simultaneously while conducts prediction tasks. Wang et al. [Wang *et al.*, 2019] consider the attention mechanism in heterogeneous graph learning, where information from multiple meta-path defined connections can be learned effectively. From the perspective of attributed graphs, SHNE [Zhang *et al.*, 2019] captures both structural closeness and unstructured semantic relations through joint optimization of heterogeneous SkipGram and deep semantic encoding. Many learning methods [Schlichtkrull *et al.*, 2018; Wang *et al.*, 2017a] on knowledge graphs can often be applied to other heterogeneous graphs.

Graph neural network. With the success of deep learning in the recent period, graph neural networks (GNNs) [Zhang, 2019] have made a lot of progress in graph representation learning. The core idea of GNN is to aggregate the feature information of the neighbors through neural networks to learn the new features that combine the independent information of the node and corresponding structural information in the graph. Most successful GNNs are based on supervised learning including GCN [Kipf and Welling, 2017a], GAT [Veličković *et al.*, 2017], GraphRNN [You *et al.*, 2018], SplineCNN [Fey *et al.*, 2018], AdaGCN [Sun *et al.*, 2019] and AS-GCN [Huang *et al.*, 2018]. The unsupervised learning GNNs can be mainly divided into two categories, i.e., random walk-based [Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Kipf and Welling, 2017b; Duran and Niepert, 2017; Hamilton *et al.*, 2017a] and mutual information-based [Veličković *et al.*, 2019].

3 Problem Formulation

In this section, we first introduce the concept of *heterogeneous graph* and the problem definition of heterogeneous graph representation learning. Next we define *meta-path based adjacency matrix*, which is useful in the following algorithm description.

Definition 3.1 (Heterogeneous Graph (HG)) *A heterogeneous graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{T}$ and an edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. Each node $v \in \mathcal{V}$ belongs to one particular node type in the node type set $\mathcal{T} : \phi(v) \in \mathcal{T}$, and each edge $e \in \mathcal{E}$ belongs to a particular edge type in the edge type set $\mathcal{R} : \psi(e) \in \mathcal{R}$. Heterogeneous graphs have the property that $|\mathcal{T}| + |\mathcal{R}| > 2$. The attributes and content of nodes can be encoded as initial feature matrix X.*

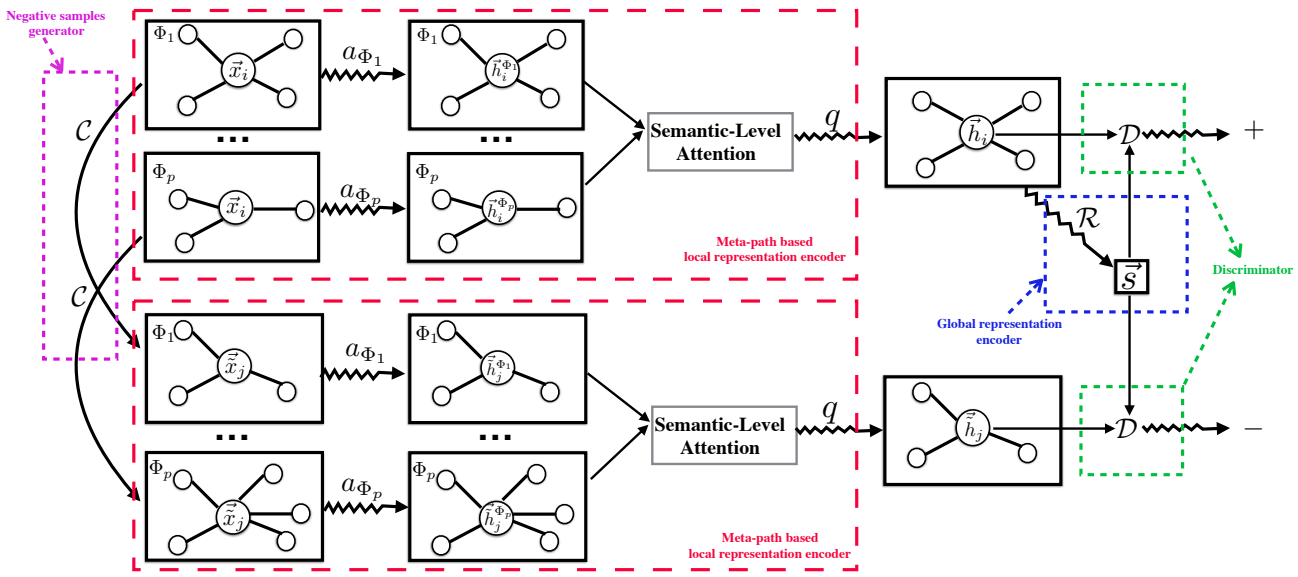


Figure 2 The high-level structure of *HDGI*. Local representation encoder is a hierarchical structure: learning node representations in terms of every meta-path based adjacency matrix respectively and then aggregating them through semantic-level attention. Global representation encoder $\mathcal{R}$ outputs a graph-level summary vector $\vec{s}$. Negative samples generator $\mathcal{C}$ is responsible for generating negative nodes. The discriminator $\mathcal{D}$ maximizes mutual information between positive nodes and the graph-level summary.

Problem Definition. (**Heterogeneous Graph Representation Learning**): Given a heterogeneous graph $\mathcal{G}$ and the initial feature matrix X, the representation learning task in $\mathcal{G}$ is to learn the low dimensional node representations $H \in \mathbb{R}^{|\mathcal{V}|\times d}$ which contains both structure information of $\mathcal{G}$ and node attributes of X. The learned representation H can be applied to downstream graph-related tasks such as node classification and node clustering, etc. We focus on learning the representation of one type of node in this paper. We represent such a set of nodes as the target-type nodes $\mathcal{V}_t$.

In a heterogeneous graph, two neighboring nodes can be connected by different types of edges. Meta-paths [Sun *et al.*, 2011], which represent node and edge types between two neighboring nodes in an HG, have been proposed to model such rich information. Formally, a path $v_1 \xrightarrow{R_1} v_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{n-1}} v_n$ is defined as a meta-path between nodes v_1 and v_n, wherein $R = R_1 \circ R_2 \circ \cdots \circ R_{n-1}$ defines the composite relations between node v_1 and v_n [Dong *et al.*, 2017]. In this paper, we intend to utilize symmetric and undirected meta paths to denote the closeness among target-type nodes $\mathcal{V}_t$, which can help simplify the problem setting.

We represent the set of meta paths used in this paper as $\{\Phi_1, \Phi_2, \cdots, \Phi_P\}$, where Φ_i denotes the i-th meta path type. For example, in Figure 1(d), Paper-Author-Paper (PAP) and Paper-Subject-Paper (PSP) are two types of meta-paths, which contain the semantic "papers written by the same author" and "papers belonging to the same subject" respectively.

Definition 3.2 (Meta-path based Adjacency Matrix) *For meta-path definition Φ_i, if there exists a path instance between node $v_i \in \mathcal{V}_t$ and $v_j \in \mathcal{V}_t$, we call that v_i and v_j are "connected neighbors" based on Φ_i. Such neighborhood information can be represented by a meta-path based adjacent matrix $A^{\Phi_i} \in \mathbb{R}^{|\mathcal{V}_t|\times|\mathcal{V}_t|}$, where $A^{\Phi_i}_{ij} = A^{\Phi_i}_{ji} = 1$ if v_i, v_j are connected by meta-path Φ_i and $A^{\Phi_i}_{ij} = A^{\Phi_i}_{ji} = 0$ otherwise.*

4 *HDGI* Methodology

4.1 *HDGI* Architecture Overview

A high-level illustration of the proposed *HDGI* is shown in Figure 2. The input of *HDGI* is a heterogeneous graph $\mathcal{G}$ containing N vertices whose initial d-dimension features are denoted by $X \in \mathbb{R}^{N\times d}$, and meta-path set $\{\Phi_i\}_{i=1}^P$. Based on $\{\Phi_i\}_{i=1}^P$ we can calculate the meta-path set based adjacency matrices $\{A^{\Phi_i}\}_{i=1}^P$. The meta-path based local representation encoding described in §4.2 has two steps: (1) learning individual node representation H^{Φ_i} from X and each $A^{\Phi_i}, i = 1, 2.., P$ and (2) generating node representation H by aggregating $\{H^{\Phi_i}\}_{i=1}^P$ through a semantic-level attention mechanism. A global representation encoder $\mathcal{R}$ is proposed to derive a graph summary vector $\vec{s}$ from H (see §4.3). The discriminator $\mathcal{D}$ will be trained with the objective to maximize mutual information between positive nodes and the graph-level summary $\vec{s}$. In §4.4 we elaborate the mutual information based discriminator $\mathcal{D}$ and the negative sample generator $\mathcal{C}$.

4.2 Meta-path based local representation encoder

Meta-path specific graph node representation learning. Each of $A^{\Phi_i}, i = 1, 2, ..P$ can be viewed as a homogeneous graph. At this step our target is to derive a node representation containing the information of initial node feature X and A^{Φ_i}, with a node-level encoder:

$$H^{\Phi_i} = a_{\Phi_i}(X, A^{\Phi_i}) \tag{1}$$

Two kinds of encoder are considered in this work. The first is Graph Convolutional Network (GCN) [Kipf and Welling, 2017a]. The nodes representation obtained by GCN is:

$$H^{\Phi_i} = (D^{\Phi_i}{}^{-\frac{1}{2}} A^{\tilde{\Phi}_i} D^{\Phi_i}{}^{-\frac{1}{2}}) X W^{\Phi_i} \tag{2}$$

where $A^{\tilde{\Phi}_i} = A^{\Phi_i} + I$, D^{Φ_i} is the diagonal node degree matrix of $A^{\tilde{\Phi}_i}$. Matrix $W^{\Phi_i} \in \mathbb{R}^{d \times F}$ is the filter parameter matrix.

The second encoder we consider is Graph Attention module (GAT) [Veličković *et al.*, 2017]. For the m-th node, its K-head attention output can be computed as:

$$\vec{h}_m^{\Phi_i} = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{j \in \mathcal{N}_m^{\Phi_i}} \alpha_{mj}^{\Phi_i,k} W^{\Phi_i} \vec{x}_j \Big) \tag{3}$$

where $\Vert$ is the concatenation operator, W^{Φ_i} is the linear transformation parameter matrix and $\mathcal{N}_m^{\Phi_i}$ is neighbor set defined by Φ_i. $\alpha_{mj}^{\Phi_i,k}$ is the normalized attention coefficient computed by the k-th attention mechanism.

After the node-level learning, we obtain the set of node representations $\{H^{\Phi_i}\}_{i=1}^{P}$. They are aggregated to get a heterogeneous graph based node representation.

Heterogeneous graph node representation learning. The representations learned based on the specific meta-path contain only the semantic-specific information. The key issue to accomplish the aggregation is exploring how much each meta-path should contribute to the final representations. Here we add a semantic attention layer L_{att} to learn the weights:

$$\{\beta^{\Phi_1}, \beta^{\Phi_2}, \ldots, \beta^{\Phi_P}\} = L_{att}(H^{\Phi_1}, H^{\Phi_2}, \ldots, H^{\Phi_P}) \tag{4}$$

Specifically, the importance of the meta-path Φ_i is calculated by

$$e^{\Phi_i} = \frac{1}{N} \sum_{n=1}^{N} \tanh(\vec{q}^{\mathrm{T}} \cdot [W_{sem} \cdot \vec{h}_n^{\Phi_i} + \vec{b}]) \tag{5}$$

where W_{sem} is a linear transformation parameter matrix. β^{Φ_i} is obtained by normalizing $\{e^{\Phi_i}\}_{i=1}^{P}$ with a softmax function:

$$\beta^{\Phi_i} = \mathrm{softmax}(e^{\Phi_i}) = \frac{\exp(e^{\Phi_i})}{\sum_{j=1}^{P} \exp(e^{\Phi_j})} \tag{6}$$

The heterogeneous graph node representation H is obtained by a linear combination of $\{H^{\Phi_i}\}_{i=1}^{P}$, that is

$$H = \sum_{i=1}^{P} \beta^{\Phi_i} \cdot H^{\Phi_i} \tag{7}$$

Our semantic attention layer is inspired by HAN [Wang *et al.*, 2019], but there are still some differences in the learning direction. HAN utilizes classification cross-entropy as the loss function, the learning direction is guided by known labels in the training set. However, the attention weights learned in *HDGI* are guided by the binary cross-entropy loss which indicates whether the node belongs to the original graph. Therefore, the weights learned in *HDGI* serve for the existence of a node. Because no classification label involves, the weights get no bias from the known labels.

The representations H serve as the final output local features. The global representation encoder leverages the representations H to output the graph-level summary which will be described in the following part.

4.3 Global Representation Encoder

The learning objective of *HDGI* is to maximize the mutual information between local representations and the global representation. The local representations of nodes are included in H, and we need the summary vector $\vec{s}$ to represent the global information of the entire heterogeneous graph. Based on H, we examined three candidate encoder functions:

Averaging encoder function. Our first candidate encoder function is the averaging operator, where we simply take the mean of the node representations to output the graph-level summary vector $\vec{s}$:

$$\vec{s} = \sigma\left(\frac{1}{N} \sum_{i=1}^{N} \vec{h}_i \right) \tag{8}$$

Pooling encoder function. In this pooling encoder function, each node's vector is independently fed through a fully-connected layer. An elementwise max-pooling operator is applied to summary the information from the nodes set:

$$\vec{s}_{pool} = max(\{\sigma(W_{pool}\vec{h}_i + b), i \in \{1, 2, \ldots, N\}) \tag{9}$$

where max denotes the element-wise max operator and σ is a nonlinear activation function.

Set2vec encoder function. The final encoder function we examine is Set2vec [Oriol Vinyals, 2016], which is based on an LSTM architecture. Because the original set2vec in [Oriol Vinyals, 2016] works on ordered node sequences, but here we need a summary of the graph concluding comprehensive information from each node instead of merely graph structure. Therefore, we apply the LSTMs to a random permutation of the node's neighbor on an unordered set.

4.4 *HDGI* Learning

Mutual information based discriminator. Belghazi et al. proved that the KL-divergence admits the Donsker-Varadhan representation and the f-divergence representation as dual representations in [Belghazi *et al.*, 2018]. The dual representations provide a lower-bound to the mutual information of random variables X and Y:

$$\mathrm{MI}(X;Y) \geq \mathbb{E}_{\mathbb{P}_{XY}}[T_\omega(x,y)] - log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Y}[e^{T_\omega(x,y)}]) \tag{10}$$

Here, $\mathbb{P}_{XY}$ is the joint distribution and $\mathbb{P}_X \otimes \mathbb{P}_Y$ is the product of margins. T_ω is a deep neural network based discriminator parametrized by ω. The expectations in equ.10 can be estimated using samples from $\mathbb{P}_{XY}$ and $\mathbb{P}_X \otimes \mathbb{P}_Y$. The expressive power of the discriminator ensures to approximate the MI with high accuracy. Here, we estimate and maximize the mutual information simultaneously by training a discriminator $\mathcal{D}$ to distinguish positive sample set $Pos = \left\{ [\vec{h}_n, \vec{s}] \right\}_{n=1}^{N}$ with negative sample set $Neg = \left\{ [\tilde{\vec{h}}_m, \vec{s}] \right\}_{m=1}^{M}$. The sample $(\vec{h}_i, \vec{s})$ is denoted as positive as node $\vec{h}_i$ belongs to the original graph (the joint distribution), and $(\tilde{\vec{h}}_j, \vec{s})$ is negative as the node $\tilde{\vec{h}}_j$ is the generated fake one (the product of marginals). The discriminator $\mathcal{D}$ is a bilinear layer:

$$\mathcal{D}(\vec{h}_i, \vec{s}) = \sigma(\vec{h}_i^{\mathrm{T}} W_D \vec{s}) \tag{11}$$

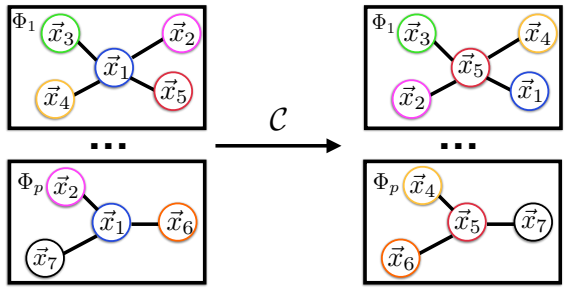


Figure 3: The example of generating negative samples

Here W_D is a learnable matrix and σ is the sigmoid activation function. In our problem, based on the approximately monotonic relationship between Jensen-Shannon divergence and mutual information [Hjelm *et al.*, 2019], we can maximize the mutual information with the binary cross-entropy loss of the discriminator:

$$\mathcal{L}(Pos, Neg, \vec{s}) = \frac{1}{N+M}\left(\sum_{n=1}^{N}\mathbb{E}_{Pos}[\log\mathcal{D}(\vec{h}_n, \vec{s})] + \sum_{m=1}^{M}\mathbb{E}_{Neg}[\log(1-\mathcal{D}(\vec{\tilde{h}}_m, \vec{s}))]\right) \tag{12}$$

In essence, the discriminator works to maximize the mutual information between a high-level global representation and local representations(node-level), which encourages the encoder to learn the information presenting in all globally relevant locations. The information about a class label can be one of the cases. The above loss can be optimized through the gradient descent, and the representations of nodes can be learned when the optimization is completed.

Negative samples generator. The negative sample set $\left\{[\vec{\tilde{h}}_m, \vec{s}]\right\}_{m=1}^{M}$ is composed of the samples that do not exist in the heterogeneous graph. We extend the negative sample geneation approach proposed in [Veličković *et al.*, 2019] to heterogeneous graph setting.

In heterogeneous graph $\mathcal{G}$, we have rich and complex structural information characterized by meta-path based adjacency matrices. Our negative samples generator

$$\tilde{X}, \{A^{\Phi_1}, A^{\Phi_2}, \ldots, A^{\Phi_P}\} = \mathcal{C}(X, \{A^{\Phi_1}, A^{\Phi_2}, \ldots, A^{\Phi_P}\}) \tag{13}$$

keeps all meta-path based adjacency matrices unchanged, which can make the overall structure of $\mathcal{G}$ stable. We shuffle the rows of the initial node feature matrix X, which changes the index of nodes in order to corrupt the node-level connections among them. The structure of the whole graph does not change, but the initial feature attribute corresponding to each node is changed. We provide a simple example to illustrate the procedure of generating negative samples in Figure 3.

5 Evaluation

5.1 Datasets

We evaluate the performance of *HDGI* on three heterogeneous graph data, and summarize their details in Table 1.

Table 1: Statistics of experimented datasets.

Dataset	Node-type	# Nodes	Edge-type	# Edges	feature	Meta-path
ACM	Paper (**P**) Author (**A**) Subject (**S**)	3025 5835 56	Paper-Author Paper-Subject	9744 3025	1870	PAP PSP
IMDB	Movie (**M**) Actor (**A**) Director (**D**) Keyword (**K**)	4275 5431 2082 7313	Movie-Actor Movie-Director Movie-keyword	12838 4280 20529	6344	MAM MDM MKM
DBLP	Author (**A**) Paper (**P**) Conference (**C**) Term (**T**)	4057 14328 20 8789	Author-Paper Paper-Conference Paper-Term	19645 14328 88420	334	APA APCPA APTPA

- *DBLP* [Gao *et al.*, 2009]: This is a research paper set, which contains scientific publications and the corresponding authors. The target author node can be divided into 4 areas: database, data mining, information retrieval, and machine learning. We use the area of authors as the labels. The initial features are generated based on authors' profiles with the bag-of-words embeddings .
- *ACM* [Wang *et al.*, 2019]: This is another academic paper data in which target paper nodes are categorized into 3 classes: database, wireless communication and data Mining. The initial features are constructed from paper keywords with the TF-IDF based embedding techniques.
- *IMDB* [Wu *et al.*, 2016]: It is a knowledge graph data about movies (target nodes) that can be categorized into three types: Action, Comedy, and Drama. The feature of a movie is composed of {color, title, language, keywords, country, rating, year} with a TF-IDF encoding.

5.2 Experiment Setup

The most commonly used tasks to measure the quality of learned representations are node classification [Perozzi *et al.*, 2014; Grover and Leskovec, 2016; Hamilton *et al.*, 2017b] and node clustering [Dong *et al.*, 2017; Wang *et al.*, 2019]. We evaluate *HDGI* from both two types of tasks.

Compared Baselines

HDGI is compared with the following supervised and unsupervised methods:

Unsupervised methods

- *Raw Feature*: The initial features are used as embeddings.
- *Metapath2vec (M2V) [Dong* et al.*, 2017]*: A meta-path based graph embedding method for heterogeneous graph. We test all meta-paths and report the best result.
- *DeepWalk (DW) [Perozzi* et al.*, 2014]*: A random walk based graph embedding method for homogeneous graph.
- *DeepWalk+Raw Feature(DW+F)*: It concatenates the learned DeepWalk embeddings with the raw features as the final representations.
- *DGI [Veličković* et al.*, 2019]*: A mutual information based graph representation method for homogeneous graph.
- *HDGI-C*: It is a variant of *HDGI* which uses graph convolutional network to capture local representations.

Table 2: The results of node classification tasks

Available data			X	A		X, A, Y			X, A				
Dataset	Train	Metric	Raw	M2V	DW	GCN	RGCN	GAT	HAN	DW+F	DGI	**HDGI-A**	**HDGI-C**
ACM	20%	Micro-F1	0.8590	0.6125	0.5503	0.9250	0.5766	0.9178	0.9267	0.8785	0.9104	0.9178	**0.9227**
		Macro-F1	0.8585	0.6158	0.5582	0.9248	0.5801	0.9172	0.9268	0.8789	0.9104	0.9170	**0.9232**
	80%	Micro-F1	0.8820	0.6378	0.5788	0.9317	0.5939	0.9250	0.9400	0.8965	0.9175	0.9333	**0.9379**
		Macro-F1	0.8802	0.6390	0.5825	0.9317	0.5918	0.9248	0.9403	0.8960	0.9155	0.9330	**0.9379**
DBLP	20%	Micro-F1	0.7552	0.6985	0.2805	0.8192	0.1932	0.8244	0.8992	0.7163	0.8975	0.9062	**0.9175**
		Macro-F1	0.7473	0.6874	0.2302	0.8128	0.2132	0.8148	0.8923	0.7063	0.8921	0.8988	**0.9094**
	80%	Micro-F1	0.8325	0.8211	0.3079	0.8383	0.2175	0.8540	0.9100	0.7860	0.9150	0.9192	**0.9226**
		Macro-F1	0.8152	0.8014	0.2401	0.8308	0.2212	0.8476	0.9055	0.7799	0.9052	0.9106	**0.9153**
IMDB	20%	Micro-F1	0.5112	0.3985	0.3913	0.5931	0.4350	0.5985	0.6077	0.5262	0.5728	0.5482	**0.5893**
		Macro-F1	0.5107	0.4012	0.3888	0.5869	0.4468	0.5944	0.6027	0.5293	0.5690	0.5522	**0.5914**
	80%	Micro-F1	0.5900	0.4203	0.3953	0.6467	0.4476	0.6540	0.6600	0.6017	0.6003	0.5861	**0.6592**
		Macro-F1	0.5884	0.4119	0.4001	0.6457	0.4527	0.6550	0.6586	0.6049	0.5950	0.5834	**0.6646**

- *HDGI-A*: This is another variant of *HDGI* which uses graph attention mechanism to learn local representations.

Supervised methods

- *GCN [Kipf and Welling, 2017a]*: A semi-supervised methods for node classification on homogeneous graphs.
- *RGCN [Schlichtkrull* et al*., 2018]*: It performs representation learning on all nodes labeled with entity types in heterogeneous graphs. We report the best performance from all meta-paths and the graph ignoring type information.
- *GAT [Veličković* et al*., 2017]*: GAT applies the attention mechanism in homogeneous graphs for node classification.
- *HAN [Wang* et al*., 2019]*: HAN employs node-level attention and semantic-level attention to capture the information from all meta-paths.

For methods designed for homogeneous graphs, i.e., *DeepWalk*, *DGI*, *GCN*, *GAT*, we do not consider graph heterogeneity and construct meta-path based adjacency matrix, then we report the best performance. We test all meta-paths for *Metapath2vec* and report the best result. For *RGCN*, because our task is to learn the representations of target-type nodes, the cross-entropy loss is calculated by the classification in target-type nodes only.

Reproducibility

For the proposed *HDGI* including *HDGI-C* and *HDGI-A*, we optimize the model with Adam [Kingma and Ba, 2015]. The dimension of node-level representations in *HDGI-C* is set as 512 and the dimension of $\vec{q}$ is set as 8. For *HDGI-A*, we set the dimension of node-level representations as 64 and the attention head is set as 4. The dimension of $\vec{q}$ is set as 8 as well. We employ Pytorch to implement our model and conduct experiments in the server with 4 GTX-1080ti GPUs. Code is available at https://github.com/YuxiangRen/Heterogeneous-Deep-Graph-Infomax.

5.3 Performance Comparison

Node classification task

In the node classification task, we train a logistic regression classifier for unsupervised learning methods, while the supervised methods output the classification result as end-to-end models. We conduct the experiments with two different training ratios (20% and 80%). The ratios of validation set and test set are fixed at 10%. To keep the results stable, we repeat the classification process for 10 times and report the average Macro-F1 and Micro-F1 in Table 2. We observe that *HDGI-C* outperforms all other unsupervised learning methods. When competing with the supervised learning methods (designed for homogeneous graphs like *GCN* and *GAT*), *HDGI* can perform much better. This observation proves that the type and semantic information are very important and need to be handled carefully instead of directly ignoring them in heterogeneous graphs. The result of *RGCN* is suboptimal, because the original RGCN is a featureless approach and we follow the code to assign a one-hot vector to each node.

Table 3: Evaluation results on the node clustering task

Data	ACM		DBLP		IMDB	
Method	NMI	ARI	NMI	ARI	NMI	ARI
DeepWalk	25.47	18.24	7.40	5.30	1.23	1.22
Raw Feature	32.62	30.99	11.21	6.98	1.06	1.17
DeepWalk+F	32.54	31.20	11.98	6.99	1.23	1.22
Metapath2vec	27.59	24.57	34.30	37.54	1.15	1.51
DGI	41.09	34.27	59.23	61.85	0.56	2.6
HDGI-A	**57.05**	**50.86**	52.12	49.86	0.8	1.29
HDGI-C	54.35	49.48	**60.76**	**62.67**	**1.87**	**3.7**

In addition, the unified learning of all types of nodes in the same latent space, is beneficial to entity type classification but may not be applicable to label classification. *HDGI* is also competitive with the result reported from *HAN* which is designed for heterogeneous graphs. The reason should be that *HDGI* can capture more global structural information when exploring the mutual information in reconstructing the representation, while supervised loss based GNNs overemphasize the direct neighborhoods [Veličković *et al.*, 2019]. This observation, on the other hand, suggests that the features learned through supervised learning in graph structures may have limitations, either from the structure or a task-based preference. These limitations can affect the learning representations from a more general perspective badly.

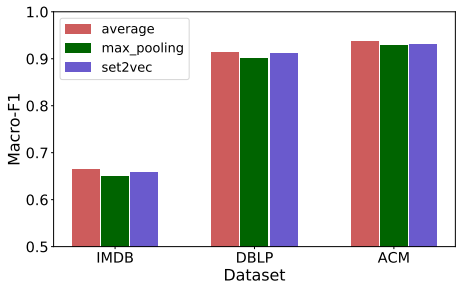
Node clustering task

In the node clustering task, we use the K-Means to conduct the clustering based on the learned representations. The number of clusters K is set as the number of the target node

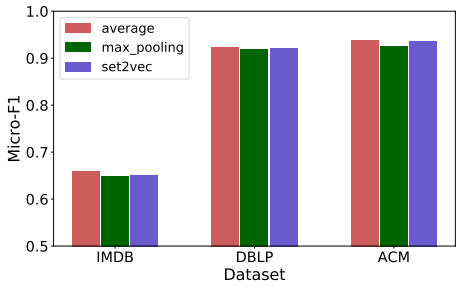
classes. We do not use any label in this unsupervised learning task and make the comparison among all unsupervised learning methods. We also repeat the clustering process for 10 times and report the average NMI and ARI in Table 2. *DeepWalk* cannot perform well because they are not able to handle the graph heterogeneity. *Metapath2vec* cannot handle diversitified semantic information simultaneously, which makes the representations not effective enough. The verification based on node clustering tasks also demonstrates that *HDGI* can learn effective representations by considering the structural information, the semantic information and the node independent information simultaneously.

HDGI-A vs HDGI-C

From the comparison between *HDGI-C* and *HDGI-A* in node classification tasks, the results reflects some interesting findings. *HDGI-C* achieves better performance than *HDGI-A* in all experiments, which means that the graph convolution works better than the attention mechanism in capturing local network structures. The reason might be that the graph attention mechanism is strictly limited to the direct neighbors of nodes, the graph convolution considering hierarchical dependencies can see farther. This analysis can also be verified by the results of the clustering task.



(a) Macro-F1



(b) Micro-F1

Figure 4: The comparison between different global representation encoder functions

Different global representation encoder functions

We present the results of *HDGI-C* with different global representation encoder functions working on the node classification task in Figure 4. The simple average function performs the best compared with other functions. However, we can find that this advantage is very subtle. In fact, each function can perform well on experimental datasets. But for larger and more complex heterogeneous graphs, a specified and sophisticated function may perform better. The design of the global encoder function for heterogeneous graphs with different scales and structures is an open question, which is worthy of further discussion.

6 Conclusion

In this paper, we propose an unsupervised graph neural network, *HDGI*, which learns node representations in heterogeneous graphs. *HDGI* combines several state-of-the-art techniques. It employs convolution-style GNNs along with a semantic-level attention mechanism to capture individual local representations of nodes. Through maximizing the local-global mutual information, *HDGI* learns high-level representations containing graph-level structural information. It exploits the structure of meta-path to model the connection semantics in heterogeneous graphs. We demonstrate the effectiveness of learned representations for both node classification and clustering tasks on three heterogeneous graphs. *HDGI* is particularly competitive in node classification tasks with state-of-the-art supervised methods, where they have the additional supervised label information. We are optimistic that mutual information maximization is a promising future direction for unsupervised representation learning.

References

[Belghazi *et al.*, 2018] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *ICML*, 2018.

[Cui *et al.*, 2018] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, 2018.

[Dong *et al.*, 2017] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144. ACM, 2017.

[Duran and Niepert, 2017] Alberto Garcia Duran and Mathias Niepert. Learning graph representations with embedding propagation. In *Advances in neural information processing systems*, pages 5119–5130, 2017.

[Fey *et al.*, 2018] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018.

[Fu *et al.*, 2017] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1797–1806. ACM, 2017.

[Gao *et al.*, 2009] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Advances in Neural Information Processing Systems*, pages 585–593, 2009.

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

[Hamilton *et al.*, 2017a] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[Hamilton *et al.*, 2017b] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. In *IEEE Data Engineering Bulletin*, 2017.

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

[Huang *et al.*, 2018] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *Advances in Neural Information Processing Systems*, pages 4558–4567, 2018.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation*, 2015.

[Kipf and Welling, 2017a] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representation*, 2017.

[Kipf and Welling, 2017b] T. N. Kipf and M. Welling. Variational graph auto-encoders. In *International Conference on Learning Representation*, 2017.

[Oriol Vinyals, 2016] Manjunath Kudlur Oriol Vinyals, Samy Bengio. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representation*, 2016.

[Ou *et al.*, 2016] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1105–1114. ACM, 2016.

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

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[Sun *et al.*, 2011] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

[Sun *et al.*, 2019] Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Adagcn: Adaboosting graph convolutional networks into deep models. *arXiv preprint arXiv:1908.05081*, 2019.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei.

Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Veličković *et al.*, 2019] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *International Conference on Learning Representation*, 2019.

[Wang *et al.*, 2017a] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[Wang *et al.*, 2017b] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Wang *et al.*, 2019] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032. ACM, 2019.

[Wu *et al.*, 2016] Chao-Yuan Wu, Alex Beutel, Amr Ahmed, and Alexander J Smola. Explaining reviews and ratings with paco: Poisson additive co-clustering. In *WWW*, pages 127–128. ACM, 2016.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Chang. Network representation learning with rich text information. In *International Joint Conference on Artificial Intelligence*, 2015.

[You *et al.*, 2018] Jiaxuan You, Rex Ying, Xiang Ren, William L Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. *arXiv preprint arXiv:1802.08773*, 2018.

[Zhang *et al.*, 2016] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Collective classification via discriminative matrix factorization on sparsely labeled networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1563–1572. ACM, 2016.

[Zhang *et al.*, 2019] Chuxu Zhang, Ananthram Swami, and Nitesh V. Chawla. Shne: Representation learning for semantic-associated heterogeneous networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 690–698. ACM, 2019.

[Zhang, 2018] Jiawei Zhang. Social network fusion and mining: a survey. *arXiv preprint arXiv:1804.09874*, 2018.

[Zhang, 2019] Jiawei Zhang. Graph neural networks for small graph and giant network representation learning: An overview. *arXiv preprint arXiv:1908.00187*, 2019.