Self-supervised Deep Heterogeneous Graph Neural Networks with Contrastive Learning

Zhiping Li[1,2], Fangfang Yuan[1](✉), Cong Cao[1](✉), Dakui Wang[1], Jiali Feng[1,2], Baoke Li[1,2], and Yanbing Liu[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{lizhiping, yuanfangfang, caocong, wangdakui, fengjiali, libaoke, liuyanbing}@iie.ac.cn

Abstract. Heterogeneous graph neural networks have shown superior capabilities on graphs that contain multiple types of entities with rich semantic information. However, they are usually (semi-)supervised learning methods which rely on costly task-specific labeled data. Due to the problem of label sparsity on heterogeneous graphs, the performance of these methods is limited, prompting the emergence of some self-supervised learning methods. However, most of self-supervised methods aggregate meta-path based neighbors without considering implicit neighbors that also contain rich information, and the mining of implicit neighbors is accompanied by the problem of introducing irrelevant nodes. Therefore, in this paper we propose a self-supervised deep heterogeneous graph neural networks with contrastive learning (DHG-CL) which not only preserves the information of implicitly valuable neighbors but also further enhances the distinguishability of node representations. Specifically, (1) we design a cross-layer semantic encoder to incorporate information from different high-order neighbors through message passing across layers; and then (2) we design a graph-based contrastive learning task to distinguish semantically dissimilar nodes, further obtaining discriminative node representations. Extensive experiments conducted on a variety of real-world heterogeneous graphs show that our proposed DHG-CL outperforms the state-of-the-arts.

Keywords: Heterogeneous graph neural networks · Self-supervised learning · Contrastive learning.

1 INTRODUCTION

Heterogeneous graphs are widely present in real-world networks, which can model various types of entities and their relations, such as academic networks, social networks, etc. Fig. 1(a) shows an example of a heterogeneous graph. Recently, heterogeneous graph neural networks (HGNNs) have become an emerging tool for mining heterogeneous graph-structured data by aggregating the properties of neighbors. Precisely because of preserving both attribute and structural information, HGNNs show superior performance in various graph-data mining tasks such as node classification, link prediction, and graph classification. In fact, HGNNs are usually trained under the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_19

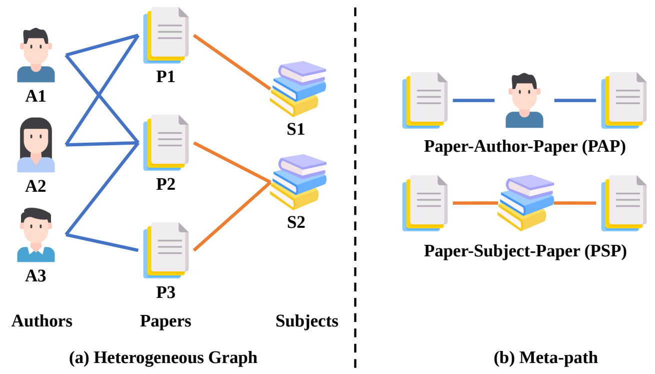


Fig. 1. A toy example of a heterogeneous graph (ACM) and its meta-path. (a) A heterogenous graph ACM consists three types of nodes (author (A), paper (P) and subject (S)) and two types of relations. (b) Two meta-paths involved in ACM (i.e., Paper-Author-Paper and Paper-Subject-Paper).

(semi-)supervised learning paradigm, which means that the process requires the task-specific labeled data. In most real-world scenarios, it is very difficult and expensive to obtain labeled data. For example, in the case of microbiome networks, labeling these data requires domain-specific knowledge which is often scarce or non-existent [33].

Motivated by methods [19, 20], self-supervised learning capable of training deep models on unlabeled data promises to be a solution. Contrastive learning as a typical method of self-supervised learning has drawn massive attention for its outstanding performance [1, 4, 23, 29], exploiting instance discrimination as a pretext task to learn more discriminative representations. Recently, some efforts have been devoted to investigating the potential of contrastive self-supervised learning on heterogeneous graphs. For example, DMGI [17] using mutual information and HeCo [28] using co-contrastive learning achieve desirable results on heterogeneous graphs. However, they only focus on aggregating meta-path based neighbors without considering implicit neighbors that also contain rich information.

To fully utilize the information of implicit neighbors and reduce the influence of irrelevant neighbors on heterogeneous graphs, in this paper, we take an attempt to perform a contrastive self-supervised learning on heterogeneous graphs. However, mining the valuable neighbors on heterogeneous graphs is a non-trivial problem, presenting us with two key challenges:

1. *How to fully mine the relevant neighbors?* It is well known that meta-paths [22] can describe different semantic contents between the nodes and their neighbors. However, due to the complexity of heterogeneous graphs, there are many deep implicit semantics on heterogeneous graphs and it may be informative for the learning of node representations. For example, in Fig. 1, it only uses the local neighbors of the paper P1 to predict the topic of P1 through the meta-path PAP. However, the paper P3 and P1's neighbor P2 belong to the same subject and P3 is also helpful to predict the topic of P1. That is to say, P3 is the implicit valuable neighbor of P1.

Therefore, modeling these implicit higher-order semantics between the nodes and their neighbors is critical for HGNNs.

2. *How to alleviate the influence of irrelevant neighbors?* We note that for each node, increasing the order of neighbors will lead to an increase in the number of irrelevant neighbors. Contrastive learning aiming to pulling positive samples together and pushing apart negative samples can be used to distinguish irrelevant neighbors. However, there is a high correlation between nodes in the graph, so we cannot treat all other nodes as negative samples like in images [1] and sentences [4]. Therefore, it is imperative to construct a graph-based contrastive learning task on heterogeneous graphs.

To address the above challenges, we propose a self-supervised deep heterogeneous graph neural networks with contrastive learning, named DHG-CL, which not only preserves the information of implicitly valuable neighbors but also further enhances the distinguishability of node representations. Specifically, we first design a projection function on heterogeneous graphs that maps different types of nodes into the same low-dimensional space. Next, to address the first challenge, we design a cross-layer semantic encoder, including a semantic-aware attention mechanism and a cross-layer message passing mechanism, which focuses on implicitly aggregating high-order neighbors in different semantic spaces and explores the deep semantics. Then, to address the second challenge, we introduce a graph-based contrastive learning task to distinguish irrelevant neighbors, further learning discriminative representations. More specifically, we construct a graph-based sampling strategy: (1) Inspired by the data augmentation [4], we pass the same node to the encoder twice with standard dropout to obtain a positive pair, aiming to generate a different contrastive object without changing the graph topology, so that we can get a strong supervision signal. (2) Because of the high correlation between nodes on the graph, we only treat nodes that are semantically irrelevant as negative samples. To delineate semantic irrelevance, we take the nodes with the number of meta-paths less than the threshold to the target node as negative samples. To summarize, the main contributions of this paper are as follows:

– We propose a self-supervised deep heterogeneous graph neural networks with contrastive learning. Unlike previous meta-path based methods, DHG-CL can fully mine the higher-order valuable neighbors on heterogeneous graphs to achieve a more informative node representation.
– We propose a graph-based contrastive learning task on heterogeneous graphs. Specifically, we introduce a dropout mask and semantic based negative sampling strategy to further learn discriminative representations.
– We conduct rich experiments on four real-world datasets and the results demonstrate that the proposed DHG-CL significantly outperforms the state-of-the-arts.

2 RELATED WORK

In this section, we review recent developments in heterogeneous graph neural networks and contrastive learning.

2.1 Heterogeneous Graph Neural Networks

Graph neural networks have attracted extensive attention due to their superior performance in modeling graph-structured data. Many researchers [26, 30] have made a detailed summary of them. Some works [11, 24] propose to use convolutional neural networks and attention mechanisms for homogeneous graph. In recent years, some HGNNs [2,6–8,27] are proposed to learn node representations on heterogeneous graphs. For example, Wang et al. [27] propose a heterogeneous graph attention network named HAN consisting of node-level and semantic-level attention. Ji et al. [8] propose a deeper architecture and improve the node-level aggregating process to alleviate the semantic confusion. However, since the above HGNNs usually rely on a large amount of labeled data, they fail to fully perform well in sparse labeled scenarios. In this paper, we perform a self-supervised learning on heterogeneous graphs to learn the deep structural properties of graphs.

2.2 Contrastive Learning

The methods of contrastive learning have shown their success in self-supervised learning by distinguishing positive and negative samples [13,14]. In CV [1,5] and NLP [4,16], many excellent contrastive learning methods have emerged. For example, Chen et al. [1] propose SimCLR, which utilizes data augmentation techniques to generate two related views as a positive pair by randomly transforming a given image. Gao et al. [4] propose SimCSE to perform data augmentation through dropout operations. In graph learning, studies on contrastive learning [9, 25, 28] are proposed to learn node representations. Veličković et al. [25] propose DGI to maximize the mutual information between global representation and local patches. On heterogeneous graphs, Park et al. [17] propose DMGI to integrate the node embeddings by introducing the consensus regularization framework and the universal discriminator. Wang et al. [28] propose HeCo, which constructs a cross-view contrastive learning task after learning node embeddings from two views (network schema and meta-path views). However, both of them ignore the mining of deep implicit semantics, failing to explore neighbors other than the meta-path based neighbors. To go one step further, we implement a cross-layer semantic encoder to explore implicitly valuable neighbors, and then design graph-based contrastive learning to reduce the influence of irrelevant neighbors.

3 PRELIMINARY

In this section, we will present several basic concepts of heterogeneous graphs.

Definition 1. ***Heterogeneous Graph.*** *A heterogeneous graph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges. It is also associated with type mapping functions, including a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge type mapping function $\varphi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote a set of node types and a set of edge types, where $|\mathcal{A}| + |\mathcal{R}| > 2$.*

Definition 2. ***Meta-path.*** *A meta-path $\mathcal{P}$ is defined as a path which consists of a set of nodes and edges. It is also in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$(abbreviated as $A_1 A_2 \cdots A_{l+1}$), which can also be described as $R_1 \circ R_2 \circ \cdots \circ R_l$, where $\circ$ denote a combination operator on relations.*

Definition 3. ***Semantic context.*** *Given a meta-path $\mathcal{P}$, the semantic context $\mathcal{N}_i^{\mathcal{P}}$ of node i is defined as a set of neighbors connected to node i via meta-path $\mathcal{P}$.*

4 THE PROPOSED DHG-CL MODEL

In this section, we will introduce the proposed DHG-CL in detail and Fig. 2 presents the overview of DHG-CL. We encode deep implicit semantics on heterogeneous graphs through a semantic-aware attention mechanism and a cross-layer message passing mechanism. To further enhance the distinguishability of embeddings, we introduce dropout mask and semantic based negative sampling strategy to construct a graph-based contrastive learning task. Finally, DHG-CL iteratively updates node embeddings via optimizing the contrastive loss.

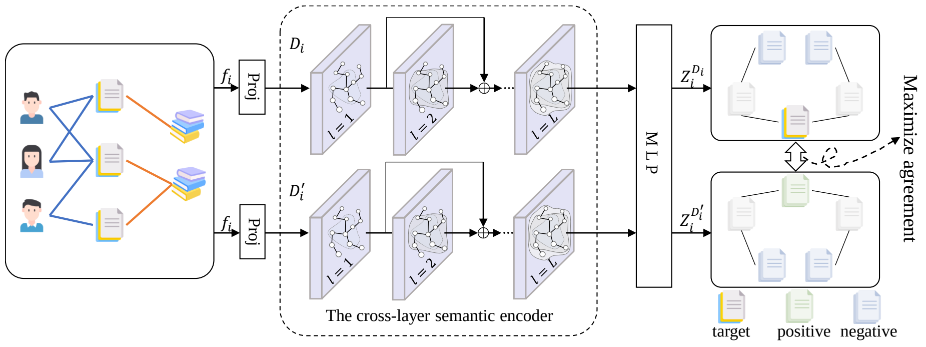


Fig. 2. The overview of the proposed DHG-CL.

4.1 Node Transformation

On heterogeneous graphs, different types of nodes lie in different feature spaces due to the heterogeneity of nodes. Therefore, we first apply a type-specific transformation to project different types of nodes into the same feature space. Specifically, we design a transformation matrix W_{ϕ_i} for node i with type ϕ_i:

$$h_i = \sigma \left(W_{\phi_i} \cdot f_i \right), \tag{1}$$

where h_i is the projected embedding of node i, σ is the activation function and f_i is the feature of node i. h_i can also be viewed as the node embedding h_i^0.

4.2 Cross-layer Semantic Encoder

To capture the deep implicit semantics, we design a cross-layer semantic encoder, illustrated as Fig. 3. Now, we give a brief introduction to it, including neighbor aggregation and message combination:

$$h_i^l \leftarrow \mathrm{Combine}^l\left(\mathrm{Aggregate}^l\left(h_j^{l-1}\right), h_i^{l-1}\right), \tag{2}$$

where $\mathrm{Aggregate}^l(\cdot)$ is the neighbor aggregation function in layer l, $\mathrm{Combine}^l(\cdot)$ is the message combination function in layer l, h_j^{l-1} is the output embedding of node j in layer $l-1$, node $j \in \mathcal{N}_i$, denoted as $\left\{\mathcal{N}_i^{\mathcal{P}_1}, \ldots, \mathcal{N}_i^{\mathcal{P}_m}\right\}$, $\mathcal{N}_i^{\mathcal{P}_n}$ is the set of neighbors of node i in meta-path $\mathcal{P}_n$.

Neighbor Aggregation Since there are multiple semantic contexts, in order to obtain sufficient expressive ability, we design a semantic-aware attention mechanism. To be specific, we do not simply fuse them in the same space [27], but instead maintain its own semantic space for the different semantics. Therefore, we design a semantic projection to map semantic context into their own space:

$$h_{j,\mathcal{P}}^l = W_{\mathcal{P}}^l \cdot h_j^{l-1}, \tag{3}$$

where $W_{\mathcal{P}}^l$ is the semantic-specific transformation matrix, node $j \in \mathcal{N}_i^{\mathcal{P}}$.

After that, different neighbors will make different contributions to the target node, so we design an intra-semantic attention to preserve the important information of neighbors as much as possible:

$$e_{ij,\mathcal{P}}^l = \mathrm{LeakyReLU}\left({a_{\mathcal{P}}^l}^{\mathrm{T}} \cdot [h_{i,\mathcal{P}}^l \| h_{j,\mathcal{P}}^l]\right), \tag{4}$$

where $e_{ij,\mathcal{P}}^l$ indicates the importance of node j to node i in meta-path $\mathcal{P}$, $a_{\mathcal{P}}^l$ is the semantic-specific attention vector for meta-path $\mathcal{P}$, $\|$ is the concatenate operation, $\mathrm{LeakyReLU}(\cdot)$ is the activation function (with negative slope $a = 0.01$).

Then we perform mask attention which means we only compute $e_{ij,\mathcal{P}}^l$ for nodes $j \in \mathcal{N}_i^{\mathcal{P}}$ to inject the graph structure into the model. After obtaining the different importance of nodes, we normalize them to get the weight $\alpha_{ij,\mathcal{P}}^l$ by softmax function:

$$\alpha_{ij,\mathcal{P}}^l = \frac{\exp\left(\mathrm{LeakyReLU}\left({a_{\mathcal{P}}^l}^{\mathrm{T}} \cdot [h_{i,\mathcal{P}}^l \| h_{j,\mathcal{P}}^l]\right)\right)}{\sum\limits_{k \in \mathcal{N}_i^{\mathcal{P}}} \exp\left(\mathrm{LeakyReLU}\left({a_{\mathcal{P}}^l}^{\mathrm{T}} \cdot [h_{i,\mathcal{P}}^l \| h_{k,\mathcal{P}}^l]\right)\right)}. \tag{5}$$

Next, we aggregate the context information with the corresponding weights:

$$\tilde{h}_{i,\mathcal{P}}^l = \sigma\left(\sum_{j \in \mathcal{N}_i^{\mathcal{P}}} \alpha_{ij,\mathcal{P}}^l \cdot h_{j,\mathcal{P}}^l\right), \tag{6}$$

where $\tilde{h}^l_{i,\mathcal{P}}$ is the learned embedding of node i in semantic $\mathcal{P}$. For example, as shown in Fig. 3, we learn the embedding of node i with semantics "co-author" and "co-subject" in ACM. Then, we design a concatenation operation to fuse the semantics in different spaces, and then map the fused embedding of node i back to its original space:

$$\tilde{h}^l_i = W^l_o \cdot \text{Concatenate}\left(\tilde{h}^l_{i,\mathcal{P}_1}, \ldots, \tilde{h}^l_{i,\mathcal{P}_m}\right), \tag{7}$$

where W^l_o is the transformation matrix, $\tilde{h}^l_i$ is the aggregated embedding of node i in layer l.

Message Combination For meta-path based HGNNs, they only need one layer to aggregate the information of neighbor j into node i. However, in this way the model can only learn a single manually designing semantic, ignoring the implicit higher-order neighbors. To capture implicit higher-order proximity information, we need a deeper model. Therefore, we stack our model to L layers, which enables nodes to reach a large proportion of nodes on the graph and mines the deeper semantics. In addition, to preserve the semantic information of each layer as much as possible, we design a cross-layer message passing mechanism:

$$h^l_i = h^{l-1}_i + \tilde{h}^l_i, \tag{8}$$

where h^l_i is the output embedding of node i in layer l.

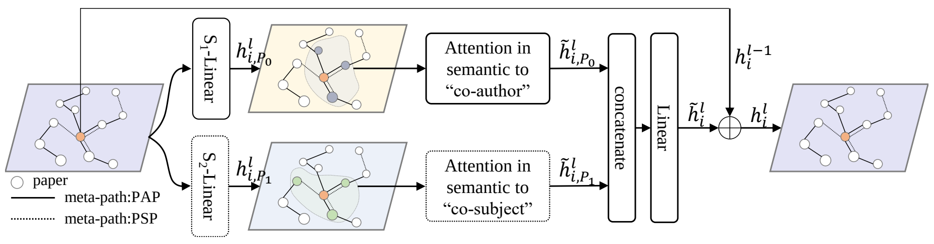


Fig. 3. The cross-layer semantic encoder.

4.3 Graph-based Contrastive Learning

On heterogeneous graphs, there are various implicit relationships between nodes, which usually contain rich semantics. With the proposed encoder, we successfully fuse the rich information among nodes. However, as the order of neighbors increases, the target node will fuse the information of many irrelevant nodes. To enhance the distinguishability of representations, we need to push away nodes that have little semantic relationship with the target node.

The customized meta-path is usually based on domain knowledge, which reflects the known high-order semantics of nodes under the specific task. For example, the meta-path PAP contains a semantic: two papers published by the same author are more similar in topic, which is more helpful to distinguish the category of the paper. Therefore, the number of meta-paths connected between two nodes reveals their similarity. That is, the higher the number is, the more similar they are. Given nodes i and j, if node j has C_n^{ij} instances to reach node i through meta-path $\mathcal{P}_n$, we define a function to measure the number of meta-path connections between nodes i and j:

$$C_{ij} = \sum_{n=1}^{m} C_n^{ij}. \tag{9}$$

Motivated by SimCSE [4], the key factor for us to obtain positive pairs is to independently sample x_i and x_i^+ by using the dropout mask. Specifically, we pass the same node to the pre-trained encoder twice. By exploiting the standard dropout twice, we can obtain two different representations as a positive pair without changing the graph topology. Then, we take nodes with C_{ij} less than T_{neg} as negative samples. The model learns the contrastive information by predicting the positive sample among the negative samples.

Before calculating the contrastive loss, we feed embeddings we get from the encoder into a multilayer perceptron to project them into the contrastive learning space:

$$Z_i = W_2 \cdot \sigma\left(W_1 \cdot h_i^L + b_1\right) + b_2, \tag{10}$$

where W_1 and W_2 are the transformation matrices, h_i^L is the output embedding of node i from encoder. For the contrastive learning task, we simply feed the same input to the encoder twice. In this way, we can obtain two embeddings with different dropout masks and we define them as $Z_i^{D_i}$ and $Z_i^{D_i^{'}}$, where D_i and $D_i^{'}$ are the different dropout masks. Finally, we calculate the contrastive loss as follows:

$$\mathcal{L}_i = -\log \frac{e^{\mathrm{sim}(Z_i^{D_i}, Z_i^{D_i^{'}})/\tau}}{\sum\limits_{j \in S_i \setminus \{i\}} e^{\mathrm{sim}(Z_i^{D_i}, Z_j^{D_j})/\tau} + \sum\limits_{j \in S_i} e^{\mathrm{sim}(Z_i^{D_i}, Z_j^{D_j^{'}})/\tau}}, \tag{11}$$

where S_i is the set of positive and negative samples of node i, $S_i = \{i, j | j \in V \, and \, C_{ij} < T_{neg}\}$, T_{neg} is the threshold for sampling negative samples, τ is a temperature parameter, $\mathrm{sim}(Z_1, Z_2)$ is the cosine similarity $\frac{Z_1^{\mathrm{T}} \cdot Z_2}{||Z_1|| \cdot ||Z_2||}$.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets We conduct experiments on four real-world networks. The detailed descriptions are summarized in Table 1.

- **ACM** [32]. ACM is extracted from KDD, SIGMOD, SIGCOMM, MobiCOMM, and VLDB. The target nodes are papers which are divided into three classes: Database, Wireless Communication, and Data Mining.

Table 1. The statistics of four datasets.

Dataset	Nodes	Edges	Meta-path
ACM	paper (P):4,019 author (A):7,167 subject (S):60	P-A:13,407 P-S:4,019	PAP PSP
DBLP	author (A):4,057 paper (P):14,328 conference (C):20 term (T):7,723	P-A:19,645 P-C:14,328 P-T:85,810	APA APCPA APTPA
Freebase	movie (M):3,492 actor (A):33,401 direct (D):2,502 writer (W):4,459	M-A:65,341 M-D:3,762 M-W:6,414	MAM MDM MWM
IMDB	movie (M):3,676 actor (A):4,353 direct (D):1,678	M-A:11,028 M-D:3,676	MAM MDM

- **DBLP** [3]. DBLP is extracted from the computer science bibliography website. The target nodes are authors which are divided into four classes: Database, Data Mining, Artificial Intelligence, and Information Retrieval.
- **Freebase** [12]. Freebase is a dataset about movies extracted from Freebase. The target nodes are movies which are divided into three classes: Action, Comedy and Drama.
- **IMDB** [32]. IMDB is a subset of dataset IMDB. The target nodes are movies which are divide into three classes: Action, Comedy, and Drama.

Baselines We compare the proposed DHG-CL with three categories of baselines, including: two unsupervised homogeneous methods (i.e., DeepWalk, DGI), four unsupervised heterogeneous methods (i.e., Mp2vec, HERec, DMGI, HeCo), and a semi-supervised heterogeneous method (i.e., HAN), to verify the effectiveness of DHG-CL.

- **DeepWalk** [18]. It performs random walk on homogeneous graphs and learns the representations of nodes through the skip-gram model.
- **DGI** [25]. It maximizes the mutual information between the graph-level summary and the local patches.
- **Mp2vec** [2]. It performs meta-path based random walk on heterogeneous graphs and learns the representations of nodes through the skip-gram model.
- **HERec** [21]. It utilizes a meta-path based random walk strategy to filter node sequences and applies DeepWalk to embed the heterogeneous graphs.
- **DMGI** [17]. It minimizes the disagreements among node embeddings and employs a universal discriminator to discriminate the graph-level summary and local patches.
- **HeCo** [28]. It adopts network schema and meta-path structure as two views to perform contrastive learning on heterogeneous graphs.
- **HAN** [27]. It adopts hierarchical attention mechanism to model both node-level and semantic-level importance.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_19

Implementation Details For all baselines, we use the settings in their original paper and modify a few parameters. Specifically, for random walk based methods like DeepWalk, Mp2vec and HERec, we set the walk length to 20, the window size to 5, the number of walks to 10 and the number of negative samples to 5. For Deepwalk, we ignore the heterogeneity of nodes and test its performance on the whole heterogeneous graph. For DGI, Mp2vec and HERec, we test all the meta-paths for them and report the best performance.

For the proposed DHG-CL, we adopt Adaptive Moment Estimation (Adam) [10] optimizer, set the dimension of node embeddings to 64, tune the learning rate from 0.0005 to 0.001, and tune the patience for early stopping from 20 to 50. For the cross-layer semantic encoder, we set the dimension of the semantic space to 64, and the number of layers to 2. For the contrast task, we tune dropout including feature dropout from 0.1 to 0.9, attention dropout from 0.1 to 0.9, temperature coefficient τ from 0.5 to 0.9, and negative threshold from 3 to 5. To reduce randomness, we perform all methods 10 times and report the average results.

Table 2. Comparison results (%$\pm\sigma$) for node classification.

Datasets	Metrics	Split	HAN	DeepWalk	DGI	Mp2vec	HERec	DMGI	HeCo	DHG-CL
ACM	Ma-F1	20	87.61±0.5	66.04±2.1	82.24±3.4	68.05±1.0	68.67±0.7	87.99±0.5	88.33±0.2	**89.64±0.3**
		40	87.68±0.6	69.70±1.2	84.01±2.6	66.30±0.4	66.64±0.6	88.23±0.4	87.28±0.4	**90.71±0.2**
		60	86.28±0.8	63.64±2.0	84.79±2.5	67.77±0.8	68.25±1.5	89.48±0.3	88.97±0.4	**90.88±0.1**
	Mi-F1	20	87.53±0.5	62.46±3.9	81.79±3.9	67.98±1.9	68.82±1.2	87.30±0.6	88.38±0.3	**89.28±0.4**
		40	87.89±0.6	71.87±2.8	83.66±3.1	67.81±0.3	68.87±2.1	87.78±0.4	87.02±0.4	**90.51±0.3**
		60	86.67±0.5	67.69±4.7	84.36±2.9	70.53±1.0	71.15±1.7	89.00±0.4	88.44±0.5	**90.55±0.3**
DBLP	Ma-F1	20	89.28±2.4	82.14±0.7	88.95±0.3	89.56±0.8	87.74±1.3	89.46±0.1	91.06±0.3	**92.24±0.7**
		40	89.86±1.9	85.01±0.4	87.98±0.4	90.71±0.6	89.24±0.9	89.44±0.1	89.90±0.5	**91.07±0.8**
		60	89.55±1.5	85.69±0.3	90.87±0.4	91.52±0.8	89.90±0.7	88.68±1.0	90.90±0.4	**91.96±0.5**
	Mi-F1	20	89.84±2.4	83.00±0.5	90.04±0.3	90.12±0.8	88.51±1.2	90.28±0.1	91.76±0.3	**92.74±0.7**
		40	90.24±1.9	85.26±0.4	88.74±0.3	91.12±0.6	89.81±0.9	90.11±0.1	90.22±0.6	**91.46±0.9**
		60	90.72±1.1	86.55±0.4	91.85±0.3	92.28±0.7	90.86±0.6	89.20±0.8	91.77±0.3	**92.71±0.6**
Freebase	Ma-F1	20	57.75±0.9	50.15±2.5	55.76±0.9	52.26±1.9	55.48±1.5	54.56±0.5	59.33±0.9	**61.74±0.8**
		40	55.51±1.2	50.87±2.1	54.33±1.4	53.59±1.2	57.31±1.8	51.93±0.8	61.15±0.5	**61.28±1.1**
		60	56.13±0.9	51.50±1.3	53.48±0.7	51.42±1.8	53.69±1.5	50.21±1.0	**60.93±0.5**	60.62±1.4
	Mi-F1	20	61.22±1.7	59.67±1.7	63.85±2.2	54.36±2.1	57.89±1.7	61.60±2.1	61.95±1.1	**66.56±0.5**
		40	57.60±1.5	61.94±1.7	61.76±3.6	55.86±1.5	59.84±1.9	62.86±3.2	63.88±0.8	**65.91±0.9**
		60	58.94±1.2	61.39±1.4	64.47±2.1	53.78±1.9	56.34±1.6	58.85±2.9	63.94±0.8	**66.40±1.2**
IMDB	Ma-F1	20	39.38±1.8	35.94±0.5	36.20±0.6	36.47±1.5	34.38±1.4	36.86±0.9	39.98±0.9	**45.13±1.0**
		40	40.30±1.9	36.01±1.0	39.56±0.4	36.88±1.3	36.89±1.5	37.93±0.7	40.76±0.8	**41.49±0.7**
		60	**44.79±1.3**	33.63±1.0	40.82±0.5	37.05±1.9	35.25±1.1	37.22±1.2	41.79±0.8	41.90±0.9
	Mi-F1	20	39.98±1.4	35.38±1.1	36.39±0.6	36.73±1.4	35.93±1.7	38.07±0.7	40.14±1.0	**44.74±1.0**
		40	40.50±1.8	36.71±1.2	39.49±0.5	36.97±1.3	37.49±1.3	37.92±0.7	40.96±0.8	**41.75±1.1**
		60	**45.27±1.3**	35.32±0.5	40.96±0.5	37.08±1.9	37.15±1.1	37.87±1.2	41.87±0.8	43.80±1.3

5.2 Node Classification

After obtaining the learned embeddings of nodes by pre-training model, we feed them into a linear classifier to predict their labels. Following [28], we choose 20, 40, 60 labeled nodes for each class as the training set, 1000 nodes as the validation set, and

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_19

1000 nodes as the test set. Then, we use common evaluation metrics: Macro-F1, Micro-F1 and report the test performance when it performs best in the validation set. The comparison results are shown in Table 2.

From Table 2, we can see that DHG-CL achieves the best performance in most cases. Comparing DeepWalk and Mp2vec, DGI and DMGI, we can see that unsupervised methods for heterogeneous graphs generally outperform unsupervised methods for homogeneous graphs, which demonstrates the benefits of graph heterogeneity. We can also see that DHG-CL outperforms HeCo and DMGI, indicating that encoding deep semantic information and introducing data augmentation technology have brought certain performance improvements. Moreover, DHG-CL outperform the semi-supervised method HAN, demonstrating the potential capabilities of self-supervised HGNNs.

5.3 Node Clustering

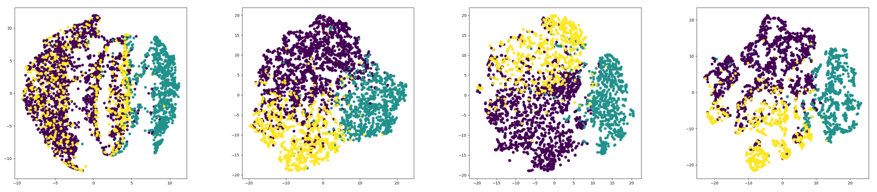
To more comprehensively evaluate the embeddings of nodes learned by DHG-CL, we also conduct a node clustering task. In this task, we utilize the K-means algorithm to perform clustering of nodes and set the number of clusters as the number of classes. Then, we adopt normalized mutual information (NMI) and adjusted rand index (ARI) to assess the quality of the clustering results. Since initializing the centroids will affect the performance of K-means, we repeat the process for 10 times and report the average, as shown in Table 3. Similarly, the proposed DHG-CL significantly outperforms other methods in most cases, which further proves the effectiveness of DHG-CL. Since HAN is trained with label guidance, we do not compare with it.

Table 3. Comparison results for node clustering.

Methods	ACM		DBLP		Freebase		IMDB	
	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
DeepWalk	36.59	28.77	66.17	71.25	13.47	14.92	0.35	0.08
DGI	50.13	45.27	69.88	75.63	16.26	17.25	0.09	0.08
Mp2vec	37.75	30.21	73.74	78.81	16.50	17.24	0.47	0.44
HERec	35.97	28.90	72.19	77.75	16.30	17.23	0.44	1.04
DMGI	50.05	40.38	67.05	71.72	16.10	16.29	0.73	**1.17**
HeCo	57.92	56.45	71.11	76.64	15.23	16.88	0.81	0.84
DHG-CL	**60.57**	**59.07**	**77.37**	**81.57**	**16.74**	**17.93**	**0.88**	0.96

5.4 Visualization

For a more intuitive evaluation, we conduct the task of visualization on ACM dataset. We use t-SNE [15] algorithm to visualize embeddings and choose different colors for different labels.



(a) DeepWalk (b) DGI (c) HeCo (d) DHG-CL

Fig. 4. Visualization paper embedding on ACM. Each point indicates one paper and its color indicates the topic.

From Fig. 4, we can observe that Deepwalk and DGI designed for homogeneous graphs do not perform well and have blurry boundaries, which may lead to confusion among nodes. In contrast, the heterogeneous graph methods perform much better than the above homogeneous graph methods. It demonstrates that graph heterogeneity contains rich information for node classification, node clustering tasks, etc. For heterogeneous graph method, DHG-CL has higher intra-class similarity and a clearer boundary than HeCo to distinguish nodes.

5.5 Variant Analysis

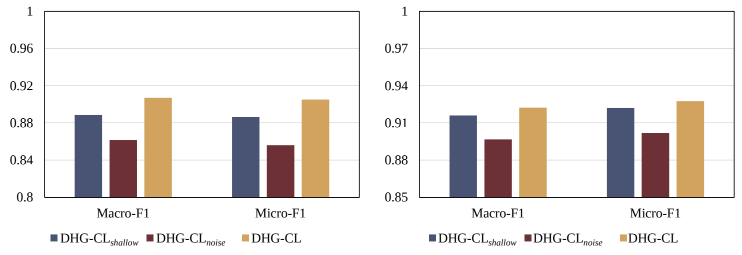
In order to verify the effectiveness of each part of DHG-CL, we design two variants of DHG-CL as follows:

- DHG-CL$_{shallow}$. To verify the effectiveness of cross-layer semantic encoder, we remove the cross-layer message passing mechanism in the encoder and set the number of layers to 1. Therefore, the DHG-CL$_{shallow}$ can only preserve shallow graph information and cannot capture higher-order semantics.
- DHG-CL$_{noise}$. To verify the effectiveness of contrasting learning strategy in DHG-CL, we introduce other common data augmentation techniques. In specific, we maintain our negative sampling strategy, and then we sample positive sample from the new graph constructed by removing a portion of the edges [31].

We use the same parameters for them and conduct comparison between them and DHG-CL on ACM and DBLP. The comparison results of 20 labeled nodes per class are shown in Fig. 5. Obviously, DHG-CL significantly outperforms their variants. DHG-CL performs better than DHG-CL$_{shallow}$, indicating that it is necessary to preserve both shallow and implicit higher-order semantics and some helpful information is also in higher-order semantics. DHG-CL performs better than DHG-CL$_{noise}$, which proves that dropout mask as a novel method is more robust than data augmentation technique which modifies (e.g. add, remove nodes/edges) the inherent structure of the graph.

5.6 Parameter Analysis

In this section, we investigate the sensitivity of parameters and report the Macro-F1 values of node classification on ACM dataset with different parameters.

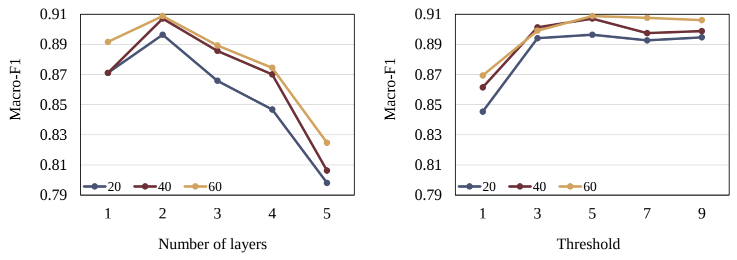


(a) ACM (b) DBLP

Fig. 5. The comparison of DHG-CL and its variants.

Number of layers L. L describes the depth of the heterogeneous graph, which determines the high-order proximity of the nodes. We vary the value of it and report the results in Fig. 6(a). We can see that with the increase of the number of layers, the performance of DHG-CL firstly goes up and then decreases drastically when L is higher than 2. This is because as the number of layers increases, the unhelpful information will bring more negative gains.

Threshold T_{neg}. T_{neg} describes the minimum number of meta-paths between nodes, which affects the number of negative samples. We explore the performance of DHG-CL with various thresholds. From Fig.6(b), we can see that with the growth of the threshold, the performance increases first and then starts to decrease when we set $T_{neg} = 5$. The reason is that the larger the threshold is, the more negative samples are selected, which will include some semantically similar samples as negative samples.



(a) Number of layers L (b) Threshold T_{neg}

Fig. 6. Parameter sensitivity of DHG-CL w.r.t. number of layers L and threshold T_{neg}.

6 CONCLUSION AND FUTURE WORK

In this work, we propose a novel self-supervised deep heterogeneous graph neural networks with contrastive learning, named DHG-CL. DHG-CL employs a cross-layer semantic encoder and the graph-based contrastive learning to preserve the information of implicitly valuable neighbors and further enhance the distinguishability of representations. In cross-layer semantic encoder, we design a semantic-aware attention mechanism and a cross-layer message passing mechanism to capture deep implicit semantics. In graph-based contrastive learning, we innovatively introduce dropout mask and semantic based negative sampling strategy to further obtain discriminative representation. Experimental results on four real-world heterogeneous graphs demonstrate the effectiveness of DHG-CL.

Acknowledgements This work is supported by Key Research and Development Program Projects of Xinjiang (No. 2022B03010-2), Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDC02030400).

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
2. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 135–144 (2017)
3. Fu, X., Zhang, J., Meng, Z., King, I.: Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of The Web Conference 2020. pp. 2331–2341 (2020)
4. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
5. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
6. Hong, H., Guo, H., Lin, Y., Yang, X., Li, Z., Ye, J.: An attention-based graph neural network for heterogeneous structural learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4132–4139 (2020)
7. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of The Web Conference 2020. pp. 2704–2710 (2020)
8. Ji, H., Wang, X., Shi, C., Wang, B., Yu, P.: Heterogeneous graph propagation network. IEEE Transactions on Knowledge and Data Engineering (2021)
9. Jiang, X., Lu, Y., Fang, Y., Shi, C.: Contrastive pre-training of gnns on heterogeneous graphs. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 803–812 (2021)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Li, X., Ding, D., Kao, B., Sun, Y., Mamoulis, N.: Leveraging meta-path contexts for classification in heterogeneous information networks. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). pp. 912–923. IEEE (2021)
13. Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering (2021)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_19

14. Liu, Y., Pan, S., Jin, M., Zhou, C., Xia, F., Yu, P.S.: Graph self-supervised learning: A survey. arXiv preprint arXiv:2103.00111 (2021)
15. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
16. Meng, Y., Xiong, C., Bajaj, P., Bennett, P., Han, J., Song, X., et al.: Coco-lm: Correcting and contrasting text sequences for language model pretraining. Advances in Neural Information Processing Systems **34** (2021)
17. Park, C., Kim, D., Han, J., Yu, H.: Unsupervised attributed multiplex network embedding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5371–5378 (2020)
18. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
19. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
20. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), `https://proceedings.neurips.cc/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf`
21. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering **31**(2), 357–370 (2018)
22. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the VLDB Endowment **4**(11), 992–1003 (2011)
23. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: European conference on computer vision. pp. 776–794. Springer (2020)
24. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
25. Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. ICLR (Poster) **2**(3), 4 (2019)
26. Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., Yu, P.S.: A survey on heterogeneous graph embedding: methods, techniques, applications and sources. arXiv preprint arXiv:2011.14867 (2020)
27. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The world wide web conference. pp. 2022–2032 (2019)
28. Wang, X., Liu, N., Han, H., Shi, C.: Self-supervised heterogeneous graph neural network with co-contrastive learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1726–1736 (2021)
29. Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F., Ma, H.: Clear: Contrastive learning for sentence representation. arXiv preprint arXiv:2012.15466 (2020)
30. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems **32**(1), 4–24 (2020)
31. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems **33**, 5812–5823 (2020)
32. Zhao, J., Wang, X., Shi, C., Liu, Z., Ye, Y.: Network schema preserving heterogeneous information network embedding. In: International Joint Conference on Artificial Intelligence (IJCAI) (2020)
33. Zitnik, M., Sosič, R., Leskovec, J.: Prioritizing network communities. Nature communications **9**(1), 1–9 (2018)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_19