symmetry

Article

Self-Supervised Graph Representation Learning via Information Bottleneck

Junhua Gu [1,2], Zichen Zheng [1], Wenmiao Zhou [1], Yajuan Zhang [1], Zhengjun Lu [3,*] and Liang Yang [1,2]

1 School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China; jhgu@hebut.edu.cn (J.G.); 201932805009@stu.hebut.edu.cn (Z.Z.); 202032803129@stu.hebut.edu.cn (W.Z.); zhangyajuan@scse.hebut.edu.cn (Y.Z.); 2018046@hebut.edu.cn (L.Y.)
2 Hebei Province Key Laboratory of Big Data Calculation, Tianjin 300401, China
3 Defense Engineering Institute AMS, PLA, Beijing 100036, China
* Correspondence: 2010042@hebut.edu.cn or luzj@sohu.com

Abstract: Graph representation learning has become a mainstream method for processing network structured data, and most graph representation learning methods rely heavily on labeling information for downstream tasks. Since labeled information is rare in the real world, adopting self-supervised learning to solve the graph neural network problem is a significant challenge. Currently, existing graph neural network approaches attempt to maximize mutual information for self-supervised learning, which leads to a large amount of redundant information in the graph representation and thus affects the performance of downstream tasks. Therefore, the self-supervised graph information bottleneck (SGIB) proposed in this paper uses the symmetry and asymmetry of graphs to establish comparative learning and introduces the information bottleneck theory as a loss training model. This model extracts the common features of both views and the independent features of each view by maximizing the mutual information estimation between the local high-level representation of one view and the global summary vector of the other view. It also removes redundant information not relevant to the target task by minimizing the mutual information between the local high-level representations of the two views. Based on the extensive experimental results of three public datasets and two large-scale datasets, it has been shown that the SGIB model can learn higher quality node representations and that several classical network analysis experiments such as node classification and node clustering can be improved compared to existing models in an unsupervised environment. In addition, an in-depth network experiment is designed for in-depth analysis, and the results show that the SGIB model can also alleviate the over-smoothing problem to a certain extent. Therefore, we can infer from different network analysis experiments that it would be an effective improvement of the performance of downstream tasks through introducing information bottleneck theory to remove redundant information.

Keywords: graph neural network; graph representation learning; self-supervised learning; information bottleneck; mutual information estimation; node classification; node clustering

check for updates

Citation: Gu, J.; Zheng, Z.; Zhou, W.; Zhang, Y.; Lu, Z.; Yang, L. Self-Supervised Graph Representation Learning via Information Bottleneck. *Symmetry* **2022**, *14*, 657. https://doi.org/10.3390/sym14040657

Academic Editor: Juan Alberto Rodríguez Velázquez

Received: 27 February 2022
Accepted: 22 March 2022
Published: 24 March 2022

1. Introduction

A graph is a kind of data structure that models a set of nodes and the edges between them, where the node represents a specific class of entity and the edge represents the connection between them. As many learning tasks require processing information that contains large numbers of objects and their relationships, graph-structured data have received increasing attention. Graph-structured data can be utilized in a variety of applications in the real world, including analysis of social networks, modeling of physical systems, learning of molecular fingerprints, prediction of protein interfaces, and classification of diseases. To better study graph-structured data, a model is needed to learn vector representations from the input graphs.

Driven by recent advances in deep learning [1], the paradigm of graph learning has shifted from structural pattern discovery to graph representation learning [2,3] in the past few years. Specifically, graph representation learning converts graph vertices, edges, or subgraphs into low-dimensional embeddings [4–6], thus preserving the important structural information of the graph. Studying graph representations can be used for many downstream tasks, such as node classification [7,8], graph classification [9,10], link prediction [11,12], etc. Most of the existing graph neural network models are built in a supervised manner, which requires a large amount of labeled input for training. However, labeling graph data turns out to be impractically time-/resources-consuming in many applications, for example, with the possibly prolonged and expensive issues that exist in in vivo animal experiments required for the pharmacological effects produced by graph-represented drug molecules. Therefore, recent research has focused on developing self-supervised learning in graph neural networks that require only limited or even no labeling.

However, the current graph neural network still has some obstacles to overcome. One of the most intractable problems is that the features of neighboring nodes may contain redundant information, which may negatively affect the prediction of the current node. In addition, the over-reliance on edge messages during training makes the trained model more vulnerable to noise and adversarial attacks on the input graph structure. There is limited research on how to solve the problem of containing redundant information in the input graph. In this paper, we believe that the key to clearing out redundant information is to employ the information bottleneck theory. The information bottleneck theory requires that the encoder contains information as little as possible while completing the downstream task. The encoder learned from the information bottleneck is expected to be more robust for redundant information removed.

A self-supervised graph representation learning via information bottleneck (SGIB) is proposed in this paper. The model uses the symmetry and asymmetry of the graph to introduce comparative learning and employs the information bottleneck as a loss training model to achieve the purpose of self-supervised learning. For the nodes in the network, all of their feature information and topological information do not necessarily play a positive role for the downstream tasks, and some pieces of information even may have a negative impact. Ideally, the encoded node vector contains all the information necessary for the downstream task, so in the process of training the model, the encoded node vector is required to contain as little redundant information as possible.

Specifically, the SGIB model trains the encoder by maximizing the mutual information between the node-level representation of one view and the graph-level representation of the other view, while minimizing the mutual information between the node-level representations of the two views. Such a strategy enables the optimized model to process the downstream task by extracting the common features of both views and independent features of each view, while removing redundant information that is irrelevant to the target task. Therefore, the graph representations learned by SGIB will be more robust and distinguishable. In summary, the main contributions of this study are summarized as follows:

1. Introducing information bottleneck into contrast learning, thus achieving the purpose of self-supervised learning.
2. Considering the neglect of redundant data in past studies, this paper proposes the use of the information bottleneck as the objective function of the optimization model. The information bottleneck was applied by maximizing the mutual information between one view node level representation and another view graph level representation, while minimizing the mutual information between two view node level representations.
3. A variety of network analysis experiments, including node classification and node clustering, were conducted on three public datasets as well as two large-scale datasets. Numerous experiments show that the method outperforms the best existing methods. In addition, an in-depth analysis of the model is conducted, and the experimental results show that SGIB can alleviate the over-smoothing problem to a certain extent.

2. Related Work

In recent years, many researchers have investigated the problem of how to build unsupervised learning through consistency. The most representative solution is the deep graph infomax (DGI) [13], which first embeds an input graph, then summarizes the input graph into a vector by a readout function, and finally maximizes the mutual information between the vector representation of the input graph nodes and the vector representation at the graph level. Following DGI, graph representation learning via graphical mutual information maximization (GMI) [14] proposes a node-level objective comparison, which maximizes the node representation with the input attribute features while maximizing the mutual information between the node representation and the topological information. Contrastive multi-view representation learning on graphs (MVGRL) [15] proposes a graph diffusion method to expand another input graph, constructs subgraphs by uniform sampling, and then compares the node representation with the global embedding of the two views to obtain the vector representation of the graph. Deep graph contrastive representation learning (GRACE) [16] extends the idea of maximizing graph mutual information across nodes and subgraphs, generates two associated graphs by random destruction, and uses the contrast loss function to maximize the node representation in the two views. Graph contrastive learning with adaptive augmentation (GCA) [17] learns the graph representation by creating new data with reasonable transformations and maximizing the mutual information under different expansions using feature consistency. Heterogeneous graph information bottleneck (HGIB) [18] utilizes information bottleneck to implement the consensus hypothesis between different meta-paths in an unsupervised manner. Subgraph information bottleneck (SIB) [19] recognize subgraphs by removing redundancy and noise from information bottleneck in the field of graph classification. Variational graph information bottleneck (VGIB) [20] introduces the noise injection method to recognize subgraphs based on graph information bottleneck.

Common among methods that do not apply information bottleneck is the reliance on maximizing mutual information between whole and part, part and part, and different views originating from the same network to perform unsupervised training. However, none of the above works using a consistent problem-solving strategy can handle the redundant data contained in the input data. The method of information bottleneck theory cannot solve the node-level task of the homogeneous graph. To address these problems, this paper proposes a self-supervised graph representation learning method based on an information bottleneck, which can extract both the common features of two views and independent features of each view, while removing redundant information irrelevant to the target task, resulting in a node representation of higher quality.

3. Preliminaries

3.1. Homogeneous Graph

Given an undirected graph $G = (V, E)$, V represents the set of nodes and E represents the set of edges. Information about the relationship between nodes is provided in the form of an adjacency matrix, $A \in \mathbb{R}^{N\times N}$. $X = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_N\}$ denotes a set of node features of the input data, N represents the number of nodes in the graph, and $\vec{x}_i \in \mathbb{R}^F$, represents the features of node i. In all experiments of this paper, it is assumed that the edges between nodes are unweighted; i.e., if there are edges $i \to j$ in the graph, then $A_{ij} = 1$, otherwise $A_{ij} = 0$.

3.2. Mutual Information Estimation

Mutual information (MI) is a Shannon entropy-based measure of the degree of interdependence among random variables. Unlike the common similarity measures, mutual information can capture the nonlinear correlation between variables, so it can be considered as a measure of the true dependence between variables. For two random variables X and Y, the mutual information between them is shown as follows:

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{1}$$

$H(X)$ is the entropy of information for X, while $H(X \mid Y)$ means under the condition of a known random variable Y, the conditional entropy of a random variable X. From the perspective of probability, mutual information is derived from the joint probability distribution $p(x,y)$ and the marginal probability distribution $p(x)$,$p(y)$ of random variables X and Y.

However, since the mutual information can be computed only in the discrete case and in the continuous case with a finite number of known distributions, its lower bound is usually estimated using some known algorithms. A neural network-based mutual information estimation model, mutual information neural estimator (MINE) [21], uses a Donsker-Varadhan estimator based on KL divergence to derive a lower bound on the mutual information, where the function T is usually the neural network:

$$I(X;Y) \geq \sup_{T \in \mathcal{F}} \mathbb{E}_{P_{XY}}[T] - \log\left(\mathbb{E}_{P_X \otimes P_Y}\left[e^T\right]\right). \tag{2}$$

In this paper, the mutual information estimation is applied to the local vector representation and global vector representation of nodes to achieve the self-supervised learning of the network.

3.3. Information Bottleneck

The core principle of information bottleneck (IB) theory is that the optimal graph representation should contain the minimum and sufficient information to complete the downstream prediction task [22]. In other words, the amount of information about the task does not change due to the encoding process, as shown in Equation (3):

$$I(x;y) = I(h;y), \tag{3}$$

where x is the input data, h is the obtained graph representation, and y is the label. In order to make the graph representation robust, the information bottleneck principle [23] tries to discard all information from the input as the input information is not helpful for the task of predicting the labels. The information bottleneck requires the graph representation to provide the maximum amount of information about the target to make the prediction accurate, while also preventing the graph representation from obtaining redundant information from the data that is not relevant to the prediction. Therefore, the information bottleneck [24] minimizes the mutual information between the data x and its representation h, while maximizing the mutual information between the representation h and the label y. The objective function is shown in Equation (4):

$$R_{IB}(\theta) = I(y;h) - \beta I_\theta(x;h), \tag{4}$$

where θ is denoted as the parameter of the neural network and β is the hyperparameter of the control weights. The second part can be split into two parts according to the chain rule of mutual information, as shown in Equation (5):

$$I(x;h) = I(x;h|y) + I(y;h), \tag{5}$$

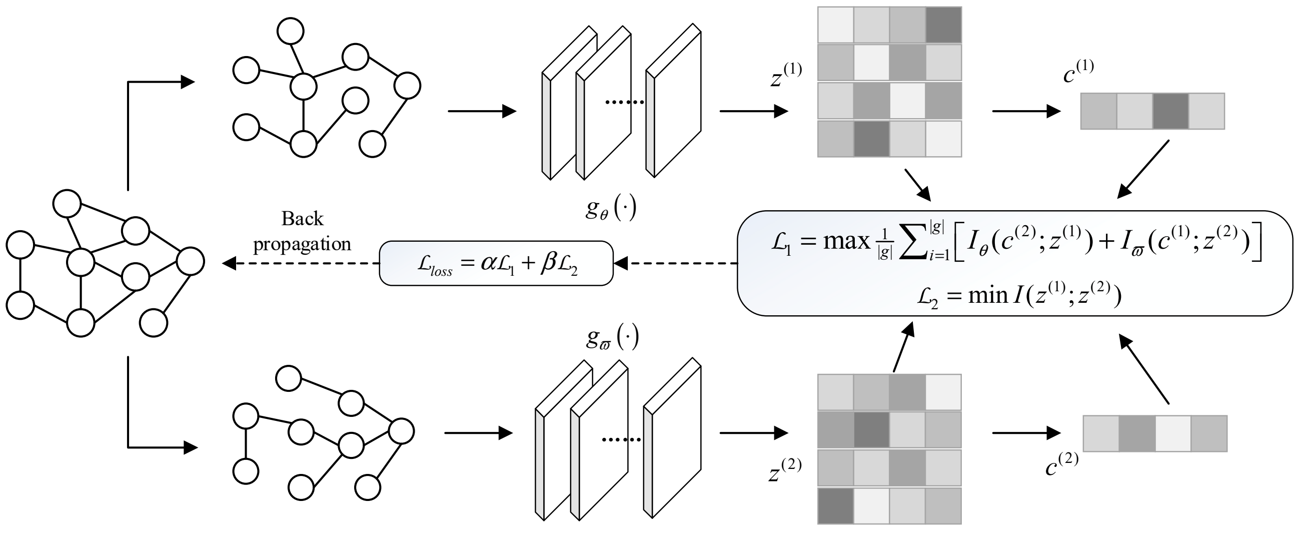
where the second term does not vary according to h, because the representation h contains the minimum and sufficient information that can accomplish the target task, as shown by Equation (3). The first term represents the information that is not useful for the target task, i.e., redundant information. Therefore, minimizing mutual information $I(x;h)$ is equivalent to minimizing redundant information $I(x;h\,|\,y)$ [25], and rewriting Equation (4) gives the formula for the information bottleneck, as shown in Equation (6):

$$R_{IB}(\theta) = I(y;h) - \beta I(x;h|y). \tag{6}$$

However, this paper focuses on training neural networks that can remove redundant information when the labels of downstream tasks are not available. This part will be elaborated on in Section 4.

4. Self-Supervised Graph Representation Learning via Information Bottleneck

SGIB is a self-supervised graph representation learning model based on information bottleneck, and the model framework is shown in Figure 1. The SGIB model is divided into three modules. Firstly, two random Dropedge [26] operations are performed to obtain two subgraphs different from the original image, and contrast learning is used to extend the information bottleneck to self-supervised learning. Secondly, to extract the common features of the two subgraphs and independent features of each view, SGIB compares the mutual information between the graphical encoding from the first-order neighbor and the graph-level representation of the other subgraph, respectively. In addition, to remove redundant information irrelevant to the target task, SGIB also compares the mutual information between the first-order graphical encodings of the two subgraphs. Finally, the information bottleneck is used as the loss function of the model to complete the training and optimization of the objective function.



$z^{(1)}$
$c^{(1)}$
$g_\theta(\cdot)$
Back propagation
$\mathcal{L}_{loss} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2$
$\mathcal{L}_1 = \max \frac{1}{|g|}\sum_{i=1}^{|g|}\left[I_\theta(c^{(2)};z^{(1)}) + I_\varpi(c^{(1)};z^{(2)})\right]$
$\mathcal{L}_2 = \min I(z^{(1)};z^{(2)})$
$g_\varpi(\cdot)$
$z^{(2)}$
$c^{(2)}$

Figure 1. Overall structure of SGIB.

4.1. Self-Supervised Information Bottleneck

In order to train a neural network that can remove redundant information and obtain high-quality node representations without downstream task labeling as agnostic, we use the contrast learning method in self-supervised learning. Contrast learning is used to learn the feature representation by comparing the data with positive and negative samples in the feature space. In order to obtain information bottleneck without labeling information, we apply the contrast learning method so that the two perspectives are "labeled" with each other:

$$R_{IB}(\theta) = I_\theta(h^{(2)};h^{(1)}) - \beta I_\theta(x;h^{(1)}), \tag{7}$$

where $h^{(1)}$ and $h^{(2)}$ are vector representations of the generalized perspective one and perspective two, respectively. Maximizing the RIB requires maximizing the first term while minimizing the second term. The second term can be split into two representations according to the mutual information chain rule:

$$I(x;h^{(1)}) = I(x;h^{(1)}|h^{(2)}) + I(h^{(2)};h^{(1)}). \tag{8}$$

The first term in Equation (7) represents the specific information from the input feature x in the vector representation $h^{(1)}$ of viewpoint one, and this term should be minimized, while maximizing the information of the vector representation of viewpoint one and viewpoint two in the second term. The joint action of these two steps reaches the maximum, so rewrite Equation (6) to obtain the generalized self-supervised information bottleneck:

$$R_{IB}(\theta) = I_\theta(h^{(2)};h^{(1)}) - \beta I_\theta(x;h^{(1)}|h^{(2)}). \tag{9}$$

The difficulty of contrast learning is constructing positive and negative example samples. In this study, Dropedge is used to expand two different subgraphs so that they are each other's positive samples, and the graph obtained after random row perturbation of the original adjacency matrix is used as a negative sample. The Dropedge algorithm is a process that expands the original data. In order to obtain two different output graphs G_1 and G_2, the edges in the original graph are randomly discarded, which is carried out by forcing some of the non-zero elements in the adjacency matrix A to zero at each training cycle with drop rates of p_1 and p_2, respectively (p_1 and p_2 can be equal). In this paper, the adjacency matrix after Dropedge is denoted as A_{drop}, and then, the adjacency matrices corresponding to graphs G_1 and G_2 are denoted as A_{drop1} and A_{drop2}, which are related to the initial adjacency matrix A as follows:

$$A_{\text{drop1}} = A - A'_1, \tag{10}$$

$$A_{\text{drop2}} = A - A'_2, \tag{11}$$

where A'_i is a sparse matrix with a size equal to the initial adjacency matrix. Each epoch in the training process performs an independent Dropedge for data augmentation, so that a different A_{drop} is generated each time, and the data change can be avoided as much as possible. In addition, since the over-smoothing phenomenon is severe in unsupervised learning, the SGIB model uses Dropedge to expand the original data, which can better alleviate the problem of over-smoothing and overfitting in deep networks.

4.2. Encoders

In this paper, the commonly used graph convolutional neural network (GCN) is chosen as the base encoder. As shown in the top and bottom views of Figure 1, we set up separate encoders $g_\theta(\bullet)$ and $g_\omega(\bullet)$ for each subgraph. The two subgraphs are propagated and mapped by one layer of GCN to obtain their respective node representations as follows:

$$Z^{(1)} = \sigma\left(\left(\widetilde{D}^{(1)}\right)^{-\frac{1}{2}}\widetilde{A}^{(1)}\left(\widetilde{D}^{(1)}\right)^{-\frac{1}{2}}X\theta\right), \tag{12}$$

$$Z^{(2)} = \sigma\left(\left(\widetilde{D}^{(2)}\right)^{-\frac{1}{2}}\widetilde{A}^{(2)}\left(\widetilde{D}^{(2)}\right)^{-\frac{1}{2}}X\omega\right) \tag{13}$$

where $\widetilde{A} = A + I$ represents the adjacency matrix with self-loop and $\widetilde{D}$ denotes the degree matrix $\widetilde{A}$ with the diagonal elements as the degrees of the nodes. θ and ω are the two different learnable parameters corresponding to encoders $g_\theta(\bullet)$ and $g_\omega(\bullet)$, respectively. σ is the nonlinear rectification function PRelu or Relu. The node-level vector representations $z^{(1)}$ and $z^{(2)}$ are clustered into graph-level vector representations employing a readout function $\varphi(\cdot) : \mathbb{R}^{n\times d} \to \mathbb{R}^d$. Finally, the aggregated graph-level vector representations are mapped between 0 and 1 to obtain the final graph-level vector representations $c^{(1)}$ and $c^{(2)}$ that can be trained and optimized.

Since the SGIB model uses the contrast learning method in self-supervised learning, it requires the selection of positive and negative example samples. We choose the graph-level vector representation $c^{(2)}$ as the positive example sample of the node-level vector representation $z^{(1)}$, and similarly choose the graph-level vector representation $c^{(1)}$ as the positive example sample of the node-level vector representation $z^{(2)}$. The negative example sample is the graph after transforming the original graph node positions. Specifically, a random row transformation of the original graph adjacency matrix is denoted as A_{shuf}. The perturbed adjacency matrix A_{shuf} is encoded by the neural network $g_\theta(\bullet)$ to obtain the representation $z'^{(1)}$ as the negative sample of the representation $z^{(1)}$ (the negative sample of viewpoint two is obtained after encoding by the neural network $g_\omega(\bullet)$). Finally, $z^{(1)} + z^{(2)}$ is returned to the downstream task.

4.3. Training and Optimization

The essence of the information bottleneck is to retain the valuable information to the prediction label while discarding the information redundant to the prediction label, and the information bottleneck in the case of multiple inputs is shown in Figure 2. A and B stand for two inputs, Y stands for the target task (label), 1 stands for the independent features required for the target task in input A but not in input B, 2 stands for the common features required for the target task in input A and B, 3 stands for the independent features required for the target task in input B but not in input A, and 4stands for the information related to the target task in neither input A nor B, i.e., redundant information.

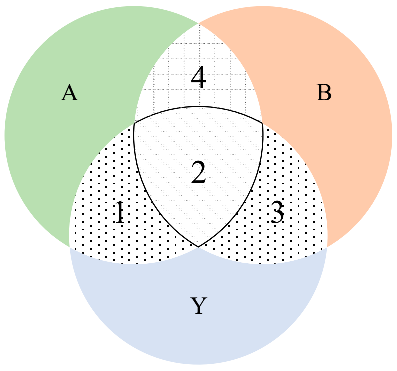


Figure 2. Information bottleneck with multi-input. (A) Input data for view one; (B) Input data for view two; (Y) The target tasks or labels.

In order to train the encoder end-to-end and learn node representations for agnostic downstream tasks, we use mutual information to measure the dependencies between these three, following the theory of information bottleneck in the multi-input case. The encoder is trained by maximizing the common and independent features of the two subgraphs (i.e., parts 1, 2, and 3) while minimizing the redundant information of the two subgraphs that is irrelevant to the target task (i.e., part 4). For the two input subgraphs, the information bottleneck formula can be reformulated as follows:

$$R_{IB}^{(1)}(\theta) = I_\theta\left(z^{(1)};c^{(2)}\right) - \beta I_\theta\left(z^{(2)};z^{(1)} \middle| \, c^{(2)}\right), \tag{14}$$

$$R_{IB}^{(2)}(\omega) = I_\omega\left(z^{(2)};c^{(1)}\right) - \beta_2 I_\omega\left(z^{(1)};z^{(2)} \middle| \, c^{(1)}\right) \tag{15}$$

θ and ω are the learnable parameters in the two GCNs, respectively. The loss function of the SGIB model is expressed as the average of $R^{(1)}$ and $R^{(2)}$, as shown in Equation (16), β_1 and β_2 are two hyperparameters:

$$L_{loss}(\theta,\omega,\beta_1,\beta_2) = \frac{I_\theta\left(z^{(1)};c^{(2)}\right)+I_\omega\left(z^{(2)};c^{(1)}\right)}{2} - \frac{\beta_1 I_\theta\left(z^{(2)};z^{(1)}\middle|c^{(2)}\right)+\beta_2 I_\omega\left(z^{(1)};z^{(2)}\middle|c^{(1)}\right)}{2}, \tag{16}$$

where $I_\theta(z^{(1)}; c^{(2)})$ can be transformed as follows to obtain its lower bound:

$$\begin{aligned} I_\theta\left(z^{(1)}; c^{(2)}\right) &= I_{\theta\omega}\left(c^{(2)}; c^{(1)} z^{(1)}\right) - I_{\theta\omega}\left(c^{(2)}; c^{(1)} \middle| z^{(1)}\right) \\ &= I_{\theta\omega}\left(c^{(2)}; c^{(1)} z^{(1)}\right) \\ &= I_{\theta\omega}\left(c^{(2)}; c^{(1)}\right) + I_{\theta\omega}\left(c^{(2)}; z^{(1)} \middle| c^{(1)}\right) \\ &\geq I_{\theta\omega}\left(c^{(2)}; c^{(1)}\right). \end{aligned} \tag{17}$$

When the graph representations c1 and c2 do not contain redundant information, i.e., $I_{\theta\omega}(c^{(2)};z^{(1)} \mid c^{(1)}) = 0$, the above inequality equation holds. Similarly, the lower bound of $I_\omega(z^{(2)};c^{(1)})$ can be deduced:

$$I_\omega(z^{(2)}; c^{(1)}) \geq I_{\theta\omega}(c^{(1)}; c^{(2)}). \tag{18}$$

The second term in Equation (16), where $I_\theta(z^{(2)};z^{(1)} \mid c^{(2)})$ can be transformed to obtain the upper bound, is as follows:

$$I_\theta(z^{(2)}; z^{(1)} | c^{(2)}) = I_{\theta\omega}(z^{(2)}; z^{(1)}) - I_{\theta\omega}(z^{(2)}; z^{(1)}; c^{(2)}) \leq I_{\theta\omega}\left(z^{(2)}; z^{(1)}\right). \tag{19}$$

Similarly, the upper bound of $I_\omega(z^{(1)};z^{(2)} \mid c^{(1)})$ can be deduced:

$$I_\theta(z^{(1)}; z^{(2)} | c^{(1)}) \leq I_{\theta\omega}(z^{(1)}; z^{(2)}). \tag{20}$$

In summary, the loss bounds of the two parts are obtained. The loss function of the model is obtained after adding the balance of weight parameters, as shown in Equation (21):

$$L_{loss}(\theta, \omega, \alpha, \beta) \geq I_{\theta\omega}(c^{(1)}; c^{(2)}) + \beta I_{\theta\omega}(z^{(1)}; z^{(2)}). \tag{21}$$

From Section 3.2, it is clear that the mutual information cannot be calculated precisely, so its lower bound is usually estimated using some known algorithms. In this paper, the goal of optimization is not to obtain a specific value but to maximize the mutual information, so there are other choices of non-KL divergence, such as the Jensen-Shannon mutual information estimator (JSD) [27] and the noise-contrast estimator (infoNCE) [28]. We chose the JSD estimator because the noise contrast estimator is more sensitive to the number of negative samples. Specifically, the effect of infoNCE decreases as the number of negative samples decreases. The JSD equation is as follows:

$$\hat{I}_{\varpi}^{(JSD)}(h_i; c) = \mathbb{E}_{\mathbb{P}}[-sp(-D_{\varpi}(h_i; c))] - \mathbb{E}_{\widetilde{\mathbb{P}}}\left[-sp\left(D_{\varpi}\left(\widetilde{h}_i; c\right)\right)\right]. \tag{22}$$

In the above equation, D_ϖ is the discriminator, ϖ is the discriminator parameter, $\widetilde{\mathbb{P}}$ is the negative sample distribution, and $sp(z) = \ln(1 + e^z)$ is the softplus activation function.

5. Experimental Analysis and Results

To demonstrate the effectiveness of the SGIB model, a variety of network analysis experiments, including node classification and node clustering, are performed on three widely used citation networks as well as two large-scale datasets. Numerous experiments show that SGIB outperforms the best available methods. In addition, an in-depth analysis of the model is performed, and the experimental results show that the method can also alleviate the over-smoothing problem to some extent.

The experimental design and results analysis in this section will be carried out in three parts. The first part applies the SGIB model to five public data sets to verify whether the network performance is improved after applying the information bottleneck theory to remove redundant information. The second part verifies whether the over-smoothing problem can be effectively mitigated in the SGIB model with the Dropedge algorithm. The third part verifies whether the improvement in the SGIB model is consistent for different label ratios.

5.1. Datasets and Implementation Details

Datasets: The statistics of the datasets used for node-level tasks are shown in Table 1. **Citation Networks.** A network consisting of papers and their relationships. Edges are the relationships between papers that include citation relationships, common authors, etc. Nodes are also the number of papers in the dataset, and features are the characteristics of each paper. **Coauthor Networks.** Coauthor-CS and Coauthor-Phy are two collaborative author networks based on the Microsoft Academic Graph for the KDD Cup 2016 Challenge. In these networks, nodes represent authors who are connected by an edge and if they have co-authored a paper. The node features represent the paper keywords of each author's paper, and the class labels indicate the most active research areas of each author.

Table 1. Statistics of the datasets used in experiments.

Datasets	Nodes	Edges	Features	Classes	Train/Val/Test Nodes
Cora	2708	5429	1433	7	140/500/1000
Citeseer	3327	4732	3703	6	120/500/1000
Pubmed	19,717	44,338	500	3	60/500/1000
Coauthor-CS	18,333	81,894	6805	15	450/450/17,433
Coauthor-Phy	34,493	247,962	8415	5	150/150/34,193

Implementation Details: SGIB follows the experimental setup of previous state-of-the-art methods, and for the node classification task, the experimental setup of DGI is followed, and the average classification accuracy and standard deviation of the test nodes are obtained by a linear classification model after 50 training sessions. For the node clustering task, the learned node representations are clustered using the K-means algorithm and the F1 score (F1), the average normalization (NMI), and the tuning index (ARI) are obtained for an average of 50 runs. SGIB is trained using a one-layer message-passing network with an Adam optimizer and an initial learning rate of 0.001. In addition, early stopping with a patience of 30 is also utilized. The original map sampling probability is chosen from {0.8, 0.9}. Graph representation dimension is chosen from {256, 512}. Local overall mutual information weights and local and local mutual information weights are chosen from {0.5, 0.9, 1.0}. The software and hardware information involved in this experiment is as follows: Operating system: Ubuntu 7.5.0-3Ubuntu1-18.04 CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 ghz; graphics card: NVIDIA Quadro RTX 5000 16 GB, CUDA 10.2, Pytorch 1.7.0.

5.2. Node Classification

The citation network uses a standard node division ratio. The co-author network selects 30 nodes per class as the training set, 30 nodes per class as the validation set, and the remaining nodes as the test set. Table 2 reports the mean node classification accuracy of our method and other baselines. The supervised models include MLP, LogReg, label propagation (LP) [29], Chebyshev [30], GCN, graph attention networks (GAT), and mixed model networks (MoNet) [31]. The unsupervised models include DGI, GMI, GRACE, GCA, Graph InfoClust (GIC) [32], MVGRL.

Table 2 shows that compared to the six state-of-the-art unsupervised methods, the SGIB model improves in all five datasets and achieves the best results in four of them. The improvement is 0.5 percentage points on Cora, 0.8 percentage points on Pubmed, 0.9 percentage points on Coauthor-CS, and 0.6 percentage points on Coauthor-Phy. The most representative unsupervised model DGI and SGIB were selected to compare the distribution of experimental results. For each model, 20 experimental results are taken for analysis, as shown in Figure 3. As can be seen from the figure, SGIB generally improves the experimental results, which is consistent with DGI. In this study, we believe that the main reason for the performance improvement is the minimization of mutual information between the node-level vector representations of view 1 and view 2. Therefore, the node representation maximizes the extraction of the most important universal and unique information in the graph, while eliminating redundant information.

Table 2. Accuracies in percent on node classification.

	Methods	Input	Cora	Citeseer	Pubmed	Coauthor-CS	Coauthor-Phy
Supervised	MLP	X,Y	58.2 ± 2.1	59.1 ± 2.3	70.0 ± 2.1	88.3 ± 0.7	88.9 ± 1.1
	LogReg	X,A,Y	57.1 ± 2.3	61.0 ± 2.2	64.1 ± 3.1	86.4 ± 0.9	86.7 ± 1.5
	LP	A,Y	68.0 ± 0.2	45.3 ± 0.2	63.0 ± 0.5	74.3 ± 0.0	90.2 ± 0.5
	Chebyshev	X,A,Y	81.2 ± 0.5	69.8 ± 0.5	74.4 ± 0.3	91.5 ± 0.0	92.1 ± 0.3
	GCN	X,A,Y	81.5 ± 0.2	70.3 ± 0.3	**79.0 ± 0.4**	**91.8 ± 0.1**	**92.6 ± 0.7**
	GAT	X,A,Y	**83.0 ± 0.7**	**72.5 ± 0.7**	**79.0 ± 0.3**	90.5 ± 0.7	91.3 ± 0.6
	MoNet	X,A,Y	81.3 ± 1.3	71.2 ± 2.0	78.6 ± 2.3	90.8 ± 0.6	92.5 ± 0.9
Unsupervised	DGI	X,A	81.7 ± 0.6	71.5 ± 0.7	77.3 ± 0.6	90.0 ± 0.3	91.3 ± 0.4
	GMI	X,A	80.7 ± 0.7	71.1 ± 0.2	78.0 ± 1.0	91.0 ± 0.0	OOM
	GRACE	X,A	80.0 ± 0.4	71.7 ± 0.6	79.5 ± 1.1	90.1 ± 0.8	92.3 ± 0.6
	GCA	X,A	80.5 ± 0.5	71.3 ± 0.4	78.6 ± 0.6	91.3 ± 0.4	93.1 ± 0.3
	GIC	X,A	81.7 ± 1.5	71.9 ± 1.4	77.3 ± 1.9	89.4 ± 0.4	93.1 ± 0.7
	MVGRL	X,A	82.8 ± 1.0	**72.7 ± 0.5**	79.6 ± 0.8	91.0 ± 0.6	93.2 ± 1.0
	SGIB	X,A	**83.3 ± 0.7**	71.7 ± 0.8	**80.4 ± 0.6**	**92.2 ± 0.5**	**93.8 ± 0.8**

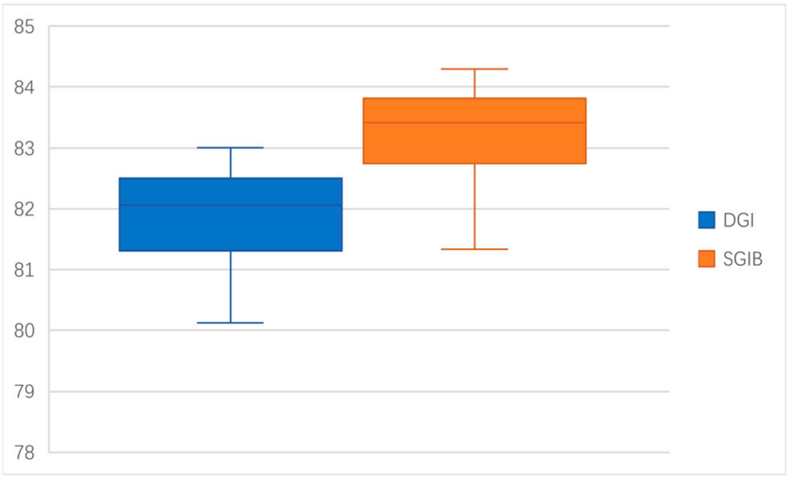


Figure 3. Box-plot of DGI and SGIB.

According to the experimental results, the SGIB model works better on large data sets, which may be because large data sets contain more data diversity and more dispersed features, so that more redundant information is generated in the encoding process. Thus, the model proposed in this paper shows stronger competitiveness when it can remove the task-irrelevant information from the data.

Compared with SGIB, unsupervised learning models such as DGI, GMI, and MVGRL only use mutual information to maximize learning graph representations, i.e., only the most dominant information in the graph is learned, so they are slightly weaker than SGIB models in terms of classification performance. Moreover, in the case of no label information, the proposed method in this study can still show experimental results that rival or even slightly outperform supervised learning models such as GAT and GCN, which is because the high quality and rich information in the graph learned by the SGIB model is sufficient to support the classification task, which will guarantee the performance in the downstream task.

5.3. Node Clustering

For the node clustering task, in addition to the four unsupervised methods mentioned in Section 5.2, seven models were selected for comparison, including K-means, spectral clustering, deep representation for graph Ccustering (DNGR) [33], relational topic models (RTM) [34], robust multi-view spectral clustering (RMSC) [35], text-associated DeepWalk (TADW) [36], and variational graph auto-encoders (VGAE) [37]. The clustering results of Cora, Citeseer, and Pubmed are shown in Table 3.

Table 3. Node clustering result in Micro F1, NMI and ARI.

Methods	Cora			Citeseer			Pubmed		
	F1	**NMI**	**ARI**	**F1**	**NMI**	**ARI**	**F1**	**NMI**	**ARI**
K-means	0.368	0.321	0.230	0.409	0.305	0.279	0.195	0.001	0.002
Spectral	0.318	0.127	0.031	0.299	0.056	0.010	0.271	0.042	0.002
DeepWalk	0.392	0.327	0.243	0.270	0.088	0.092	0.670	0.279	**0.299**
DNGR	0.340	0.318	0.142	0.300	0.180	0.044	0.467	0.155	0.054
RTM	0.307	0.230	0.169	0.342	0.239	0.203	0.444	0.194	0.148
RMSC	0.331	0.255	0.090	0.320	0.139	0.049	0.421	0.255	0.222
TADW	0.481	0.441	0.332	0.414	0.291	0.228	0.335	0.001	0.001
GAE	0.595	0.429	0.347	0.327	0.176	0.124	0.660	0.277	0.279
VGAE	0.609	0.436	0.346	0.308	0.156	0.093	0.634	0.229	0.213
DGI	0.707	0.544	0.472	0.714	0.479	0.485	0.667	**0.307**	0.277
GMI	0.701	0.542	0.495	0.667	0.419	0.418	0.644	0.239	0.225
SGIB	**0.714**	**0.546**	**0.505**	**0.716**	**0.487**	**0.487**	**0.673**	**0.307**	0.279

From the perspective of quantitative analysis, the performance of the SGIB model outperforms the other models in almost all metrics. Moreover, from the perspective of qualitative analysis, as shown in Figure 4 based on the visualized t-SNE plots on the three datasets, the SGIB model shows more discernible clusters, and the separability and tightness of its clustering results are more evident. The feasibility and rationality of the SGIB model can be seen from both qualitative and quantitative perspectives.

5.4. Ablation Experiment

In order to verify the role played by information bottleneck in the SGIB model, ablation experiments are also conducted in this paper. The experiments are conducted to remove the effect of the Dropedge algorithm by performing random edge deletion operations on two other unsupervised models, and verifying whether a higher quality vector representation is possible after removing redundant information. This study selected the extant classical unsupervised model DGI, GMI was combined with the Dropedge algorithm for the experiments, and the setup of the ablation experiments followed that of DGI. The experimental results are shown in Table 4.

Table 4. Ablation experiments.

Methods	Cora	Citeseer	Pubmed
DGI	82.3	71.8	76.8
DGI + Dropedge	82.9	**72.0**	79.5
GMI	80.7	71.1	78.0
GMI + Dropedge	81.9	69.7	78.2
SGIB	**83.3**	71.7	**80.4**

From the above experimental results, we can see that the SGIB model still shows competitiveness after removing the effect of Dropedge. For example, after combining the DGI model with the Dropedge algorithm on the Pubmed dataset, there is a significant performance improvement of 2.7 percentage points for the node classification task, and it can be inferred that the random deletion of edges in the network can remove some redundant information to a certain extent, which makes the model performance improve. The SGIB model also has a nearly 1 percentage point improvement compared to the DGI + Dropedge model, which is entirely due to the application of information bottleneck and proved to be effective in removing redundant information from the representation of the data, thus improving the performance of the model. Moreover, the experimental results on the large-scale dataset Pubmed show more significant improvement than the other two small-scale datasets, for the larger the dataset and the more redundant information exists, the more significant the performance improvement of the SGIB model.

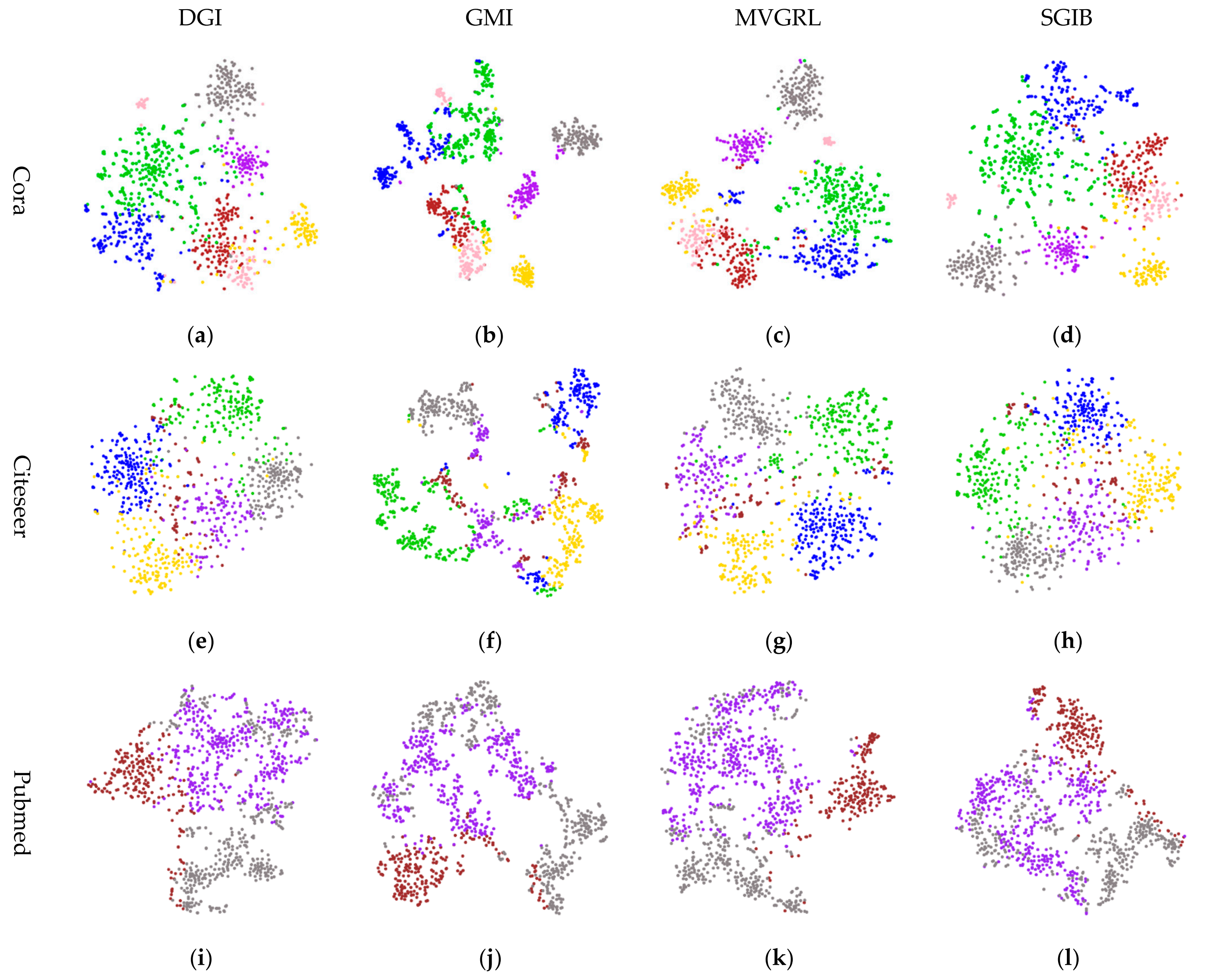


Figure 4. The visualization of the embeddings obtained from DGI, GMI, MVGRL and SGIB. (**a**) DGI for Cora; (**b**) GMI for Cora; (**c**) MVGRL for Cora; (**d**) SGIB for Cora; (**e**) DGI for Citeseer; (**f**) GMI for Citeseer; (**g**) MVGRL for Citeseer; (**h**) SGIB for Citeseer; (**i**) DGI for Pubmed; (**j**) GMI for Pubmed; (**k**) MVGRL for Pubmed; (**l**) SGIB for Pubmed.

5.5. Node Classification with Various Depths

In this subsection, the performance of the SGIB model with other mainstream unsupervised models for node classification is mainly verified when the model depth is gradually increased. Compared with the DGI, GMI, and MVGRL models, the SGIB model is more useful for mitigating network over-smoothing. To ensure the fairness of the experiments, the same set of hyperparameters and the same discard rate p are applied, and the network depth is from layer 1 to layer 32. The experimental results are shown in Table 5.

Table 5. Node classification AC with various depths.

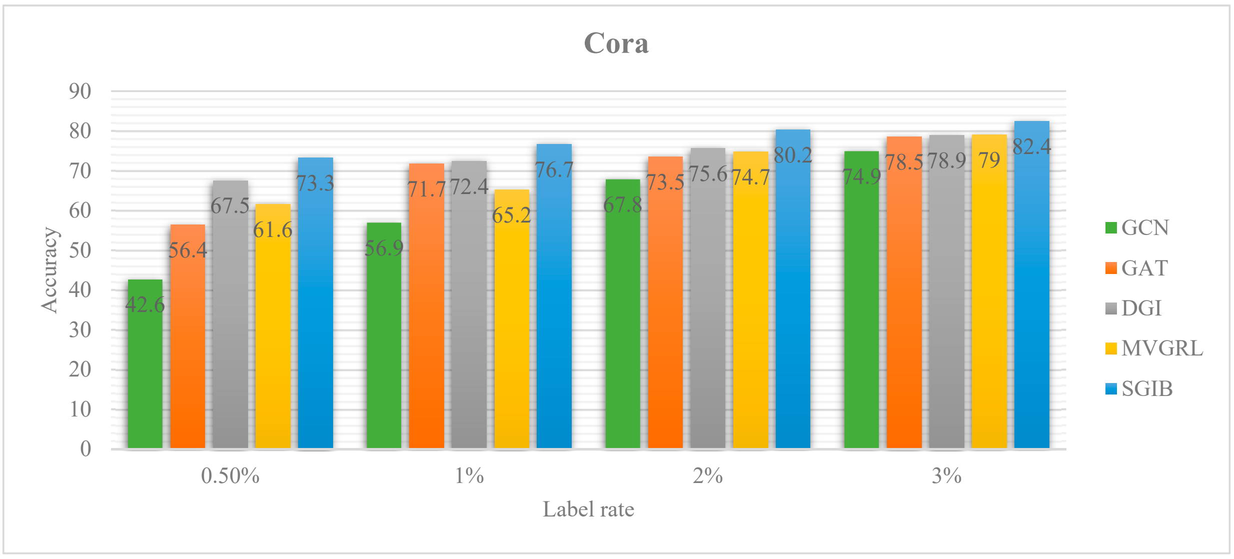
Data	Methods	Depths					
		1	**2**	**4**	**8**	**16**	**32**
Cora	DGI	82.30	79.36	73.10	21.87	20.01	16.43
	DGI + D	82.90	79.00	72.60	36.48	21.60	16.05
	GMI	—	80.70	74.06	37.81	16.22	16.06
	GMI + D	—	**81.95**	77.07	38.01	15.75	16.15
	MVGRL	82.80	81.74	78.20	28.38	22.04	16.82
	SGIB	**83.32**	80.80	**79.07**	**65.54**	**23.80**	**20.86**
Citeseer	DGI	71.80	70.24	62.34	28.18	20.39	17.10
	DGI + D	72.02	70.51	64.85	32.59	21.25	17.07
	GMI	—	71.10	58.80	38.18	20.70	17.06
	GMI + D	—	69.70	54.76	39.60	24.41	19.83
	MVGRL	**72.70**	69.28	60.29	52.96	33.02	18.32
	SGIB	71.73	**71.29**	**67.50**	**58.11**	**35.51**	**20.11**
Pubmed	DGI	76.80	73.80	65.23	50.56	45.21	34.97
	DGI + D	79.48	71.83	64.94	51.20	41.84	34.89
	GMI	—	78.00	75.50	61.41	44.36	34.67
	GMI + D	—	78.18	74.62	61.32	36.39	34.81
	MVGRL	79.60	75.30	67.78	36.28	34.40	34.12
	SGIB	**80.44**	**80.10**	**75.50**	**61.58**	**45.60**	**43.66**

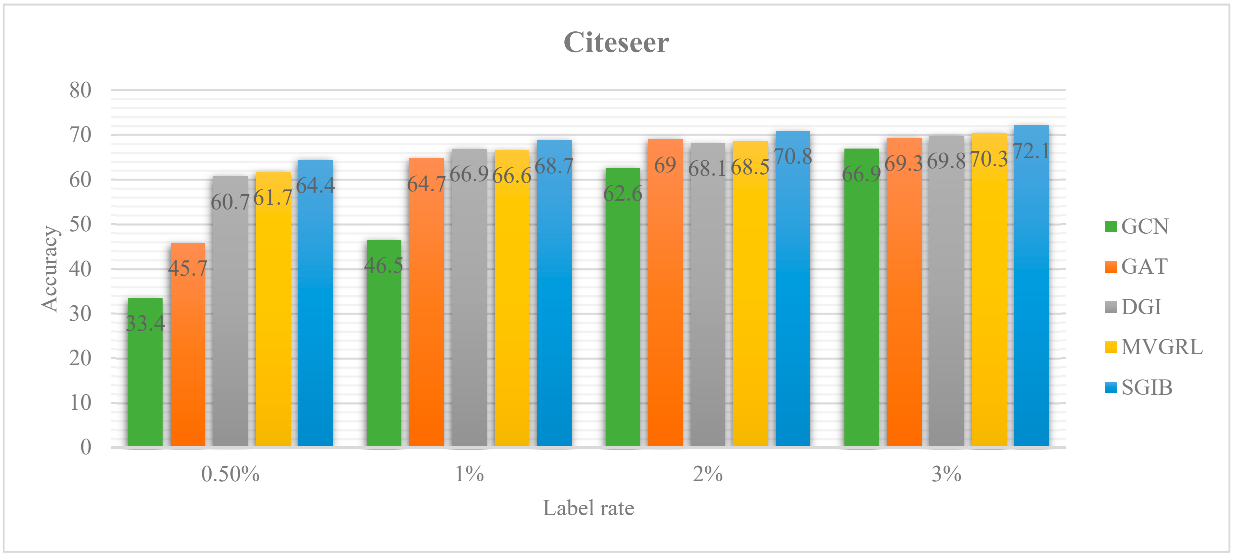
The experimental comparison revealed that for the two-layer network Cora and Citeseer datasets generally did not work as well as the GMI model, which may be due to the original setting of the GMI model as a two-layer network. As the depth of the model network increases, the SGIB model presents optimal results for all results from 4 to 32 layers. In particular, it has a significant performance improvement of 9.5 percentage points on the Pubmed dataset, from which it is inferred that the SGIB model may be better at mitigating over-smoothing on large datasets.

5.6. Limited Labeled Training

This section focuses on the performance comparison of the SGIB model with other mainstream graph neural network models (GCN, GAT, DGI, MVGRL) on node classification tasks at low labeling rates. Thus, the improvement of SGIB model performance is verified to be consistent under different labeling rates. The labeling rates were set to 1%, 2%, and 3% on the Cora and Citeseer datasets, and 0.03%, 0.05%, and 0.1% on the Pubmed dataset.

From the above experimental results, as shown in Figure 5, it can be concluded that the SGIB model can achieve excellent results in the case of different labeling rates as well, and the improvement is consistent in the face of different labeling rates. In all three data sets, the lower the labeling rate, the more obvious the improvement effect is. For example, on the Cora dataset with a 1% labeling rate, there is a 4.3 percentage point improvement, which indicates that the SGIB model relies more on the self-supervised approach to obtain information about the nodes themselves.





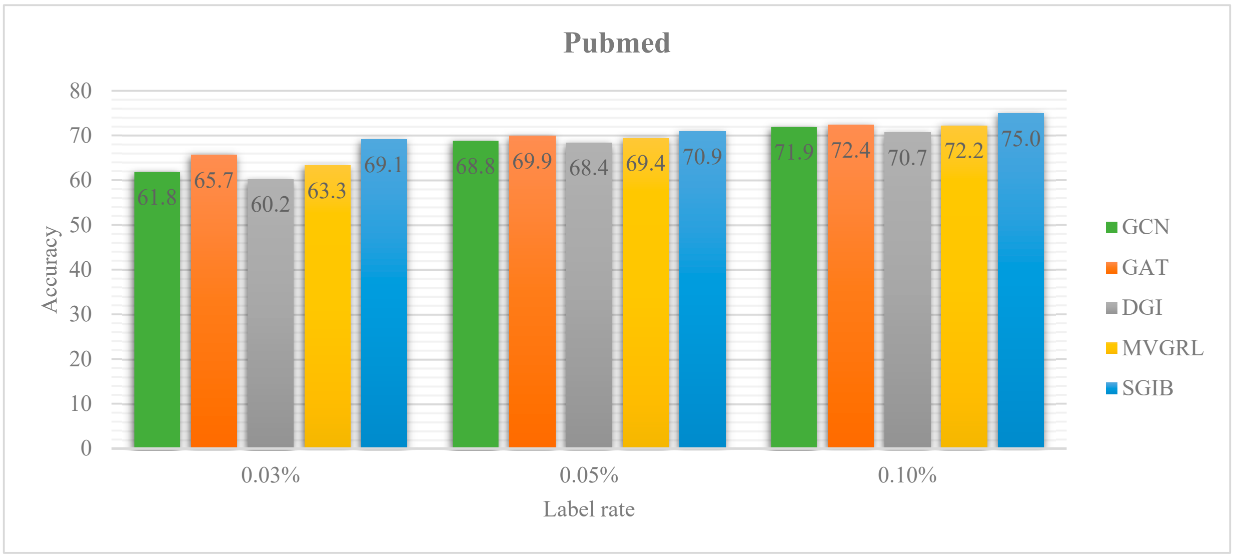


Figure 5. Node classification results with limited training labels on Cora, Citeseer, and Pubmed.

6. Conclusions

In this paper, we propose a graph network representation model based on information bottleneck to address the problem that the neighboring node features in the input graph may contain useless information, and we implement the self-supervised SGIB algorithm using the contrast learning method. The SGIB algorithm introduces the information bottleneck theory so that the vector representation learned by the encoder from the graph structure data contains minimum and sufficient input information. In this study, experiments were conducted on five datasets, and the experimental results show that (1) the SGIB algorithm removes redundant information based on the extraction of common and independent features of multiple inputs, making the encoder less susceptible to noisy data; (2) the Dropedge algorithm for contrast learning can alleviate the overfitting problem to a certain extent; (3) the performance improvement of the SGIB algorithm compared with the state-of-the-art supervised learning algorithm and an unsupervised learning algorithm is verified in a large number of experiments, and it is found that the SGIB algorithm performs better on large-scale data sets.

Author Contributions: Conceptualization, J.G.; methodology, L.Y.; software, Z.Z.; validation, Z.Z.; writing-original draft preparation, Z.Z.; writing-review and editing, W.Z.; supervision, Y.Z.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under Grant 61972442; in part by the Key Research and Development Project of Hebei Province of China under Grant 20350802D and 20310802D; in part by the Natural Science Foundation of Hebei Province of China under Grant F2020202040; in part by the Natural Science Foundation of Tianjin of China under Grant 20JCYBJC00650.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
3. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the 30th Advances in Neural Information Processing Systems Conference, Long Beach, CA, USA, 4–9 December 2017; pp. 1024–1034.
4. Pan, S.; Wu, J.; Zhu, X.; Long, G.; Zhang, C. Task sensitive feature exploration and learning for multitask graph classification. *IEEE Trans. Cybern.* **2016**, *47*, 744–758. [CrossRef] [PubMed]
5. Chen, D.; Nie, M.; Zhang, H.; Wang, Z.; Wang, D. Network embedding algorithm taking in variational graph autoencoder. *Mathematics* **2022**, *10*, 485. [CrossRef]
6. Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; Zhang, C. Adversarially regularized graph autoencoder for graph embedding. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 2609–2615.
7. Zhang, Y.; Pal, S.; Coates, M.; Ustebay, D. Bayesian graph convolutional neural networks for semi-supervised classification. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 5829–5836.
8. Qiu, J.; Tang, J.; Ma, H.; Dong, Y.; Wang, K.; Tang, J. Deepinf: Social influence prediction with deep learning. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2110–2119.
9. Bai, Y.; Ding, H.; Bian, S.; Chen, T.; Sun, Y.; Wang, W. Simgnn: A neural network approach to fast graph similarity computation. In Proceedings of the 12th ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 384–392.
10. Sun, F.Y.; Hoffmann, J.; Verma, V.; Tang, J. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.

11. Liben-Nowell, D.; Kleinberg, J. The link-prediction problem for social networks. *J. Assoc. Inf. Sci. Technol.* **2007**, *58*, 1019–1031. [CrossRef]
12. Zhang, M.H.; Chen, Y.X. Link prediction based on graph neural networks. In Proceedings of the 31st Advances in Neural Information Processing Systems Conference, Montréal, QC, Canada, 3–8 December 2018; pp. 5171–5181.
13. Velickovic, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep graph infomax. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
14. Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; Huang, J. Graph representation learning via graphical mutual information maximization. In Proceedings of the 20th Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 259–270.
15. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020; Volume 119, pp. 4116–4126.
16. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Deep graph contrastive representation learning. *arXiv* **2020**, arXiv:2006.04131.
17. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Graph contrastive learning with adaptive augmentation. In Proceedings of the 21th Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 2069–2080.
18. Yang, L.; Wu, F.; Zheng, Z.; Niu, B.; Gu, J.; Wang, C.; Cao, X.; Guo, Y. Heterogeneous graph information bottleneck. In Proceedings of the 30th International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–27 August 2021; pp. 1638–1645.
19. Yu, J.; Xu, T.; Rong, Y.; Bian, Y.; Huang, J.; He, R. Recognizing predictive substructures with subgraph information bottleneck. *arXiv* **2021**, arXiv:2103.11155. [CrossRef] [PubMed]
20. Yu, J.; Cao, J.; He, R. Improving subgraph recognition with variational graph information bottleneck. *arXiv* **2021**, arXiv:2112.09899.
21. Belghazi, M.I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; Hjelm, R.D. Mutual information neural estimation. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 530–539.
22. Achille, A.; Soatto, S. Emergence of invariance and disentanglement in deep representations. *J. Mach. Learn. Res.* **2018**, *19*, 1947–1980.
23. Tishby, N.; Pereira, F.C.; Bialek, W. The information bottleneck method. *arXiv* **2000**, arXiv:physics/0004057.
24. Alemi, A.A.; Fischer, I.; Dillon, J.V.; Murphy, K. Deep variational information bottleneck. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
25. Federici, M.; Dutta, A.; Forré, P.; Kushman, N.; Akata, Z. Learning robust representations via multi-view information bottleneck. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
26. Rong, Y.; Huang, W.; Xu, T.; Huang, J. Dropedge: Towards deep graph convolutional networks on node classification. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
27. Nowozin, S.; Cseke, B.; Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. In Proceedings of the 29th Advances in Neural Information Processing Systems Conference, Barcelona, Spain, 5–10 December 2016; pp. 271–279.
28. Oord, A.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
29. Zhu, X.; Ghahramani, Z.; Lafferty, J.D. Semi-supervised learning using gaussian fields and harmonic functions. In Proceedings of the 20th International Conference on Machine Learning, Washington, DC, USA, 21–24 August 2003; pp. 912–919.
30. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 29th Advances in Neural Information Processing Systems Conference, Barcelona, Spain, 5–10 December 2016; pp. 3837–3845.
31. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5424–5434.
32. Mavromatis, C.; Karypis, G. Graph infoclust: Maximizing coarse-grain mutual information in graphs. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 25th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Virtual Event, 11–14 May 2021*; Karlapalem, K., Cheng, H., Ramakrishnan, N., Agrawal, R.K., Reddy, P.K., Srivastava, J., Chakraborty, T., Eds.; Springer: Cham, Switzerland, 2021; Volume 12712, pp. 541–553.
33. Cao, S.; Lu, W.; Xu, Q. Deep neural networks for learning graph representations. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1145–1152.
34. Chang, J.; Blei, D. Relational topic models for document networks. In Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009; Volume 5, pp. 81–88.
35. Xia, R.; Pan, Y.; Du, L.; Yin, J. Robust multi-view spectral clustering via low-rank and sparse decomposition. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–13 July 2014; pp. 2149–2155.
36. Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; Chang, E. Network representation learning with rich text information. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 2111–2117.
37. Kipf, T.N.; Welling, M. Variational graph auto-encoders. *arXiv* **2016**, arXiv:1611.07308.