MVRACE: Multi-view Graph Contrastive Encoding for Graph Neural Network Pre-training

Bo Wu [1] (wubochn@ruc.edu.cn)
Xun Liang[2] (xliang@ruc.edu.cn)
Xiangping Zheng[3] (xpzheng@ruc.edu.cn)
Yuhui Guo[4] (yhguo@ruc.edu.cn)
Xuan Zhang[5] (zhangxuanalex@ruc.edu.cn)
Hui Tang[6] (huitang@ruc.edu.cn)
Renmin University of China
Beijing, China

Jun Wang[7] (junwang@swin.edu.au)
Swinburne University of Technology
Melbourne, Australia

Abstract

Graph neural networks (GNNs) have become a *de facto* paradigm for graph representation learning. Generally, GNNs are trained in an end-to-end manner with supervision, requiring considerable task-specific labeled data. To reduce the labeling burden, recent works leverage self-supervised tasks to pre-train an expressive GNN model on abundant unlabeled data and finetune the trained model on downstream datasets with only a few labels. However, existing GNN pre-training approaches only concentrate on a single view for graph self-supervised learning while ignoring the rich semantic information in graphs, leading to the lack of sample utilization efficiency during the pre-training process. To tackle such challenges, we propose a multi-view graph contrastive encoding for graphs during GNN pre-training, called MVRACE. The critical insight is that we construct node and graph-level views to capture local attribute information and global structure in a graph. Concretely, the node-level view utilizes graph centrality and encodes the *r*-ego network to capture the local-whole relationship in a graph. The graph-level view aims to encode graph pairs to explore different graph structures and empower the discrimination ability of the GNN encoder. In addition, we combine multi-views with a joint contrastive loss function to integrate node- and graph-semantic information simultaneously. Comprehensive experiments on multiple domain datasets demonstrate that our approach can significantly yield competitive performance compared to state-of-the-art methods. **Keywords:** graph neural network; graph contrastive encoding; multi-view

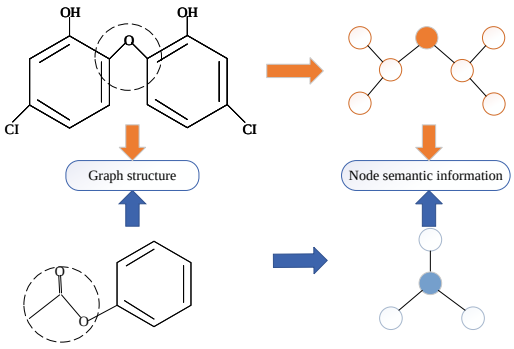


Figure 1: The illustration of multi-view graph contrastive learning. The node-level view encode substructure to describe the key node semantic information. The graph-level view encode the whole graph information and attempt to discriminate different graph structure.

Generally, GNNs are often trained in a (semi-) supervised manner, which is arduously expensive and sometimes infeasible to access sufficient labeled data for domain-specific tasks. For example, it is hard to collect enough correct annotation for biological proteins and chemical molecules due to hardware measurement limitations or human errors (Che, 2021; Gedela, Bobby, & Bhatt, 2022). These natural science fields have still faced the challenge of the lack of ground truth. Inspired by remarkable success of pre-trained language models (Devlin, Chang, & L., 2019), some advances (Z. Hu, Dong, Wang, Chang, & Sun, 2020; Qiu et al., 2020; Lu, Jiang, Fang, & Shi, 2021; W. Hu et al., 2020) concentrate on pre-training GNNs and learn the transferable knowledge from self-supervised tasks to downstream tasks, which further facilitates GNN performance and reduce the labeling burden. However, existing GNN pre-training methods simply create positive and negative samples in graph-based self-supervised tasks, while they ignore the rich semantic information at the node and graph levels, and lack the ability to distinguish similar graph structures. At the node level, there are always some key nodes in the graph. The local structures formed by these nodes and neighbors usually become an important component of a graph, which requires us to put more efforts on extracting and identifying these local structures. In addition, the graph structure is diverse and constantly changing, and the description of a single view is likely to be incomplete and one-sided. Therefore, we need to strengthen the ability of GNNs to distinguish graphs with different structures, especially for graphs with similar structures but not the same type, such as isomers in chemical molecules. Take Figure 1 as an example, some very important atoms and chemical bonds (i.e., oxygen atom and large π bond) dominate in a molecule graph, which determines the unique properties of chemical molecules. If we construct the positive and negative sample pairs according to the graph data augmentaion strategy (You et al., 2020) (e.g., dropping nodes, deleting edges), it is likely to destroy the key atoms and chemical bonds, which reduces the ability of GNNs to capture key local features and relationships. Meanwhile, these two chemical molecules are very similar in structure, due to both of them containing *benzene rings*, which brings difficulties to GNN learning the structure of the graph and may cause misclassfication on graphs. The combination of multi-view information inside the graph is often overlooked by above methods so that they have a deficiency in the scalability and robustness.

In M. Goldwater, F. K. Anggoro, B. K. Hayes, & D. C. Ong (Eds.), *Proceedings of the 45th Annual Conference of the Cognitive Science Society.*

To overcome these limitations, we propose a novel Multi-View gRAph Contrastive Encoding for GNN pre-training, MVRACE for brevity, to explore rich multiview information among nodes and graphs. Multi-view representation learning is able to generate multiple views of a same object, which has achieved state-of-the-art performance on image classification (Chen, Kornblith, Norouzi, & Hinton, 2020; Bachman, Hjelm, & Buchwalter, 2019). Inspired by this, we firstly introduce two new views that reflect local and global semantic information. We construct node and graph-level encoding and incorporate attribute of nodes and structure of graph to capture the different semantic relation in the graph. Furthermore, we introduce a self-supervised technique to learn graph representations by contrastive learning based on different views. To help MVRACE learn the commonality between two views well, we conbine node and graph contrastive learning loss to optimize the model and balance information from different views better. Finally, we conduct a series of experiments about on large-scale graph datasets, which shows the superiority of our proposed methods. Our methods open the direction of designing more complementary views for GNN pre-training, which is significant and can learn the essential and inherent feature of the graph. In summary, our contributions are summarized as follows:

- We propose a multi-view graph contrastive encoding model for GNNs pre-training, called MVRACE. MVRACE can effectively mine the semantic relationship at the node- and graph- level, which can learn complex interaction patterns among nodes and graphs.
- We design an effective contrastive learning method to learn high-quality node embeddings from different views. Our method is unsupervised and can effectively capture the features and structure information of the graph without the help of data labels.
- We conduct extensive experiments on large-scale graph datasets in different domains for pre-training and evaluate it on downstream task. The experiment results shows that our approach achieves competitive or better performance compared to baselines, even with few labeled training instances.

Related Work

Graph Neural Network Pre-training

GNNs aggregate the features of neighbor nodes and iteratively update node representations. pre-training GNN intends to transfer knowledge from source domain graphs to target domain graphs. (W. Hu et al., 2020) utilizes the self-supervised strategies to generate node and graph embeddings and combines them to obtain local and global semantic information of the entire graph. GCC (Qiu et al., 2020) extracts positive and negative subgraph instances and distinguish them with contrast learning. GPT-GNN (Z. Hu et al., 2020) masks the attributes of a graph and designs a generation task to capture the structural and semantic properties of the graph. L2P-GNN (Lu et al., 2021) constructs parent tasks and child tasks to make up the gap in the optimization objectives between pre-training and fine-tuning.

Graph Contrastive Learning

Contrastive learning is one of the state-of-the-art self-supervised representation learning algorithms, which can capture inherent attribute of the data by constructing positive and negative sample pairs (Jaiswal, Babu, Zadeh, Banerjee, & Makedon, 2020; Jing, Zhu, Zang, & Wang, 2023). Recently, graph contrastive learning attracts a lot of attention, especially in the situation of label missing and complex graph structure (Zheng et al., 2023; Tang, Liang, Guo, Zheng, & Wu, 2022; Xu, Deng, Xie, & Ji, 2023). (You et al., 2020) designs a graph augmentation strategy and performs mutual information maximization to complete graph contrastive learning. (Chu, Wang, Shi, & Jiang, 2021) is absorbed in negative graph samples and utilizes curriculum Learning (Bengio, Louradour, Collobert, & Weston, 2009) to select more representive negative samples. In addition, (Mo, Peng, Xu, Shi, & Zhu, 2022) introduces a multiplet loss to explores the complementary information between the structural information and neighbor information, which enlarges inter-class variation and reduces intra-class variation.

Preliminaries

GNNs. Let $G = (V, E, \mathbf{X}, \mathbf{A})$ be a graph with nodes V and edges E, where $\mathbf{X} \in \mathbb{R}^{n \times d_0}$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ are feature matrix and adjacency matrix, respectively. n is the number of nodes and d_0 denotes the dimension of initial feature. GNNs leverage the aggregator function to cpature neighbor feature and iteratively update node representation (Gilmer, Schoenholz, & Riley, 2017), i.e.,

$$\mathbf{h}_i^k = \Theta\left(\mathbf{h}_i^{k-1}, \delta_{j \in \mathcal{N}_i} \phi^k\left(\mathbf{h}_i^{k-1}, \mathbf{h}_j^{k-1}, \mathbf{e}_{i,j}\right)\right), \tag{1}$$

where $\mathbf{h}_i^k$ represents the k-th iteratively embedding of the node i. $\mathbf{h}_i^0 = \mathbf{x}_i$ denotes the initial feature. $\mathbf{e}_{i,j}$ represents the edge feature between node i and j. Θ and ϕ represent differentiable functions or network layers, such as multi-layer perceptron (MLP). δ determines how to aggregate neighbor node features such as sum, mean or max (Fey & Lenssen, 2019). To obtain graph representation $\mathbf{h}_G$, the *READOUT* function pools node features from the final iteration k (W. Hu et al., 2020), i.e.,

$$\mathbf{h}_G = READOUT(\mathbf{h}_i | i \in G), \tag{2}$$

where *READOUT* is a permutation-invariant function, such as averaging, maxing or a more sophisticated graph-level pooling layers (Ying & You, 2018; Zhang, Cui, & Neumann, 2018).

Pre-training and Fine-tuning. The conventional GNN pre-training process can be divided into a two step paradigm: (1) Training GNNs on abundant graph datasets to abtain prior knowledge, and perform the gradient descent to update parameters. (2) Saving the trained parameters, then fine-tuning

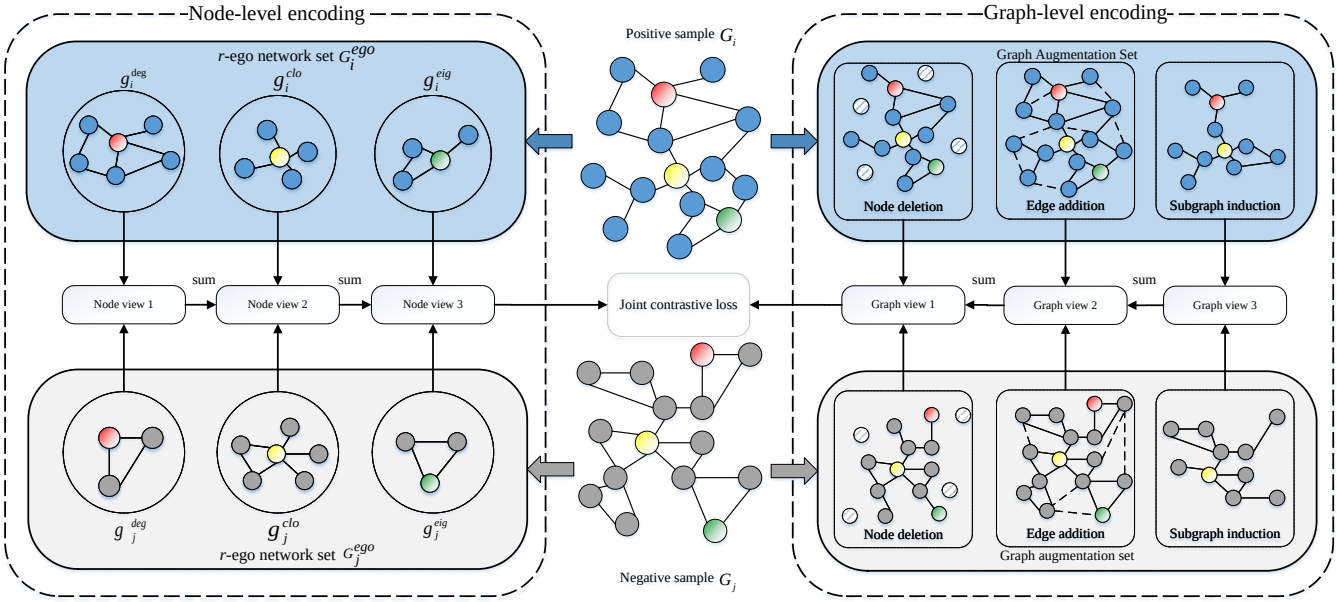


Figure 2: The Overall framework of MVRACE. We create a multi-view encoding mechanism consisting of the node and graph level. In the middle are the original graph and the negative sample graph. On the left is the node-level encoding, including three ego network comparisons that describe the centrality of the diagram. On the right is the graph-level encoding, including three graph structure comparisons with different graph enhancement strategies. Finally, the total loss function is obtained by summing the node- and graph-level contrastive loss.

the GNNs on the downstream datasets. According to the description of (Lu et al., 2021), We can describe the step 1 and 2 by Eq. (3) and (4), respectively, i.e.,

$$\theta_0 = \arg\min_{\theta} \mathcal{L}^{pre}\left(f_\theta; \mathcal{D}^{pre}\right), \tag{3}$$

$$\theta_1 \overset{update}{\longleftarrow} \theta_0 - \eta \nabla_{\theta_0} \mathcal{L}^{fine}\left(f_{\theta_0}; \mathcal{D}^{fine}\right), \tag{4}$$

where $\mathcal{L}, f, \mathcal{D}^{pre}, \mathcal{D}^{fine}$, and η denote loss function, GNNs, pre-training datasets, downstream datasets and learning rate, respectively.

Multi-view Graph Contrastive Encoding

In this section, we develop a novel multi-view contrastive encoding strategy for GNN pre-training, named MVRACE, which consists of node- and graph-level encoding. Figure 2 presents an overall framework of our method.

Node-level Encoding

Existing graph contrastive learning methods ignore the local structure in a graph, of which rich semantic information is stored in the nodes and their neighbors. The local structure is closely related to the whole of the graph and can become the symbolic feature of the graph. Therefore, we first define the r-ego network and graph centrality to find significant nodes and utilize the r-ego network to include nodes and their neighbors. We encode the r-ego network with a local-whole contrastive loss to obtain node-level semantic information.

Definition 1. **r-ego network.** The Ego network originates from social networks (Li, Wang, & Chang, 2014), which consists of a focal node (i.e., ego) and plus the neighbors directly connecting to the node. Given a vertex v, we can define the r-ego network as a subgraph $S_v = \{(u,e) \mid d(v,u) \leq r\}$. The $d(v,u)$ represents the shortest path distance between node v and u in graph G.

Definition 2. **Graph centrality.** The importance of each node is different in a graph, which forces us to select the most representative nodes to capture the pivotal local structure. Graph centrality (Bergamini, Borassi, Crescenzi, Marino, & Meyerhenke, 2019) can be used to measure the importance of nodes. We utilize the following graph centrality measurement to sort the importance of nodes and pick up the node with maximum centrality.

- **Degree centrality.** The degree centrality is the most direct measure of node centrality in graph analysis. In the graph, the greater the degree of a node, the higher the degree centrality of the node, which means that the node is more important in the graph, formulated as follows:

$$C^{deg} = max\{deg(i)\}, \tag{5}$$

where $deg(i)$ represents the degree of node i.

- **Closeness centrality.** The closeness centrality denotes the shortest distance from a node to all other nodes. The central node has the best visual field in the graph and can

quickly perceive what is happening at other nodes and the direction of information flow, formulated as:

$$C^{clo} = max\{\frac{1}{\sum_j d_{ij}}\}(i \neq j), \tag{6}$$

where d_{ij} denotes the shortest distance between node i and j.

- **Eigenvector centrality.** The eigenvector centrality $\mathbf{x}$ describes the eigenvector of the adjacency matrix with the largest eigenvalue λ, i.e., the solution of $\mathbf{Ax} = \lambda\mathbf{x}$, which can be defined as:

$$C^{eig} = \lambda, \tag{7}$$

where $\mathbf{A}$ denotes the adjacency matrix and $\mathbf{x}$ represents the eigenvector.

It should be noted that our purpose is not to utilize the graph centrality directly for graph representation learning but to take out the node corresponding to the centrality so as to export the ego network of the central node, namely $\{node^{deg}, node^{clo}, node^{eig} | C^{deg}, C^{clo}, C^{eig}\}$. Then we derive the r-ego networks from these nodes. For each graph, it has the three r-ego network, marked as r-ego network set $G^{ego} = \{g^{deg}, g^{clo}, g^{eig}\}$.

The node-level encoding task treats each ego-network instance as a distinct class of its own and learns to discriminate between these instances from different graph, which can explore the local-whole relationship in a graph and capture the representative features at the node level. To this end, we adopt the InfoNCE loss (van den Oord, Li, & Vinyals, 2018). The InfoNCE loss can be described as a process that only leverages its own information to distinguish positive and negative samples. Given a query key G_i, we calculate the values of positive ego samples G_i^{ego} and the values of K negative ego sample $G_j^{ego}(j \neq i)$, so as to maximize the value of positive sample relative to query key, defined as:

$$\mathcal{L}^{node} = -\log \frac{\exp\left(\mathbf{q}_i^\top \mathbf{h}_i^{ego}/\tau\right)}{\sum_{j=0}^{K} \exp\left(\mathbf{q}_i^\top \mathbf{h}_j^{ego}/\tau\right)}, ego \in \{deg, clo, eig\}, \tag{8}$$

where τ is the temperature hyper-parameter. f_θ represent GNN encoder and obtain the node embeddings corresponding to query instance and key instance (i.e., positive ego samples and negative ego samples), marked as $\mathbf{q}_i = f_\theta(G_i)$, $\mathbf{h}_i^{ego} = f_\theta(G_i^{ego})$, $\mathbf{h}_j^{ego} = f_\theta(G_j^{ego})$, respectively.

Disscussion. In some graphs, the most important node calculated from the graph centrality may be the same. In this case, we will merge the same nodes and export the corresponding ego networks for coding calculation to reduce the time of calculating the graph centrality and rank the importance of nodes. The time complexity of degree centrality, closeness centrality and eigenvector centrality are $O(|V| \cdot |E|)$, $O(|V|^2)$ and $O\left(\frac{|V|}{\log|\lambda_1/\lambda_2|}\right)$, respectively.

Graph-level Encoding

The graph-level view aims to encode the structure and attribute of the whole graph and strengthen the recognition ability of the GNN encoder for graphs with different structures during the pre-training process. In order to achieve our goal, we use a graph augmentation strategy, which is widely used to expand the number of samples and construct different sample variants to strengthen the ability of the model. We employ a predicting task whether two augmented graphs originate from the same original graph or not to enhance the discrimination of the GNN encoder.

Graph Augmentation. To perform graph-level encoding on graphs, we employ graph augmentation to obtain positive and negative sample pairs at the graph level without affecting the semantics label. Given a collection of original graphs $G \in \{G_n : n \in N\}$. We formulate the augmented graph $\hat{G} \sim q(\hat{G} \mid G)$, where $q(\cdot \mid G)$ is the augmentation distribution conditioned on the original graph, which is pre-defined, representing the human prior for data distribution. For instance, some graphs can only delete nodes because the edges are sparse. We select and use the graph augmentation method introduced by (You et al., 2020). The following are graph augmentation strategies.

- **Node deletion.** As shown in Figure 2, we randomly delete a certain portion of nodes marked with slashes.
- **Edge addition.** As shown in Figure 2, we perturb the connectivities in G by randomly adding a certain ratio of edges, marked as the dotted line. It implies that the semantic meaning of G has certain robustness to the edge connectivity pattern variances.
- **Subgraph Induction.** As shown in Figure 2, we sample a subgraph from G using a random walk, which assumes the structure saved in the subgraph can effectively represent the original graph.

The same as the node-level encoding, we still use the InfoNCE loss (van den Oord et al., 2018) function to distinguish different graphs. It can be represented as follows,

$$\mathcal{L}^{graph} = -\log \frac{\exp\left(\mathbf{z}_i^\top \hat{\mathbf{z}}_i/\tau\right)}{\sum_{j=0}^{K} \exp\left(\mathbf{z}_i^\top \mathbf{z}_j/\tau\right)}, \tag{9}$$

where τ is the temperature hyper-parameter. f_θ is the GNN encoder to obtain the graph embeddings corresponding to query instance and key instance (i.e., original graph embedding augmented graph embedding), marked as $\mathbf{z}_i = f_\theta(G_i)$, $\hat{\mathbf{z}}_i = f_\theta(\hat{G}_i)$, $\mathbf{z}_j = f_\theta(G_j)$, respectively.

Pre-training Schema

Figure 2 shows the overall scheme of the pre-training framework. MVRACE employs the pre-training contrastive learning at both the node- and graph- levels on the graph G. We minimize the following loss:

$$\mathcal{L} = \mathcal{L}^{\text{node}} + \lambda\mathcal{L}^{\text{graph}}, \tag{10}$$

where λ is the balance coefficient.

Experiment

Datasets

We conduct experiments on two domain datasets: biology and bibliography. In biology , we follow the (W. Hu et al., 2020) setting in PPI (Zitnik, Feldman, & Leskovec, 2019) dataset, of which about 306K protein ego-networks is used for pre-training. In fine-tuning, we predict 40 fine-grained biological functions with 88K labeled subgraphs derived from PPI. The bibliographic graphs called PreDBLP, constructed by (Lu et al., 2021), contains 1,054,309 paper subgraphs in 31 field (e.g., artificial intelligence, data mining). We follow the setting that (Lu et al., 2021) divides 794,862 subgraphs for pre-training and 299,447 subgraphs for fine-tuning, of which the task is to predict the research field from 6 different categories.

Baselines

To contextualize the empirical results of MVRACE on the benchmarks, we compare against six baselines. (1) **No pre-train** means training and testing GNNs on the downstream datasets directly. (2) **Infomax** (Velickovic, Fedus, & H., 2019) aims to maximize mutual information between local node representations and global graph representation. (3) **EdgePred** (Hamilton, Ying, & Leskovec, 2017) predicts whether there is an edge between nodes. (4) **AttrMasking** (W. Hu et al., 2020) learns the regularities of the node or edge attributes distributed by leveraging the graph structure. (5) **Context Prediction.** (W. Hu et al., 2020) use ssubgraphs to predict their surrounding graph structures. (6) **L2P-GNN** (Lu et al., 2021) constructs task set and optimizing the GNN pre-training process with node- and graph-level adaptation.

Settings

Pre-training. We pre-train four GNN architectures on each dataset for 100 epochs and use Adam (Kingma & Ba, 2015) as optimization with learning rate of 0.001, weight decay of 1e-4. For a fair comparison, we fix the hidden representation dimension to be 300. In addition, we set the r-ego network to 1 hop in MVRACE, which means the nearest neighbors. The default value of balance coefficient λ is 1.

Fine-tuning. We fine-tune the pre-trained GNNs on downstream datasets. The optimizer is still Adam (Kingma & Ba, 2015) with the learning rate of 0.001 and weight decay of 1e-4. For all domain datasets, we split the downstream datasets with an 8:1:1 ratio for train, validation, and test sets. All downstream experiments are repeated with ten random seeds, and we report the mean with standard deviation following (Lu et al., 2021). (Our hardware environment: CPU: Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, GPU: NVIDIA RTX 3090@24GB, Memory: 128GB.)

Performance on Downstream Tasks

Table 1 compares the performance of MVRACE and state-of-the-art pre-training baselines, w.r.t. four different GNN architectures. After pre-training our MVRACE and the baselines on the biology and bibliography dataset, we apply the pre-trained model to the fine-tuning datasets to predict graph labels. We evaluate the test performance with average ROC-AUC across the downstream task. We make the following observations.

Overall, the MVRACE consistently yields the best performance among all methods across different GNN architectures, which brings an average ROC-AUC improvement by 0.41%-3.04% compared to suboptimal method L2P-GNN. We believe that such significant improvements can be attributed to the rich node- and graph- semantic information captured during the pre-training process. Compared to the no pre-train baselines, our MVRACE significantly improves the performance by 1.91%-9.67% on two datasets. The improvements suggest that the multi-view contrastive encoding on graphs is capable of learning transferable and informative knowledge for the downstream tasks.

At the same time, in the face of different GNN architectures, the effect of different pre-training methods is also very different. On the biology dataset, GCN, GraphSAGE and GIN have a large improvement without pre training, but GAT has a small improvement. On the PreDBLP dataset, GCN, GraphSAGE and GAT have a large increase, while GIN has a small increase. However, in general, MVRACE can still maintain considerable performance improvement in the face of different GNN frameworks, which further shows the effectiveness of our utilization of local and global graph structure and semantic capture.

Model Analysis

In this section, we investigate the inherent mechanism of MVRACE. As the node- and graph-level encoding play pivotal roles in MVRACE, we conduct the ablation experiment to explore their respective impact. Furthermore, the influence of different parameter values on the model's performance is verified by parameter analysis. Finally, we investigate the improvement of MVRACE in fine-tuning and the sample utilization. Since similar trends are observed for different GNNs architectures, we only report the GIN results.

Ablation Study. We conduct an ablation study to explore the effectiveness of node- and graph-level encoding in pre-training, respectively. We compare three ablated variants, namely MVRACE-node, MVRACE-node (random), and MVRACE-graph. The MVRACE-node only includes the node-level encoding component, and MVRACE-node (random) replaces the graph centrality nodes with random nodes. The MVRACE (graph) only includes the graph-level encoding component. As Table 2 shows, the three ablated variants perform better than no pre-training baselines on Biology and PreDBLP datasets, illustrating that either the node-level encoding or graph-level encoding is beneficial for pre-training. The mean and standard deviation values of MVRACE-node (random) are lower than MVRACE-node, denoting that encoding graph centrality nodes could further learn the significant node information. In addition, our complete MVRACE is superior to all variants, indicating capturing local and

Table 1: Experimental results ROC-AUC (mean ± std in percent) of different pre-training strategies w.r.t. various GNN architectures. The improvements are relative to the respective GNN without pre-training.

Model	Biology				PreDBLP			
	GCN	GraphSAGE	GAT	GIN	GCN	GraphSAGE	GAT	GIN
No pre-train	63.22±1.06	65.72±1.23	68.21±1.26	64.82±1.21	62.18±0.43	61.03±0.65	59.63±2.32	69.01±0.23
EdgePred	64.72±1.06	67.39±1.54	67.37±1.31	65.93±1.65	65.44±0.42	63.60±0.21	55.56±1.67	69.43±0.07
DGI	64.33±1.14	66.69±0.88	68.37±0.54	65.16±1.24	65.57±0.36	63.34±0.73	61.30±2.17	69.34±0.09
ContextPred	64.56±1.36	66.31±0.94	66.89±1.98	65.99±1.22	66.11±0.16	62.55±0.11	58.44±1.18	69.37±0.21
AttrMasking	64.35±1.23	64.32±0.78	67.72±1.16	65.72±1.31	65.49±0.52	62.35±0.58	53.34±4.77	68.61±0.16
L2P-GNN	66.48±1.59	69.89±1.63	69.15±1.86	70.13±0.95	66.58±0.28	65.84±0.37	62.24±1.89	70.79±0.17
MVRACE	**68.50**±1.21	**71.12**±1.54	**69.51**±1.47	**71.09**±1.10	**67.40**±0.53	**66.11**±0.33	**63.89**±1.34	**71.98**±0.41
Gain	8.35%	8.22%	1.91%	9.67%	8.39%	8.32%	7.14%	4.30%

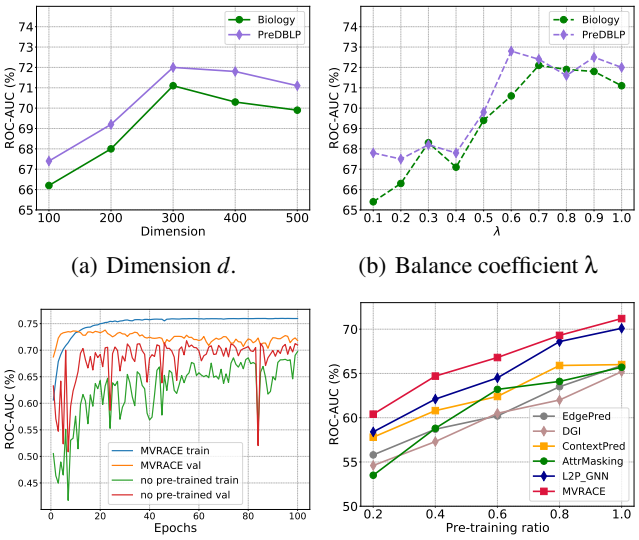
Table 2: Ablation study

Variants	Dataset	
	Biology	PreDBLP
MVRACE-node (random)	66.76±1.41	69.15±1.23
MVRACE-node	69.15±1.20	69.21±0.64
MVRACE-graph	68.42±1.21	69.51±0.45
MVRACE	71.09±1.10	71.98±0.41

global semantic information is necessary, while the lack of any of them will lead to a sharp decline in performance.

Parameter Analysis. We present the sensitivity analysis of two important hyper-parameters employed by MVRACE, namely dimension d and balance coefficient λ. We adjust the dimension $d \in \{100, 200, 300, 400, 500\}$ and analyze the effects of different dimensions. As Figure 3(a) shows, we observe that MVRACE achieves the optimal performance when the dimension is 300 and is generally stable around the optimal setting, indicating that MVRACE is robust w.r.t. the representation dimensions. Besides, we vary the balance coefficient λ from 0.1 to 1.0 with increments of 0.1. The results in Figure 3(b) shows that we should keep λ in [0.6-1.0] to obtain reliable performance.

Stability Analysis In this section, we analyze the improvement of fine-tuning on the biology dataset. As Figure 3(b) shows, we compare MVRACE to no pre-trained GIN. Whether in the training set or the validation set, we can find that the ROC-AOC curve of GIN without pre-training is more tortuous, and the final optimal value is much lower than our method. The above two observations further verify the effectiveness and stability of our strategy in GNN pre-training. Since MVRACE could utilize the rich semantic information at the node- and graph-level, we would like to know the performance of MVRACE with a different number of labeled training data by adjusting the training set ratio during pre-training. As Figure 3(d) shows that our MVRACE achieves high performance compared to other methods, validating its superiority in the scenario with few labeled data.



(a) Dimension d. (b) Balance coefficient λ

(c) Fine-tuning stability. (d) Different pre-training ratio.

Figure 3: The results of ablation study, parameter analysis and stability analysis.

Conclusion

In this paper, we introduce a multi-view contrastive encoding mechanism for graph neural network pre-training called MVRACE. In MVRACE, we consider the rich semantic relationships contained in the graph and define the key nodes and their local substructures by leveraging graph centrality. In addition, we use the graph augmentation strategy to create graph-level contrastive learning to enhance the ability of GNNs to distinguish different graph structures. We combine the node- and graph-level semantic information with a joint loss to integrate rich multi-view graph information. Extensive experiments demonstrate that MVRACE significantly outperforms the state of the art and can improve the stability of fine-tuning as well as pre-training sample utilization.

Acknowledgments

This work was supported by National Natural Science Foundation of China (62072463, 71531012), Research Seed Funds of School of Interdisciplinary Studies of Renmin University of China, National Social Science Foundation of China (18ZDA309), and Opening Project of State Key Laboratory of Digital Publishing Technology of Founder Group. Xun Liang is the corresponding author of this paper.

References

Bachman, P., Hjelm, R. D., & Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. In *Neurips* (pp. 15509–15519).

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *ICML 2009, montreal, quebec, canada, june 14-18, 2009* (Vol. 382, pp. 41–48). ACM.

Bergamini, E., Borassi, M., Crescenzi, P., Marino, A., & Meyerhenke, H. (2019). Computing top-k closeness centrality faster in unweighted graphs. *ACM Trans. Knowl. Discov. Data*, *13*(5), 53:1–53:40. Retrieved from https://doi.org/10.1145/3344719 doi: 10.1145/3344719

Che, Y. (2021). Comprehensive statistical analysis on estimated errors of averagine model for intact proteins. In *BIOSTEC* (pp. 136–139). SCITEPRESS.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. In *ICML* (Vol. 119, pp. 1597–1607). PMLR.

Chu, G., Wang, X., Shi, C., & Jiang, X. (2021). Cuco: Graph representation with curriculum contrastive learning. In Z. Zhou (Ed.), *IJCAI* (pp. 2300–2306).

Devlin, J., Chang, M. W., & L., K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Fey, M., & Lenssen, J. E. (2019). Fast graph representation learning with PyTorch Geometric. In *Iclr workshop*.

Gedela, G. S., Bobby, J., & Bhatt, N. (2022). Chemistry inspired molecular representation for property prediction. In *CODS-COMAD* (pp. 290–291).

Gilmer, J., Schoenholz, S. S., & Riley, P. F. (2017). Neural message passing for quantum chemistry. In *ICML*.

Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Neurips*.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. S., & Leskovec, J. (2020). Strategies for pre-training graph neural networks. In *ICLR*.

Hu, Z., Dong, Y., Wang, K., Chang, K., & Sun, Y. (2020). GPT-GNN: generative pre-training of graph neural networks. In *KDD* (pp. 1857–1867).

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2020). A survey on contrastive self-supervised learning.

Jing, M., Zhu, Y., Zang, T., & Wang, K. (2023). Contrastive self-supervised learning in recommender systems: A survey. *CoRR*, *abs/2303.09902*. Retrieved from https://doi.org/10.48550/arXiv.2303.09902 doi: 10.48550/arXiv.2303.09902

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR*.

Li, R., Wang, C., & Chang, K. C. (2014). User profiling in an ego network: co-profiling attributes and relationships. In *WWW* (pp. 819–830).

Lu, Y., Jiang, X., Fang, Y., & Shi, C. (2021). Learning to pre-train graph neural networks. In *AAAI* (pp. 4276–4284).

Mo, Y., Peng, L., Xu, J., Shi, X., & Zhu, X. (2022). Simple unsupervised graph representation learning. In *AAAI* (pp. 7797–7805).

Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., ... Tang, J. (2020). GCC: graph contrastive coding for graph neural network pre-training. In *KDD* (pp. 1150–1160).

Tang, H., Liang, X., Guo, Y., Zheng, X., & Wu, B. (2022). Graph fine-grained contrastive representation learning. In *IEEE international conference on acoustics, speech and signal processing, ICASSP 2022, virtual and singapore, 23-27 may 2022* (pp. 3478–3482). IEEE. Retrieved from https://doi.org/10.1109/ICASSP43922.2022.9746085 doi: 10.1109/ICASSP43922.2022.9746085

van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. , *abs/1807.03748*.

Velickovic, P., Fedus, W., & H., W. L. (2019). Deep graph infomax. In *ICLR*.

Xu, X., Deng, C., Xie, Y., & Ji, S. (2023). Group contrastive self-supervised learning on graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, *45*(3), 3169–3180. Retrieved from https://doi.org/10.1109/TPAMI.2022.3177295 doi: 10.1109/TPAMI.2022.3177295

Ying, R., & You, J. (2018). Hierarchical graph representation learning with differentiable pooling. *arXiv*.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph contrastive learning with augmentations. In *Neurips*.

Zhang, M., Cui, Z., & Neumann, M. (2018). An end-to-end deep learning architecture for graph classification. In *Aaai*.

Zheng, X., Liang, X., Wu, B., Wang, J., Guo, Y., Zhang, S., & Ma, Y. (2023). Modeling high-order relation to explore user intent with parallel collaboration views. In X. Wang et al. (Eds.), *Database systems for advanced applications - 28th international conference, DASFAA 2023, tianjin, china, april 17-20, 2023, proceedings, part II* (Vol. 13944, pp. 489–504). Springer. Retrieved from https://doi.org/10.1007/978-3-031-30672-3_33 doi: 10.1007/978-3-031-30672-3_33

Zitnik, M., Feldman, M. W., & Leskovec, J. (2019). Evolution of resilience in protein interactomes across the tree of life. *NAS*.