Multi-Scale Subgraph Contrastive Learning

Yanbei Liu[1] , **Yu Zhao**[2] , **Xiao Wang**[3*] , **Lei Geng**[1] and **Zhitao Xiao**[1*]
[1]School of Life Sciences, Tiangong University
[2]School of Electronics and Information Engineering, Tiangong University
[3]School of Software, Beihang University
{liuyanbei, zy0720, genglei, xiaozhitao}@tiangong.edu.cn, xiaowang@bupt.edu.cn

Abstract

Graph-level contrastive learning, aiming to learn the representations for each graph by contrasting two augmented graphs, has attracted considerable attention. Previous studies usually simply assume that a graph and its augmented graph as a positive pair, otherwise as a negative pair. However, it is well known that graph structure is always complex and multi-scale, which gives rise to a fundamental question: *after graph augmentation, will the previous assumption still hold in reality?* By an experimental analysis, we discover the semantic information of an augmented graph structure may be not consistent as original graph structure, and whether two augmented graphs are positive or negative pairs is highly related with the multi-scale structures. Based on this finding, we propose a multi-scale subgraph contrastive learning method which is able to characterize the fine-grained semantic information. Specifically, we generate global and local views at different scales based on subgraph sampling, and construct multiple contrastive relationships according to their semantic associations to provide richer self-supervised signals. Extensive experiments and parametric analysis on eight graph classification real-world datasets well demonstrate the effectiveness of the proposed method.

1 Introduction

Recently, graph neural networks (GNNs) have become a primary representation learning technique for dealing with many kinds of complex systems, ranging from the Internet and transportation graphs to biochemical interactions and social networks [Kavun *et al.*, 2012; Zhou *et al.*, 2020; Tang and Liu, 2010]. Many real world applications usually require the graph-level representations, such as predicting molecular properties in drugs [Chen *et al.*, 2019a], forecasting protein functions in biological networks [Jiang *et al.*, 2017], and predicting properties of circuits in circuit design [Zhang *et al.*, 2019]. Therefore, GNNs, which are able to learn the graph-level representations, play an important role in these real applications.

Most existing GNNs belong to the supervised learning paradigm, which requires a lot of labeled graphs. However, in many practical applications, collecting a large amount of labeled graph needs to consume a lot of resources. For example, in the field of chemistry, properties of chemical molecules are often obtained through density functional theory calculations, which require expensive computational resources [Jain *et al.*, 2016]. Therefore, Graph Contrastive Learning (GCL), one typical self-supervised paradigm, attracts considerable attention. The general framework for GCL is to maximize the consistency of augmented views from the same anchor graph (positive pair), while minimizing the consistency of views from different anchor graphs (negative pair) [You *et al.*, 2020; Suresh *et al.*, 2021]. Therefore, the key to graph contrastive learning is to ensure the semantic information matching between different augmented views, that is, views with similar semantics have similar representations.

There have been proposed many different graph augmentation strategies for GCL, e.g., node dropping [You *et al.*, 2020], edge perturbation [Suresh *et al.*, 2021], attribute masking [Jin *et al.*, 2021], subgraph sampling [You *et al.*, 2020]. However, one fundamental question is that *will the semantics of two augmented graphs still match in practice once the graph structure changes?* It is well known that the graph structure is very complex, and different substructures may have their functional implications [Newman, 2013]. For example, in a social network, different communities may indicate factions, interest groups; communities in a metabolic network might correspond to functional units, cycles, or circuits. Since the graph augmentation strategies essentially change the graph structures, it is hard to ensure the semantic information of different graph augmentations is matched.

Here, to provide more evidence for the above analysis, we perform an experiment to closely check the semantic relationship between different graph structures. Specifically, we select different substructures with different sizes on four real-world data, and then examine their semantic similarities (details can be seen in Section 2). The results clearly show that different graph structures have different semantics, and more importantly, the complex semantic information is positively related with the size of structure, i.e., larger subgraphs usually

*Corresponding author

present larger semantic similarities. This well indicates that we cannot simply assume the semantic information of augmented structures are already matched, while more complex relationships between different augmented structures need to be carefully considered for an effective GCL, and forcibly requiring two augmented graphs with different semantics to be matched may largely mislead the GCL model.

In this paper, we propose a novel Multi-Scale SubGraph Contrastive Learning method, which models the multi-scale semantic information in different augmented subgraphs. First, our experiment in Section 2 reveals that despite the graph structure is complex and different subgraphs have different semantics, their relationships can be generally divided into two facts: larger subgraphs, representing global view, usually have larger similarities, while smaller subgraphs, representing local view, usually have smaller similarities. These findings motivate us to employ different learning strategies based on the two facts. Specifically, we employ subgraph sampling to generate global and local views. We expect to pull the global representations of same anchor graph close to each other, while also encouraging similarity between global and local views. Meanwhile, we encourage the local representations to maintain a certain distance in the feature space. Finally, we introduce a regressor to measure the similarity between local views to avoid the unreliability of traditional distance measures in high-dimensional spaces.

Our contribution can be summarized as follows:

- We study the roles of multi-scale augmented graphs in GCL and verify that the basic requirement on augmented subgraphs of GCL may not always hold in practice, i.e., not all the augmented subgraphs are semantically matched.
- We propose a novel multi-scale subgraph contrastive learning method for graph-level representation learning. Our model is able to consider multi-scale information of graph data and formulate different learning strategies according to its semantic associations.
- We conduct comprehensive experiments on eight real-world datasets, and show that the proposed method achieves state-of-the-art performance on both unsupervised and semi-supervised graph classification tasks.

2 An Experimental Investigation

In this section, we employ data augmentation to obtain information at different scales in the graph dataset, and analyze its semantic similarity, aiming to obtain the semantic association between information at different scales. We take four molecule graph dataset (MUTAG, NCI1, DD, PROTEINS) in the TUDataset [Morris *et al.*, 2020] as examples. First, we trained a 5-layer graph isomorphic network [Xu *et al.*, 2019] with a hidden dimension of 32 on a single dataset via the Adam [Kingma and Ba, 2014] optimizer in a supervised training paradigm. Second, we augment the original graph with a subgraph sampling strategy based on random walks, and generate subgraphs at different scales by controlling the number of nodes in the augmented view [You *et al.*, 2020]. Since the starting point of the random walk is randomly selected, the generated subgraph is not fixed. We generate two

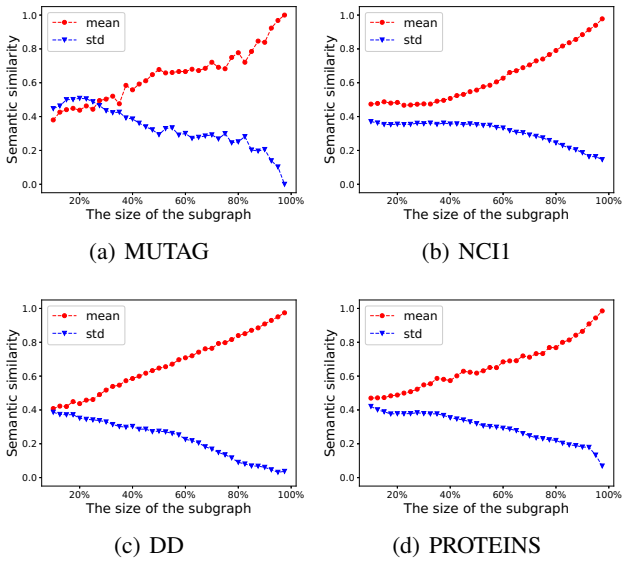


Figure 1: Semantic similarity of subgraphs of different sizes.

augmented views for each graph in the dataset, and treat the augmented views from the same original graph as subgraph pair. Then, we remove the classifier of the graph isomorphic network, input subgraph pair generated by each graph into the network to obtain its feature vectors, and calculate the cosine similarity between the two feature vectors. We do the same for every graph in the dataset. Finally, we take this similarity as the semantic similarity between views and compute the mean and variance of the semantic similarity between subgraph pairs generated from all original graphs in this dataset.

Figure 1 shows the semantic similarity between subgraph pairs at different scales. It can be seen that, firstly, as the size of subgraph increases, the mean value of the semantic similarity between the generated subgraph pairs increases continuously, which means the content that the larger subgraph describe is more similar on a semantic level. Secondly, the variance of the semantic similarity between subgraph pairs is decreasing, which means that the content changes of their descriptions are also decreasing. The above phenomenon implies that small-scale subgraphs always describe different content in the graph and vary greatly, while large-scale subgraphs describe more similar content, so the semantic association between view pairs at different scales is not uniform. Current GCL methods usually perform simple perturbations on the graph structure to obtain augmented views and consider them to have similar semantics. However, the semantic information of augmented views at different scales is different, which requires us to distinguish them at a finer granularity.

3 Methodology

Problem definition. Given an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ denotes the set of $|\mathcal{V}|$ nodes and $\mathcal{E} = e_{ij} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ indicates the adjacency matrix where each entry

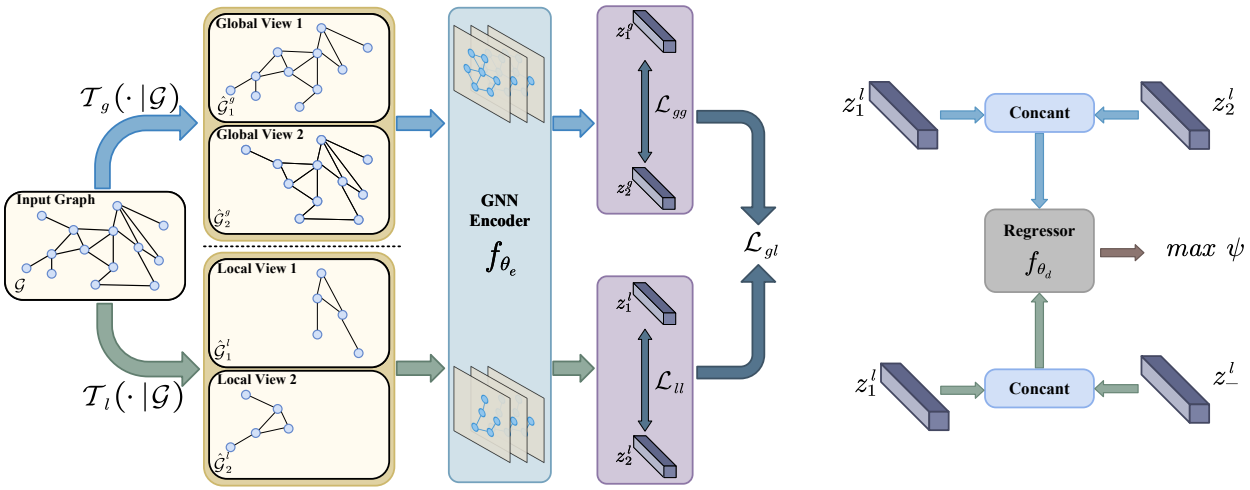


Figure 2: The overall architecture of MSSGCL (left). The original graph generates global views and local views pairs through random walks with controlled number of nodes, which are then fed into the encoder to obtain global and local representations. We maximize the similarity between global views and the similarity between global and local views by optimizing $\mathcal{L}_{gg}$ and $\mathcal{L}_{gl}$. The dissimilarity between local views is encouraged by optimizing the output of a learned similarity measure $\mathcal{L}_{ll}$. f_{θ_e} is the GNN-based encoder, which includes a backbone network followed by a multi-layer perceptron. f_{θ_d} (right) is a learnable regressor to measure the similarity between local views.

e_{ij} is the linkage relation between nodes i and j. For self-supervised graph representation learning, given a bunch of unlabeled graphs $G = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_M\}$, our goal is to learn a representation vector z_i for each graph through supervisory signals obtained from the data itself. The resulting representation vector z_i can be used in different types of downstream tasks, such as graph classification.

Overall framework. In this section, we will introduce our multi-scale subgraph based graph contrastive learning framework. As shown in Figure 2, it consists of three major components, including generation of multi-scale subgraphs, graph-level representation learning and multi-scale contrastive loss. Given a $\mathcal{G}$, we first apply graph augmentation techniques to obtain two sets of subgraphs with different sizes, namely global views and local views. Then, we utilize GNN-based encoders to learn their representations z^g and z^l, respectively. Subsequently, we perform the hierarchical local-global contrastive learning to optimize model parameters by the gradient descent.

3.1 Generation of Multi-Scale Subgraphs

In this section, we focus on graph-level data augmentation to obtain multi-scale subgraphs. Given a graph $\mathcal{G} \in \{\mathcal{G}_m : m \in M\}$, we define the augmented graph as $\hat{\mathcal{G}} \sim \mathcal{T}\left(\hat{\mathcal{G}} \,|\mathcal{G}\right)$, where $\mathcal{T}\left(\cdot\,|\mathcal{G}\right)$ is a predefined augmentation over the original graph, representing the human prior knowledge of graph data. Generally speaking, there are four common graph augmentation methods, namely node dropping, attribute masking, edge perturbation and subgraph sampling. Previous study [You *et al.*, 2020] has shown that, compared with attribute masking and edge perturbation, subgraph sampling can benefit downstream tasks across different categories of graph datasets. So we mainly employ subgraph sampling as data augmentation strategy. As mentioned above, there are large differences in semantic similarity between pairs of subgraphs with different scales. Large-scale subgraph pairs have high semantic similarity and small variance, while small-scale subgraph pairs have the opposite. Therefore, we can form local and global subgraph sampling strategies $\mathcal{T}_l\left(\cdot\,|\mathcal{G}\right)$ and $\mathcal{T}_g\left(\cdot\,|\mathcal{G}\right)$ by limiting the scale of generated views. Then we perform the two subgraph sampling strategies on the given graph $\mathcal{G}$, and obtain two global views $\left\{\hat{\mathcal{G}}_i^g\right\}_{i=1}^2$ and two local views $\left\{\hat{\mathcal{G}}_i^l\right\}_{i=1}^2$ for a single graph.

3.2 Graph-level Representation Learning

After acquiring the different scale augmentation views, we further learn their latent representations. Here we employ GNNs as the encoder to obtain node representations by iteratively aggregating neighbor information. Next, we will take the global view $\hat{\mathcal{G}}_i^g$ as an example to illustrate the representation learning process, which is exactly the same for the local view. Given an augmented graph $\hat{\mathcal{G}}_i^g$ with its feature matrix $X \in \mathbb{R}^{|\mathcal{V}| \times N}$, where $x_n = X\left[n,:\right]^{\mathrm{T}}$ is the N-dimensional feature vector of node v_n. In general, in a K-layer GNN, the node representations of k-layer can be formalized as:

$$\begin{aligned} a_n^{(k)} &= AGGREGATE^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}\left(n\right)\right\}\right), \\ h_n^{(k)} &= COMBINE^{(k)}\left(h_n^{(k-1)}, a_n^{(k)}\right), \end{aligned} \tag{1}$$

where $h_n^{(k)}$ is the representation of node v_n at k-th layer, with $h_n^{(0)} = x_n$. $\mathcal{N}\left(n\right)$ is the set of neighbors of node v_n. The $AGGREGATION^{(k)}\left(\cdot\right)$ and $COMBINE^{(k)}\left(\cdot\right)$ are important functional components of GNN. After the K-layer

propagation, the $READOUT$ function aggregates the feature vectors of all nodes to get the representations of the entire graph which can be used for downstream tasks:

$$f\left(\hat{\mathcal{G}}_i^g\right) = READOUT\left(\left\{h_n^{(K-1)} : v_n \in \mathcal{V}_i\right\}\right). \quad (2)$$

Similar to contrastive learning in the computer visual domain [Chen *et al.*, 2020], a non-linear transformation $g(\cdot)$ is used to map graph-level representations into the latent space to enhance the performance:

$$z_i^g = g\left(f\left(\hat{\mathcal{G}}_i^g\right)\right). \quad (3)$$

Through a similar procedure, we can obtain the representations of the local view as follows:

$$z_i^l = g\left(f\left(\hat{\mathcal{G}}_i^l\right)\right). \quad (4)$$

3.3 Multi-Scale Contrastive Loss

There are significant differences in the semantic similarity between subgraph pairs at different scales. Existing methods may suffer from some drawbacks in applying contrastive learning between subgraph pairs. For example, GraphCL directly pulls the representations distance between small-scale subgraphs. Direct application of such existing contrastive learning strategies creates noisy and potentially contradictory constraints that complicate the learning process and affect performance. To address this, we introduce two subgraph pairs of different sizes by limiting the number of nodes, namely global view and local view. After that, we optimize the global-to-global, local-to-global and local-to-local relationships, respectively. In the following, we denote l_s as a general contrastive loss, which is introducted as the noise-contrastive estimation loss [Oord *et al.*, 2018], and we take the global representation as an example to illustrate its optimization process:

$$l_s\left(z_1^g, z_2^g\right) = -\log \frac{\exp\left(z_1^g \cdot z_2^g/\tau\right)}{\exp\left(z_1^g \cdot z_2^g/\tau\right) + \sum \exp\left(z_1^g \cdot z_-^g/\tau\right)}, \quad (5)$$

where z_1^g and z_2^g are global representations from the same graph, z_-^g denotes the negative samples, which can be seen as global representations from other graphs in our architecture and τ is a temperature hyperparameter.

Global-to-global. Since the global view pair contains most of the content of the original graph, it owns similar semantic information. Our goal is to maximize the similarity of the global representations from the same original graph and minimize the similarity of the global representations from different original graphs. The global-to-global loss can be written as follows:

$$\mathcal{L}_{gg} = \mathbb{E}_{p(z^g)}\left[l_s\left(z_1^g, z_2^g\right)\right], \quad (6)$$

where $p\left(z^g\right)$ is the distribution of z^g.

Global-to-local. The global view owns a subgraph with large size, so it contains the content of the local view to a large extent, which ensures that the global view can share some semantic information with the local view. Therefore, we define a loss function that pulls to narrow the distance between local and global representations in the latent space and establish the connection between the local and global representations, which can be written as follows:

$$\mathcal{L}_{gl} = \mathbb{E}_{p(z^g, z^l)}\left[\sum_{i=1}^{2}\left(l_s\left(z_i^g, z_1^l\right) + l_s\left(z_i^g, z_2^l\right)\right)\right], \quad (7)$$

where $p\left(z^g, z^l\right)$ is the joint distribution of z^g and z^l.

Local-to-local. Two local views from the same original graph usually describe different contents with low semantic similarity. Thus, instead of similarity of local representations as most existing studies have done, we encourage their dissimilarity, making them farther apart in the representation space. Given a measure function l_d, we express maximizing the dissimilarity between local views as minimizing the loss:

$$\mathcal{L}_{ll} = \mathbb{E}_{p(z^l)}\left[l_d\left(z_1^l, z_2^l\right)\right]. \quad (8)$$

In principle, we can choose any similarity measurement method, such as cosine similarity, but the high dimension of feature space may lead to the learning of meaningless representations [Aggarwal *et al.*, 2001], and the semantic relation-

Algorithm 1 The training process of the MSSGCL

Input: Training set $D = \{\mathcal{G}_i\}_{i=1}^M$, batch size P, training epochs T, a GNN based encoder f, global and local augmentation distributions $\mathcal{T}_g$ and $\mathcal{T}_l$
Output:The pre-trained GNN encoder f_{θ_e}

```
Initialize GNN encoder and regressor parameters
while t < T do
  Sample graph minibatch B_G from D
  B_{Ĝ^g} ~ T_g(B_{Ĝ^g}|B_G), B_{Ĝ^l} ~ T_l(B_{Ĝ^l}|B_G)
  z^g, z^l ← Eqs.(1,2,3,4)
  for all i ∈ {1,...,P} do
    Get positive local view pair z_1^l, z_2^l
    Randomly choose a local view z_-^l
  end for
  Update θ_d to maximize the ψ(z_1^l, z_2^l) via Eq.(9)
  for all i ∈ {1,...,P} do
    Calculate similarity loss of global views
    Calculate similarity loss of global and local views
    Calculate similarity loss of local views
  end for
  Update f_{θ_e} to minimize the total loss by Eq.(11)
end while
```

Dataset	Category	Graph	Node	Edge
MUTAG	Molecules	188	17.93	19.79
NCI1	Molecules	4110	29.87	32.30
PROTEINS	Molecules	1113	39.06	72.82
DD	Molecules	1178	284.32	715.66
IMDB-B	Social Network	1000	19.77	95.63
COLLAB	Social Network	5000	74.49	2457.78
RDT-B	Social Network	2000	429.63	497.75
RDT-M5K	Social Network	5000	508.52	594.87

Table 1: Statistics of datasets.

Method	NCI1	PROTEINS	DD	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B	AVG.
WL	80.01±0.50	72.92±0.56	74.02±2.28	80.72±3.00	60.30±3.44	68.82±0.41	46.06±0.21	72.30±3.44	70.52
DGK	80.31±0.46	73.30±0.82	74.85±0.74	87.44±2.72	64.66±0.50	78.04±0.39	41.27±0.18	66.96±0.56	70.85
sub2vec	52.84±1.47	53.03±5.55	54.33 ±2.44	61.05±15.80	55.26±1.54	71.48±0.41	36.68±0.42	55.26±1.54	55.04
node2vec	54.89±1.61	57.49±3.57	74.77±0.51	72.63±10.20	54.57±0.37	72.76±0.92	31.09±0.14	38.60±2.30	57.10
graph2vec	73.22±1.81	73.30±2.05	70.32±2.32	83.15±9.25	71.10±0.54	75.48±1.03	47.86±0.26	71.10±0.54	70.69
InfoGraph	76.20±1.06	74.44±0.31	72.85±1.78	89.01±1.13	70.65±1.13	82.50±1.42	53.46±1.03	73.03±0.87	74.02
GraphCL	77.87±0.41	74.39±0.45	78.62±0.40	86.80±1.34	71.36±1.15	89.53±0.84	55.99±0.28	71.14±0.44	75.41
JOAO	78.07±0.47	74.55±0.41	77.32±0.54	87.35±1.02	69.50±0.36	85.29±1.35	55.74±0.63	70.21±3.08	74.75
JOAO V2	78.36±0.53	74.07±1.10	77.40±1.15	87.67±0.79	69.33±0.34	86.42±1.45	56.03±0.27	70.83±0.25	75.01
SimGRACE	79.12±0.44	75.35±0.09	77.44±1.11	89.01±1.31	71.72±0.82	89.51±0.89	55.91±0.34	71.30±0.77	76.17
MSSGCL	**81.45 ± 0.48**	**75.49 ± 0.70**	**79.73 ± 0.44**	**89.68 ± 0.57**	**73.48 ± 0.83**	**91.08±0.78**	**56.17±0.18**	**73.14±0.38**	**77.52**

Table 2: Comparison of classification accuracy with other baselines in unsupervised setting. AVG. denotes the average accuracy.

ship between different local view pairs varies greatly. Therefore, instead of using a traditional metric to push local views away from each other, we measure the similarity of local view pairs through a regressor with learnable parameters.

Specifically, we exploit the intuition that although local views from different graphs may contain the same semantic content, in general we still expect local views from the same graph to be more closely related to each other than that from different graphs. To implement this expectation, we utilize a learnable regressor $f_{\theta_d} := \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$ that gives a similarity measure between local views. The parameters of the regressor θ_d can be trained in conjunction with the parameters of the encoder. Therefore, we train the regressor by maximizing the cost function:

$$\psi\left(z_1^l, z_2^l\right) = \mathbb{E}_{p\left(z_1^l, z_2^l\right)}\left[f_{\theta_d}\left(z_1^l, z_2^l\right)\right] - \mathbb{E}_{p\left(z_1^l \otimes z_-^l\right)}\left[f_{\theta_d}\left(z_1^l, z_-^l\right)\right], \tag{9}$$

where z_-^l is the negative sample of z_1^l, i.e., the local representations from different graphs, which can be obtained by random sampling from the same batch. And $p\left(z_1^l \otimes z_-^l\right)$ is the product of two marginal distributions. After that, we take the trained regressor as our metric function:

$$\ell_d = f_{\theta_d'},\ s.t.\ \theta_d^{'} = arg\max_{\theta_d} \psi\left(z_1^l, z_2^l\right). \tag{10}$$

In general, we use the regressor to make local representations from the same graph more similar than those from different graphs. Then, we train the encoder to minimize the metric value between local representation pairs from the same graph to account for their low semantic similarity.

In summary, our model can be summarized as a bi-level iterative optimization problem and the final loss function can be written as follows:

$$\min \mathcal{L}_{gg} + \lambda_1 \mathcal{L}_{gl} + \lambda_2 \mathcal{L}_{ll}, \tag{11}$$

where λ_1 and λ_2 are hyper-parameters to balance different loss terms. The implementation details of our framework are provided in Algorithm 1.

4 Experiment

In this section, we compare the proposed method with other advanced models in unsupervised and semi-supervised learning settings to evaluate its performance. Furthermore, we perform ablation experiments to demonstrate the effectiveness of various components of the proposed method.

4.1 Setup

Datasets. We adopt the TUDataset benchmark [Morris *et al.*, 2020], which contains different types of graphs, i.e., molecules and social networks, whose details can be shown in Table 1.

Implementation details. In our framework, we set the global view size to be 80% of the whole and the local view size to be 20% of the whole for molecular graphs, and 90% of the global view size and 10% of the local view size for social networks. The measurement function between local views is composed of a 5-layer MLPs with batch normalization and RELU activation functions. Its output is fed into a Sigmoid function, which outputs a scalar to indicate the similarity between two local views.

4.2 Unsupervised Representation Learning

Experiments setting. We follow the work [Sun *et al.*, 2020] to evaluate the performance of the proposed method on unsupervised graph representation learning, where the model learns graph-level representations only through the supervision signals provided by the data itself without relying on labels. After that, a SVM classifier is used to evaluate the quality of the representations. In addition to the SOTA graph kernel based methods, WL [Shervashidze *et al.*, 2011], DGK [Yanardag and Vishwanathan, 2015], for example, we also compare the proposed method with other eight advanced graph self-supervised learning methods, including node2vec [Grover and Leskovec, 2016], sub2vec [Adhikari *et al.*, 2018], graph2vec [Narayanan *et al.*, 2017], Infograph [Sun *et al.*, 2020], GraphCL [You *et al.*, 2020], JOAO [You *et al.*, 2021] and SimGRACE [Xia *et al.*, 2022]. For our model, we adopt GIN as the encoder, and a sum pooling is used as the readout function. We use 10-fold cross validation accuracy to report classification performance. Experiments are repeated 5 times.

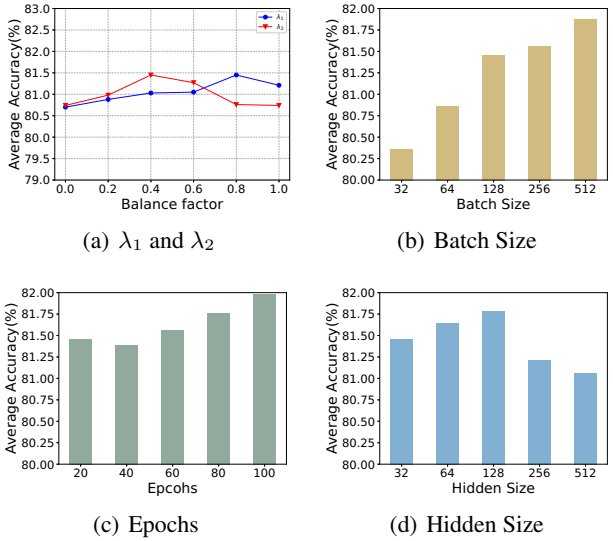
Results analysis. The results of the downstream graph classification task are shown in Table 2. Although graph kernel-based methods can perform well on a single dataset, they cannot be extended to all datasets. Similar to our method, GraphCL constructs comparison paths between small-scale subgraph pairs, but it ignores rich global information and cannot achieve better performance. However, MSSGCL can

LR.	Method	NCI1	PROTEINS	DD	COLLAB	RDT-B	RDT-M5K	AVG.
1%	No-pretrain	60.72 ± 0.45	-	-	57.46 ± 0.25	-	-	59.09
	Augmentations	60.49 ± 0.46	-	-	58.40 ± 0.97	-	-	59.45
	GAE	61.63 ± 0.84	-	-	63.20 ± 0.67	-	-	62.42
	Infomax	62.72 ± 0.65	-	-	61.70 ± 0.77	-	-	62.21
	ContextPred	61.21 ± 0.77	-	-	57.60 ± 2.07	-	-	59.41
	GraphCL	62.55 ± 0.86	-	-	64.57 ± 1.15	-	-	63.56
	JOAO	61.97 ± 0.72	-	-	63.71 ± 0.84	-	-	62.84
	JOAO V2	62.52 ± 1.16	-	-	64.51 ± 2.21	-	-	63.52
	SimGRACE	64.21 ± 0.65	-	-	64.28 ± 0.98	-	-	64.25
	MSSGCL	**64.73 ± 0.75**	-	-	**65.02 ± 0.78**	-	-	**64.88**
10%	No-pretrain	73.72 ± 0.24	70.40 ± 1.54	73.56 ± 0.41	73.71 ± 0.27	86.83 ± 0.27	51.33 ± 0.44	71.56
	Augmentations	73.59 ± 0.32	70.29 ± 0.64	74.30 ± 0.81	74.19 ± 0.13	87.74 ± 0.39	52.01 ± 0.20	72.02
	GAE	74.36 ± 0.24	70.51 ± 0.17	74.54 ± 0.68	75.09 ± 0.19	87.69 ± 0.40	33.58 ± 0.13	69.30
	Infomax	**74.86 ± 0.26**	72.27 ± 0.40	75.78 ± 0.34	73.76 ± 0.29	88.66 ± 0.95	53.61 ± 0.31	73.16
	ContextPred	73.00 ± 0.30	70.23 ± 0.63	74.66 ± 0.51	73.69 ± 0.37	84.76 ± 0.52	51.23 ± 0.84	71.26
	GraphCL	73.63 ± 0.25	74.17 ± 0.34	76.17 ± 1.37	74.23 ± 0.21	89.11 ± 0.19	52.55 ± 0.45	73.48
	JOAO	74.48 ± 0.27	72.13 ± 0.92	75.69 ± 0.67	75.30 ± 0.32	88.14 ± 0.25	52.83 ± 0.54	73.10
	JOAO V2	74.86 ± 0.39	73.31 ± 0.48	75.81 ± 0.73	75.53 ± 0.18	88.79 ± 0.65	52.71 ± 0.28	73.50
	SimGRACE	74.60 ± 0.41	74.03 ± 0.51	76.48 ± 0.52	74.74 ± 0.28	88.96 ± 0.62	53.94 ± 0.64	73.78
	MSSGCL	74.77 ± 0.31	**75.86 ± 0.52**	**78.99 ± 0.18**	**76.12 ± 0.13**	**90.58 ± 0.34**	**54.36 ± 0.24**	**75.11**

Table 3: Comparison of classification accuracy with other baselines in semi-supervised setting. AVG. denotes the average accuracy.

Method	NCI1	DD	COLLAB	RDT-B
MSSGCL	81.45±0.48	79.73±0.44	73.40±0.72	91.08±0.78
w/o global-global	80.27±0.51	78.62±0.49	71.92±1.10	88.87±2.42
w/o global-local	80.70±0.40	79.37±1.18	73.08±0.71	89.75±1.00
w/o local-local	80.74±0.51	79.45±0.40	72.86±0.54	89.83±1.41

Table 4: Ablation study on four benchmark datasets.



(a) λ_1 and λ_2 (b) Batch Size (c) Epochs (d) Hidden Size

Figure 3: Classification accuracies of MSSGCL on NCI1 under different hyper-parameters.

achieve good performance on all datasets, outperforming all other baseline models. This can be attributed to the fact that our method considers multi-scale views of the graph and combines multi-scale features according to the semantic relationships between views to form a favorable feature space.

4.3 Semi-supervised Representation Learning

Experiments setting. For semi-supervised setting, we pre-train a GNN in an unsupervised manner with all data, and then fine-tune the GNN with a certain percentage of labels on the same datasets. Since the pre-training and fine-tuning of graph-level tasks in semi-supervised learning are less studied in the past, we additionally introduce several network embedding methods: GAE [Kipf and Welling, 2016], local global representation consistency enforcement [Velickovic *et al.*, 2019] and ContextPred [Hu *et al.*, 2020]. The rest of the baselines also include SOTA graph self-supervised learning methods, such as GraphCL [You *et al.*, 2020], JOAO [You *et al.*, 2021] and SimGRACE [Xia *et al.*, 2022]. Following the settings in GraphCL, we employ 5-layer Residual Graph Convolutional Network (ResGCN) [Chen *et al.*, 2019b] with 128 hidden dimensions as our backbone network, and adopt 10-fold cross validation. Experiments are repeated 5 times.

Results analysis. For the semi-supervised graph classification task, results are shown in Tabel 3, where two subtasks are reported with label rates of 1% and 10%, respectively. For 1% label rate setting, MSSGCL outperforms all the baseline models. For 10% label rate setting, MSSGCL greatly outperforms the previous baselines and achieves the optimal performance on 6 out of 7 datasets. Compared with GraphCL, which only considers single size subgraph, MSSGCL achieves an average 2% improvement.

4.4 Ablation Study

In this section, we will study three contrastive relations, i.e., global-global, global-local, and local-local. We construct different variants by removing different loss terms to verify their effectiveness.

As can be seen from results in Table 4, when the model removes the global-global term, the performance drops significantly because the term contains rich global information. When the local-local contrastive relationship is added to the loss term, the model performance can be improved to a certain extent, which clearly shows that our regressor is effective. And when the model considers three contrast relationships at the same time, the model performance can reach the optimum.

4.5 Hyper-parameters Study

Balance factor for the loss. In this section, we investigate the sensitivity of hyper-parameters λ_1 and λ_2. Figure 3(a) shows the classification accuracy of MSSGCL with different values. With the increase of λ_1, the performance of the model has been continuously improved, but it tends to be saturated after a certain level, which shows that establishing the connection between local and global can help the model learn better representations. On the other hand, properly encouraging the dissimilarity of local representations can promote the performance of the model, but when λ_2 is too large, the performance of the model will decrease. We believe that this is because there is still a low semantic similarity between local view pairs, excessive dissimilarity can impair the quality of learned representations.

Batch size, hidden size and epochs. Figure 3(b) and Figure 3(c) show the performance of MSSGCL at different batch sizes and epochs. From results of two figures, it can be seen that larger batch size and training epochs lead to better performance, which is consistent with the findings of the work [Chen *et al.*, 2020]. The possible reason is that larger batch size provides more samples for comparison. Similarly, training longer time will generate more negative samples. Figure 3(d) shows the sensitivity of the hidden size. We can see that with the increasing of hidden dimension from 32 to 128, the performance gradually improves. As the hidden dimension continues to increase, the performance begins to degrade, which may be caused by the model overfitting.

5 Related Work

5.1 Graph Neural Network

In recent years, graph neural networks have emerged as a promising method for analyzing graph due to their powerful expressive power. They mainly follow the mechanism of message passing (or neighborhood aggregation) [Gilmer *et al.*, 2017]. Each node captures the attribute and structural information of neighbor nodes through message passing to update its own node representations, and then a shared linear transform is used to map the representations of nodes into a new feature space. After the iteration of k-layer, the representation vectors of nodes can capture the information of k-hop neighbors. Graph Convolutional Network (GCN) [Welling and Kipf, 2017] adopts the information of 1-hop neighbors to update node features, where the weight of each neighbors depends on the node degree. Graph Attention Network (GAT) [Veličković *et al.*, 2018] considers the weight difference of neighbors through attention mechanism. Graph Isomorphic Networks (GIN) [Xu *et al.*, 2019], inspired by the Weisfeiler-Lehman (WL) kernel, use simple summation operations and multilayer perceptrons (MLPs) to achieve the most powerful discriminative capabilities. These methods mainly focus on supervised learning, which means that a large amount of labeled graph is required. However, obtaining manually annotated labels is expensive in terms of time and labor, so our method mainly focuses on unsupervised/self-supervised learning.

5.2 Graph Contrastive Learning

Contrastive learning has been widely used in the field of computer vision with promising results. Affected by this, some recent studies have begun to introduce contrastive learning into the field of graph learning. The basic idea is to promote the embedding of augmentation views generated from the same instance to be closer, while those from different instances are opposite. Encoders trained in this way can be used for downstream tasks. DGI [Velickovic *et al.*, 2019] opens a precedent for graph contrastive learning, which treats node and graph-level representations as postive pairs and maximize their mutual information. MVGRL [Hassani and Khasahmadi, 2020] further improves model performance by extending DGI to multiple views and cross-contrasting between them through graph diffusion. Sub-Con [Jiao *et al.*, 2020] learns node representations by sampling the subgraph and taking the central node and subgraph as a postive sample pair. GraphCL[You *et al.*, 2020] proposes four data augmentation methods for graphs, and proves subgraph sampling is an augmentation method beneficial to different types of datasets. Cuco [Chu *et al.*, 2021] combines curriculum learning with contrastive learning, and proposes a scoring and a pacing functions to automatically select negative samples during training. Although the above methods have made good progress in GCL, they ignore the semantic association between augmented views, while our method can adopt different learning strategies according to the semantic information of views at different scales.

6 Conclusion

In this paper, we investigate the semantic association among subgraphs at different scales, and propose a novel multi-scale subgraph contrastive learning method. Based on the semantic association, we define two different types of subgraph, i.e., global view and local view. We construct a variety of contrastive relations between views, and implement different learning strategies to achieve mutual matching of semantic information between augmented views. We conduct graph classification experiments on eight real-world datasets, and the experimental results demonstrate that the proposed method can outperform the state-of-the-arts in unsupervised and semi-supervised learning.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No.62172052, 61901297), the Special Foundation for Beijing Tianjin Hebei Basic Research Cooperation (J210008, 21JCZXJC00170, H2021202008), and The Open Project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University (No. MMC20210).

References

[Adhikari *et al.*, 2018] Bijaya Adhikari, Yao Zhang, Naren Ramakrishnan, and B Aditya Prakash. Sub2vec: Feature learning for subgraphs. In *PAKDD*, pages 170–182, 2018.

[Aggarwal *et al.*, 2001] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, pages 420–434, 2001.

[Chen *et al.*, 2019a] Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.

[Chen *et al.*, 2019b] Ting Chen, Song Bian, and Yizhou Sun. Are powerful graph neural nets necessary? a dissection on graph classification. *arXiv preprint arXiv:1905.04579*, 2019.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[Chu *et al.*, 2021] Guanyi Chu, Xiao Wang, Chuan Shi, and Xunqiang Jiang. Cuco: Graph representation with curriculum contrastive learning. In *IJCAI*, pages 2300–2306, 2021.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126, 2020.

[Hu *et al.*, 2020] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *ICLR*, 2020.

[Jain *et al.*, 2016] Anubhav Jain, Yongwoo Shin, and Kristin A Persson. Computational predictions of energy materials using density functional theory. *Nature Reviews Materials*, 1(1):1–13, 2016.

[Jiang *et al.*, 2017] Biaobin Jiang, Kyle Kloster, David F Gleich, and Michael Gribskov. Aptrank: an adaptive pagerank model for protein function prediction on bi-relational graphs. *Bioinformatics*, 33(12):1829–1836, 2017.

[Jiao *et al.*, 2020] Yizhu Jiao, Yun Xiong, Jiawei Zhang, Yao Zhang, Tianqi Zhang, and Yangyong Zhu. Sub-graph contrast for scalable self-supervised graph representation learning. In *ICDM*, pages 222–231, 2020.

[Jin *et al.*, 2021] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *IJCAI*, 2021.

[Kavun *et al.*, 2012] Sergii V Kavun, Irina V Mykhalchuk, Nataliya I Kalashnykova, and Oleksandr G Zyma. A method of internet-analysis by the tools of graph theory. In *Intelligent Decision Technologies*, pages 35–44, 2012.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *NeurIPS*, 2016.

[Morris *et al.*, 2020] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv:2007.08663*, 2020.

[Narayanan *et al.*, 2017] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv:1707.05005*, 2017.

[Newman, 2013] Mark EJ Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, 2013.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Shervashidze *et al.*, 2011] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[Sun *et al.*, 2020] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2020.

[Suresh *et al.*, 2021] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*, pages 15920–15933, 2021.

[Tang and Liu, 2010] Lei Tang and Huan Liu. Graph mining applications to social network analysis. *Managing and Mining Graph Data*, pages 487–513, 2010.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR*, 2019.

[Welling and Kipf, 2017] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Xia *et al.*, 2022] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *WWW*, pages 1070–1079, 2022.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Yanardag and Vishwanathan, 2015] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *KDD*, pages 1365–1374, 2015.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, pages 5812–5823, 2020.

[You *et al.*, 2021] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *ICML*, pages 12121–12132, 2021.

[Zhang *et al.*, 2019] Guo Zhang, Hao He, and Dina Katabi. Circuit-gnn: Graph neural networks for distributed circuit design. In *ICML*, pages 7364–7373, 2019.

[Zhou *et al.*, 2020] Fan Zhou, Qing Yang, Ting Zhong, Dajiang Chen, and Ning Zhang. Variational graph neural networks for road traffic prediction in intelligent transportation systems. *IEEE Transactions on Industrial Informatics*, 17(4):2802–2812, 2020.