SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2025, Vol. 68, Iss. 1, 112101:1–112101:14
https://doi.org/10.1007/s11432-022-3952-3

GPS: graph contrastive learning via multi-scale augmented views from adversarial pooling

Wei JU[1], Yiyang GU[1], Zhengyang MAO[1], Ziyue QIAO[2], Yifang QIN[1], Xiao LUO[3*], Hui XIONG[2] & Ming ZHANG[1*]

[1]*School of Computer Science, National Key Laboratory for Multimedia Information Processing, Peking University, Beijing 100871, China;*
[2]*Artificial Intelligence Thrust, The Hong Kong University of Science and Technology, Guangzhou 511453, China;*
[3]*Department of Computer Science, University of California, Los Angeles CA 90095, USA*

Received 3 December 2022/Revised 2 August 2023/Accepted 9 November 2023/Published online 24 December 2024

Abstract Self-supervised graph representation learning has recently shown considerable promise in a range of fields, including bioinformatics and social networks. A large number of graph contrastive learning approaches have shown promising performance for representation learning on graphs, which train models by maximizing agreement between original graphs and their augmented views (i.e., positive views). Unfortunately, these methods usually involve pre-defined augmentation strategies based on the knowledge of human experts. Moreover, these strategies may fail to generate challenging positive views to provide sufficient supervision signals. In this paper, we present a novel approach named graph pooling contrast (GPS) to address these issues. Motivated by the fact that graph pooling can adaptively coarsen the graph with the removal of redundancy, we rethink graph pooling and leverage it to automatically generate multi-scale positive views with varying emphasis on providing challenging positives and preserving semantics, i.e., strongly-augmented view and weakly-augmented view. Then, we incorporate both views into a joint contrastive learning framework with similarity learning and consistency learning, where our pooling module is adversarially trained with respect to the encoder for adversarial robustness. Experiments on twelve datasets on both graph classification and transfer learning tasks verify the superiority of the proposed method over its counterparts.

Keywords graph representation learning, graph neural networks, graph contrastive learning, graph augmentations, graph pooling

Citation Ju W, Gu Y Y, Mao Z Y, et al. GPS: graph contrastive learning via multi-scale augmented views from adversarial pooling. Sci China Inf Sci, 2025, 68(1): 112101, https://doi.org/10.1007/s11432-022-3952-3

1 Introduction

With the prevalence of graph-structured data [1–5], it is vital to develop effective representations of whole graphs for various real-world applications such as protein/molecular property prediction [6,7], drug discovery [8–10], traffic forecasting [11–13], and recommender systems [14–16]. Graph neural networks (GNNs) have recently emerged as powerful tools for learning graph representations in fully-supervised or semi-supervised scenarios [17–20]. However, obtaining a large number of label annotations is often challenging, particularly in highly specialized domains such as biochemistry [9]. While the number of labeled graphs may be restricted, unlabeled graphs are quite straightforward to acquire in practice. Hence, plenty of efforts have been directed towards self-supervised graph representation learning, which explores unlabeled graphs to alleviate the dependency on massive label annotations.

Motivated by the recent progress in computer vision [21, 22] and recommender systems [23–25], recent researches attempt to integrate contrastive learning to representation learning in the graph machine learning [26–30]. The primary principle underlying graph contrastive learning (GCL) methods is to maximize the mutual information (MI) [31] between the input graph and its representation. Specifically, these approaches anticipate that a graph has a representation that is similar to its own augmented view and distinct from other graphs. Thus, these methods can provide discriminative graph-level representations, which are beneficial for a variety of downstream applications.

* Corresponding author (email: xiaoluo@cs.ucla.edu, mzhang_cs@pku.edu.cn)

Despite their superior performance, existing self-supervised methods rely on handcrafted augmentation strategies to provide positive views for comparison. Common strategies include node dropping, edge perturbation, attribute masking, graph diffusion [29] and subgraph [28]. These handcrafted strategies, however, have the following drawbacks. First, current methods are inconvenient to apply to different datasets since they require expert knowledge to select appropriate strategies for preserving semantics. Edge perturbation, for example, has been empirically demonstrated to benefit social networks but harm certain biological molecules, while node dropping and subgraph are typically beneficial across datasets [28]. Moreover, when dealing with datasets from unknown domains, we may require extensive trials to determine the appropriate augmentation strategies, making it inefficient for practical applications. Second, these pre-defined strategies could fall short of generating challenging positive views to provide sufficient supervision signals. In particular, we expect that augmented samples can fully discard redundant information from different perspectives, implying the representations of challenging positives are far from those of the original graphs. If the augmented views are close to the original samples, the representation collapse may even occur, resulting in trivial outputs.

Graph pooling is another central area of research for graph representation learning, which originated from the traditional convolutional neural network (CNN) for extracting information efficiently. Graph pooling can be divided into TopK-based methods [32, 33] and cluster-based methods [17, 34], which can effectively learn to reduce the redundant information while preserving semantics. Specifically, they either select important nodes from the original graph or group nodes into clusters and coarsen the graph. To sum up, graph pooling has the potential to improve GCL since it can adaptively remove the redundancy of the graph from different perspectives. However, existing researches typically study different pooling manners in supervised scenarios [35]. It is still unclear how to integrate graph pooling methods into GCL by automatically providing effective augmented views.

In this paper, we propose a novel approach named graph pooling contrast (GPS) by leveraging learnable graph pooling to generate positive views for effective contrastive learning. Apart from introducing a graph encoder for producing effective graph representations, we involve two graph pooling modules to generate positive views with different emphases on providing challenging positives and preserving semantics, i.e., strongly-augmented view and weakly-augmented view. On the one hand, we directly maximize the similarity of a graph and its weakly-augmented view in a hard manner. On the other hand, we explore the semantics involved in strongly-augmented views by consistency learning between the similarity of two views in a soft manner. Further, our two pooling modules are adversarially trained with the graph encoder for adversarial robustness and efficiency. Finally, we conduct extensive experiments to empirically validate the effectiveness of our proposed approach GPS, validating the superiority over state-of-the-art baselines on graph classification and transfer learning tasks.

2 Related work

Representation learning aims to learn effective representations of graph topology and node attributes, which can be categorized into matrix factorization-based, random walk-based, and neural network-based. Matrix factorization-based methods [36, 37] directly adopt classic techniques for dimension reduction. Random walk-based methods such as DeepWalk [38] and Node2Vec [39] model probabilities of co-occurrence pairs using noise-contrastive estimation [40]. Neural network-based methods, especially GNNs, have attracted increasing interest in recent years. With the development of representation learning, various GNNs [2–4, 41–45] have achieved state-of-the-art performance. Generally, a GNN shares a common spirit: extracting local structural features by message passing [46, 47] where nodes iteratively aggregate messages from neighboring nodes through edges. With GPS, besides learning effective graph representations derived from GNNs, we also benefit from graph pooling to automatically generate multi-scale view augmentations.

Contrastive learning on graphs has become a dominant component in self-supervised learning on graphs. Inspired by previous success in visual representation learning, some recent studies [26, 28, 48–51] marry the power of contrastive learning and GNNs, and have shown competitive performance. The key idea of these methods is to maximize the agreement between semantics-invariant transformations of the graphs. GCA [49] generates different views by incorporating various priors for graph topology and semantics. GraphCL [28] explores the augmentations from the aspects of node dropping, edge perturbation, attribute masking, and subgraph sampling. However, existing studies typically involve inflexible and pre-defined

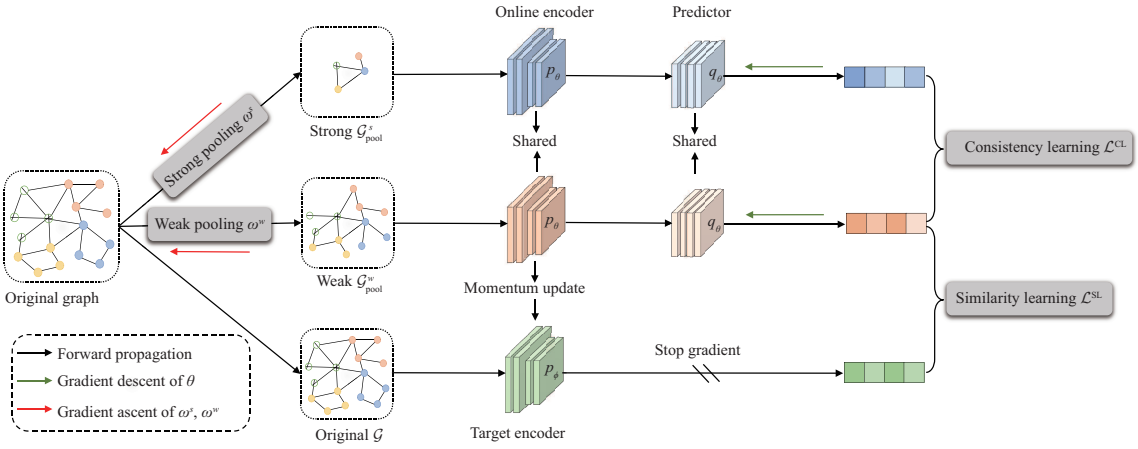

Figure 1 (Color online) Illustration of the proposed framework GPS. We first generate two positive views via our two pooling modules. Then, the two augmented views are fed into the online network while the original graph is fed into the target network. Our contrastive learning framework captures similarity learning and consistency learning, where the graph pooling modules are adversarially trained with respect to the encoder.

augmentation strategies based on the knowledge of human experts, while our approach leverages learnable multi-scale graph pooling to generate positive views for contrastive learning.

Graph pooling is a central component of a range of GNN architectures [17, 32–34, 52]. It is originated from the traditional CNNs and reduces the number of parameters of CNNs by downsampling and summarizing from the representations, which makes the training process highly efficient. Similarly, some studies try to generalize pooling operations to graphs for extracting effective information of the whole graph hierarchically, and these graph pooling methods can be boiled down to two categories: TopK-based pooling and cluster-based pooling.

TopK-based pooling aims to select the most important nodes from the original graph and use these nodes to construct a new graph. SAGPool [32] leverages the self-attention mechanism [53] to select nodes by considering both node features and graph topology. In gPool [33], the nodes are selected via mapping the node feature into the importance scores. They share a similar idea to learn a sorting vector based on node representations using GNNs, which indicates the importance of different nodes.

Cluster-based pooling tries to utilize an assignment matrix to achieve pooling by assigning nodes to different clusters and coarsen the graph hierarchically. DiffPool [17] treats graph pooling as a node clustering problem and introduces a differentiable pooling module to decide the pooled graph topology. ASAP [34] learns a sparse soft cluster assignment for nodes to cluster local subgraphs hierarchically for effectively capturing the graph substructure.

Our framework rethinks the powerful capability of graph pooling and makes the first attempt to leverage learnable graph pooling to derive augmented views in an adversarial manner.

3 Methodology

In this section, we propose GPS, a novel GCL method, and the overall architecture is shown in Figure 1. The positive views play a critical role in GCL and deserve a careful design. Previous methods usually generate positive views by handcrafted augmentation strategies, which require expert knowledge and fail to generate challenging positives for providing sufficient supervision signals. To address these problems, we leverage graph pooling techniques to construct positive views with a varying focus on challenging positives and semantic preservation, i.e., strongly-augmented view and weakly-augmented view, respectively. We also develop a unified GCL framework including similarity learning and consistency learning to make the best of two views, with our graph pooling modules being adversarially trained with respect to the graph encoder. Next, we will go into the specific components of our proposed GPS.

3.1 Preliminaries and notations

Definition 1 (Graph)**.** Define a graph as $\mathcal{G} = (V, E, X, A)$, where V represents the node set and E represents the edge set. $X \in \mathbb{R}^{|V| \times d_0}$ is the node feature matrix (i.e., the v-th row of X is the feature

vector x_v of v-th node) and $A \in \mathbb{R}^{|V|\times|V|}$ denotes the adjacent matrix of the graph.

Definition 2 (Unsupervised graph representation learning)**.** Given a set of unlabeled graphs $\mathcal{S} = \{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$, the primary objective is to develop a graph encoder that can generate an embedding vector $z_m \in \mathbb{R}^d$ for each graph $\mathcal{G}_m$, without relying on any label information. These learned graph embeddings $\{z_1, \ldots z_M\}$ will be applied for downstream tasks such as graph classification.

3.2 GNN-based encoder

We mainly utilize GNNs as our graph encoder due to their superior performance. GNNs typically follow the message-passing scheme to encode the structural and attributive information into node representations [46]. In particular, the propagation of the k-th layer of a K-layer GNN is described as follows:

$$h_v^{(k)} = \mathrm{COM}_\theta^{(k)}\left(h_v^{(k-1)}, \mathrm{AGG}_\theta^{(k)}\left(\left\{h_u^{(k-1)}\right\}_{u\in\mathcal{S}(v)}\right)\right), \tag{1}$$

where $h_v^{(k)}$ represents the embedding of node v at layer k, and $\mathcal{S}(v)$ is the neighbors of v. $\mathrm{AGG}_\theta^{(k)}$ is a function that aggregates information from neighbors, $\mathrm{COM}_\theta^{(k)}$ is a function that updates node features by combining the neighbor features and the feature of the node itself. Finally, the graph-level representation $g_\theta(\mathcal{G})$ is learned from node-level representations through an READOUT function, calculated as

$$g_\theta(\mathcal{G}) = \mathrm{READOUT}\left(\left\{h_v^{(K)}\right\}_{v\in V}\right), \tag{2}$$

where READOUT could be a straightforward permutation invariant approach such as averaging or a more well-designed graph-level pooling function like connected layers [46].

3.3 Graph pooling module

Different from previous methods which introduce pre-defined augmentations to generate positive views, we leverage learnable graph pooling to generate augmented views adaptively and automatically. In formulation, we generate positive views as follows:

$$\mathcal{G}_{\mathrm{pool}} = \mathrm{Pool}(\mathcal{G}, \rho), \tag{3}$$

where ρ denotes the ratio of nodes to be kept. There are several advanced graph pooling methods to construct $\mathrm{Pool}(\cdot, \rho)$, which can be divided into two categories as shown in Figure 2. Next, we introduce the details of these two categories in our framework respectively.

TopK-based pooling. In TopK-based pooling methods [32,33], attention mechanisms are typically adopted for adaptively selecting the nodes to be kept. In our implementation, we involve a graph encoder to generate self-attention scores $Z \in \mathbb{R}^{|V|\times 1}$ for all nodes. Then, we select the top $\lceil\rho|V|\rceil$ nodes based on the value of Z to generate an index set idx. The calculation considers both topological information and node attributes. Finally, the pooled graph G_{pool} are denoted as follows:

$$X_{\mathrm{pool}} = X_{\mathrm{idx},:} \odot Z_{\mathrm{idx}}, A_{\mathrm{pool}} = A_{\mathrm{idx,idx}}, \tag{4}$$

where $X_{\mathrm{idx},:}$ denotes the node-wise indexed feature matrix, $\odot$ denotes the broadcasted element-wise product and $A_{\mathrm{idx,idx}}$ denotes the row-wise and column-wise indexed adjacency matrix. The pooled vertex and edge set can be inferred from X and A.

Cluster-based pooling. Cluster-based methods [17,34] leverage graph clustering to coarsen the input graph. In our framework, we reutilize this idea and generate a cluster assignment matrix $S \in R^{|V|\times\lceil\rho|V|\rceil}$, where each row corresponds to one node while each column corresponds to one cluster. Formally, the pooled graph $\mathcal{G}_{\mathrm{pool}}$ can be denoted as follows:

$$X_{\mathrm{pool}} = S^{\mathrm{T}}X, A_{\mathrm{pool}} = S^{\mathrm{T}}AS, \tag{5}$$

where $X_{i,:}$ denotes embedding of the i-th cluster and A_{ij} denotes the connectivity strength between cluster i and cluster j. We generate the cluster assignment in an adaptive manner. Following [17], we generate S by another learnable GNN with a softmax activation function.

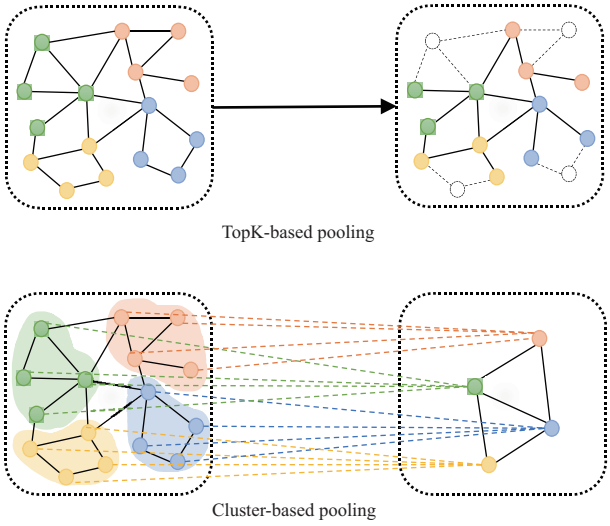

Figure 2 (Color online) Illustration of the graph pooling methods.

3.4 Contrastive learning framework

In the contrastive learning framework, a critical issue is how to generate positive views for input graphs. On the one hand, we need to generate augmented views with the most removal of redundant information. Hence, their representations should be far from these input graphs for generating challenging views and providing sufficient supervision signals for contrastive tasks, which could prevent representation collapse during optimization. On the other hand, augmented graphs should preserve crucial semantic information. Since there is a trade-off between challenging positives and semantic preserving, we generate a strongly-augmented view and a weakly-augmented view for different emphases. Formally, we introduce two different ratios $\rho_1 > \rho_2$ for two augmented views $\mathcal{G}^w_{\text{pool}} = \text{Pool}(\mathcal{G}, \rho_1)$ and $\mathcal{G}^s_{\text{pool}} = \text{Pool}(\mathcal{G}, \rho_2)$. Then, we leverage different patterns to explore information from two views.

Motivation for introducing strongly-augmented view and weakly-augmented view. Motivated by [54], we introduce two different ratios ρ for two augmented views via graph pooling. We encourage to capture different semantic information from the two complementary views, and expect the patterns embedded in strong-augmented views could contribute to contrastive learning by enhancing the generalizability of learned representations. To the best of our knowledge, this could be the first work to introduce weak and strong augmentations into the graph domains.

Similarity learning for weakly-augmented views. Our weakly-augmented views focus on preserving semantic information, and thus we propose a contrastive task in a hard manner. Previous approaches tend to bring different views of the same instance closer while pushing views of different samples further away [21, 22]. In comparison, the latest contrastive learning method BYOL [55] relies only on positive views and achieves superior performance. Inspired by this, we introduce an online encoder p_θ and a target encoder p_ϕ sharing the same architecture. Moreover, an additional predictor q_θ is applied to the online network, which implies an asymmetric architecture. Then, we feed the original graph $\mathcal{G}$ and weakly augmented graph $\mathcal{G}^w_{\text{pool}}$ into the target encoder and online encoder respectively, producing the representations $z = p_\phi(\mathcal{G})$ and $h^w = q_\theta(p_\theta(\mathcal{G}^w_{\text{pool}}))$. We minimize the cosine distance of two representations and the total loss in a batch $\mathcal{B}$ ($|\mathcal{B}| = B$) is

$$\mathcal{L}^{\text{SL}} = \frac{1}{B}\sum_{\mathcal{G}\in\mathcal{B}} 1 - \frac{z \cdot h^w}{||z||_2||h^w||_2}. \tag{6}$$

Consistency learning for strongly-augmented views. Our strongly-augmented views are aggressive since strong augmentation could distort topological patterns and attributes. Hence, directly doing contrastive learning, which employs a "hard" manner to achieve alignment from two views may lead to sub-optimal results. Nevertheless, strongly-augmented views can still provide some useful clues such as important motifs or subgraphs. To make the best of these clues, we develop a novel consistency learning (i.e., distributional divergence minimization) to achieve semantics consistency in a "soft" way by considering the relation of each point to samples in the same batch. Formally, after obtaining representations

of strongly-augmented graphs, i.e., $h^s = q_\theta(p_\theta(\mathcal{G}^s_{\text{pool}}))$, the similarity distribution of strongly-augmented views can be calculated by comparison with other graphs in a mini-batch as

$$\mu^b = \frac{\exp\left(\cos\left(h^s, z_b\right)/\tau\right)}{\sum_{\mathcal{G}_{b'}\in\mathcal{B}} \exp\left(\cos\left(h^s, z_{b'}\right)/\tau\right)}, \tag{7}$$

where z_b denotes the b-th representation in the mini-batch, τ is a temperature parameter set to be 0.5 as in [56] and $\cos(\cdot,\cdot)$ denotes the cosine similarity. In a similar way, the distribution of weakly-augmented graphs can be written as follows:

$$\nu^b = \frac{\exp\left(\cos\left(h^w, z_b\right)/\tau\right)}{\sum_{\mathcal{G}_{b'}\in\mathcal{B}} \exp\left(\cos\left(h^w, z_{b'}\right)/\tau\right)}. \tag{8}$$

Instead of hard similarity learning, we encourage the consistency between two distributions $\mu = [\mu^1, \ldots, \mu^B]$and $v = [\nu^1, \ldots, \nu^B]$ using Kullback-Leibler (KL) divergence. In formulation, the consistency learning loss is written as

$$\mathcal{L}^{\text{CL}} = \frac{1}{B}\sum_{\mathcal{G}\in\mathcal{B}} \frac{1}{2}(D_{\text{KL}}(\mu||\nu) + D_{\text{KL}}(\nu||\mu)), \tag{9}$$

where $D_{\text{KL}}(\cdot||\cdot)$ denotes the KL divergence of two distributions. Instead of directly enforcing view h^s close to z, we propose a soft contrastive task to keep the similarity structure consistent. In this way, we explore information in strongly-augmented views while alleviating the impacts of semantic loss.

Adversarial learning for robustness. Adversarial training has shown great success in improving the model robustness [57,58]. In this inspirit, we leverage adversarial learning to train the graph pooling module for generating effective positive views, aiming to produce augmented graphs that are distinct from the original ones while preserving their semantic information. This way maximally enhances the optimization of contrastive learning, facilitating the learning of discriminative graph representations. Specifically, the graph pooling module is trained against the graph encoder module in an adversarial manner. The adversarial objective for weakly-augmented views is formulated in a minimax form as

$$\min_{\theta}\max_{\omega^w} \mathcal{L}^{\text{SL}}(\theta, \omega^w), \tag{10}$$

where ω^w denotes the parameters in the graph pooling module for weak augmentations. From (10), we can observe that the graph encoder and the graph pooling module are two mutually interacted. On the one hand, the graph pooling module is trained to generate complex and robust views for effective representations. On the other hand, the graph encoder is optimized to continuously enhance the discrimination ability by minimizing the distance between input and its challenging and robust positive views. Unfortunately, directly minimizing the objective function as in (10) is nontrivial to find a saddle point solution. Following the optimization scheme in adversarial networks [59], we employ a pair of gradient descent and gradient ascent applied to update parameters in the graph encoder and graph pooling module, respectively. Formally, the updating process can be formulated as

$$\begin{cases} \omega^w \longleftarrow \omega^w + \eta\dfrac{\partial\mathcal{L}^{\text{SL}}(\theta, \omega^w)}{\partial\omega^w}, \\ \theta \longleftarrow \theta - \eta\dfrac{\partial\mathcal{L}^{\text{SL}}(\theta, \omega^w)}{\partial\theta}, \end{cases} \tag{11}$$

where η denotes the learning rate. As for strongly-augmented views, we leverage a consistency learning objective instead of a similarity learning objective to train the graph pooling module, since we seek to release the bias bought by weakly-augmented views. As a result, the optimization scheme is defined as

$$\begin{cases} \omega^s \longleftarrow \omega^s + \eta\dfrac{\partial\mathcal{L}^{\text{CL}}(\theta, \omega^s)}{\partial\omega^s}, \\ \theta \longleftarrow \theta - \eta\dfrac{\partial\mathcal{L}^{\text{CL}}(\theta, \omega^w, \omega^s)}{\partial\theta}, \end{cases} \tag{12}$$

Algorithm 1 Training procedure of GPS.

Input: Unlabeled data $\{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$, encoder parameter θ, momentum parameter ϕ, graph pooling module ω^w and ω^s.
Output: Momentum graph encoder g_ϕ.

```
Initialize θ, φ, ω^w and ω^s.
while not convergence do
    Sample B graphs for B;
    Generate G^w_pool and G^s_pool for G ∈ S;
    Calculate the similarity learning loss by Eq. (6);
    Calculate the consistency learning loss by Eq. (9);
    Update θ, ω^w and ω^s by Eq. (13);
    Update φ by momentum update in Eq. (14).
end while
```

where ω^s denotes the parameters in the graph pooling module for strong augmentations. The updated rules in (11) and (12) are summarized in a mini-batch for back-propagation updating as

$$\begin{cases} \omega^w \longleftarrow \omega^w + \eta \dfrac{\partial \mathcal{L}^{\mathrm{SL}}(\theta, \omega^w)}{\partial \omega^w}; \\ \omega^s \longleftarrow \omega^s + \eta \dfrac{\partial \mathcal{L}^{\mathrm{CL}}(\theta, \omega^s)}{\partial \omega^s}; \\ \theta \longleftarrow \theta - \eta \dfrac{\partial \mathcal{L}^{\mathrm{SL}}(\theta, \omega^w)/\partial \theta + \partial \mathcal{L}^{\mathrm{CL}}(\theta, \omega^w, \omega^s)}{\partial \theta}. \end{cases} \tag{13}$$

Empirical convergence can be obtained in our experiments, in accordance with the findings of other adversarial models [57, 58]. The momentum update is adopted in the graph encoding branch as

$$\phi \leftarrow \gamma\phi + (1-\gamma)\theta. \tag{14}$$

Here, we set the momentum coefficient γ to 0.99 following [21]. The parameters ϕ undergo smooth evolution through momentum updates to enhance optimization stability. The training procedure of the algorithm is shown in Algorithm 1.

4 Experiment

4.1 Experimental setup

Datasets. We evaluate our proposed GPS on two tasks: graph classification and transfer learning tasks on twelve datasets from TU datasets [60] and open graph benchmark (OGB) datasets [61]. For TU datasets, we adopt three bioinformatics datasets (MUTAG, PROTEINS, NCI1) and three social network datasets (IMDB-B, IMDB-M, REDDIT-M-5K) for the graph classification task. For OGB datasets, we select six molecular datasets (BBBP, ToxCast, ClinTox, BACE, HIV, MUV) for molecular property prediction under transfer learning settings.

Baselines. We conduct a comprehensive comparison of our GPS with three distinct groups of methods: (1) supervised methods including GraphSage [62], GCN [63], GIN [64] and GAT [65]; (2) kernel methods including shortest path kernel (SP) [66], graphlet kernel (GK) [67], Weisfeiler-Lehman kernel (WL) [68]; (3) unsupervised methods including Node2Vec [39], Sub2Vec [69], Graph2Vec [70], InfoGraph [26], GraphCL [28], JOAO [56], AD-GCL [71], SimGRACE [72], and GraphCLA [73].

Implementation details. For our approach, we use a 2-layer GIN [64] as our GNN-based encoder. We set the hidden dimension of GIN as 512 and the number of training epochs as 50. The batch size is set to 128. The ratios in graph pooling modules are set to 0.4 and 0.9 for the strongly-augmented view and weakly-augmented view, respectively. These two hyper-parameters will be discussed in Subsection 4.4.

4.2 Experimental results

As shown in Table 1, we evaluate the effectiveness of our GPS for graph classification, compared to various baselines. We can draw the following conclusions:

• Overall, from the results, it can be observed that our proposed model GPS shows superior performance across all six datasets. GPS consistently performs better than other unsupervised baselines by a

Table 1 Performance of unsupervised learning on bioinformatics and social network classification over five runs (averaged accuracy with standard deviation). The best results are shown in boldface.

	Method	MUTAG	PROTEINS	NCI1	IMDB-B	IMDB-M	REDDIT-M-5K
Supervised	GraphSage	85.1 ± 7.6	75.3 ± 2.4	77.7 ± 1.5	72.3 ± 5.3	50.9 ± 2.2	43.8 ± 3.2
	GCN	85.6 ± 5.8	75.2 ± 3.6	80.2 ± 2.0	74.0 ± 3.4	51.9 ± 3.8	20.0 ± 0.0
	GIN	**89.4 ± 5.6**	**76.2 ± 2.8**	**82.7 ± 1.7**	**75.1 ± 5.1**	**52.3 ± 2.8**	**57.6 ± 1.5**
	GAT	**89.4 ± 6.1**	74.7 ± 4.0	66.6 ± 2.2	70.5 ± 2.3	47.8 ± 3.1	45.9 ± 0.1
Kernel	SP	85.2 ± 2.4	–	73.5 ± 0.1	55.6 ± 0.2	38.0 ± 0.3	39.6 ± 0.2
	GK	81.7 ± 2.1	–	66.0 ± 0.1	65.9 ± 1.0	43.9 ± 0.4	41.0 ± 0.2
	WL	80.7 ± 3.0	72.9 ± 0.6	–	72.3 ± 3.4	47.0 ± 0.5	46.1 ± 0.2
Unsupervised	Node2Vec	72.6 ± 10.2	57.5 ± 3.6	54.9 ± 1.6	50.2 ± 0.9	36.0 ± 0.7	–
	Sub2Vec	61.1 ± 15.8	53.0 ± 5.6	52.8 ± 1.5	55.3 ± 1.5	36.7 ± 0.8	36.7 ± 0.4
	Graph2Vec	83.2 ± 9.6	73.3 ± 2.1	73.2 ± 1.8	71.1 ± 0.5	46.3 ± 1.4	47.9 ±0 .3
	InfoGraph	89.0 ± 1.1	74.4 ± 0.3	76.2 ± 1.1	71.1 ± 0.9	49.7 ± 0.5	53.5 ± 1.0
	GraphCL	86.8 ± 1.3	74.4 ± 0.5	77.9 ± 0.4	71.1 ± 0.4	48.5 ± 0.6	56.0 ± 0.3
	JOAO	87.3 ± 1.0	74.6 ± 0.4	78.1 ± 0.5	70.2 ± 3.1	–	55.7 ± 0.6
	AD-GCL	89.3 ± 1.5	73.6 ± 0.7	69.7 ± 0.5	71.6 ± 1.0	49.0 ± 0.5	54.9 ± 0.4
	SimGRACE	89.1 ± 1.4	74.9 ± 0.7	79.1 ± 0.5	71.6 ± 0.7	48.7 ± 0.7	55.9 ± 0.4
	GraphCLA	89.3 ± 0.4	74.5 ± 0.6	73.0 ± 0.6	72.3 ± 0.5	49.5 ± 0.4	–
	GPS-TopK (ours)	**89.9 ± 0.7**	**75.1 ± 0.4**	79.1 ± 0.6	73.5 ± 0.7	51.4 ± 0.6	**56.3 ± 0.2**
	GPS-Cluster (ours)	89.5 ± 1.2	74.7 ± 0.5	**79.5 ± 0.4**	**73.8 ± 1.1**	**51.7 ± 0.5**	55.9 ± 0.4

significant margin. The strong performance demonstrates the effectiveness of the proposed multi-scale pooling framework for effective GCL.

• A general observation is that supervised algorithms still have the highest performance. Interestingly, even compared with the supervised ones, our approach GPS achieves competitive performance in 5 out of 6 datasets and outperforms supervised results on dataset MUTAG. Moreover, among all the supervised algorithms, we can see that GIN consistently outperforms other GNN models on all datasets, which verifies the superiority of GIN with strong representation capability. This justifies the reason why we choose GIN as the base GNN-based encoder.

• The performance of traditional kernel methods is inferior to most unsupervised methods, which suggests that these methods may be ineffective in capturing effective information of the graph topology and node attributes. Moreover, the features derived from kernel methods are typically heuristic, which leads to worse generalization ability and sub-optimal performance.

• By integrating the idea of contrastive learning into GNNs, recent state-of-the-art methods (InfoGraph, GraphCL, JOAO, AD-GCL, SimGRACE, and GraphCLA) have obtained high enough performance, which pushes away the other unsupervised baselines (Node2Vec, Sub2Vec, Graph2Vec), sufficiently showing the superiority of instance discrimination principle in the contrastive learning.

• Among two variants based on different graph pooling techniques, we can see that GPS-TopK and GPS-Cluster stand out as two robust variants. They achieve top-tier or competitive performance across all datasets. Compared to existing state-of-the-arts, their superior results validate the effectiveness of our framework, which explores learnable graph pooling to derive augmented views in an adversarial manner.

4.3 Ablation study

Then, we compare GPS with its three variants to validate the effectiveness of each component.

• **GPS w/o weak.** We remove the weakly-augmented view and train the model with similarity learning using the strongly-augmented view since consistency learning requires both views.

• **GPS w/o strong ($\mathcal{L}^{\mathrm{CL}}$).** We remove the strongly-augmented view and the model is simply trained with similarity learning using the weakly-augmented view.

• **GPS w/o $\mathcal{L}^{\mathrm{SL}}$.** We remove the similarity learning loss and the model is simply trained with consistency learning using both views.

• **GPS w/o adv.** We remove the adversarial learning in the graph pooling modules. The pooling modules are updated with gradient descent along with the encoder.

We compare the performance of different variants and then plot the results in Figure 3. From the figure, we can draw the following conclusions. First, the results of GPS are consistently better than all the other

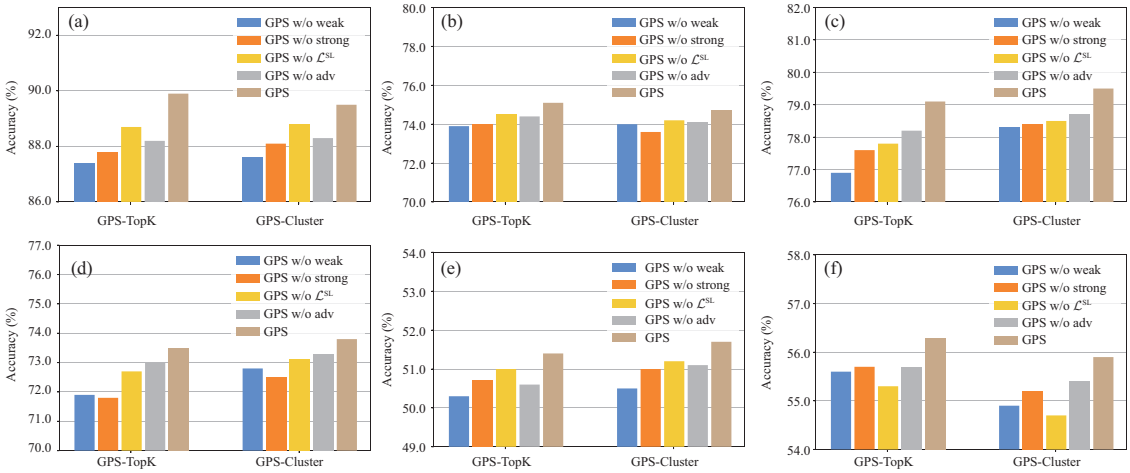

Figure 3 (Color online) Performance of ablation study of several model variants on all six datasets. (a) MUTAG; (b) PROTEINS; (c) NCI1; (d) IMDB-B; (e) IMDB-M; (f) REDDIT-M-5K.

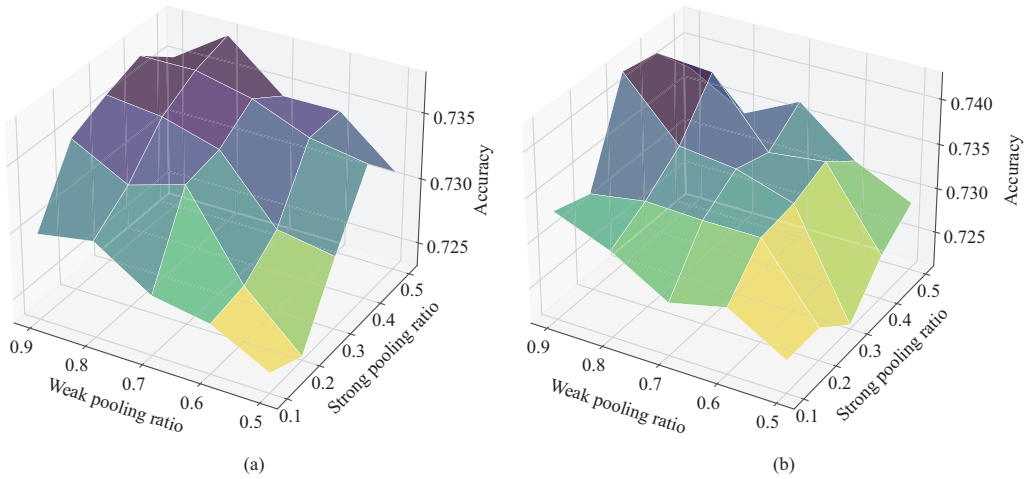

Figure 4 (Color online) Analysis of graph pooling ratio on IMDB-B. (a) GPS-TopK; (b) GPS-Cluster.

four variants, indicating that both our multi-scale graph pooling and adversarial learning are effective for GCL. Second, the results of GPS w/o weak and GPS w/o strong are usually inferior to GPS w/o adv on most datasets, which verifies the usefulness of the two view augmentations. Third, GPS w/o strong is generally better than GPS w/o weak on most datasets, which implies that weakly-augmented views as well as similarity learning play a more important role in this framework. Fourth, we observe that removing the strongly-augmented view is equivalent to removing $\mathcal{L}^{\mathrm{CL}}$. It can be noticed that regardless of which loss is removed (GPS w/o strong ($\mathcal{L}^{\mathrm{CL}}$) or GPS w/o $\mathcal{L}^{\mathrm{SL}}$), the performance of our proposed method deteriorates significantly, demonstrating the significance of our proposed diverse losses. Additionally, GPS w/o $\mathcal{L}^{\mathrm{SL}}$ outperforms $\mathcal{L}^{\mathrm{CL}}$ in five out of six datasets, highlighting the importance of emphasizing strongly-augmented and weakly-augmented views for learning discriminative graph representations.

4.4 Sensitivity analysis

In this subsection, we investigate the sensitivity of parameters graph pooling ratio ρ and batch size B.

Analysis of graph pooling ratio. We test the effect of the graph pooling ratio ρ, which controls the ratio of the augmented graph. We vary ρ_1 and ρ_2 as $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\{0.5, 0.6, 0.7, 0.8, 0.9\}$, respectively. The results of our two variants on IMDB-B are shown in Figure 4. We can observe that for two variants, generally, with the decrease of ρ_1 or ρ_2 while the other ratio is fixed, the performance tends to decrease slowly. Maybe the reason is that the small ratio of graph pooling is prone to distort topological patterns and attributes. However, note that for GPS-TopK, the performance difference caused by different parameter combinations is less than 0.01, and for GPS-Cluster, the performance is relatively stable when

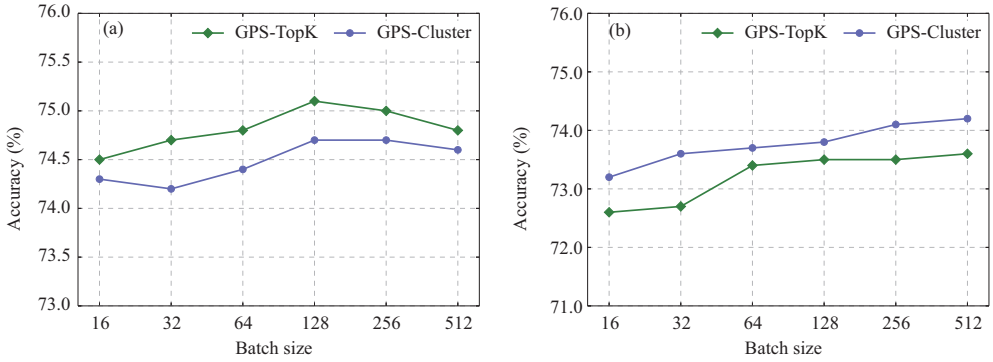

Figure 5 (Color online) Analysis of batch size on PROTEINS and IMDB-B. (a) PROTEINS; (b) IMDB-B.

Table 2 Clustering performance on four graph property prediction benchmarks. The best results are shown in boldface.

Method	DD			IMDB-B			REDDIT-B			REDDIT-M-12K		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
InfoGraph	0.008	0.558	−0.006	0.041	0.538	0.005	0.016	0.508	0.000	0.045	0.205	0.003
GraphCL	0.019	0.573	−0.009	0.046	0.545	0.008	0.033	0.519	0.001	0.096	0.181	0.021
CuCo	0.012	0.562	−0.010	0.001	0.507	0.000	0.018	0.510	0.000	0.003	0.192	0.002
JOAO	0.012	0.578	−0.004	0.042	0.543	0.008	0.034	0.520	0.001	0.003	0.183	0.001
RGCL	0.014	0.565	−0.009	0.047	0.546	0.007	0.017	0.509	0.001	0.003	0.092	0.001
SimGRACE	0.001	0.589	0.003	**0.049**	0.559	0.007	0.024	0.513	0.001	0.062	0.210	0.005
GPS	**0.020**	**0.594**	**0.004**	0.048	**0.0565**	**0.009**	**0.035**	**0.523**	**0.002**	**0.113**	**0.220**	**0.035**

the parameters are not too large or small, as shown in the plateau in Figure 4(b). We conjecture that it is beneficial to performance via generating augmented views with the removal of redundant information and preserving semantics in an adversarial way. We hence conclude that our proposed framework GPS is generally insensitive to these parameters, demonstrating the robustness to hyperparameter tuning and easing the parameter selection for our framework.

Analysis of batch size. Next, we evaluate the effect of the batch size B, and vary it in the range of $\{16, 32, 64, 128, 256, 512\}$. The results are shown in Figure 5. It can be seen that for PROTEINS, with the increase of B, the performance tends to first increase and then decrease. A too-small B would lead to a lack of intra-batch sample diversity and fail to provide an effective similarity distribution while a large B may introduce too many noise samples. For IMDB-B, we can observe that an increasing batch size consistently enhances performance. This is because a sufficiently large batch can more effectively represent the entire dataset, encompassing a wider range of diverse samples to facilitate the learning of discriminative representations for the target samples. It is worth noting that an excessively large batch size could potentially lead to issues related to space complexity.

4.5 Graph-level clustering

To further demonstrate the discrimination of the learned graph representations, we conduct the experiment of graph-level clustering [50] on four datasets, including one DD, IMDB-B, REDDIT-B, and REDDIT-M-12K. We compare our GPS with several competitive baselines: InfoGraph [26], GraphCL [28], CuCo [27], JOAO [56], RGCL [74] and SimGRACE [72]. Here we adopt three widely-used evaluation indicators to measure the clustering performance: normalized mutual information (NMI) [75], clustering accuracy (ACC) [76] and adjusted rand index (ARI) [77]. These evaluation indicators cover various aspects of clustering outcomes. NMI and ACC have a range of $[0, 1]$, whereas ARI ranges in $[-1, 1]$. Higher values indicate better performance across all three evaluation indicators.

The quantitative results of graph-level clustering are reported in Table 2, it can be observed that our proposed GPS consistently demonstrates superior performance compared to other GCL approaches across all four datasets under three evaluation indicators. This showcases the exceptional effectiveness of our framework in graph-level clustering. This might be attributed to our multi-scale augmented views, which capture complementary information and learn more discriminative representations through adversarial learning, thereby better serving the clustering task.

Table 3 Performance of transfer learning on molecular property prediction over five runs (ROC-AUC with standard deviation). The best results are shown in boldface.

Method	BBBP	ToxCast	ClinTox	BACE	HIV	MUV	Avg.	Rank
No Pre-Train	65.8 ± 4.5	63.4 ± 0.6	58.0 ± 4.4	70.1 ± 5.4	75.3 ± 1.9	71.8 ± 2.5	67.4	10
EdgePred [78]	67.3 ± 2.4	64.1 ± 0.6	64.1 ± 3.7	79.9 ± 0.9	76.3 ± 1.0	74.1 ± 2.1	71.0	9
AttrMasking [78]	64.3 ± 2.8	64.2 ± 0.5	71.8 ± 4.1	79.3 ± 1.6	77.2 ± 1.1	74.7 ± 1.4	71.9	5
ContextPred [78]	68.0 ± 2.0	63.9 ± 0.6	65.9 ± 3.8	79.6 ± 1.2	77.3 ± 1.0	**75.8 ± 1.7**	71.8	6
GraphPartition [79]	70.3 ± 0.7	63.2 ± 0.3	64.2 ± 0.5	79.6 ± 1.8	77.1 ± 0.7	75.4 ± 1.7	71.6	7
InfoGraph [26]	68.8 ± 0.8	62.7 ± 0.4	69.9 ± 3.0	75.9 ± 1.6	76.0 ± 0.7	75.3 ± 2.5	71.4	8
GraphCL [28]	69.7 ± 0.7	62.4 ± 0.6	76.0 ± 2.7	75.4 ± 1.4	78.5 ± 1.2	69.8 ± 2.7	72.0	4
JOAO [56]	70.2 ± 1.0	62.9 ± 0.5	81.3 ± 2.5	77.3 ± 0.5	76.7 ± 1.2	71.7 ± 1.4	73.4	3
AD-GCL [71]	70.0 ± 1.0	63.1 ± 0.7	79.8 ± 3.5	78.5 ± 0.8	78.3 ± 1.0	72.3 ± 1.6	73.7	2
GPS-TopK	**71.5 ± 0.9**	**64.4 ± 0.3**	**82.1 ± 2.9**	**80.1 ± 0.8**	**79.0 ± 1.1**	75.6 ± 1.7	**75.5**	**1**

4.6 Transfer learning

In this subsection, we evaluate the generalization of our proposed method on molecular property prediction for transfer learning. Following [78], our model is pre-trained on a large-scale ZINC15 dataset (two million unlabeled molecules) and later fine-tuned on six OGB [61] datasets to test out-of-distribution performance. Here, we only consider GPS-TopK for illustration. We adopt four common pre-training strategies (No Pre-Train, EdgePred, AttrMasking, and ContextPred in [78]) and five state-of-the-art techniques (GraphPartition [79], InfoGraph [26], GraphCL [28], JOAO [56], and AD-GCL [71]) to study the transferability of the various pre-training strategies.

The results are shown in Table 3 [26, 28, 56, 71, 78, 79]. It can be observed that GPS-TopK significantly outperforms various baselines in five out of the six datasets, and achieve Top-1 performance in terms of average ROC-AUC among ten baselines, which fully shows the excellent generalization capacity of our framework. Moreover, it is worth mentioning that compared to the No Pre-Train, our method improves 41.6% and 14.3% on ClinTox and BACE respectively, indicating the effectiveness of the discrimination ability of the contrastive learning principle. Compared with the stronger baselines GraphCL [28], JOAO [56] and AD-GCL [71], we can see those different methods may have their own preference for different datasets due to their specific characteristics such as binding affinity, toxicity and adverse reactions. However, our method can consistently outperform these baselines in most datasets, indicating the effectiveness of the multi-scale pooling. The above results show that our method GPS can learn effective graph-level representations which achieve superior out-of-distribution performance.

4.7 Semi-supervised learning

Lastly, we evaluate our proposed model for semi-supervised learning on two large-scale OGB datasets [60] ogbg-ppa and ogbg-code to test the scalability of our framework. The ogbg-ppa dataset, which consists of 158100 proteins, is extracted from the protein-protein association networks of 1581 different species. The ogbg-code dataset is a collection of abstract syntax trees obtained from 452741 Python method definitions. Here, we only consider GPS-TopK for illustration. Our model is pre-trained on one dataset using self-supervised learning and later fine-tuned based on 3% and 10% label supervision on the same dataset following the setting in [56] and compare it with GraphCL [28] and JOAO [56].

The results are reported in Table 4. From the table, we can see that our GPS-TopK significantly outperforms all the baselines on two large-scale OGB datasets, which again demonstrates the strength and scalability of our proposed model. Maybe the reason is that GraphCL and JOAO adopt the empirically pre-defined rules for augmentation selection while our framework leverages learnable graph pooling to automatically provide effective augmented views.

5 Conclusion

In this study, we explore self-supervised graph representation learning by presenting a novel framework GPS. Specifically, GPS leverages learnable graph pooling to automatically generate multi-scale positive views, which emphasize preserving semantics and providing challenging positives via strongly-augmented

Table 4 Performance of semi-supervised learning on large-scale OGB datasets (accuracy on ogbg-ppa, F1 on ogbg-code at 3% and 10% label rate respectively). The best results are shown in boldface.

Rate	Method	ogbg-ppa	ogbg-code
3%	GraphCL	44.3 ± 5.2	12.0 ± 0.3
	JOAO	47.8 ± 4.6	11.7 ± 0.6
	GPS-TopK	$\mathbf{49.6 \pm 3.9}$	$\mathbf{13.2 \pm 0.4}$
10%	GraphCL	55.8 ± 0.9	20.9 ± 0.3
	JOAO	60.1 ± 1.2	21.4 ± 0.5
	GPS-TopK	$\mathbf{63.2 \pm 1.1}$	$\mathbf{23.1 \pm 0.5}$

view and weakly-augmented view, respectively. Moreover, we develop a joint contrastive learning framework that incorporates both views to explore similarity learning and consistency learning, where our graph pooling modules are adversarially trained with the encoder for robustness and efficiency. Extensive experiments well showcase the superiority of our proposed GPS over state-of-the-art baselines on twelve real-world datasets.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant Nos. 62306014, 62276002) and China Postdoctoral Science Foundation (Grant No. 2023M730057).

References

1 Cao W M, Zheng C T, Yan Z Y, et al. Geometric deep learning: progress, applications and challenges. Sci China Inf Sci, 2022, 65: 126101
2 Wu L Y, Liu D, Guo X, et al. Multi-scale spatial representation learning via recursive Hermite polynomial networks. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022. 1465–1473
3 Chen D, Wang M, Chen H, et al. Cross-modal retrieval with heterogeneous graph embedding. In: Proceedings of the 30th ACM International Conference on Multimedia, 2022. 3291–3300
4 Wang Y, Wu L. Beyond low-rank representations: orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering. Neural Netw, 2018, 103: 1–8
5 Wang Y, Jin W, Derr T. Graph neural networks: self-supervised learning. In: Graph Neural Networks: Foundations, Frontiers, and Applications. Berlin: Springer, 2022. 391–420
6 Jiang B, Kloster K, Gleich D F, et al. AptRank: an adaptive PageRank model for protein function prediction on bi-relational graphs. Bioinformatics, 2017, 33: 1829–1836
7 Ju W, Liu Z, Qin Y, et al. Few-shot molecular property prediction via hierarchically structured learning on relation graphs. Neural Netw, 2023, 163: 122–131
8 Kojima R, Ishida S, Ohta M, et al. kGCN: a graph-based deep learning framework for chemical structures. J Cheminform, 2020, 12: 1
9 Hao Z, Lu C, Huang Z, et al. ASGN: an active semi-supervised graph neural network for molecular property prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020. 731–752
10 Rozemberczki B, Hoyt C T, Gogleva A, et al. ChemicalX: a deep learning library for drug pair scoring. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022. 3819–3828
11 Qu A, Wang Y, Hu Y, et al. A data-integration analysis on road emissions and traffic patterns. In: Proceedings of the Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI, 2020. 503–517
12 Fang Z, Long Q, Song G, et al. Spatial-temporal graph ode networks for traffic flow forecasting. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021. 364–373
13 Zhao Y, Luo X, Ju W, et al. Dynamic hypergraph structure learning for traffic flow forecasting. In: Proceedings of the 39th International Conference on Data Engineering (ICDE), Anaheim, 2023
14 Qin Y, Wu H, Ju W, et al. A diffusion model for POI recommendation. 2023. ArXiv:2304.07041
15 Qin Y, Ju W, Wu H, et al. Learning graph ode for continuous-time sequential recommendation. 2023. ArXiv:2304.07042
16 Wang Y, Zhao Y, Zhang Y, et al. Collaboration-aware graph neural network for recommender systems. In: Proceedings of the 1st Learning on Graphs Conference, 2022
17 Ying Z, You J, Morris C, et al. Hierarchical graph representation learning with differentiable pooling. 2018. ArXiv:1806.08804
18 Ju W, Yang J, Qu M, et al. KGNN: harnessing kernel-based networks for semi-supervised graph classification. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining, 2022. 421–429
19 Jin T S, Dai H Q, Cao L J, et al. Deepwalk-aware graph convolutional networks. Sci China Inf Sci, 2022, 65: 152104
20 Luo X, Ju W, Gu Y, et al. Toward effective semi-supervised node classification with hybrid curriculum pseudo-labeling. ACM Trans Multimedia Comput Commun Appl, 2024, 20: 1–19
21 He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 9729–9738
22 Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning, 2020. 1597–1607
23 Zhou C, Ma J, Zhang J, et al. Contrastive learning for debiased candidate generation in large-scale recommender systems. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021. 3985–3995

24 Wei Y, Wang X, Li Q, et al. Contrastive learning for cold-start recommendation. In: Proceedings of the 29th ACM International Conference on Multimedia, 2021. 5382–5390
25 Xiao S T, Shao Y X, Li Y W, et al. LECF: recommendation via learnable edge collaborative filtering. Sci China Inf Sci, 2022, 65: 112101
26 Sun F Y, Hoffmann J, Verma V, et al. InfoGraph: unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In: Proceedings of the Internation Conference on Learning Representations, 2020
27 Chu G, Wang X, Shi C, et al. CuCo: graph representation with curriculum contrastive learning. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence, 2021. 2300–2306
28 You Y, Chen T, Sui Y, et al. Graph contrastive learning with augmentations. 2020. ArXiv:2010.13902
29 Hassani K, Khasahmadi A H. Contrastive multi-view representation learning on graphs. In: Proceedings of the International Conference on Machine Learning, 2020. 4116–4126
30 Wang X, Liu N, Han H, et al. Self-supervised heterogeneous graph neural network with co-contrastive learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021. 1726–1736
31 Linsker R. Self-organization in a perceptual network. Computer, 1988, 21: 105–117
32 Lee J, Lee I, Kang J. Self-attention graph pooling. In: Proceedings of the International Conference on Machine Learning, 2019. 3734–3743
33 Gao H, Ji S. Graph U-Nets. In: Proceedings of the International Conference on Machine Learning, 2019. 2083–2092
34 Ranjan E, Sanyal S, Talukdar P. ASAP: adaptive structure aware pooling for learning hierarchical graph representations. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 5470–5477
35 Sun Q, Li J, Peng H, et al. SUGAR: subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In: Proceedings of the Web Conference, 2021. 2081–2091
36 Ahmed A, Shervashidze N, Narayanamurthy S, et al. Distributed large-scale natural graph factorization. In: Proceedings of the 22nd International Conference on World Wide Web, 2013. 37–48
37 Cai D, He X F, Han J W, et al. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell, 2011, 33: 1548–1560
38 Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 701–710
39 Grover A, Leskovec J. Node2Vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 855–864
40 Gutmann M, Hyvärinen A. Noise-contrastive estimation: a new estimation principle for unnormalized statistical models. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010. 297–304
41 Fan W F, He K, Li Q, et al. Graph algorithms: parallelization and scalability. Sci China Inf Sci, 2020, 63: 203101
42 Guo Q P, Qiu X P, Xue X Y, et al. Syntax-guided text generation via graph neural network. Sci China Inf Sci, 2021, 64: 152102
43 Ju W, Luo X, Qu M, et al. TGNN: a joint semi-supervised framework for graph-level classification. 2023. ArXiv:2304.11688
44 Mao Z, Ju W, Qin Y, et al. RAHNet: retrieval augmented hybrid network for long-tailed graph classification. 2023. ArXiv:2308.02335
45 Zou C, Han A, Lin L, et al. A simple yet effective framelet-based graph neural network for directed graphs. IEEE Trans Artif Intell, 2024, 5: 1647–1657
46 Gilmer J, Schoenholz S S, Riley P F, et al. Neural message passing for quantum chemistry. In: Proceedings of the International Conference on Machine Learning, 2017. 1263–1272
47 Ju W, Fang Z, Gu Y, et al. A comprehensive survey on deep graph representation learning. 2023. ArXiv:2304.05055
48 Velickovic P, Fedus W, Hamilton W L, et al. Deep graph infomax. 2019. ArXiv:1809.10341
49 Zhu Y, Xu Y, Yu F, et al. Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference, 2021. 2069–2080
50 Ju W, Gu Y, Chen B, et al. GLCC: a general framework for graph-level clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 4391–4399
51 Yi S, Ju W, Qin Y, et al. Redundancy-free self-supervised relational learning for graph clustering. IEEE Trans Neural Netw Learn Syst, 2024. doi: 10.1109/TNNLS.2023.3314451
52 Wang Y G, Li M, Ma Z, et al. Haar graph pooling. In: Proceedings of the International Conference on Machine Learning, 2020. 9952–9962
53 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. 2017. ArXiv:1706.03762
54 Wang X, Qi G J. Contrastive learning with stronger augmentations. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 5549–5560
55 Grill J B, Strub F, Altché F, et al. Bootstrap your own latent — a new approach to self-supervised learning. 2020. ArXiv:2006.07733
56 You Y, Chen T, Shen Y, et al. Graph contrastive learning automated. In: Proceedings of the International Conference on Machine Learning, 2021. 12121–12132
57 Kong K, Li G, Ding M, et al. FLAG: adversarial data augmentation for graph neural networks. 2020. ArXiv:2010.09891
58 Jiang H, He P, Chen W, et al. SMART: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. 2177–2190
59 Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. 2017. ArXiv:1701.04862

60 Morris C, Kriege N M, Bause F, et al. TUDataset: a collection of benchmark datasets for learning with graphs. 2020. ArXiv:2007.08663

61 Hu W, Fey M, Zitnik M, et al. Open graph benchmark: datasets for machine learning on graphs. 2020. ArXiv:2005.00687

62 Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. 2017. ArXiv:1706.02216

63 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: Proceedings of the Internation Conference on Learning Representations, 2017

64 Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks? In: Proceedings of the Internation Conference on Learning Representations, 2019

65 Veličković P, Cucurull G, Casanova A, et al. Graph attention networks. In: Proceedings of the International Conference on Learning Representations, 2018

66 Borgwardt K M, Kriegel H P. Shortest-path kernels on graphs. In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05), 2005

67 Shervashidze N, Vishwanathan S, Petri T, et al. Efficient graphlet kernels for large graph comparison. In: Proceedings of the Artificial Intelligence and Statistics, 2009. 488–495

68 Shervashidze N, Schweitzer P, van Leeuwen E J, et al. Weisfeiler-Lehman graph kernels. J Mach Learn Res, 2011, 12: 2539–2561

69 Adhikari B, Zhang Y, Ramakrishnan N, et al. Sub2Vec: feature learning for subgraphs. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2018. 170–182

70 Narayanan A, Chandramohan M, Venkatesan R, et al. Graph2Vec: learning distributed representations of graphs. 2017. ArXiv:1707.05005

71 Suresh S, Li P, Hao C, et al. Adversarial graph augmentation to improve graph contrastive learning. 2021. ArXiv:2106.05819

72 Xia J, Wu L, Chen J, et al. SimGRACE: a simple framework for graph contrastive learning without data augmentation. In: Proceedings of the ACM Web Conference, 2022. 1070–1079

73 Pu X, Zhang K, Shu H, et al. Graph contrastive learning with learnable graph augmentation. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023. 1–5

74 Li S, Wang X, Zhang A, et al. Let invariant rationale discovery inspire graph contrastive learning. In: Proceedings of the International conference on machine learning, 2022. 13052–13065

75 Strehl A, Ghosh J. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. J Mach Learn Res, 2002, 3: 583–617

76 Li T, Ding C. The relationships among various nonnegative matrix factorization methods for clustering. In: Proceedings of the 6th International Conference on Data Mining (ICDM'06), 2006. 362–371

77 Hubert L, Arabie P. Comparing partitions. J Classif, 1985, 2: 193–218

78 Hu W, Liu B, Gomes J, et al. Strategies for pre-training graph neural networks. In: Proceedings of the International Conference on Learning Representations, 2019

79 You Y, Chen T, Wang Z, et al. When does self-supervision help graph convolutional networks? In: Proceedings of the International Conference on Machine Learning, 2020. 10871–10880