Neural Networks 158 (2023) 359–368

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Unsupervised graph-level representation learning with hierarchical contrasts

Wei Ju [a,1], Yiyang Gu [a,1], Xiao Luo [b,*], Yifan Wang [a], Haochen Yuan [a], Huasong Zhong [c], Ming Zhang [a,*]

[a] School of Computer Science, Peking University, Beijing, 100871, China
[b] Department of Computer Science, University of California, Los Angeles, 90095, USA
[c] Alibaba Group, Hangzhou, 311100, China

ARTICLE INFO

Article history:
Received 28 September 2022
Received in revised form 8 November 2022
Accepted 13 November 2022
Available online 26 November 2022

Keywords:
Graph representation learning
Graph contrastive learning
Graph neural networks
Unsupervised learning

ABSTRACT

Unsupervised graph-level representation learning has recently shown great potential in a variety of domains, ranging from bioinformatics to social networks. Plenty of graph contrastive learning methods have been proposed to generate discriminative graph-level representations recently. They typically design multiple types of graph augmentations and enforce a graph to have consistent representations under different views. However, these techniques mostly neglect the intrinsic hierarchical structure of the graph, resulting in a limited exploration of semantic information for graph representation. Moreover, they often rely on a large number of negative samples to prevent collapsing into trivial solutions, while a great need for negative samples may lead to memory issues during optimization in graph domains. To address the two issues, this paper develops an unsupervised graph-level representation learning framework named Hierarchical Graph Contrastive Learning (HGCL), which investigates the hierarchical structural semantics of a graph at both node and graph levels. Specifically, our HGCL consists of three parts, i.e., node-level contrastive learning, graph-level contrastive learning, and mutual contrastive learning to capture graph semantics hierarchically. Furthermore, the Siamese network and momentum update are further involved to release the demand for excessive negative samples. Finally, the experimental results on both benchmark datasets for graph classification and large-scale OGB datasets for transfer learning demonstrate that our proposed HGCL significantly outperforms a broad range of state-of-the-art baselines.

1. Introduction

Graph-structured data is pervasive in a broad range of domains, such as social networks (Cai, Gong, Shen, Ma, & Jiao, 2014), sensor networks (Wang, Liao, Wang, Huang, & Chen, 2016), and transportation networks (Ali, Zhu, & Zakarya, 2022; Zhang, Cao, Huang, Shi, & Zhou, 2022). Among graph machine learning problems, learning informative representations of the whole graph is critical for many applications including predicting protein functionality in biological networks (Jiang, Kloster, Gleich, & Gribskov, 2017) and inferring molecular properties in drug discovery (Hao et al., 2020; Kojima et al., 2020). The primary challenge of graph-level representation learning is to explore effective whole-graph embeddings that capture both node attributes as well as topological information.

Nowadays, most approaches have made great efforts to extend the convolution operation to graph-structured data, and fall under the umbrella of graph neural networks (GNNs), showing great success in learning graph representations (Baek, Kang, & Hwang, 2021; Lee, Lee, & Kang, 2019; Rassil, Chougrad, & Zouaki, 2022; Ying et al., 2018). These GNN methods extract critical topological features and node attributes using neighbor-aware message passing mechanism in learning node representations (Gilmer, Schoenholz, Riley, Vinyals, & Dahl, 2017; Jiang, Chen, Wang, & Luo, 2022; Ju, Luo et al., 2022; Xie, Zhang, Gong, Tang, & Han, 2020), all of which are integrated into a graph-level representation for various downstream applications (Duan et al., 2022; Ju, Qin et al., 2022). Nevertheless, these approaches mostly require massive task-specific labels which are scarce in many domains (Hao et al., 2020; Sun, Hoffmann, Verma, & Tang, 2020). Even worse, label annotation is also extremely time-consuming

* Corresponding authors.
E-mail addresses: xiaoluo@cs.ucla.edu (X. Luo), mzhang_cs@pku.edu.cn (M. Zhang).

[1] Equal contribution with the order determined by flipping a coin.

https://doi.org/10.1016/j.neunet.2022.11.019
0893-6080/

and labor-intensive, making supervised learning methods hard to be applied in real-world scenarios. For example, living animal experiments are required to identify the pharmacological effect of molecular graphs. Meanwhile, there always exist a vast number of unlabeled samples available in real-world circumstances. As such, unsupervised graph-level representation learning methods are expected to be proposed as a key technique to alleviate the reliance on label information in practice.

Early unsupervised methods such as graph kernels usually adopt handcraft features which may lead to poor generalization and thus fail to achieve satisfactory performance. Motivated by the recent success in self-supervised representation learning in computer vision (Chen, Kornblith, Norouzi, & Hinton, 2020; He, Fan, Wu, Xie, & Girshick, 2020) and natural language processing (Devlin, Chang, Lee, & Toutanova, 2019), several works have introduced this technique to graph representation learning (Luo, Ju, Qu, Gu, et al., 2022; Sun et al., 2020; You, Chen, Shen, & Wang, 2021; You, Chen, Sui et al., 2020). The underlying idea behind graph contrastive learning (GCL) is to augment graph samples from different views. With the guidance of self-supervised learning, these methods encourage a graph to have similar representations to its augmented view compared with other graphs. Thus, these approaches are capable of generating effective graph-level representations, which are beneficial for various downstream applications.

Even though previous GCL methods have achieved promising performance, they are prone to suffer from two critical limitations as follows, which could render low-quality graph-level representations and sub-optimal performance:

- **Neglection of hierarchical semantics.** Existing approaches typically fall short of adequately exploring hierarchical structural characteristics. As we know, a node is the most fundamental structural property in a graph, and a graph-level representation is computed by organically aggregating all node representations. Moreover, node representations are capable of capturing different scales of patch information in the whole graph after multiple graph convolutional layers. Consequently, node representations are critical for obtaining informative graph-level representations. However, current GCL methods only concentrate on graph-level information but neglect node-level exploration of the graph-structured data (You, Chen, Sui et al., 2020).
- **Dependency of massive negative samples**. To prevent collapsing into trivial solutions (i.e., generate the same representation for all graphs), the bulk of them heavily rely on excessive negative samples. In GCL, negative nodes and graphs are used to serve as crucial regulators. This issue could lead to large memory cost during model optimization, which may be even unaffordable due to the domain specificity (You, Chen, Sui et al., 2020). Recently, some works have developed several contrastive learning algorithms on images that do not need negative samples (Chen & He, 2021; Grill et al., 2020). However, they have not been investigated in GCL methods to alleviate the dependence on a large number of negative samples.

In this study, our paper presents a principled framework called Hierarchical Graph Contrastive Learning (HGCL) for unsupervised graph-level representation learning. To address the **limitation 1**, our approach models not only the structural semantics of the entire graph (i.e., graph-level semantics) but also substructures of different granularities (i.e., patch-level semantics), which are embedded in node representations at all depths of the GNNs. Technically, our HGCL consists of three parts: (i) Node-level contrastive learning for informative patch-level representations; (ii) Graph-level contrastive learning for discriminative graph-level representations; (iii) Mutual contrastive learning to enhance the unity of multi-scale representations. To overcome the **limitation 2**, our approach proposes to adopt a Siamese architecture as our backbone in both node-level and graph-level contrastive learning frameworks. In brief, our design is comprised of two networks, dubbed online network and target network, which communicate and learn from each other. The consistency between node (graph) representations is encouraged across different views from two graph neural networks, respectively. The core of the Siamese architecture is introducing a predictor on top of the online network to design an asymmetric architecture, and the momentum update for the target network is involved to encourage encoding gradual information, which can empirically avoid collapsed solutions (Grill et al., 2020). Apart from contrasting representations across the two networks, we enhance the representation learning within the online encoder by proposing within-network contrastive learning loss to regularize the training of bootstrapping contrastive learning objectives. At last, the HGCL maximizes the mutual information between the node-level representations and graph-level representation to enhance the unity of hierarchical representations from both local and global views. Our proposed model HGCL is validated on various graph classification benchmark datasets and large-scale OGB datasets. Experimental results show the superiority of our HGCL against a wide range of state-of-the-art baselines on both graph classification task and transfer learning task. To summarize, the contributions of this work are as follows:

- This paper introduces a unified unsupervised graph-level representation learning framework HGCL, among which we simultaneously model both patch-level semantics and graph-level semantics to mutually enhance each other via multiple types of contrastive learning.
- To avoid collapsed solutions in graph contrastive learning, our proposed HGCL leverages bootstrapping in the Siamese network and conducts GCL both across two networks and within the online network.
- Comprehensive experiments are conducted to evaluate the effectiveness of our model. Experimental results demonstrate that our proposed HGCL significantly outperforms various state-of-the-art methods.

2. Related work

2.1. Graph representation learning

Graph neural networks (GNNs), particularly graph convolutional networks (Kipf & Welling, 2017), have shown extraordinary capabilities to encode graph-structured data. Recently, the vast majority of graph representation learning techniques adopt the GNNs as the backbone and achieve promising performance. Graph representation learning can be categorized into node-level and graph-level representation learning. The former has been studied extensively in recent years (Jin et al., 2021; Velickovic et al., 2019; Zhu et al., 2021), while the latter is underexplored but important for a range of real-world applications. Prevailing approaches derive graph-level representations via neighbor propagation of GNNs (Gilmer et al., 2017; Ju, Yang et al., 2022; Kipf & Welling, 2017; Veličković et al., 2017), and global summarizing (Lee et al., 2019; Ying et al., 2018; Zhang, Cui, Neumann, & Chen, 2018). These methods are typically optimized in a supervised way, demanding a huge amount of task-specific labels. However, annotating labels is often too costly and thus the required labels are very scarce, making them inapplicable in reality (Hao et al., 2020). To tackle this challenge, this paper concentrates on unsupervised graph-level representation learning and explores the hierarchical structural semantics of a graph at both node and graph levels.

2.2. Graph contrastive learning

Contrastive learning (CL), which is extended from the Information Maximization principle, has achieved great success in visual domains. These CL methods typically maximize the mutual information between the input and its representation by comparing positive pairs produced via random perturbation of the original data with sampled negative counterparts (Chen et al., 2020; He et al., 2020). Increasing attempts have been made to introduce CL to graph domains (Hassani & Khasahmadi, 2020; Luo, Ju, Qu, Chen, et al., 2022; Qiu et al., 2020; Velickovic et al., 2019). As a famous early work, DGI (Velickovic et al., 2019) maximizes the mutual information between patch-level and graph-level representations on the augmented graphs. More attention has been paid to graph-level representation learning currently. These approaches are generally based on the framework of visual contrastive learning, which pushes two graph views augmented from the same sample close while enlarging the distance between graph views from different samples (Chu, Wang, Shi, & Jiang, 2021; Liu et al., 2021; You et al., 2021; You, Chen, Sui et al., 2020; Zeng & Xie, 2021). However, these methods usually suffer from the neglection of patch semantics and depend on huge negative samples, which may lead to sub-optimal performance. In this paper, the HGCL studies the hierarchical structural semantics of the graph and employs the Siamese network and momentum update to address these challenges.

2.3. Siamese network

The Siamese network is a sort of network architecture in which two or more identical subnetworks are utilized to produce and compare feature vectors for each input (Bromley et al., 1993). Numerous self-supervised visual representation learning systems have recently embraced this design and shown performance gains over previous efforts. For instance, BYOL is composed of online and offline networks that interact and learn from one another without the need for negative samples (Grill et al., 2020). SimSiam extends BYOL and can also avoid collapse when maximizing the similarity between two representations of the same sample without using negative instances (Chen & He, 2021). Moreover, it emphasizes the importance of the additional predictor in the online network and the momentum-updating procedure in the target network to avoid trivial solutions in the absence of negative examples. Inspired by recent works, our method extends the Siamese network into graph-level representation learning and achieves promising performance.

3. Preliminaries and notations

Let $G = (V, E)$ denote a graph, where V is the node set and $E \subset V \times V$ is the edge set. The node $v \in V$ has feature vectors associated with it, denoted by $\mathbf{x}_v \in \mathbb{R}^F$ where F denotes the feature dimension. Unsupervised graph-level representation learning is a fundamental task with a wide range of applications, including predicting the mechanical characteristics of molecules and determining the functionality of chemical compounds.

Definition (*Unsupervised Graph-level Representation Learning*)**.** Given the M unlabeled graphs $\{G_1, \ldots, G_M\}$ available, the goal is to learn a GNN-based encoder to generate an effective representation $\mathbf{z}_m \in \mathbb{R}^d$ for each graph G_m without label guidance, where d is embedding dimension. The generated graph representations $\{\mathbf{z}_1, \ldots \mathbf{z}_M\}$ will be evaluated on a series of downstream tasks including graph classification and transfer learning.

4. Methodology

4.1. Framework overview

Existing methods typically neglect the hierarchical structural semantics of the graph in nature. Moreover, they usually depend on a huge number of negative samples during model optimization. To address these key limitations, our HGCL has two simple yet effective designs that are different from existing contrastive learning methods: (i) Hierarchical self-supervision to preserve features at multiple granularities; (ii) Siamese architecture to release the dependency of huge negative samples for avoiding representation collapse. The above two different designs make our HGCL better explore hierarchical structural semantics in graph-structured data via contrastive learning.

4.2. GNN-based encoder

Recently, graph neural networks (GNNs) have been proposed to aggregate feature information of node neighborhood (Kipf & Welling, 2017; Veličković et al., 2017), which have gained increasing attention as powerful tools for a wide range of downstream tasks. Most prevailing methods learn representations of graph-structured data by stacking graph convolution layers, which typically fall into the neighborhood message passing mechanisms (Gilmer et al., 2017). In more detail, following the definition of GNNs in Xu, Hu, Leskovec, and Jegelka (2019), the basic idea of a GNN is to learn node embedding $\mathbf{h}_v^k$ for node v at the kth layer based on an iterative aggregation of neighborhoods, where kth iteration of message passing can be calculated as follows:

$$\begin{aligned}\mathbf{h}_{N(v)}^{(k)} &= \mathcal{AGG}_\theta^{(k)}\left(\left\{\mathbf{h}_u^{(k-1)}, \forall u \in N(v)\right\}\right)\\ \mathbf{h}_v^{(k)} &= \mathcal{COM}_\theta^{(k)}\left(\mathbf{h}_v^{(k-1)}, \mathbf{h}_{N(v)}^{(k)}\right),\end{aligned} \tag{1}$$

where $N(v)$ denotes the neighbors of v. There are varied definitions of $\mathcal{AGG}_\theta^{(k)}$ and $\mathcal{COM}_\theta^{(k)}$ that can be achieved (Hamilton, Ying, & Leskovec, 2017; Kipf & Welling, 2017; Veličković et al., 2017). Finally, the feature vectors at all layers of the GNN are summarized into a single vector:

$$\mathbf{h}_v = \mathcal{CAT}(\{\mathbf{h}_v^k\}_{k=1}^K), \tag{2}$$

where $\mathcal{CAT}$ denotes the vector concatenation operation. In this way, each vector can capture patch information at different scales centered at each node. We stack the node representation into an embedding matrix $\mathbf{H} \in \mathbb{R}^{|V| \times d}$ as:

$$\mathbf{H} = g_\theta(G), \tag{3}$$

where θ is the parameter of the encoder. At last, a readout function is adopted to generate a graph-level representation:

$$\widetilde{\mathbf{h}} = \mathcal{OUT}\left(\{\mathbf{h}_v\}_{v \in V}\right), \tag{4}$$

in which $\widetilde{\mathbf{h}}$ is the graph-level representation and $\mathcal{OUT}$ could be averaging or a more complicated graph pooling operation (Lee et al., 2019; Ying et al., 2018; Zhang et al., 2018).

4.3. Graph augmentations

Data augmentation derives novel rational data via applying certain transformations without changing the semantics (Chen et al., 2020), which is crucial for contrastive learning. Recent graph contrastive learning methods also develop a range of strategies for graphs. In this paper, our HGCL involves node-level contrastive learning and thus we aim to maintain all nodes in augmentation schemes for convenience. Specifically, the HGCL adopts four strategies to generate augmented views of each graph while keeping all nodes (You et al., 2021) as shown in Fig. 1:

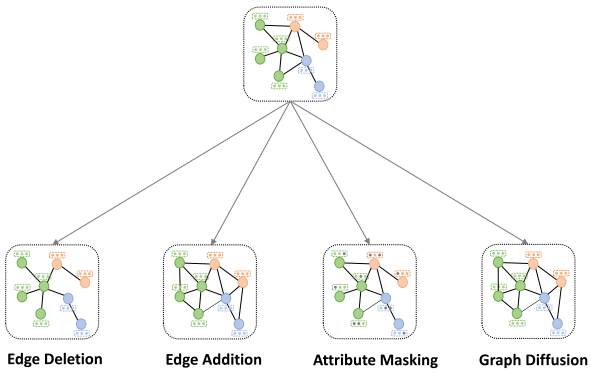


Fig. 1. Illustration of our graph augmentation strategies.

- **Edge Deletion** randomly eliminates some edges from the graph. It assumes that the semantics of the graph is immune to changes in edge connection patterns.
- **Edge Addition** selects two nodes at random. If they are not connected directly but can be reached by a path, we add an edge between the selected two nodes.
- **Attribute Masking** chooses nodes randomly and then masks some of their attributes at random. It is premised on the likelihood that the graph semantics would be resilient with incomplete node attributes.
- **Graph Diffusion** are typically used to generate a congruent view by the Personalized PageRank kernel. The derived augmentation is helpful to provide more comprehensive global information. Let A, D and I denote the adjacency matrix, the degree matrix, and the identity matrix respectively, the diffusion matrix is computed as:

$$S = \alpha \left(I - (1-\alpha)D^{-1/2}AD^{-1/2}\right)^{-1}, \tag{5}$$

in which $\alpha \in (0, 1)$ is a randomly chosen coefficient (Hassani & Khasahmadi, 2020).

In this paper, the proposed HGCL randomly choose one of four graph augmentation strategies to generate two correlated views for each sample. Then, we introduce our graph contrastive learning framework as below.

4.4. Hierarchical graph contrastive learning

As is well known, graphs intrinsically exhibit a diverse range of structural properties, including nodes, edges to subgraphs. The local substructures in a graph always consist of critical characteristics and prominent patterns. Thus, learning about the local substructures and the whole graph is both very important for graph-level representation learning, which can reflect hierarchical structural topology information. The framework of our proposed HGCL is shown in Fig. 2.

4.4.1. Node-level contrastive learning

Existing unsupervised node-level representation learning methods contrast node representations between different views, which is capable of learning effective node representation for downstream tasks. We argue that node representations are also vital for learning graph-level representations due to two principal reasons. On the one hand, node representations may propagate topological structure information from local perspectives. On the other hand, graph-level representations are created directly from node-level representations, implying a tight link between them. As such, we seek to learn informative node representations via contrastive learning for better learning of the whole graph.

Inspired by recent works (Grill et al., 2020; Zbontar, Jing, Misra, LeCun, & Deny, 2021), to avoid the dependency on a large number of negative samples, the HGCL utilizes a Siamese network, which contains two GNNs, namely the online network g_θ and the target network g_ϕ. Two networks have the same encoder architecture but the online network has an additional predictor p_θ on top of the online network g_θ. To produce the bootstrapping contrastiveness, we begin with node representations from one viewpoint in the online network, and maximize the cosine similarity to corresponding representations from another perspective in the target network. Moreover, our approach incorporates additional negative samples to further enhance the fundamental bootstrapping loss as shown in Fig. 3.

Specifically, for each graph G, the HGCL first generates two graph views $\hat{G}^1$ and $\hat{G}^2$, then obtain the node embedding matrices $\mathbf{H}^1 = g_\theta(\hat{G}^1)$ and $\mathbf{H}^2 = g_\phi(\hat{G}^2)$ via the online network and the target network, respectively. Different from existing graph contrastive learning methods, we employ an asymmetric framework where the online network outputs the node embedding matrix $\mathbf{Z}^1 = p_\theta(\mathbf{H}^1)$. Then, the InfoNCE loss (Oord, Li, & Vinyals, 2018) is adopted to distinguish the node embeddings of the same node in two different views from other node embeddings in the graph using both the online network and target network. For any node v, its embedding $\mathbf{z}_v^1$ generated from the online network is viewed as an anchor. The embedding of the same node generated from the target network $\mathbf{h}_v^2$ is viewed as a positive sample while the other embeddings from the same network are treated as negative samples. Formally, the cross-network node-level contrastive learning loss for each node v is defined as:

$$\mathcal{L}^{node,c,1}(v, G) = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_v^1, \mathbf{h}_v^2\right)\right)}{\sum_{v' \in G} \exp\left(\text{sim}\left(\mathbf{z}_v^1, \mathbf{h}_{v'}^2\right)\right)}, \tag{6}$$

where $\text{sim}(\mathbf{z}_v^1, \mathbf{h}_v^2)$ denote the cosine similarity of $\mathbf{z}_v^1$ and $\mathbf{h}_v^2$.

Remark. Note that our framework still involves negative samples, which are essential and natural to bring pairwise relationships in graph-structured data. However, our design of asymmetric architecture does not need to depend on large-scale negative samples to avoid representation collapse. In summary, our model only involves a small size of the negatives that are innate to provide essential signals for more discriminative representation learning.

Moreover, we can switch two augmented inputs of two graph encoders, and get another contrastive loss as:

$$\mathcal{L}^{node,c,2}(v, G) = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_v^2, \mathbf{h}_v^1\right)\right)}{\sum_{v' \in G} \exp\left(\text{sim}\left(\mathbf{z}_v^2, \mathbf{h}_{v'}^1\right)\right)}. \tag{7}$$

The final contrastive learning objective is obtained as the average of two losses over all nodes in V in each graph G:

$$\mathcal{L}^{node,c}(G) = \frac{1}{2|V|} \sum_{v \in V} [\mathcal{L}^{node,c,1}(v, G) + \mathcal{L}^{node,c,2}(v, G)], \tag{8}$$

where $\mathcal{L}^{node,c,1}(v, G)$ and $\mathcal{L}^{node,c,2}(v, G)$ are two symmetric loss to contrast two node views from graph augmentations across different encoder networks.

Following He et al. (2020) and Jin et al. (2021), the target network does not conduct gradient updating during optimization directly. In contrast, the parameters in the target network are updated via a momentum updating strategy as:

$$\phi \leftarrow \eta\phi + (1-\eta)\theta, \tag{9}$$

where η is a momentum coefficient. In this way, parameters in the target network are evolved smoothly.

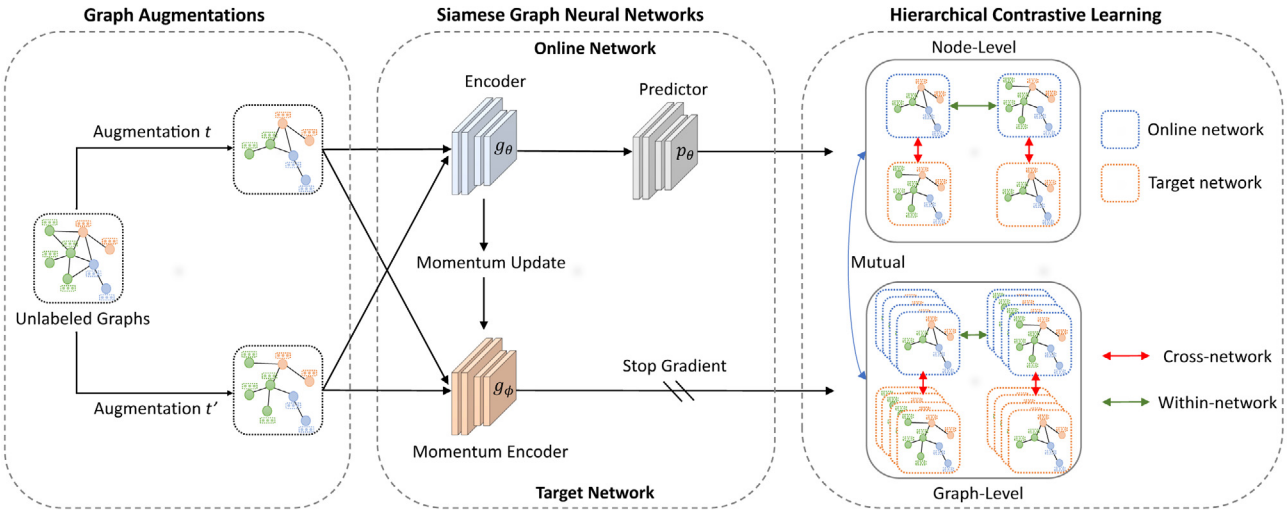


Fig. 2. Overall framework of the HGCL. Our proposed HGCL first leverages graph augmentation to derive two augmented graph views, which are then fed into the online network and target network to generate hierarchical representations, respectively. Our framework integrates node-level contrastive learning, graph-level contrastive learning, and mutual contrastive learning to learn effective graph-level representations. Moreover, gradient updating and momentum updating are conducted for the online network and target network, respectively.

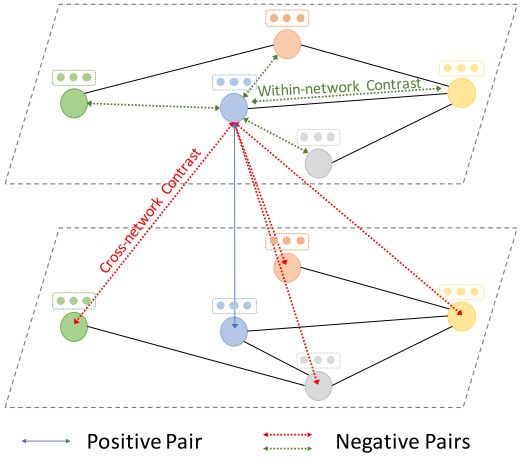


Fig. 3. Illustration of node-level contrastive learning. Cross-network contrast compares pair representations produced by both the online and target networks, while within-network contrast discriminates pair representations from two views in the online network.

However, the above contrastive loss is conducted across two different networks, while the node views within the same network have not been considered yet. Since only the online network is updated by back-propagation, we further contrast the outputs of the online encoder, which acts as an extra regularization for generating informative node representations. Similarly, we distinguish the node embeddings of the same node in two different views from other node embeddings in the graph only using the online network. Specifically, the within-network node-level contrastive loss is defined as:

$$\mathcal{L}^{node,w,1}(v,G) = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_v^1, \mathbf{z}_v^2\right)\right)}{\sum_{v' \in G} \exp\left(\text{sim}\left(\mathbf{z}_v^1, \mathbf{z}_{v'}^2\right)\right)}. \tag{10}$$

Also, two augmented inputs can be switched:

$$\mathcal{L}^{node,w,2}(v,G) = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_v^2, \mathbf{z}_v^1\right)\right)}{\sum_{v' \in G} \exp\left(\text{sim}\left(\mathbf{z}_v^2, \mathbf{z}_{v'}^1\right)\right)}. \tag{11}$$

The final regularization contrastive learning objective is obtained as the average of two regularization losses over all nodes in V in each graph G, formally given by:

$$\mathcal{L}^{node,w}(G) = \frac{1}{2|V|} \sum_{v \in V} [\mathcal{L}^{node,w,1}(v,G) + \mathcal{L}^{node,w,2}(v,G)]. \tag{12}$$

In a nutshell, both the cross-network node-level contrastive loss and within-network node-level contrastive loss are combined to form the final loss for each graph G:

$$\mathcal{L}^{node}(G) = \mathcal{L}^{node,c}(G) + \mathcal{L}^{node,w}(G). \tag{13}$$

4.4.2. Graph-level contrastive learning

Following the popular scheme of graph contrastive learning (You et al., 2021; You, Chen, Sui et al., 2020), the HGCL are encouraged to contrast graph-level representations between different views to enhance the model training. Recall that there are two generated augmented graphs $\hat{G}^1$ and $\hat{G}^2$ for each graph, we thus attempt to summarize all node representations with an extra readout function on top of node embedding matrices, i.e., $\mathbf{H}^1$ and $\mathbf{H}^2$. In this way, we can generate graph-level representations $\widetilde{\mathbf{h}}^1$ and $\widetilde{\mathbf{h}}^2$ for augmented graphs $\hat{G}^1$ and $\hat{G}^2$. The predictor is also adopted to output the embedding $\widetilde{\mathbf{z}}^1 = p_\theta(\widetilde{\mathbf{h}}^1)$. The InfoNCE loss (Oord et al., 2018) is adopted to maximize the similarity between positive sample pairs $\{\widetilde{\mathbf{z}}^1, \widetilde{\mathbf{h}}^2\}$ compared with negative pairs. Technically, we construct a minibatch of B graphs, containing $2B$ augmented graphs $\{\hat{G}_b^1, \hat{G}_b^2\}_{b=1}^B$. For each positive pair $\hat{G}_b^1$ and $\hat{G}_b^2$, the other $(B-1)$ augmented samples in the minibatch are regarded as negatives. After re-annotating $\widetilde{\mathbf{z}}^1$ and $\widetilde{\mathbf{h}}^2$ as $\widetilde{\mathbf{z}}_b^1$ and $\widetilde{\mathbf{h}}_b^2$ for the bth graph in the minibatch, the HGCL contrasts two graph representations across two networks for the bth graph as below:

$$\mathcal{L}^{graph,c,1}(G_b) = -\log \frac{\exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_b^1, \widetilde{\mathbf{h}}_b^2\right)\right)}{\sum_{b'=1}^B \exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_{b'}^1, \widetilde{\mathbf{h}}_{b'}^2\right)\right)}. \tag{14}$$

Similarly, we switch two augmented inputs:

$$\mathcal{L}^{graph,c,2}(G_b) = -\log \frac{\exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_b^2, \widetilde{\mathbf{h}}_b^1\right)\right)}{\sum_{b'=1}^B \exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_{b'}^2, \widetilde{\mathbf{h}}_{b'}^1\right)\right)}. \tag{15}$$

The final graph-level contrastive learning objective for each graph is obtained as the average of two graph contrastive learning losses, formally given by:

$$\mathcal{L}^{graph,c}(G_b) = \frac{1}{2}[\mathcal{L}^{graph,c,1}(G_b) + \mathcal{L}^{graph,c,2}(G_b)]. \tag{16}$$

Moreover, we also consider the ties between two views within the online network, serving as a regularization for generating effective graph representations. Formally,

$$\begin{aligned}\mathcal{L}^{graph,w,1}(G_b) &= -\log \frac{\exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_b^1, \widetilde{\mathbf{z}}_b^2\right)\right)}{\sum_{b'=1}^{B}\exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_{b'}^1, \widetilde{\mathbf{z}}_{b'}^2\right)\right)} \\ \mathcal{L}^{graph,w,2}(G_b) &= -\log \frac{\exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_b^2, \widetilde{\mathbf{z}}_b^1\right)\right)}{\sum_{b'=1}^{B}\exp\left(\text{sim}\left(\widetilde{\mathbf{z}}_{b'}^2, \widetilde{\mathbf{z}}_{b'}^1\right)\right)} \\ \mathcal{L}^{graph,w}(G_b) &= \frac{1}{2}[\mathcal{L}^{graph,w,1}(G_b) + \mathcal{L}^{graph,w,2}(G_b)].\end{aligned} \tag{17}$$

The overall objective function for each graph G_b is defined as the sum of cross-network graph-level contrastive loss and within-network graph-level contrastive loss as:

$$\mathcal{L}^{graph}(G_b) = \mathcal{L}^{graph,c}(G_b) + \mathcal{L}^{graph,w}(G_b). \tag{18}$$

4.4.3. Mutual contrastive learning

After considering both node-level and graph-level contrastive learning, these two types of representations may have a gap that hinders unified representation learning. To alleviate this issue, the proposed HGCL enhances the unity of hierarchical representations by maximizing the mutual information between node-level representations and the global graph-level representations. Since node representations can capture patch information at different scales via message passing, this encourages the encoder to produce consistent graph representations for both the local patch and the whole graph. Specifically, given a batch of graphs, our framework employs the InfoNCE loss to distinguish the two representations of the same graph from other graph embeddings in the minibatch. Since local representations and global representations have a huge difference, representation collapse is not likely to happen. Thus, we do not need to involve the target encoder here. Specifically, for the graph G_b and node $v \in G_b$, the mutual contrastive loss is defined as:

$$\mathcal{L}^{mutmal}(v, G_b) = -\log \frac{\exp\left(\text{sim}\left(\mathbf{z}_v, \widetilde{\mathbf{z}}_b\right)\right)}{\sum_{b'=1}^{B}\exp\left(\text{sim}\left(\mathbf{z}_v, \widetilde{\mathbf{z}}_{b'}\right)\right)}, \tag{19}$$

where $\mathbf{z}_v$ and $\widetilde{\mathbf{z}}_b$ denote the node embedding of node v and graph embedding respectively and B denotes the batch size.

The final mutual contrastive loss for the whole mini-batch is generated by using all possible combinations of global and local patch representations across all graph instances in a batch as:

$$\mathcal{L}^{mutmal} = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{V_b}\sum_{v\in V_b}\mathcal{L}^{mutmal}(v, G_b). \tag{20}$$

4.4.4. Optimization

Finally, our framework minimizes three kinds of contrastive learning objectives to learn hierarchical graph-level representations. In a nutshell, the overall objective function of the HGCL for all graphs in a minibatch can be defined as:

$$\mathcal{L} = \frac{1}{3}\left(\frac{1}{B}\sum_{b=1}^{B}(\mathcal{L}^{node}(G_b) + \mathcal{L}^{graph}(G_b)) + \mathcal{L}^{mutmal}\right). \tag{21}$$

Our objective function is optimized by the minibatch stochastic gradient descent (SGD) method. The whole learning procedure of the HGCL is summarized in Algorithm 1.

Algorithm 1 Learning Algorithm of the HGCL

Input: Unlabeled graphs $\{G_1, \cdots, G_M\}$.
Parameter: Online network parameter θ and target network parameter ϕ.
Output: Target network g_ϕ.

```
Initialize network parameters θ and φ;
while not convergence do
    Construct a minibatch using B samples in the training set;
    for the i-th sample in the minibatch do
        Derive Ĝ_i and Ĝ'_i via a random graph augmentation;
        Obtain node-level representations H^1, H^2 and graph-level
        representations h̃^1, h̃^2 through Siamese network;
        Calculate overall objective function by Eq. (21);
        Update θ by back-propagation;
    end for
    Update φ by momentum update in Eq. (9);
end while
```

5. Experiments

5.1. Experimental settings

Datasets. To evaluate the superiority of our proposed HGCL, we experiment with six widely-used graph classification datasets from TU datasets[2] including three bioinformatics datasets (MUTAG, DD, and PROTEINS) and three social network datasets (IMDB-B, IMDB-M, and COLLAB).

Baselines. To demonstrate the effectiveness of our proposed HGCL, we choose ten baselines from two categories: graph kernel methods (Shortest Path (SP) Kernel (Borgwardt & Kriegel, 2005), Graphlet Kernel (GK) (Shervashidze, Vishwanathan, Petri, Mehlhorn, & Borgwardt, 2009), and Weisfeiler–Lehman (WL) Kernel (Shervashidze, Schweitzer, Van Leeuwen, Mehlhorn, & Borgwardt, 2011)); Unsupervised learning methods (Node2Vec (Grover & Leskovec, 2016), Graph2Vec (Narayanan et al., 2017), InfoGraph (Sun et al., 2020), GraphCL (You, Chen, Sui et al., 2020), JOAO (You et al., 2021), AD-GCL (Suresh, Li, Hao, & Neville, 2021), and RGCL (Li et al., 2022)).

Parameter Settings. In the experiments, we adopt the same encoder architecture GIN (Xu et al., 2019) on all the datasets following InfoGraph (Sun et al., 2020), consisting of two graph convolutional layers with 512 hidden neurons and one sum-pooling layer. The batch size is set to 32. The momentum coefficient η is set to 0.99 following He et al. (2020). As for the graph contrastive learning baselines, we also use GIN to provide a fair comparative study. By closely obeying the assessment criterions used in earlier researches (Sun et al., 2020; You, Chen, Sui et al., 2020), we evaluate the classification accuracy over ten folds using the cross-validation with LIBSVM (Chang & Lin, 2011). We repeat the procedure five times with different random seeds and report the mean accuracy (in %) and standard deviation.

5.2. Experimental results

We conduct a detailed comparison of our proposed HGCL to state-of-the-art baselines and report the quantitative findings of various approaches in Table 1. We make the following observations based on the table:

- From the comparison of kernel methods, it can seen that their performance varies significantly among datasets. For example, WL significantly outperforms the other two methods on social network datasets but performs worse than those on bioinformatics datasets. Perhaps this is because that the semantic gap between datasets in different domains is large, and hand-crafted features cannot always be associated with desired semantic information due to their inability to be learned automatically.

[2] https://ls11-www.cs.tu-dortmund.de/staff/morris/graphkerneldatasets

Table 1
Performance of our model against compared methods on bioinformatics and social network datasets over 5 runs (Averaged accuracy with a standard deviation).

Methods		Datasets					
		MUTAG	DD	PROTEINS	IMDB-B	IMDB-M	COLLAB
Kernel	SP	85.2 ± 2.4	75.5 ± 3.5	75.0 ± 0.5	55.6 ± 0.2	38.0 ± 0.3	49.8 ± 1.2
	GK	81.7 ± 2.1	72.5 ± 3.8	71.6 ± 0.5	65.9 ± 1.0	43.9 ± 0.4	56.3 ± 0.6
	WL	80.7 ± 3.0	74.0 ± 2.3	72.9 ± 0.5	72.3 ± 3.4	47.0 ± 0.5	69.3 ± 3.4
Unsupervised	Node2Vec	72.6 ± 10.2	–	–	50.2 ± 0.9	36.0 ± 0.7	–
	Graph2Vec	83.2 ± 9.6	70.3 ± 2.3	73.3 ± 2.0	71.1 ± 0.5	46.3 ± 1.4	71.1 ± 0.5
	InfoGraph	89.0 ± 1.1	72.9 ± 1.8	74.4 ± 0.5	71.1 ± 0.9	49.7 ± 0.5	70.7 ± 1.1
	GraphCL	86.8 ± 1.3	78.6 ± 0.4	74.3 ± 0.4	71.1 ± 0.4	48.5 ± 0.6	71.4 ± 1.2
	JOAO	87.3 ± 1.0	77.3 ± 0.5	74.5 ± 0.4	70.2 ± 3.1	–	69.5 ± 0.4
	AD-GCL	89.3 ± 1.5	74.5 ± 0.5	73.6 ± 0.7	71.6 ± 1.0	49.0 ± 0.5	73.3 ± 0.6
	RGCL	87.7 ± 1.0	78.9 ± 0.5	75.0 ± 0.4	71.9 ± 0.8	–	70.9 ± 0.7
	HGCL (Ours)	**90.1 ± 0.8**	**79.2 ± 0.6**	**75.5 ± 0.5**	**73.9 ± 0.7**	**51.3 ± 0.5**	**75.8 ± 0.4**

- As can be observed, traditional unsupervised learning methods (i.e., Node2vec and Graph2vec) achieve worse performance compared with other graph contrastive learning techniques, indicating that utilizing efficient graph neural networks can capture important graph structural information for downstream representation learning.
- Among graph contrastive learning baselines, AD-GCL achieves the best results on the majority of datasets. Perhaps the reason is that AD-GCL automatically learns the graph augmentations, and incorporates adversarial learning into graph contrastive learning, which provides a challenging view for generating discriminative representations.
- Our framework HGCL consistently achieves the best performance on all datasets, indicating the superiority of our framework. When compared with the state-of-the-art approach AD-GCL, the performance improvement on DD and IMDB-M is 6.31% and 4.69%, respectively. Given the abundant diversity in various kinds of datasets, our improvement is rather considerable, showing the superiority of our HGCL.

Discussion on the Improvement. Though graph contrastive learning has been broadly explored in earlier researches, these approaches typically suffer from the neglection of hierarchical semantics and dependency of excessive negative samples, while our HGCL provides two critical components to overcome the above two limitations: (i) Introduction of multiple levels of graph contrastive learning. Apart from graph-level contrastive learning, our HGCL also introduces both node-level contrastive learning and mutual contrastive learning for a comprehensive exploration of hierarchical semantics. (ii) Introduction of Siamese architecture. The asymmetric architecture is trained with momentum update, alleviating the representation collapse issue that could happen during the optimization phase.

5.3. Ablation study

To see how different components of our proposed HGCL affect the performance, we conduct an ablation study to validate the contribution of each component. Specifically, three model variants are introduced as follows:

- HGCL w/o node: It removes the node-level graph contrastive learning objective $\mathcal{L}^{node}$.
- HGCL w/o graph: It removes the graph-level graph contrastive learning objective $\mathcal{L}^{graph}$.
- HGCL w/o mutual: It removes the mutual graph contrastive learning objective $\mathcal{L}^{mutual}$.

Results are reported in Table 2, and different datasets have similar findings, which are summarized as follows:

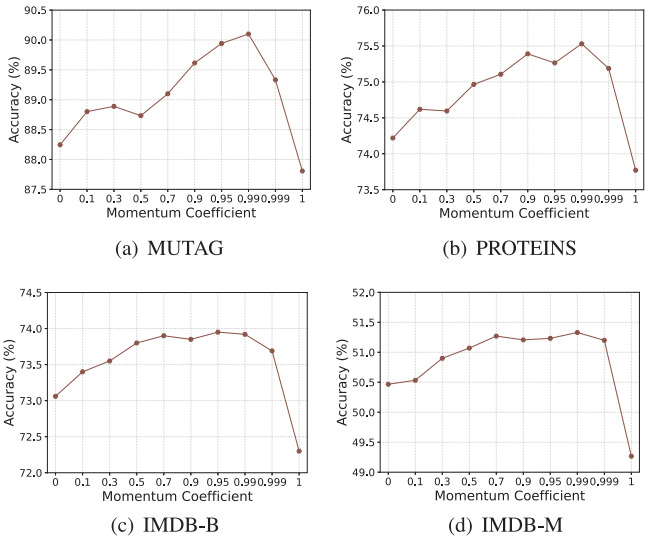


Fig. 4. Performance *w.r.t.* momentum coefficients η.

- To begin, we notice consistent performance increment when all components are organically integrated when comparing our full model to three designed variants, validating the importance and necessity of each component in our proposed HGCL, and thus they are capable of providing a significant contribution to the remarkable performance.
- Second, HGCL w/o graph shows the worst performances among the three variants, which indicates that graph-level contrastive learning is still the most important to generate the whole graph representations. This is in accordance with our intuition since the other two components have indirect effects on graph-level representation learning.

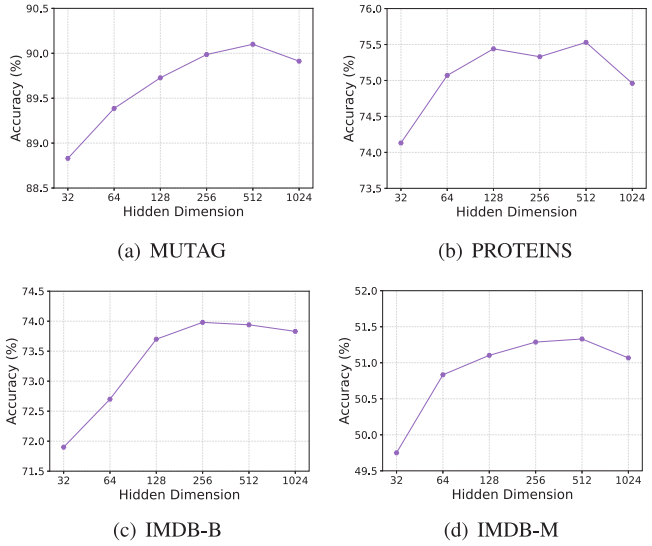
5.4. Parameter sensitivity

Here we investigate the sensitivity of our proposed HGCL to hyper-parameters. Specifically, we study the influence of varying different momentum coefficients and embedding dimensions in hidden layers on four representative datasets MUTAG, PROTEINS, IMDB-B, and IMDB-M. Experimental results show that the findings on other datasets are similar.

Effect of Momentum Coefficient. First, we examine the influence of momentum coefficient η by varying η in the range of $\{0, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99, 0.999, 1\}$ while keeping all other parameters fixed in Fig. 4. It can be observed that too large or too small η may hurt the model performance with the

Table 2
Ablation study of several model variants (in %).

Methods	Datasets					
	MUTAG	DD	PROTEINS	IMDB-B	IMDB-M	COLLAB
HGCL w/o node	89.2 ± 1.1	78.1 ± 0.9	74.4 ± 0.7	73.4 ± 0.8	50.7 ± 0.5	75.2 ± 0.9
HGCL w/o graph	88.7 ± 0.9	77.6 ± 1.1	74.1 ± 0.4	72.8 ± 1.0	50.4 ± 0.8	74.3 ± 0.3
HGCL w/o mutual	89.5 ± 1.2	78.3 ± 0.7	74.9 ± 0.3	73.3 ± 0.6	50.8 ± 0.6	74.6 ± 0.7
Full model	**90.1** ± **0.8**	**79.2** ± **0.6**	**75.5** ± **0.5**	**73.9** ± **0.7**	**51.3** ± **0.5**	**75.8** ± **0.4**



(a) MUTAG (b) PROTEINS

(c) IMDB-B (d) IMDB-M

Fig. 5. Performance *w.r.t.* hidden dimension d.

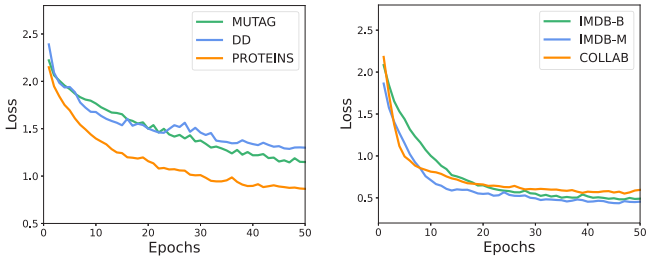
following explanation. When $\eta = 1$, the target network is not updated, which results in quite poor performance. When $\eta = 0$, the update of the target network is too unstable to reach the optimal performance. The empirical value of 0.99 seems to be appropriate for satisfactory results.

Effect of Hidden Dimension. Then, we investigate the influence of varying embedding dimensions in hidden layers d on four datasets. We use different d in $\{32, 64, 128, 256, 512, 1024\}$ with all other parameters fixed. The results are illustrated in Fig. 5. It can be shown that when we increase d from 32 to 512, the classification performance consistently becomes better. However, further increasing d may not be beneficial to the model prediction accuracy. The potential reason could be that a large hidden dimension would effectively bring a stronger representation capability of the model, but too large dimensions may result in over-fitting and poor generalization.

5.5. Empirical convergence

In this part, we plot the training curves of our proposed HGCL in Fig. 6. The report is recorded on all datasets, which can be categorized into two main groups: bioinformatics and social networks. Though the convergence is not guaranteed theoretically, we can observe that on all datasets, our proposed HGCL consisting of multiple levels of graph contrastive learning works well and achieves empirical convergence in practice, validating the effectiveness of exploring the hierarchical structural semantics of a graph at both node and graph levels.

Furthermore, it can be observed that all datasets can converge within 20 epochs except the instability for MUTAG and PROTEINS datasets, which further proves the advantage of the fast convergence rate of our proposed HGCL. The volatility of curves for MUTAG and PROTEINS may attribute to the small size of these



(a) Bioinformatics (b) Social Networks

Fig. 6. The training curves of our proposed approach HGCL on bioinformatics and social networks datasets.

two datasets. Maybe the reason for the fast convergence rate on all datasets is that our hierarchical self-supervision keeps features at multiple granularities, so that the generated graph representations could always be informative for downstream tasks. Additionally, the Siamese network and momentum update are further involved to reduce the requirement for huge negatives and avoid the representation collapse, and makes graph-level representation learning more stable and efficient.

5.6. Transfer learning

To evaluate our HGCL on large-scale datasets, we perform transfer learning for predicting molecular properties. Following Hu, Liu et al. (2020), we pre-train our HGCL using self-supervised learning on a large-scale ZINC15 database (Gaulton et al., 2012; Mayr et al., 2018) and later fine-tune it on various Open Graph Benchmark (OGB) datasets (Hu, Fey et al., 2020) to test out-of-distribution performance. In this way, we study the transferability of the various pre-training strategies.

Datasets. We evaluate our model HGCL on six benchmark OGB molecule property prediction datasets in the experiment. For graph-level self-supervised pre-training, we adopt a subset of the ZINC15 database, which involves two million unlabeled molecules following the previous settings (Hu, Liu et al., 2020). For downstream classification tasks, six large-scale OGB datasets in Moleculenet (Wu et al., 2018) are utilized to validate model performance, with the scaffold split scheme adopted for dataset split (Chen, Sheridan, Hornak, & Voigt, 2012).

Experiments Settings. For pre-training, GIN are used as our GNN-based encoder with 300 hidden units for performance evaluation as indicated in Xu et al. (2019). For fine-tuning, an extra linear predictor is trained on top of the pre-trained encoder with 100 training epochs. We compare our HGCL with non-pretrain (without self-supervised training on ZINC15 and with only fine-tuning) and existing graph pre-training algorithms. To be specific, apart from graph contrastive learning methods GraphCL (You, Chen, Sui et al., 2020), JOAO (You et al., 2021), AD-GCL (Suresh et al., 2021), and RGCL (Li et al., 2022), we also include five different pre-training techniques including EdgePred (Kipf & Welling, 2016), Infomax (Velickovic et al., 2019), AttrMasking (Hu, Liu et al., 2020), ContextPred (Hu, Liu et al., 2020), and GraphPartition (You, Chen, Wang, & Shen, 2020).

Table 3
Results on downstream molecular property prediction benchmarks.

Methods	Datasets					
	BBBP	ClinTox	ToxCast	MUV	HIV	BACE
No Pre-Train	65.8 ± 4.5	58.0 ± 4.4	63.4 ± 0.6	71.8 ± 2.5	75.3 ± 1.9	70.1 ± 5.4
EdgePred	67.3 ± 2.4	64.1 ± 3.7	64.1 ± 0.6	74.1 ± 2.1	76.3 ± 1.0	79.9 ± 0.9
Infomax	68.8 ± 0.8	69.9 ± 3.0	62.7 ± 0.4	75.3 ± 2.5	76.0 ± 0.7	75.9 ± 1.6
AttrMasking	64.3 ± 2.8	71.8 ± 4.1	**64.2 ± 0.5**	74.7 ± 1.4	77.2 ± 1.1	79.3 ± 1.6
ContextPred	68.0 ± 2.0	65.9 ± 3.8	63.9 ± 0.6	75.8 ± 1.7	77.3 ± 1.0	79.6 ± 1.2
GraphPartition	70.3 ± 0.7	64.2 ± 0.5	63.2 ± 0.3	75.4 ± 1.7	77.1 ± 0.7	79.6 ± 1.8
GraphCL	69.7 ± 0.7	76.0 ± 2.7	62.4 ± 0.6	69.8 ± 2.7	78.5 ± 1.2	75.4 ± 1.4
JOAO	70.2 ± 1.0	81.3 ± 2.5	63.0 ± 0.5	71.7 ± 1.4	76.7 ± 1.2	77.3 ± 0.5
AD-GCL	70.0 ± 1.1	79.8 ± 3.5	63.1 ± 0.7	72.3 ± 1.6	78.3 ± 1.0	78.5 ± 0.8
RGCL	71.4 ± 0.7	83.4 ± 0.9	63.3 ± 0.2	76.7 ± 1.0	77.9 ± 0.8	76.0 ± 0.8
HGCL (Ours)	**73.6 ± 1.3**	**84.2 ± 0.8**	64.0 ± 0.4	**78.3 ± 1.2**	**78.8 ± 0.9**	**80.2 ± 1.0**

Performance Analysis. We report the performance of the proposed HGCL with other competing baselines in the transfer learning setting in Table 3. It can be observed that our developed HGCL outperforms all other baselines on five of six datasets. Specifically, we gain a 14.4% performance increment on dataset BACE against the non-pretrain baseline, validating the effectiveness of our HGCL on large-scale transfer learning. Among competing baselines, the best performance of each dataset is scattered, indicating the significant differences in the properties of distinct downstream datasets. Our approach, however, consistently obtains the best performance among the majority of datasets. Additionally, our HGCL outperforms the best pre-training strategy GraphPartition and other contrastive learning methods, showing the superiority of our proposed HGCL.

6. Conclusion

This paper studies unsupervised graph-level representation learning, and a novel framework called the HGCL is proposed, which studies the hierarchical structural semantics of a graph at both node and graph levels. Specifically, HGCL consists of three parts, i.e., node-level contrastive learning, graph-level contrastive learning, and mutual contrastive learning to explore graph semantics in a principled way. Moreover, the Siamese network and momentum update are further involved to reduce the requirement for excessive negatives. Extensive experiments on various graph classification datasets and large-scale OGB datasets validate the superiority of the proposed framework. In the future, our works will further extend our framework to other promising domains such as healthcare, finance and security.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This paper is partially supported by the National Key Research and Development Program of China with Grant No. 2018 AAA0101902 and the National Natural Science Foundation of China (NSFC Grant Numbers 62276002 and 62106008).

References

Ali, A., Zhu, Y., & Zakarya, M. (2022). Exploiting dynamic spatio–temporal graph convolutional neural networks for citywide traffic flows prediction. *Neural Networks*, *145*, 233–247.
Baek, J., Kang, M., & Hwang, S. J. (2021). Accurate learning of graph representations with graph multiset pooling. In *ICLR*.
Borgwardt, K. M., & Kriegel, H. -P. (2005). Shortest-path kernels on graphs. In *ICDM*.
Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., et al. (1993). Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, *7*(04), 669–688.
Cai, Q., Gong, M., Shen, B., Ma, L., & Jiao, L. (2014). Discrete particle swarm optimization for identifying community structures in signed social networks. *Neural Networks*, *58*, 4–13.
Chang, C. -C., & Lin, C. -J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–27.
Chen, X., & He, K. (2021). Exploring simple siamese representation learning. In *CVPR* (pp. 15750–15758).
Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *ICML*.
Chen, B., Sheridan, R. P., Hornak, V., & Voigt, J. H. (2012). Comparison of random forest and pipeline pilot naive Bayes in prospective QSAR predictions. *Journal of Chemical Information and Modeling*, *52*(3), 792–803.
Chu, G., Wang, X., Shi, C., & Jiang, X. (2021). CuCo: Graph representation with curriculum contrastive learning. In *IJCAI*.
Devlin, J., Chang, M. -W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
Duan, Z., Xu, H., Wang, Y., Huang, Y., Ren, A., Xu, Z., et al. (2022). Multivariate time-series classification with hierarchical variational graph pooling. *Neural Networks*, *154*, 481–490.
Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., et al. (2012). ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, *40*(D1), D1100–D1107.
Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *ICML*.
Grill, J. -B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*.
Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *SIGKDD*.
Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *NeurIPS*.
Hao, Z., Lu, C., Huang, Z., Wang, H., Hu, Z., Liu, Q., et al. (2020). ASGN: An active semi-supervised graph neural network for molecular property prediction. In *KDD*.
Hassani, K., & Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. In *ICML*.
He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *CVPR*.
Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., et al. (2020). Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687.
Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., et al. (2020). Strategies for pre-training graph neural networks. In *ICLR*.
Jiang, B., Chen, S., Wang, B., & Luo, B. (2022). MGLNN: Semi-supervised learning via multiple graph cooperative learning neural networks. *Neural Networks*.
Jiang, B., Kloster, K., Gleich, D. F., & Gribskov, M. (2017). AptRank: An adaptive PageRank model for protein function prediction on bi-relational graphs. *Bioinformatics*, *33*(12), 1829–1836.

Jin, M., Zheng, Y., Li, Y. -F., Gong, C., Zhou, C., & Pan, S. (2021). Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *IJCAI*.

Ju, W., Luo, X., Ma, Z., Yang, J., Deng, M., & Zhang, M. (2022). GHNN: Graph harmonic neural networks for semi-supervised graph-level classification. *Neural Networks*, *151*, 70–79.

Ju, W., Qin, Y., Qiao, Z., Luo, X., Wang, Y., Fu, Y., et al. (2022). Kernel-based substructure exploration for next POI recommendation. arXiv preprint arXiv:2210.03969.

Ju, W., Yang, J., Qu, M., Song, W., Shen, J., & Zhang, M. (2022). KGNN: Harnessing kernel-based networks for semi-supervised graph classification. In *WSDM*.

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. In *NeurIPS workshop on bayesian deep learning*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kojima, R., Ishida, S., Ohta, M., Iwata, H., Honma, T., & Okuno, Y. (2020). kGCN: A graph-based deep learning framework for chemical structures. *Journal of Cheminformatics*, *12*, 1–10.

Lee, J., Lee, I., & Kang, J. (2019). Self-attention graph pooling. In *ICML*.

Li, S., Wang, X., Zhang, A., Wu, Y., He, X., & Chua, T. -S. (2022). Let invariant rationale discovery inspire graph contrastive learning. In *International conference on machine learning* (pp. 13052–13065). PMLR.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., et al. (2021). Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

Luo, X., Ju, W., Qu, M., Chen, C., Deng, M., Hua, X. -S., et al. (2022). DualGraph: Improving semi-supervised graph classification via dual contrastive learning. In *ICDE*.

Luo, X., Ju, W., Qu, M., Gu, Y., Chen, C., Deng, M., et al. (2022). CLEAR: Cluster-enhanced contrast for self-supervised graph representation learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J. K., Ceulemans, H., et al. (2018). Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science*, *9*(24), 5441–5451.

Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., & Jaiswal, S. (2017). Graph2vec: Learning distributed representations of graphs. arXiv preprint arXiv:1707.05005.

Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Qiu, J., Chen, Q., Dong, Y., Zhang, J., Yang, H., Ding, M., et al. (2020). GCC: Graph contrastive coding for graph neural network pre-training. In *KDD*.

Rassil, A., Chougrad, H., & Zouaki, H. (2022). Augmented graph neural network with hierarchical global-based residual connections. *Neural Networks*, *150*, 149–166.

Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., & Borgwardt, K. M. (2011). Weisfeiler–Lehman graph kernels. *Journal of Machine Learning Research*, *12*(9), 2539–2561.

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., & Borgwardt, K. (2009). Efficient graphlet kernels for large graph comparison. In *AISTATS*.

Sun, F. -Y., Hoffmann, J., Verma, V., & Tang, J. (2020). Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*.

Suresh, S., Li, P., Hao, C., & Neville, J. (2021). Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. In *ICLR*.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2019). Deep graph infomax. In *ICLR*.

Wang, H., Liao, X., Wang, Z., Huang, T., & Chen, G. (2016). Distributed parameter estimation in unreliable sensor networks via broadcast gossip algorithms. *Neural Networks*, *73*, 1–9.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., et al. (2018). MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, *9*(2), 513–530.

Xie, Y., Zhang, Y., Gong, M., Tang, Z., & Han, C. (2020). MGAT: Multi-view graph attention networks. *Neural Networks*, *132*, 180–189.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? In *ICLR*.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., & Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*.

You, Y., Chen, T., Shen, Y., & Wang, Z. (2021). Graph contrastive learning automated. In *ICML*.

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

You, Y., Chen, T., Wang, Z., & Shen, Y. (2020). When does self-supervision help graph convolutional networks? In *ICML*.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*.

Zeng, J., & Xie, P. (2021). Contrastive self-supervised learning for graph classification. In *AAAI*.

Zhang, J., Cao, J., Huang, W., Shi, X., & Zhou, X. (2022). Rutting prediction and analysis of influence factors based on multivariate transfer entropy and graph neural networks. *Neural Networks*.

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

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., & Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *WWW*.