DualGraph: Improving Semi-supervised Graph Classification via Dual Contrastive Learning

Xiao Luo[1,*], Wei Ju[1,*], Meng Qu[2,3], Chong Chen[4], Minghua Deng[1,†], Xian-Sheng Hua[4] and Ming Zhang[1,†]
[1]*Peking University,* [2]*Mila - Quebec AI Institute,* [3]*Université de Montréal,* [4]*DAMO Academy, Alibaba Group*
[1]{*xiaoluo,juwei,dengmh,mzhang_cs*}*@pku.edu.cn,* [2,3]*meng.qu@umontreal.ca,* [4]{*cheung.cc,xiansheng.hxs*}*@alibaba-inc.com*

2022 IEEE 38th International Conference on Data Engineering (ICDE) | 978-1-6654-0883-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDE53745.2022.00057

***Abstract*—In this paper, we study semi-supervised graph classification, a fundamental problem in data mining and machine learning. The problem is typically solved by learning graph neural networks with pseudo-labeling or knowledge distillation to incorporate both labeled and unlabeled graphs. However, these methods usually either suffer from overconfident and biased pseudo-labels or suboptimal distillation caused by the insufficient use of unlabeled data. Inspired by the recent progress of contrastive learning and dual learning, we propose DualGraph, a principled framework to leverage unlabeled graphs more effectively for semi-supervised graph classification. DualGraph consists of a prediction module and a retrieval module to model graphs G and their labels y from opposite while complementary views (i.e., $p(y|G)$ and $p(G|y)$ respectively). The two modules are jointly trained via posterior regularization, which encourages their inter-module consistency on unlabeled graphs. Moreover, we improve model training for each module with a contrastive learning framework to encourage the intra-module consistency on unlabeled data. Experimental results on a range of publicly accessible datasets reveal the effectiveness of our DualGraph.**

***Index Terms*—Graph Classification, Graph Neural Networks, Contrastive learning, Semi-supervised Learning**

I. Introduction

Graphs have demonstrated increasing significance for representing structured and relational data in a variety of domains. Graph classification, as a fundamental problem in machine learning, attempts to predict the property of the whole graph, which has been extensively studied recently and has many downstream applications. Some important applications include predicting the quantum mechanical properties [1] and assessing the functionality of chemical compounds [2] (e.g., whether the compound is mutagen or non-mutagen).

In literature, researchers have proposed many machine learning techniques for graph classification [3]–[6]. Among them, graph neural networks (GNNs) [7]–[9] achieve the best results. The key idea of GNNs is to learn discriminative graph representations using neighbor-aware message passing algorithms [10]–[13]. To be particular, each node receives information from all of its neighbors, which is then aggregated to incrementally update the node representation. At last, a readout function is used to combine all of the node representations into a graph-level representation. In this manner, the learned graph representation can reflect the structural topology of the graph for classification.

Despite their high performance, current GNN methods often need a significant quantity of labeled data for training (e.g., molecules with known characteristics in chemical scenarios). Nevertheless, annotated labels are often prohibitively expensive [9]. For instance, labels are often generated using the costly Density Functional Theory (DFT) calculations or complicated experiments in chemical activities. Also, graph annotation in some specific domains highly relies on domain experts. Consequently, the labeled samples usually take a tiny portion of the total training data, severely limiting the performance of GNN models. In practice, there usually exist a wealth of available unlabeled graphs. Although their properties (i.e., labels) are unknown, their structures include information that may help enhance the GNN-based encoders if these unlabeled graphs can be properly used. As a consequence, in this paper, we investigate the problem of semi-supervised graph classification, which leverages both labeled data and unlabeled data to predict graph properties.

Indeed, a few semi-supervised methods have been proposed for graph classification, which combine GNNs with semi-supervised learning techniques. In general, these methods can be divided into two categories, i.e., pseudo-labeling methods [14] and knowledge distillation methods [1], [15]. Pseudo-labeling methods repeatedly annotate unlabeled graphs, and treat graphs with high-confidence predictions as additional training data to improve model training. For example, SEAL-AI [14] annotates the unlabeled data with a GNN classifier and then utilizes the most confident data to improve the other classifier with pseudo-labels. Knowledge distillation methods usually employ a teacher-student architecture [16], where the teacher model usually takes the lead for learning discriminative graph representations, the student model is fine-tuned for the downstream task. For instance, InfoGraph [15] consists of two encoders with the same architecture. Two encoders target different objectives that are trained by maximizing the mutual information of their representations.

However, the performance of existing semi-supervised graph classification methods remains unsatisfactory due to the following limitations. (1) *Biased pseudo-labels.* While the pseudo-labeling methods annotate unlabeled samples as extra training data, they are prone to generating overconfident and skewed findings (e.g., incorrect pseudo-labels [14]) for unlabeled data, especially when the amount of labeled data is often scarce. (2) *Insufficient use of unlabeled data.* In the teacher-student framework, the student model is usually fine-

[*]Equal contribution with order determined by flipping a coin. This work was done when Xiao Luo interned in Alibaba Group.
[†]Corresponding authors.

2375-026X/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDE53745.2022.00057

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

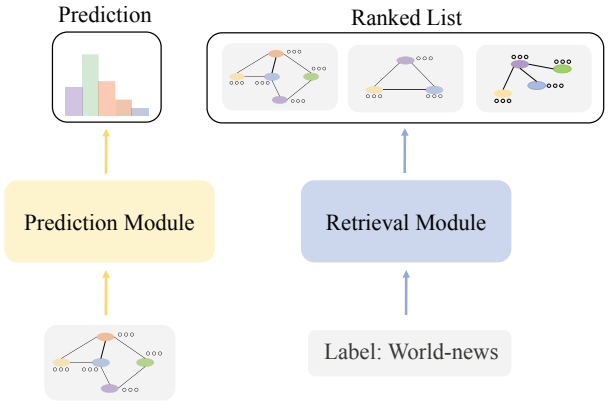


Fig. 1: Illustration of the two modules in DualGraph. Left: The prediction module predicts the label distribution of a given graph. Right: The retrieval module focuses on the dual task, which aims to rank the graphs of each given label from a ranked list by computing matching scores.

tuned in a supervised manner [1], where the unlabeled data is not fully explored. This makes it hard to improve models when labeled data is limited in practice. Therefore, we expect an approach which is able to leverage unlabeled data more efficiently and produce more reliable pseudo-labels for semi-supervised graph classification.

Inspired by the recent success of techniques on contrastive learning [17]–[19] and dual learning [20]–[23], in this paper we propose a principled framework called the DualGraph for semi-supervised graph classification. The key to graph classification is to understand the relationship between graphs G and their labels y. To better model such relationships, DualGraph adopts a prediction module and a retrieval module, which are motivated by two complementary perspectives. The prediction module predicts the labels of given graphs and thus models the distribution $p(y|G)$, while the retrieval module retrieves graphs of each given label and thereby models the distribution $p(G|y)$ as shown in Figure 1. In order to train both modules jointly and let them mutually enhance each other, we propose to maximize their inter-module consistency and intra-module consistency on unlabeled graphs. For the inter-module consistency, we leverage posterior regularization [24] and encourage both modules to derive close joint distributions on graphs and labels. In this way, both modules can mutually correct and further produce more reliable pseudo-labels on unlabeled graphs to benefit each other. For the intra-module consistency, we leverage contrastive learning techniques, which ensure that each single module gives consistent outputs on each unlabeled graph and its corresponding augmented graph. Specifically, we force the predicted labels from the prediction module to be similar on original and augmented graphs, and meanwhile the InfoNCE loss [18], [25] is used to ensure the consistency of the matching scores from the retrieval module on both of the original and augmented graphs. With this contrastive learning framework on unlabeled graphs, DualGraph is able to leverage unlabeled graphs more effectively compared with existing methods. We further propose an EM-styled algorithm for training DualGraph, and the algorithm alternates between an E-step and an M-step. In the E-step, we fix the prediction module and update the retrieval module, and in the M-step, the retrieval module is fixed to update the prediction module. We conduct extensive experiments on various widely-used datasets to evaluate the DualGraph. Experimental results show that DualGraph outperforms a wide range of state-of-the-art methods under different settings. In summary, our primary contributions are as follows:

- We propose a novel framework for semi-supervised graph classification, consisting of a prediction module and a retrieval module to encourage the consistency on unlabeled data and overcome the unreliability of pseudo-labels and scarcity of labeled data.
- We leverage dual learning with posterior regularization to encourage the inter-module consistency and take advantage of contrastive learning to encourage the intra-module consistency. Furthermore, an EM-styled algorithm is proposed to alternatively optimize both modules for consistency enhancement.
- We conduct extensive experiments on a variety of publicly available datasets to evaluate the DualGraph. Experimental results and a case study validate the efficacy of the proposed framework under different settings.

II. Related Work

A. Semi-supervised Graph Classification

Graph Classification. Predicting the properties of graphs is a fundamental issue in a variety of areas, including chemistry and biology [1], [2]. Traditional methods such as graph kernels [26]–[28], which decompose graphs into substructures and use kernel functions to capture the graph similarity, have achieved great success. However, these methods suffer from poor scalability owing to the ineffective handcrafted features. Inspired by the progress of graph neural networks (GNNs), researchers have discovered their potential in a wide range of fields [11]–[13]. Variants of GNNs have been proposed for this problem [15], [29], [30]. Among these methods, the node representations are propagated and updated using the representations of their neighbors, following the scheme of message passing. Finally, the graph-level representations can be obtained via graph pooling functions [5], [6] for downstream applications. Nonetheless, these approaches usually require a large amount of labeled data for training. Also, annotating labels is often too expensive, rendering them unsuitable for real-world applications. To tackle this issue, our work studies semi-supervised graph classification, fully leveraging the unlabeled data and overcoming the labeled data scarcity.

Semi-supervised learning. Our work is related to semi-supervised learning. Pseudo-labeling is an early technique in this vein. It adopts a trained classifier to predict the label distributions of the unlabeled samples and expand the training set with well-classified data. For instance, the entropy

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

minimization strategy [31] enforces the classifier to make predictions with low entropy on unlabeled data. Another common technique of semi-supervised learning is consistency learning [32]–[36]. These approaches assume that the model should output consistency predictions when fed perturbation to input samples. Among these, the temporal ensembling model [32], for example, utilizes an exponential moving average strategy where a mean teacher model [37] averages the parameters of networks to provide a stable prediction.

Furthermore, a few researches for semi-supervised graph classification [1], [14], [15], [19], [38] have been proposed. These works are often either based on pseudo-labeling or knowledge distillation. However, they may easily suffer from biased pseudo-labels and the suboptimal distillation resulting from insufficient use of unlabeled data. Compared with these works, our proposed DualGraph benefits from dual learning and contrastive learning, consisting of a prediction module and a retrieval module to encourage the inter-module and intra-module consistency on unlabeled graphs, and hence overcomes the unreliability of pseudo-labels and scarcity of labeled data.

B. Contrastive Learning for Graph Neural Networks

Contrastive learning (CL) is based on the principle of utilizing self-supervised information between contrastive pairs produced via random perturbation of the original data [18], [25]. Recent efforts have been made to apply CL to GNNs [15], [39]–[43]. As a pioneering effort, DGI [39] augments the original graph by simply shuffling node features and then offers a contrastive objective of maximizing the mutual information between node representations and the global graph representation. InfoGraph [15] blends supervised learning and unsupervised learning in the semi-supervised setting by employing a mean teacher approach [37]. This method consists of two encoders that are trained by maximizing the mutual information of their representations. Zhu et al. [43] developed adaptive graph augmentations to incorporate different priors of the topological and semantic property of graphs. Despite the advance of graph CL techniques, the problem of incorporating graph contrastive learning into the semi-supervised scenario has been critical and less unexplored. In this paper, we propose a novel contrastive learning framework to encourage the intra-module consistency in a semi-supervised way, while existing works fail to adapt to this setting.

C. Dual Learning

Another category of related work is dual learning [20]–[23], [44], which can be generalized to the concept of simultaneously learning the primal and dual modules to mutually enhance each other. For instance, English-French translation with French-English translation [20], code generation with code summarization [45], visual question answering with visual question generation [23] can all be combined together to exploit the duality of two tasks. Our proposed DualGraph extends the idea of dual learning into the graph-structured data. In our work, predicting the label of given graphs and retrieving the graphs of each given label are treated as dual tasks which can be enhanced collaboratively. In contrast to their supervised settings, we incorporate dual learning into semi-supervised scenarios and propose a novel framework to effectively alleviate the biased pseudo-labels. To the best of our knowledge, we are the first to perform dual learning for the graph classification task.

III. Problem Definition

In this section, we will provide formal definitions of terminologies in this paper for the sake of clarity. Then we formalize the problem definition.

Definition 1: Graph. Denote a graph as $G = (V, E, \mathbf{X})$, where V represents the vertex set and E represents the edge set. We use x_v to denote the attribute vector of the node v and $\mathbf{X} \in R^{|V| \times d}$ represents the node attribute matrix, and d is the dimension of the attribute vector.

Semi-supervised graph classification is a fundamental problem in machine learning and data mining that has various applications, including predicting the quantum mechanical properties of molecules and analyzing the functionality of chemical compounds. Given labeled and unlabeled graphs, semi-supervised graph classification attempts to predict the label distributions of unlabeled graphs. Formally, we define the semi-supervised graph classification as follows:

Definition 2: Semi-supervised Graph Classification. Consider a set of graphs $\mathcal{G} = \{\mathcal{G}^L, \mathcal{G}^U\}$, where labeled graphs are $\mathcal{G}^L = \left\{G_1, \cdots, G_{|\mathcal{G}^L|}\right\}$ and unlabeled graphs are $\mathcal{G}^U = \left\{G_{|\mathcal{G}^L|+1}, \cdots, G_{|\mathcal{G}^L|+|\mathcal{G}^U|}\right\}$. Let $\mathcal{Y}^L = \{y_1, \cdots, y_{|\mathcal{G}^L|}\}$ represent the labels corresponding to $\mathcal{G}^L$, and $\mathcal{T} = \{(G_i, y_i)\}_{i=1}^{|\mathcal{G}^L|}$ denote the graph-label pair. Semi-supervised graph classification aims to learn a multi-class classification model that can annotate labels to unlabeled graphs $\mathcal{G}^U$.

IV. Methodology

Next, we introduce our proposed framework DualGraph.

A. Framework Overview

The paper proposes the DualGraph for semi-supervised graph classification based on dual contrastive learning. Existing methods typically learn graph neural networks with pseudo-labeling or knowledge distillation. However, these methods usually either suffer from biased pseudo-labels or the suboptimal distillation resulting from the insufficient use of unlabeled data. As a result, their performance is not yet satisfactory when the labeled data is very limited.

In order to make better use of unlabeled graphs and generate more reliable pseudo-labels, DualGraph leverages the idea of dual learning and contrastive learning. More specifically, DualGraph has a prediction module to predict graph labels and a retrieval module to retrieve graphs of a given label. The two modules are trained by maximizing their inter-module and intra-module consistency on unlabeled graphs. To encourage the inter-module consistency, we leverage dual learning with posterior regularization to let both modules reach agreement on the categories of unlabeled graphs. This further allows both modules to provide cleaner pseudo-labels of unlabeled

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

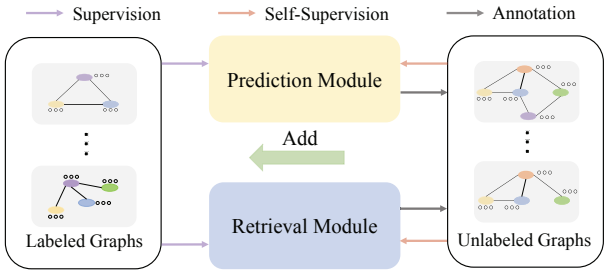


Fig. 2: A diagram of our framework. DualGraph consists of a prediction module and a retrieval module, which are trained with both supervision and self-supervision in a semi-supervised environment. Moreover, they collaborate to generate newly labeled data, which is then utilized to enhance both modules as supervised signals.

graphs to benefit each other. For the intra-module consistency, we encourage each module to give similar output on an unlabeled graph and its augmented graphs. In this way, each module is able to leverage unlabeled graphs more sufficiently to regularize the GNN-based encoders.

Formally, our framework is composed of a prediction module $\mathcal{P}_\theta$ and a retrieval module $\mathcal{Q}_\phi$, in which θ and ϕ are the model parameters. Here the prediction module attempts to predict the label for a given labeled graph. Therefore, it models the probability $p_\theta(y|G)$ for graph-label pair (G, y). By contrast, the retrieval module retrieves relevant graphs with the given label y by producing the probability $q_\phi(G|y)$ for a graph-label pair (G, y). The overall objective function is provided below:

$$\mathcal{L} = \mathcal{L}_P + \mathcal{L}_Q + \mathcal{L}_C \tag{1}$$

Our objective function contains three terms in total. In $\mathcal{L}_P$ and $\mathcal{L}_Q$, we utilize both labeled graphs and unlabeled graphs to train the prediction module and the retrieval module, respectively. As for the collaborative objective $\mathcal{L}_C$, we make full use of two modules to collaboratively select highly confident unlabeled graphs as additional labeled graphs for supervised training. An illustration of our DualGraph is shown in Figure 2 and we will go through the specifics of each component.

B. GNN-based Encoder

Graph neural networks (GNNs) [10], [29], [46] have recently emerged as promising approaches to learn the representation of graph-structured data. The embedding vector for node v at the k-th layer is denoted by $\mathbf{h}_v^k$. Prevailing methods for learning graph representations are based on graph neural networks with neural message passing mechanisms. To be more specific, for each node $v \in V$, firstly the embedding vectors from the neighbors of v at layer $k-1$ are aggregated. The node representation $\mathbf{h}_v^k$ is then computed iteratively by combining the embedding of v in the last layer with the aggregated neighbor embedding to the embedding of v of the current layer. Formally, the embedding of node v in the k-th layer is calculated as follows:

$$\mathbf{h}_{N(v)}^{(k)} = \mathcal{A}_{\theta_e}^{(k)}\left(\left\{\mathbf{h}_u^{(k-1)}, \forall u \in N(v)\right\}\right) \tag{2}$$

$$\mathbf{h}_v^{(k)} = \mathcal{C}_{\theta_e}^{(k)}\left(\mathbf{h}_v^{(k-1)}, \mathbf{h}_{N(v)}^{(k)}\right) \tag{3}$$

where $N(v)$ denotes the neighbors of v. Here $\mathcal{A}_{\theta_e}^{(k)}$ and $\mathcal{C}_{\theta_e}^{(k)}$ denote the aggregation and combination functions in the k-th layer, respectively. Finally, the graph-level representation can be attained by aggregating all node representations in the K-th layer with a readout function. Formally,

$$f_{\theta_e}(G) = \text{READOUT}\left(\left\{\mathbf{h}_v^K\right\}_{v \in V}\right) \tag{4}$$

in which $f_{\theta_e}(G)$ is the graph-level representation, θ_e is the parameter of encoder, and READOUT represents averaging or a more sophisticated graph-level pooling function [5], [6], [30]. In this paper, we follow InfoGraph [15] and adopt GIN [29] to learn node representation due to its powerful expression ability. For conciseness, the sum operation is used to produce the graph-level representation.

C. Semi-supervised Contrastive Prediction Module

In our framework, the prediction module $\mathcal{P}_\theta$ focuses on the primary task of predicting graph labels. We begin by using the GNN-based encoder mentioned earlier to get the graph embedding $\mathbf{z}$ for each graph G. After that, a multi-layer perception (MLP) classifier $\mathcal{H}_{\theta_h}(\cdot)$ is used to predict the labels for the labeled graphs. Formally,

$$\mathbf{z} = f_{\theta_e}(G) \tag{5}$$

$$p_\theta(y|G) = \mathcal{H}_{\theta_h}(\mathbf{z}) = \mathcal{H}_{\theta_h}(f_{\theta_e}(G)) \tag{6}$$

Here $\theta = \{\theta_e, \theta_h\}$, where θ_e represents parameters of the encoder and θ_h represents parameters of the MLP classifier. The prediction module consists of a supervised prediction objective (SP) and a self-supervised prediction objective (SSP). As for SP, given a graph G and its relation label y, the prediction module seeks to maximize the probability of y for graph G. The prediction module is trained using the labeled data provided, with the objective of minimizing the cross-entropy loss function specified below:

$$\mathcal{L}_{SP} = \mathbb{E}_{(G,y)\in\mathcal{T}}\left[-\log p_\theta(y \mid G)\right] \tag{7}$$

As discussed in the introduction, when labeled data is limited, the supervised loss can hardly achieve satisfactory performance. Inspired by the recent graph contrastive learning methods [19], [43], we seek to boost the performance via contrastive learning. However, existing contrastive learning methods usually target the representation in an unsupervised setting, which depend on a large number of pairwise representation comparison and thus do not fit the semi-supervised scenario. Here in SSP, we describe an alternative where we enforce consistency of predicted label for original and augmented graphs in Figure 3. We first produce the different views

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

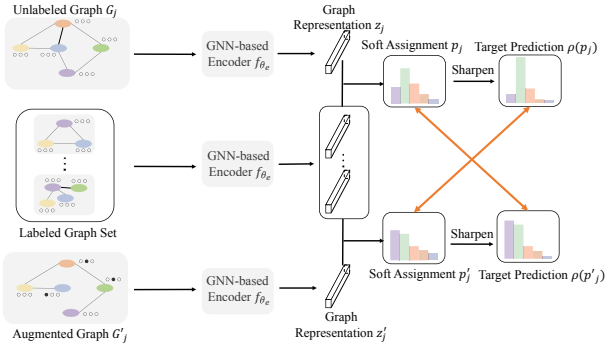


Fig. 3: Illustration of self-supervised learning for the prediction module (SSP). We first assign soft assignments to original graphs and their augmented graphs. The soft assignments are produced using a soft similarity classifier that measures the similarity to a mini-batch of labeled samples. The sharpening operation aims to produce the target prediction with a high degree of purity. Finally, we perform consistency learning by comparing the prediction of one view to the sharpened prediction of the other view.

for unlabeled graphs and then generate the label distributions non-parametrically by comparing the representations of the graph views to those of a batch of labeled graphs. Lastly, we maximize the consistency of label distributions for different views with a well-designed sharpen operation. Our method can be interpreted as a way of contrasting between original and augmented graphs by comparing their predicted label distributions instead of their features, which ensures that different views of the same unlabeled instance are assigned similar pseudo-labels [47] and helps produce discriminative representations to improve the module performance [17].

More precisely, we first perform graph augmentation to generate another graph view from the original graphs. Specifically, we adopt four typical types of basic graph alteration procedures [19], [48] as shown in Figure 4 as follows:

- **Edge deletion**: We randomly delete several edges from the graph following an i.i.d uniform distribution. It is premised on the assumption that semantic information is resistant to variations in edge connection patterns.
- **Node deletion**: We randomly pick some nodes and delete them from the graph, along with all connected edges. The dropping probability of each node also follows a default i.i.d. uniform distribution.
- **Attribute masking**: We randomly sample certain nodes and then randomly mask some of their attributes. It is based on the premise that the graph representation is likely to be robust with the partial vertex attributes.
- **Subgraph**: We use a random walk to sample a subgraph from the graph. It is predicated on the premise that the semantics of the graph can be maintained to a large extent in its local structure.

In our framework, we generate an augmented graph G'_j for

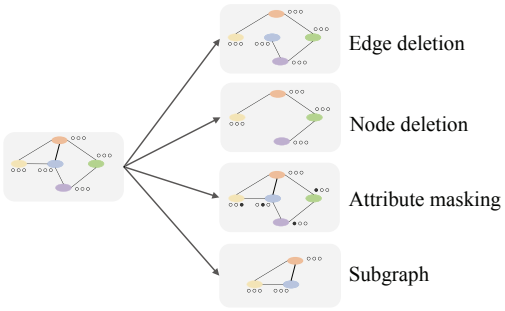


Fig. 4: Illustration of four types of basic graph alteration operations. We use four distinct types of fundamental graph alteration procedures to construct positive pairs of graphs.

each unlabeled graph G_j by randomly selecting one of the four augmentation procedures. Then we get their representations by GNN-based encoder respectively, i.e., $\mathbf{z}_j = f_{\theta_e}(G_j)$ and $\mathbf{z}'_j = f_{\theta_e}(G'_j)$. Here we seek to enhance the consistency of label distributions for two views with the help of labeled data. Since the classifier $\mathcal{H}_{\theta_h}(\cdot)$ may be inaccurate due to overfitting in the absence of adequate labeled data, we use a non-parametrically classifier by comparing the distance between unlabeled graph and labeled graph. To begin, we define a soft similarity classifier based on distance as follows:

$$\pi\left(\mathbf{z}_j, L\right) = \sum_{(G,y)\in\mathcal{T}} \left(\frac{d\left(\mathbf{z}_j, f_{\theta_e}(G)\right)}{\sum_{(G,y)\in\mathcal{T}} d\left(\mathbf{z}_j, f_{\theta_e}(G)\right)} \right) y \tag{8}$$

where y is the one-hot ground truth label vector of graph G. In our work, $d(\mathbf{z}_i, \mathbf{z}_j)$ is exponential temperature-scaled cosine $exp(\mathbf{z}_i^T \mathbf{z}_j / \|\mathbf{z}_i\| \|\mathbf{z}_j\| \tau)$ following [18] and τ is set to 0.5 indicated in [18]. In practice, we sample b labeled graphs from graph-label pair set $\mathcal{T}$ as support set B. The soft assignments for unlabeled graph G_j and its augmentation G'_j are formulated as:

$$p_j = \pi\left(\mathbf{z}_j, B\right) = \sum_{(G,y)\in B} \left(\frac{d\left(\mathbf{z}_j, f_{\theta_e}(G)\right)}{\sum_{(G,y)\in B} d\left(\mathbf{z}_j, f_{\theta_e}(G)\right)} \right) y \tag{9}$$

$$p'_j = \pi\left(\mathbf{z}'_j, B\right) = \sum_{(G,y)\in B} \left(\frac{d\left(\mathbf{z}'_j, f_{\theta_e}(G)\right)}{\sum_{(G,y)\in B} d\left(\mathbf{z}'_j, f_{\theta_e}(G)\right)} \right) y \tag{10}$$

Our soft prediction is more reliable since it makes use of the labeled graph non-parametrically instead of the unreliable classifier. Inspired by [49], we refine the soft predictions to get a target prediction by a sharpening function ρ:

$$\left[\rho\left(p_j\right)\right]_c := \frac{[p_j]_c^{1/T}}{\sum_{c=1}^{C} [p_j]_c^{1/T}}, c = 1, \ldots, \mathcal{C} \tag{11}$$

where T is a temperature parameter set to 0.5 following [49] and $\mathcal{C}$ is the number of classes. The sharpening operation can produce the strengthened target distribution since it increases the purity of the prediction and places a greater focus on data points with a confident prediction.

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

Finally, we optimize the prediction by learning from their high confidence assignment (i.e., target distribution). To be more precise, we train our model by matching the soft assignment to the target distribution. Since we have two perspectives of predictions for each unlabeled graph, we compare the prediction from one view with the sharpened prediction from the other view. Formally,

$$\mathcal{L}_{SSP} = \mathbb{E}_{G_j \in \mathcal{G}^U} [H\left(\rho\left(p_j\right), p'_j\right) + H\left(\rho\left(p'_j\right), p_j\right)] \quad (12)$$

Considering that sharpening operations can produce target distributions close to one-hot format, we use the cross-entropy loss term for H. While the KL-divergence loss term may also be adopted, we find worse performance. Our self-supervised loss makes use of the labeled data to generate predictions, which can be viewed as a hybrid of pseudo-labeling [50] and consistency learning [33]. Unlike traditional pseudo-labeling techniques [50], [51] which maintain artificial labels whose largest class probability is above a preset threshold, we use a sharpening operation [49], retaining the maximum amount of information for the unlabeled data. Consistency learning [32], [33] makes use of unlabeled data by making the assumption that when perturbed views of the same sample are fed into the model, the model should output similar predictions. We perform consistency learning by comparing the prediction of one view to the sharpened prediction of the other view, which helps generate highly confident and transformation-invariant predictions. The final objective function for the prediction module is formulated as:

$$\mathcal{L}_P = \mathcal{L}_{SP} + \mathcal{L}_{SSP} \quad (13)$$

D. Semi-supervised Contrastive Retrieval Module

The retrieval module $\mathcal{Q}_\phi$ aims to select a list of graphs with a given label. This is similar to retrieving a set of samples from a database given a query in the field of information. In our case, "sample" and "query" refer to graph and label, respectively. The retrieval task is capable of providing a complementary view for the prediction module (i.e., $p(y|G)$ and $p(G|y)$ respectively), which may assist in obtaining accurate pseudo labels for unlabeled data to enhance the prediction module. In this part, we train a retrieval module $\mathcal{Q}_\phi$ to approximate $p(G|y)$.

Recent learning-to-rank models [52], [53] usually calculate the joint probability $q_\phi(G, y)$, which is proportional to $q_\phi(G|y)$ when the probability of observing label $q(y)$ is fixed. Similar to the prediction module, our retrieval module also includes a supervised retrieval objective (SR) and a self-supervised retrieval objective (SSR). As for SR, we use the labeled graphs by minimizing the following objective function:

$$\mathcal{L}_{SR} = \mathbb{E}_{(G,y)\in\mathcal{T}} [-\log q_\phi(Q, y)] \quad (14)$$

However, it is intractable to directly optimize $\mathcal{L}_{SR}$, since the condition $\sum_{(G,y)\in\mathcal{T}} q_\phi(Q, y)$ requires traveling over all graph-label pairs in L to calculate the partition function. As a consequence, we follow the existing learning-to-rank models

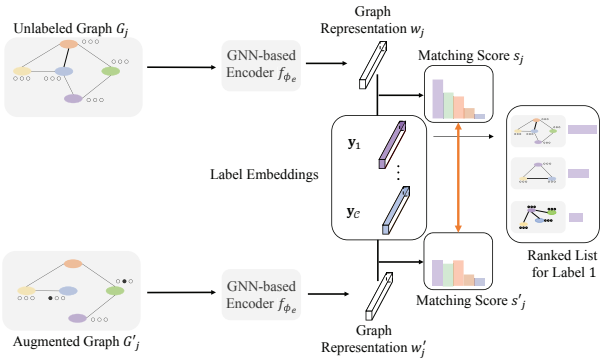


Fig. 5: Illustration of the self-supervised learning for the retrieval module. We first produce matching scores for original graphs and their augmented graphs by taking the inner product of label embeddings and graph embeddings. Then for a mini-batch, the InfoNCE loss is adopted to ensure the matching scores of two views are consistent.

[52], [53], and let $q_\phi(Q, y)$ output a score to estimate the matching degree of the graph-label pair (Q, y) instead of enforcing $q_\phi(Q, y)$ as a valid distribution. In our framework, we adopt the pointwise method [54], [55] to train the retrieval models with labeled data, which posits that each graph-label pair is assigned a score that measures their matching degree. Thus, the scores are used to determine the rank of each sample given a label. Specifically, the score is regarded as a binary supervised signal to indicate whether the graph G matches a label y. Each labeled graph is paired up with all labels, with a binary signal to indicate whether they are matching. In our framework, we start with a GNN-based encoder $f_{\phi_e}(\cdot)$ to get the embedding vector for each graph G:

$$\mathbf{w} = f_{\phi_e}(G) \quad (15)$$

Then each label y is mapped to its label embedding $\mathbf{y}$ which has the same dimension as $\mathbf{w}$. We adopt a learning-to-rank layer, which takes the inner product $q_\phi(G|y) = \sigma(\mathbf{w}^T\mathbf{y})$ as the matching score. Here $\sigma(\cdot)$ is the sigmoid activation function and $\mathbf{y}$ is the embeddings for label y. Then the loss objective is formulated as:

$$\mathbb{E}_{(G,y)\in\mathcal{T}} \left[-\log \sigma\left(\mathbf{w}^\top\mathbf{y}\right)\right] + \mathbb{E}_{(G,y')\notin\mathcal{T}} \left[-\log\left(1-\sigma\left(\mathbf{w}^\top\mathbf{y}'\right)\right)\right] \quad (16)$$

where (G, y') pairs up the graph G with incorrect label y', $\mathbf{w}$ is the graph embedding for G. As for SSR in Figure 5, similarly, we expect that the matching scores of the original graph and its augmentation are consistent. Formally, after getting the embedding $\mathbf{w}_j$ and $\mathbf{w}'_j$ of two graph views, we get the matching score vector as formulated:

$$\boldsymbol{s}_j = \sigma(\mathbf{w}_j^T\mathbf{Y}), \boldsymbol{s}'_j = \sigma(\mathbf{w}_j'^T\mathbf{Y}) \quad (17)$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_C]$ and $\boldsymbol{s}_j$ denotes the matching score for all labels. Then for a mini-batch B, we adopt the InfoNCE

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

loss [18] to enforce the matching scores of two views $\boldsymbol{s}_j$ and $\boldsymbol{s}'_j$ are consistent. Formally,

$$\mathcal{L}_{\text{SSR}} = \mathbb{E}_{G_j \in \mathcal{G}^U} -\log \frac{\exp\left(\boldsymbol{s}_j^T \boldsymbol{s}'_j/\tau\right)}{\exp\left(\boldsymbol{s}_j^T \boldsymbol{s}'_j/\tau\right) + \sum_{G_{j'} \in B} \exp\left(\boldsymbol{s}_j^T \boldsymbol{s}_{j'}/\tau\right)} \tag{18}$$

where τ is the shared temperature parameter set to 0.5. The final objective function for the retrieval module is

$$\mathcal{L}_R = \mathcal{L}_{SR} + \mathcal{L}_{SSR} \tag{19}$$

E. Collaborative Interaction Module

We have introduced the formulation of prediction and retrieval modules and how they are trained separately in a semi-supervised way by minimizing the objective function. Pseudo-labeling is a common technique that aims to fully exploit well-classified samples by selecting unlabeled data with high confidence of prediction and then treating them as labeled data [56], which boosts the model performance in semi-supervised learning. However, a single module may overfit and then produce overconfident results. When the uncertain samples have a wrong pseudo label during self-labeling, the model gradually becomes overconfident in its later prediction since the noise from misclassification accumulates and degrades the performance [57]. To overcome this drawback, in this section, we want to combine the advantages of two modules via collaborative interaction by encouraging the inter-module consistency on unlabeled data.

Note that the retrieval module is also able to annotate unlabeled data as follows:

$$q_\phi(y|G) = \frac{q_\phi(G|y) q_\phi(y)}{q_\phi(G)} \tag{20}$$

Formally, we minimize the KL divergence of prediction distribution from two modules:

$$\mathcal{L}_C = \mathbb{E}_{G \in \mathcal{G}^U} \mathrm{KL}\left(q_\phi(y \mid G) \| p_\theta(y \mid G)\right) \tag{21}$$

In this way, two modules are trained jointly and exchange their guesses for unlabeled data, which produces consistent and confident results. Following the framework of Posterior Regularization [24] which imposes constraints from the desired distribution on probabilistic distribution, we formulate the alternation optimization akin to the form of EM algorithm by regrading $q_\phi(y)$ as the desired distribution. To be specific, we approach the optimization process by sampling from unlabeled data and accepting the confident samples that both modules agree on. Next, we introduce the details of the EM algorithm.

E step. In the E-step, we update the retrieval module $\mathcal{Q}_\phi$ with the fixed prediction module $\mathcal{P}_\theta$. However, directly minimizing KL divergence is very tough. Following the weak-sleep algorithm [58], we minimize the reversed KL divergence $\mathrm{KL}\left(p_\theta(y \mid G) \| q_\phi(y \mid G)\right)$, which shares the same optimal solution as in the KL divergence. By integrating L_Q and L_C as the overall loss and then taking the derivative with respect to the module parameter ϕ, we obtain the gradient for ϕ in the following formulation:

$$\nabla_\phi \mathcal{L} = \nabla_\phi \mathcal{L}_R + \mathbb{E}_{G \in \mathcal{G}^U, y \sim p_\theta(y|G)} \left[\nabla_\phi - \log q_\phi(y \mid G)\right] \tag{22}$$

where the gradient comes additional unlabeled data annotated by the retrieval module $\mathcal{Q}_\phi$, apart from the supervised loss and self-supervised loss. Note that

$$\mathbb{E}_{q_\phi(y|G)} \left[\nabla_\theta \log q_\phi(y \mid G)\right] = 0 \tag{23}$$

Since we expect both two modules to collaboratively annotate unlabeled data, we modify the gradient as follows:

$$\nabla_\phi \mathcal{L} = \nabla_\phi \mathcal{L}_R + \mathbb{E}_{G \in \mathcal{G}^U, y \sim (p_\theta(y|G) + q_\phi(y|G))} \left[\nabla_\phi - \log q_\phi(y \mid G)\right] \tag{24}$$

Even though the gradient remains the same as in Eq.22, the new gradient leverages both modules to collaboratively produce confident results for unlabeled data, which releases the noise in a single module and therefore boosts the performance. Note that without any label information, we assume a uniform distribution for unlabeled graph $q_\phi(G)$. Consequently, we can replace $q_\phi(y|G)$ with $q_\phi(G, y)$. Similarly, directly calculating $q_\phi(G, y)$ is infeasible. As a result, we use the pointwise loss in Eq.16 instead.

M step. In the M-step, we update the prediction module $\mathcal{P}_\theta$ with the fixed retrieval module Q_ϕ. Similarly, the gradient for θ can be calculated as follows:

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_P + \mathbb{E}_{G \in \mathcal{G}^U, y \sim (q_\phi(y|G) + p_\theta(y|G))} \left[\nabla_\theta - \log p_\theta(y \mid G)\right] \tag{25}$$

In summary, two modules are updated alternatively by minimizing the overall loss $\mathcal{L}$ with regard to ϕ and θ, which corresponds to E step and M step, respectively. During each step, we incorporate confident unlabeled data for supervised learning besides self-supervised learning. To be specific, we sample confidence annotated data from unlabeled data at both E and M steps. Following the Eq.24 and Eq.25, the sampling is from the distribution $G \in \mathcal{L}^U, y \sim (q_\phi(y \mid G) + p_\theta(y \mid G))$, which is equivalent to the distribution $(G, y) \sim (q_\phi(G, y) + p_\theta(G, y))$. From the distribution, we adopt a hybrid strategy where a sample will be selected as confident samples only if it is considered credible by both the prediction and retrieval module. In the prediction module, we first assume a uniform distribution for unlabeled graph $p(G)$. Then given an unlabeled graph G and its predicted label y, we rank these samples based on the probability of predicting y and take top-m samples from the ranked list as the credible samples. As for retrieval module, sampling comes from the distribution $q_\phi(G, y)$ and we have $q_\phi(G, y) = q_\phi(G|y)q(y)$ from Bayes Rule. For the first term, given a label y, the retrieval module traverses through each unlabeled sample G and produces a matching score $q_\phi(Q, y) \propto q_\phi(G|y)$, which results in a ranked list for each label y. For the second term, $q(y)$ can be considered as prior knowledge that constrains the label distribution for the retrieved samples. Practically, we assign $q(y)$ with label distribution from the labeled dataset. For each label, we select

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Training Algorithm of DualGraph

Input: Labeled data $\mathcal{G}^L$, unlabeled data $\mathcal{G}^U$.
Parameter: Prediction module parameter θ and retrieval module parameter ϕ
Output: Jointly learned $p_\theta(y|G)$

```
Initialize model parameter θ and φ on data G^L and G^U with Eq.13 and Eq.19.
while not convergence do
    Annotate unlabeled data G^U from the distribution (G, y) ~ (q_φ(G, y) + p_θ(G, y))
    // E-step
    Optimize model parameter φ with Eq.24.
    // M-step
    Optimize model parameter θ with Eq.25.
end while
```

$m_y = mq(y)$ samples. To be specific, the top m_y samples from the ranked list for label y are selected as credible samples. Finally, following the hybrid strategy, after both modules collect a set of credible samples, the intersection of them is regarded as the final set.

The training algorithm for DualGraph is summarized in Algorithm 1. The whole algorithm processes in an iterative manner. In each iteration, we produce an additional set of annotated samples from the unlabeled set with both prediction and retrieval modules collaboratively. The iterative process ends when the model convergences or all of the unlabeled data is exhausted.

V. Experiments

In this section, we first introduce the experimental settings, then conduct extensive experiments to validate the effectiveness of our proposed method. We aim to answer the following research questions:

- **RQ1**: Compared with state-of-the-art models, does our model DualGraph achieve better performance for semi-supervised graph classification?
- **RQ2**: How do the different components of the model contribute to the performance?
- **RQ3**: How do the model hyper-parameters in DualGraph impact the final performance?
- **RQ4**: How do the different encoder architectures and types of augmentation impact the performance?
- **RQ5**: Is there any supplementary analysis that can validate the superiority of DualGraph?

A. Experimental Settings

1) Datasets: We use eight well-known benchmark datasets[1] following [15], [19], [48], To be specific, there are three bioinformatics datasets (i.e., PROTEINS [59], MSRC21 [60] and DD [61]), four social network datasets (i.e., IMDB-B, IMDB-M, REDDIT-B, REDDIT-M-5k [62]) and one scientific collaboration dataset (i.e., COLLAB [62]). For bioinformatics datasets, we seek to categorize proteins into enzymes and non-enzymes. For IMDB-B and IMDB-M, they are movie-collaboration datasets which contain actors/actresses ego-networks and we aim to predict their genres. For REDDIT-B and REDDIT-M-5k, they consist of user ego-networks and the task is to predict each user's community. For COLLAB, we aim to predict each researcher's subfield of Physics based on their collaboration ego-networks. When node attributes are not available, all-ones encoding is used as input node features in the datasets following Sun et.al [15]. The detailed statistics are summarized in Table I.

TABLE I: Details of dataset statistics.

Datasets	Category	Graph Size	Avg.Nodes	Avg.Edges
PROTEINS	Bioinformatics	1113	39.06	72.82
MSRC21	Bioinformatics	563	77.52	198.32
DD	Bioinformatics	1178	284.32	715.66
IMDB-B	Social Networks	1000	19.77	96.53
IMDB-M	Social Networks	1500	13.00	65.94
REDDIT-B	Social Networks	2000	429.63	497.75
REDDIT-M-5k	Social Networks	4999	508.52	594.87
COLLAB	Scientific Collaboration	5000	74.49	2457.78

2) Train/valid/test splits: To effectively apply the datasets to our semi-supervised scenarios, it is essential to divide the datasets in a reasonable way. Specifically, we first split graphs in each dataset into three groups: the training set, validation set, and test set with the ratio of 7:1:2. We further sample 2/7 of the graphs in the training set as the labeled set and utilized the remaining graphs as the unlabeled set. In our experiment, 50% of the labeled set is available for training as default to show the superiority of our approach in the semi-supervised scenario with limited labeled data. Also, we will vary the amounts of the labeled set in the subsequent experiments. The validation set is for hyper-parameter selection and we report the performance on the test set.

3) Baselines: The proposed DualGraph is compared to the following approaches, which fall into three categories: traditional graph approaches, traditional semi-supervised learning approaches and graph-specific semi-supervised learning approaches. The first category includes Graphlet Kernel [27], Shortest Path Kernel (SP) [26], Weisfeiler-Lehman Kernel (WL) [28], Deep Graph Kernel (DG) [62], Sub2Vec [63] and Graph2Vec [64]. Traditional semi-supervised approaches include EntMin [31], Π-Model [37], Mean-Teacher [37], VAT [35]. The last category includes InfoGraph [15], ASGN [1], JOAO [65] and CuCo [48]. To provide rigorous comparative analysis, we use the same underlying architecture (i.e., GIN [29]) when comparing traditional semi-supervised learning methods. As for JOAO and CuCo, we first get graph-level representation by graph contrastive learning and then train a multi-layer perception (MLP) classifier by labeled data as suggested in their papers [19], [48]. The parameters for all baseline methods are initialized as in the corresponding papers, and then were carefully tuned to achieve optimal performance.

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

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Summary of performance on eight benchmark graph classification datasets in terms of accuracy in percentage with standard deviations over five runs. The best performance is highlighted in boldface. Our proposed method DualGraph outperforms all the baseline methods in most cases.

Methods	PROTEINS	MSRC21	DD	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M-5k	COLLAB
Graphlet Kernel	64.8 ± 2.3	24.2 ± 2.8	53.2 ± 1.4	54.5 ± 1.7	32.3 ± 2.4	57.8 ± 2.7	34.3 ± 0.8	55.7 ± 1.1
SP Kernel	65.2 ± 2.6	29.7 ± 3.1	55.3 ± 2.1	52.0 ± 1.6	37.7 ± 1.9	68.3 ± 3.7	30.4 ± 1.3	64.1 ± 1.3
WL Kernel	63.5 ± 1.6	30.7 ± 2.9	57.3 ± 1.2	58.1 ± 2.3	33.3 ± 1.4	61.8 ± 1.3	37.0 ± 0.9	62.9 ± 0.7
DG Kernel	64.4 ± 1.7	33.1 ± 2.7	60.5 ± 0.8	55.6 ± 2.2	34.6 ± 1.3	66.2 ± 2.4	36.5 ± 2.4	61.3 ± 1.2
Sub2Vec	52.7 ± 4.5	18.3 ± 6.2	46.4 ± 3.2	44.9 ± 3.5	31.8 ± 2.7	63.5 ± 2.3	35.1 ± 1.5	60.8 ± 1.4
Graph2Vec	63.1 ± 1.8	28.4 ± 3.2	53.7 ± 1.6	61.2 ± 2.6	38.1 ± 2.2	67.7 ± 2.3	38.1 ± 1.4	63.6 ± 0.9
EntMin	62.7 ± 2.7	32.1 ± 3.3	59.8 ± 1.3	67.1 ± 3.7	37.4 ± 1.2	66.9 ± 3.5	38.7 ± 2.8	63.8 ± 1.6
Π-Model	63.2 ± 1.2	34.3 ± 2.8	61.8 ± 1.8	67.0 ± 3.4	39.0 ± 3.5	67.1 ± 2.9	39.0 ± 1.1	63.7 ± 1.0
Mean-Teacher	64.3 ± 2.1	34.6 ± 2.6	60.6 ± 1.8	66.4 ± 2.7	38.8 ± 3.6	68.7 ± 1.3	39.2 ± 2.1	63.6 ± 1.4
VAT	64.1 ± 1.2	33.8 ± 3.4	59.9 ± 2.6	67.2 ± 2.9	39.6 ± 1.4	70.8 ± 4.1	38.9 ± 3.2	64.1 ± 1.1
InfoGraph	68.2 ± 0.7	**40.6 ± 2.4**	67.5 ± 1.4	71.8 ± 2.3	42.3 ± 1.8	75.2 ± 2.4	41.5 ± 1.7	65.7 ± 0.4
ASGN	67.7 ± 1.2	38.0 ± 2.5	68.5 ± 0.6	70.6 ± 1.4	41.2 ± 1.4	73.1 ± 2.3	42.2 ± 0.8	65.3 ± 0.8
JOAO	68.7 ± 0.9	35.4 ± 2.5	67.9 ± 1.3	71.0 ± 1.9	42.6 ± 1.5	74.8 ± 1.6	42.1 ± 1.2	65.8 ± 0.4
CuCo	67.9 ± 1.8	37.2 ± 2.6	68.3 ± 1.1	71.6 ± 1.4	42.1 ± 1.3	75.2 ± 1.6	41.9 ± 2.2	66.0 ± 1.3
DualGraph	**70.1 ± 1.2**	37.3 ± 2.3	**69.8 ± 0.8**	**72.1 ± 0.7**	**44.8 ± 0.4**	**75.4 ± 1.4**	**42.9 ± 1.4**	**67.2 ± 0.6**

4) Parameter Settings: For our DualGraph, we utilize GIN [29] to parameterize the GNN encoder for both prediction and retrieval modules, which involves three graph convolutional layers and one sum-pooling layer, followed by the softmax function. The batch size is set to 64 and the number of epochs is set to 20 for all datasets. The embedding dimension of hidden layers is 32 for bioinformatics datasets and 64 for datasets of other categories. We train p_θ and q_ϕ with Adam optimizer [66] with initial learning rate 0.01 and weight decay 0.0005 in each iteration. We tune all hyper-parameters on the validation set through grid search. We fix the number of annotated instances m per iteration. Since the DualGraph model takes the joint selection of prediction and retrieval modules, retrieving a fixed amount of instances m is not guaranteed. To solve this problem, we start with a retrieval upper bound $m' = m$ for both modules to select instances. We iteratively increase the upper bound ($m' = 1.25m'$) until we can sample m unique instances from the joint set. In our experiment, m is set to 10% of the number of original unlabeled instances, corresponding to the sampling ratio r (10% by default), which means that the unlabeled data is exhausted after ten iterations. We conduct five runs of training and testing under different random seeds and report the mean accuracy and standard deviation.

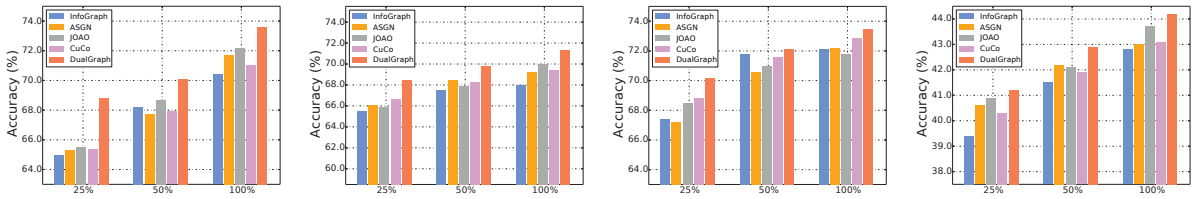
B. Performance Comparison (RQ1)

In Table II, we provide the compared results of semi-supervised graph classification using half of the labeled set. We make the following observations from the results.

- It can be seen that the majority of traditional graph approaches perform worse than other methods, indicating that the strong representation-learning capability of graph neural network methods enables the extraction of more meaningful information from structured and relational data for classification.
- The approaches that leverage traditional semi-supervised learning techniques perform worse than the graph-specific semi-supervised learning approaches, demonstrating that graph methods designed specifically for semi-supervised scenarios are superior to traditional graph methods and more beneficial to complex graph-related tasks such as classification and regression.
- Graph contrastive learning methods (CuCo and JOAO) achieve the best accuracy in most cases among previous competing approaches. The potential reason may be that they can leverage the unlabeled data more effectively compared to traditional graph approaches, which are not tailored for semi-supervised graph classification. InfoGraph and ASGN, in particular, are not as effective as them mostly. Perhaps this is due to their inability to explore unlabeled material with diverse priors. As a consequence, they will lose some important information to some degree, resulting in poor performance.
- Our proposed DualGraph achieves the best performance on most of the datasets, validating the efficacy of our framework. We attribute the remarkable performance to two factors based on the findings: (1) Introduction of dual learning. Instead of replicating another prediction module as co-training, DualGraph introduces the dual retrieval module. Two modules are jointly trained and complement each other to enhance inter-module consistency. (2) Introduction of semi-supervised contrastive learning. Instead of adopting the previous contrastive learning framework [18], we introduce a novel semi-supervised contrastive learning framework for both modules, which improves the performance for each individual module.

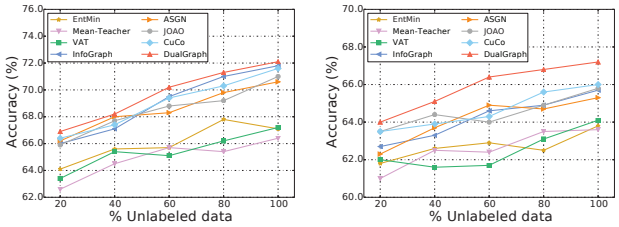
Performance on Different Amounts of Labeled Data. In Figure 6, we tune the rates of the labeled data for training to illustrate the performance of different approaches. We take the four representative datasets (i.e., PROTEINS, DD, IMDB-B and REDDIT-M-5k) as examples. Of note, traditional

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.



(a) PROTEINS (b) DD (c) IMDB-B (d) REDDIT-M-5k

Fig. 6: Performance on four datasets with various amounts of the labeled data (i.e., 25%, 50% and 100% labeled data) and all the unlabeled data. The best performance is highlighted in boldface. Increasing labeled data results in better performance.



(a) IMDB-B (b) COLLAB

Fig. 7: Performance comparisons *w.r.t.* the amount of unlabeled data. In most cases, increasing unlabeled data results in better performance.

methods are not included since they do not show competitive performance. The results show that as the number of available labeled instances grows, the performance of all approaches improves, suggesting that adding more labeled data is an effective way to enhance performance. In most cases, the proposed DualGraph achieves the best performance among all the approaches, indicating that explicitly integrating both dual learning and contrastive learning into graph classification can improve the performance of current GNN-based models.

Performance on Different Amounts of Unlabeled Data. Given the difficulty and expense of obtaining data annotation, a proper model should be able to make sufficient use of a large amount of unlabeled data to improve the performance. As a consequence, we vary different sizes of unlabeled data to compare the performance of traditional semi-supervised learning approaches and graph-specific semi-supervised learning approaches. We select the IMDB-B and COLLAB datasets as instance, and plot the test accuracy at each rate (i.e., 20%, 40%, 60%, 80% and 100%) in Figure 7. From the result, we observe that more unlabeled data can benefit InfoGraph, ASGN and DualGraph consistently, whereas the performance of other approaches could fluctuate as the quantity of unlabeled data rises. Moreover, the curve of DualGraph is mostly on top of the other models' curves, demonstrating that our DualGraph can make full use of the unlabeled data in most of the settings.

C. Ablation Study (RQ2)

In this part, we conduct ablation experiments to understand the effectiveness of each component. Particularly, we introduce a few model variants as follows:

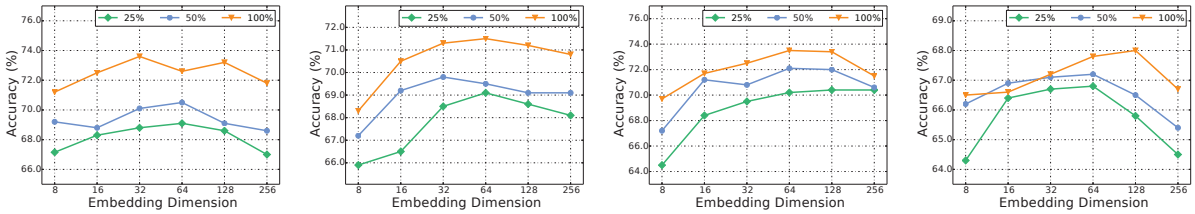
- GNN-Sup: We only use the prediction module with the supervision of the labeled data. (i.e., $\mathcal{L} = \mathcal{L}_{SP}$)
- GNN-Pred: We only use the prediction module without annotating the unlabeled data. (i.e., $\mathcal{L} = \mathcal{L}_{P}$)
- GNN-Pred-ST: We remove the retrieval module and optimize the prediction module with self-training [56]. To be precise, the model is first trained to annotate the unlabeled data; then the most confident pseudo-labeled data is treated as extra training data in the next iteration.
- GNN-Pred-Co: We adopt the co-training scheme [67] by replacing the retrieval module with the prediction module with different initialization. Specifically, the model selects the annotated data as extra labeled data based on the agreement of two modules in each iteration.
- DualGraph w/o Intra: We remove the intra-module consistency in two modules. (i.e., $\mathcal{L}_P = \mathcal{L}_{SP}$, $\mathcal{L}_R = \mathcal{L}_{SR}$)
- DualGraph w/o Inter: We remove the inter-module consistency. To be specific, we feed the pseudo-labels produced by one module into the other module without adopting the consistency loss $\mathcal{L}_C$.

The results are recorded in Table III. We summarize the following findings: (1) GNN-Pred is superior to GNN-Sup, which hence illustrates the importance of making full use of contrastive learning for unlabeled samples in the prediction module. (2) From the comparison of GNN-Pred-ST and GNN-Pred, self-training benefit the performance gain, which is in accordance with the findings in previous work [56]. (3) We can observe a consistent performance gain when comparing ensemble model GNN-Pred-Co with self-training model GNN-Pred-ST, which implies that dual learning can be beneficial to improve the performance. (4) GNN-Pred-Co shows worse performance compared with the Full Model, indicating that the retrieval module (dual task) is critical for the framework. Since two tasks are jointly optimized and complement each other to enhance inter-module consistency, our framework achieves better performance compared with the co-training scheme [67]. (5) From the comparison of DualGraph w/o Intra and the Full Model, we can see that contrastive learning is indeed a crucial and effective component in our framework and can be beneficial to improve the performance. (6) From the comparison of DualGraph w/o Inter and the Full Model, which demonstrates the advantage of the intersection strategy

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

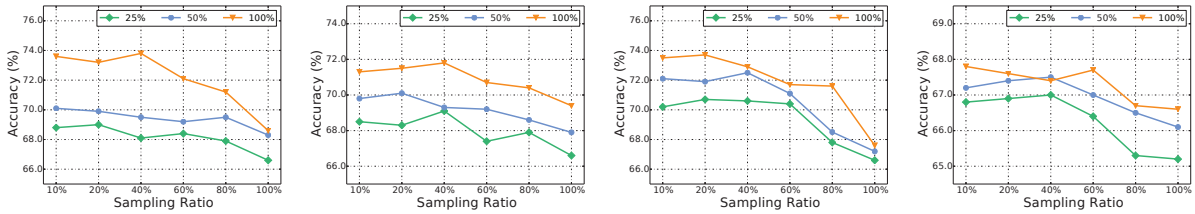
TABLE III: Ablation study of several model variants (in %). The best performance is highlighted in boldface. The results show the contribution of different components in the proposed framework.

Methods	PROTEINS	MSRC21	DD	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M-5k	COLLAB
GNN-Sup	63.3 ± 1.4	29.8 ± 2.7	62.5 ± 1.5	63.4 ± 2.1	39.2 ± 1.6	69.8 ± 1.1	38.6 ± 2.5	61.7 ± 1.5
GNN-Pred	64.3 ± 1.8	31.9 ± 3.2	61.7 ± 1.7	67.4 ± 1.4	40.5 ± 1.8	68.6 ± 1.6	39.0 ± 1.8	63.7 ± 1.2
GNN-Pred-ST	67.6 ± 0.6	32.9 ± 1.8	65.4 ± 2.7	67.6 ± 2.4	39.1 ± 0.5	71.8 ± 1.7	38.9 ± 2.4	65.7 ± 1.9
GNN-Pred-Co	68.1 ± 2.9	33.3 ± 4.5	66.9 ± 2.8	69.5 ± 1.2	41.8 ± 2.3	74.6 ± 0.8	39.9 ± 1.9	66.5 ± 0.4
DualGraph w/o Intra	68.5 ± 1.5	33.7 ± 4.8	69.3 ± 2.1	68.0 ± 1.2	39.3 ± 2.8	75.2 ± 0.7	39.4 ± 2.8	66.7 ± 0.4
DualGraph w/o Inter	68.7 ± 2.3	34.8 ± 2.3	68.1 ± 1.7	69.4 ± 3.5	42.1 ± 2.8	73.4 ± 1.3	41.0 ± 2.7	65.5 ± 1.0
Full Model	**70.1 ± 1.2**	**37.3 ± 2.3**	**69.8 ± 0.8**	**72.1 ± 0.7**	**44.8 ± 0.4**	**75.4 ± 1.4**	**42.9 ± 1.4**	**67.2 ± 0.6**



(a) PROTEINS (b) DD (c) IMDB-B (d) COLLAB

Fig. 8: Performance comparisons *w.r.t.* the embedding dimensions of hidden layers.



(a) PROTEINS (b) DD (c) IMDB-B (d) COLLAB

Fig. 9: Performance comparisons *w.r.t.* the sampling ratio.

to ease the effect of incorrect label annotations, enabling our DualGraph to obtain the superior performance.

D. Parameter Sensitivity (RQ3)

In this part, we examine the sensitivity of the DualGraph to hyper-parameters. Specifically, we investigate the effect of varying embedding dimensions of hidden layer and sampling ratio in our framework on four datasets in different settings. Here we vary the amounts of the labeled data (i.e., 25%, 50% and 100% labeled data) for each dataset.

Effect of Different Embedding Dimensions. We begin with analyzing the impact of the embedding dimensions of hidden layers d. It is well known that the higher the value of the embedding dimensions, the larger the capacity of the model. Therefore, we anticipate the model performing well as dimension rises. All other parameters are fixed to the ones that provide the best results and d is varied in $\{8, 16, 32, 64, 128, 256\}$. Figure 8 reveals the performance. It can be found that the larger embedding dimension of hidden layers often contributes to higher accuracy before saturation. However, the too-large dimension may degrade the performance owing to overfitting induced by the redundancy of parameters.

Effect of Different Sampling Ratios. Then we analyze the effect of different sampling ratios ρ. We fix all other parameters and vary ρ in $\{10\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$, results are shown in Figure 9. We observe that when ρ is small (e.g., 10% and 20%), the performance is stable relatively in different settings while a larger sampling ratio generally leads to worse performance. The reason is that a larger sampling ratio weakens the strength of iterative optimization, which aligns with our expectations.

E. Further Analysis (RQ4)

Furthermore, we investigate the impact of different encoder architectures and different types of graph augmentation.

Effect of Different Encoder Architectures. Here we study the effect of different encoder architectures. We select four types of popular graph neural networks (i.e., GCN [10], GraphSAGE [68], GAT [46] and GIN [29]) and compare their performance on four datasets in Figure 10. We can observe that GIN consistently outperforms other base models on four datasets, which validates the effectiveness of GIN with strong representation ability. This justifies the reason why we choose GIN as the base model for all GNN-based methods.

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

TABLE IV: Comparisons when contrasting different types of augmentation (in %). The best performance is highlighted in boldface. Random operation can be generally more beneficial than other deterministic augmentations on most datasets.

Methods	PROTEINS	MSRC21	DD	IMDB-B	IMDB-M	REDDIT-B	REDDIT-M-5k	COLLAB
Edge deletion	67.7 ± 2.1	32.7 ± 2.1	68.0 ± 0.8	70.0 ± 1.3	41.4 ± 0.9	74.8 ± 1.2	41.9 ± 1.3	66.3 ± 0.8
Node deletion	69.6 ± 1.6	35.4 ± 2.4	69.2 ± 1.4	71.4 ± 0.8	44.1 ± 0.5	$\mathbf{75.8 \pm 0.7}$	42.5 ± 1.5	66.7 ± 0.5
Attribute masking	68.9 ± 1.3	35.7 ± 1.9	68.8 ± 1.5	70.8 ± 0.7	43.9 ± 0.4	74.3 ± 1.4	41.6 ± 1.5	66.4 ± 0.6
Subgraph	$\mathbf{70.3 \pm 1.0}$	36.6 ± 1.8	69.4 ± 0.7	71.7 ± 0.5	44.5 ± 0.8	74.9 ± 1.4	42.5 ± 1.7	66.7 ± 0.8
Random	70.1 ± 1.2	$\mathbf{37.3 \pm 2.3}$	$\mathbf{69.8 \pm 0.8}$	$\mathbf{72.1 \pm 0.7}$	$\mathbf{44.8 \pm 0.4}$	75.4 ± 1.4	$\mathbf{42.9 \pm 1.4}$	$\mathbf{67.2 \pm 0.6}$

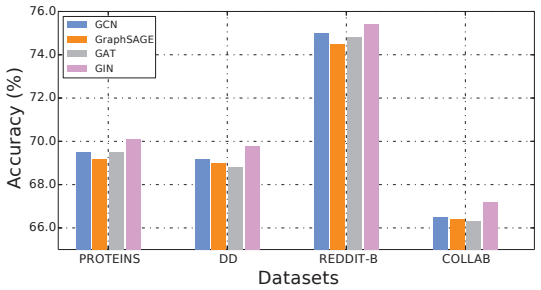


Fig. 10: DualGraph with different encoder architectures.

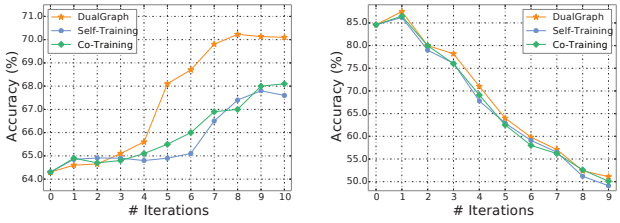


Fig. 11: Analysis of quality of newly annotated instances at each iteration. Left: Convergence curve of accuracy (in %) for different methods. Right: Accuracy of the annotated samples in each iteration. Both are evaluated on the PROTEINS dataset.

Effect of Different Types of Graph Augmentation. We also conduct research to validate the necessity of selecting graph modification operations at random during graph augmentation. We compare this to the following deterministic augmentation setting. We generate augmented graphs by doing this process once for each type of operation, such as edge deletion, node deletion, attribute masking and subgraph. The average accuracy for each operation in our framework is shown in Table IV. As can be observed, deterministic selection performs worse than random selection in most cases. The reason is that augmented graphs generated by executing random alteration procedures are more difficult to distinguish from the original graph. Contrastive learning can learn more effective and stable graph-level representations through more challenging tasks.

F. Case Study (RQ5)

To explore the fundamental cause for performance gain in DualGraph, we undertake an experiment to test the following hypothesis. A natural hypothesis is that the performance improvement is due primarily to the quality of the retrieved instances. It is under discussion how this "quality" can be defined as an objective metric. Quality is defined in our study as the accuracy of retrieved instances. Since we know the ground-truth labels of the "unlabeled data", we can evaluate the accuracy technically.

To test this hypothesis, we devised the following experiments. We assess the accuracy of semi-supervised methods trained in an iterative manner (Self-Training, Co-training and DualGraph) on the test set in each iteration, as well as the accuracy of its promoted pseudo-labeled instances.

Figure 11 depicts the outcomes of the experiments. When comparing the accuracy of retrieved instances in each iteration between DualGraph and other semi-supervised methods, the accuracy for retrieved instances in each iteration is typically greater, implying that DualGraph can pick examples of higher quality, which contributes to each iteration's performance improvement. Furthermore, we can see that the accuracy of the unlabeled samples in each iteration in our framework is greater, demonstrating that our framework is capable of selecting reliable and confident signals from unlabeled data. Lastly, for DualGraph, we can see the empirical convergence of the accuracy during alternative updating.

VI. Conclusion

In this paper, we investigate the problem of semi-supervised graph classification, which is fundamental in graph data mining and propose a novel method termed the DualGraph. DualGraph is a principled framework, which consists of a prediction module and a retrieval module to encourage the consistency on unlabeled data and overcome the unreliability of pseudo-labels and scarcity of labeled data. Both modules are jointly optimized to mutually enhance each other with an EM-styled algorithm. Specifically, we leverage posterior regularization to encourage the inter-module consistency and take advantage of contrastive learning to encourage the intra-module consistency. Experimental results on various widely-used datasets validate the superiority of our proposed framework. In future work, we will broaden the scope of our DualGraph to various promising domains including molecular networks and recommender systems.

Acknowledgment

This paper is partially supported by National Key Research and Development Program of China with Grant No. 2018AAA0101902 as well as the National Natural Science Foundation of China (NSFC Grant No. 62106008 and No. 62006004).

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] Z. Hao, C. Lu, Z. Huang, H. Wang, Z. Hu, Q. Liu, E. Chen, and C. Lee, "Asgn: An active semi-supervised graph neural network for molecular property prediction," in *KDD*, 2020.

[2] R. Kojima, S. Ishida, M. Ohta, H. Iwata, T. Honma, and Y. Okuno, "kgcn: a graph-based deep learning framework for chemical structures," *Journal of Cheminformatics*, vol. 12, pp. 1–10, 2020.

[3] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Müller, and A. Tkatchenko, "Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space," *The journal of physical chemistry letters*, vol. 6, no. 12, pp. 2326–2331, 2015.

[4] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *KDD*, 2018.

[5] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *ICML*, 2019.

[6] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *NeurIPS*, 2018.

[7] C. Lu, Q. Liu, C. Wang, Z. Huang, P. Lin, and L. He, "Molecular property prediction: A multilevel quantum interactions modeling perspective," in *AAAI*, 2019.

[8] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions," in *NeurIPS*, 2017.

[9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[11] G. Zhong and C.-M. Pun, "Latent low-rank graph learning for multimodal clustering," in *ICDE*, 2021.

[12] D. Shimin, Y. Quanming, Y. Zhang, and C. Lei, "Efficient relation-aware scoring function search for knowledge graph embedding," in *ICDE*, 2021.

[13] Z. Wang, T. Xia, R. Jiang, X. Liu, K.-S. Kim, X. Song, and R. Shibasaki, "Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks," in *ICDE*, 2021.

[14] J. Li, Y. Rong, H. Cheng, H. Meng, W. Huang, and J. Huang, "Semi-supervised graph classification: A hierarchical graph perspective," in *WWW*, 2019.

[15] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *ICLR*, 2020.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Deep Learning and Representation Learning NIPS Workshop*, 2014.

[17] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *NeurIPS*, 2020.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[19] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *NeurIPS*, 2020.

[20] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *NeurIPS*, 2016.

[21] Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, and T.-Y. Liu, "Dual supervised learning," in *ICML*, 2017.

[22] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *ICCV*, 2017.

[23] Y. Li, N. Duan, B. Zhou, X. Chu, W. Ouyang, X. Wang, and M. Zhou, "Visual question generation as dual task of visual question answering," in *CVPR*, 2018.

[24] K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar, "Posterior regularization for structured latent variable models," *The Journal of Machine Learning Research*, vol. 11, pp. 2001–2049, 2010.

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[26] K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs," in *ICDM*, 2005.

[27] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," in *AISTATS*, 2009.

[28] N. Shervashidze, P. Schweitzer, E. J. Van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels." *Journal of Machine Learning Research*, vol. 12, no. 9, pp. 2539–2561, 2011.

[29] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.

[30] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *AAAI*, 2018.

[31] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *NeurIPS*, 2005.

[32] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2017.

[33] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *NeurIPS*, 2016.

[34] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," *ICLR*, 2018.

[35] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[36] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020.

[37] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[38] W. Ju, J. Yang, M. Qu, W. Song, J. Shen, and M. Zhang, "Kgnn: Harnessing kernel-based networks for semi-supervised graph classification," in *WSDM*, 2022.

[39] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax." in *ICLR*, 2019.

[40] J. Zeng and P. Xie, "Contrastive self-supervised learning for graph classification," in *AAAI*, 2021.

[41] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *ICML*, 2020.

[42] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *KDD*, 2020.

[43] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *WWW*, 2021.

[44] H. Lin, J. Yan, M. Qu, and X. Ren, "Learning dual retrieval module for semi-supervised relation extraction," in *WWW*, 2019.

[45] B. Wei, G. Li, X. Xia, Z. Fu, and Z. Jin, "Code generation as a dual task of code summarization," in *NeurIPS*, 2019.

[46] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *ICLR*, 2017.

[47] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*, 2020.

[48] G. Chu, X. Wang, C. Shi, and X. Jiang, "Cuco: Graph representation with curriculum contrastive learning," in *IJCAI*, 2021.

[49] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016.

[50] G. J. McLachlan, "Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis," *Journal of the American Statistical Association*, vol. 70, no. 350, pp. 365–369, 1975.

[51] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.

[52] T.-Y. Liu, "Learning to rank for information retrieval," 2011.

[53] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *CVPR*, 2019.

[54] K. Crammer, Y. Singer *et al.*, "Pranking with ranking." in *NeurIPS*, 2001.

[55] P. Li, Q. Wu, and C. Burges, "Mcrank: Learning to rank using multiple classification and gradient boosting," *NeurIPS*, 2007.

[56] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013.

[57] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *NeurIPS*, 2018.

[58] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, "The" wake-sleep" algorithm for unsupervised neural networks," *Science*, vol. 268, no. 5214, pp. 1158–1161, 1995.

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.

[59] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, pp. 47–56, 2005.
[60] M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting, "Propagation kernels: efficient graph kernels from propagated information," *Machine Learning*, vol. 102, no. 2, pp. 209–245, 2016.
[61] P. D. Dobson and A. J. Doig, "Distinguishing enzyme structures from non-enzymes without alignments," *Journal of molecular biology*, vol. 330, no. 4, pp. 771–783, 2003.
[62] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *KDD*, 2015.
[63] B. Adhikari, Y. Zhang, N. Ramakrishnan, and B. A. Prakash, "Sub2vec: Feature learning for subgraphs," in *PAKDD*, 2018.
[64] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.
[65] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *ICML*, 2021.
[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
[67] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *ACL*, 1998.
[68] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NeurIPS*, 2017.

Authorized licensed use limited to: Peking University. Downloaded on October 11,2022 at 03:35:38 UTC from IEEE Xplore. Restrictions apply.