Hypergraph-enhanced Dual Semi-supervised Graph Classification

Wei Ju [1] **Zhengyang Mao** [1] **Siyu Yi*** [2] **Yifang Qin** [1] **Yiyang Gu** [1] **Zhiping Xiao** [3] **Yifan Wang** [4] **Xiao Luo** [3] **Ming Zhang*** [1]

Abstract

In this paper, we study semi-supervised graph classification, which aims at accurately predicting the categories of graphs in scenarios with limited labeled graphs and abundant unlabeled graphs. Despite the promising capability of graph neural networks (GNNs), they typically require a large number of costly labeled graphs, while a wealth of unlabeled graphs fail to be effectively utilized. Moreover, GNNs are inherently limited to encoding local neighborhood information using message-passing mechanisms, thus lacking the ability to model higher-order dependencies among nodes. To tackle these challenges, we propose a **H**ypergraph-**E**nhanced Du**AL** framework named HEAL for semi-supervised graph classification, which captures graph semantics from the perspective of the hypergraph and the line graph, respectively. Specifically, to better explore the higher-order relationships among nodes, we design a hypergraph structure learning to adaptively learn complex node dependencies beyond pairwise relations. Meanwhile, based on the learned hypergraph, we introduce a line graph to capture the interaction between hyperedges, thereby better mining the underlying semantic structures. Finally, we develop a relational consistency learning to facilitate knowledge transfer between the two branches and provide better mutual guidance. Extensive experiments on real-world graph datasets verify the effectiveness of the proposed method against existing state-of-the-art methods.

[1]School of Computer Science, National Key Laboratory for Multimedia Information Processing, Peking University-Anker Embodied AI Lab, Peking University, China [2]School of Statistics and Data Science, Nankai University, China [3]Department of Computer Science, University of California, Los Angeles, USA [4]School of Information Technology & Management, University of International Business and Economics, China. Correspondence to: Siyu Yi <siyuyi@mail.nankai.edu.cn>, Ming Zhang <mzhang_cs@pku.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Graph classification, which involves identifying the class labels of graphs, is a significant problem with diverse practical applications in various fields. Data originating from domains such as bioinformatics, chemoinformatics, and social network analysis, can naturally be represented as graphs. For example, molecules in chemoinformatics can be represented as graphs by viewing atoms as nodes and chemical bonds between pairs of atoms as edges. The objective of this task is to effectively recognize the class label of each graph, such as predicting the quantum mechanical properties (Hao et al., 2020; Ju et al., 2023a) and assessing the functionality of chemical compounds (Kojima et al., 2020).

To solve this problem, early methods leverage the idea of graph kernels that compute a similarity measure between graphs by comparing their substructures (Kashima et al., 2003; Shervashidze et al., 2009; 2011), and have been proven to effectively capture the structural properties of graphs. Despite their efficacy, graph kernels may face challenges in scalability and computational efficiency when dealing with large datasets or complex graphs. Recently, graph neural networks (GNNs) (Kipf & Welling, 2016; Ju et al., 2024a) have emerged as a prominent and powerful paradigm for graph classification (Mao et al., 2023; Yi et al., 2023b; Luo et al., 2023c). The key idea of GNNs is to learn effective graph representations by iteratively aggregating information from neighboring nodes (Gilmer et al., 2017), which have achieved remarkable success.

However, most prevailing methods typically follow the framework of supervised learning, which demands a substantial amount of labeled graphs to train GNN models. However, in the field of graph analytics, obtaining labeled graphs can be a costly endeavor. Annotating graph data requires expert domain knowledge, manual efforts, and often extensive human involvement, making the process time-consuming and expensive. For instance, molecular labels are often acquired through costly Density Functional Theory (DFT) calculations or generated from complex experiments in the field of chemistry (Hao et al., 2020; Ju et al., 2023a). This scarcity of labeled graphs and the high cost of annotation pose significant challenges for developing accurate and robust GNNs for graph classification.

This inspires us to study semi-supervised graph classification, where we leverage both labeled and unlabeled graphs. Despite the unavailability of properties (*i.e.*, labels) in the unlabeled graphs, their structures contain valuable information that could potentially enrich the capabilities of GNNs if utilized effectively. Actually, there are several approaches along this line (Li et al., 2019; Hao et al., 2020; Luo et al., 2022; Ju et al., 2022; 2023b). ASGN (Hao et al., 2020) adopts a teacher-student framework to fully utilize labeled and unlabeled graphs. DualGraph (Luo et al., 2022) incorporates contrastive learning to encourage the consistency of unlabeled graphs in a dual manner. KGNN (Ju et al., 2022) and TGNN (Ju et al., 2023b) unify the GNNs and graph kernels in a semi-supervised framework.

Despite the encouraging performance achieved by existing methods, they still suffer from two key limitations. *First*, GNNs are restricted to capturing only low-order local neighborhood information and struggle to model high-order dependencies between nodes. For instance, in chemoinformatics, GNNs may be unable to effectively capture the complex interactions and long-range dependencies between atoms in a molecule, thereby potentially limiting their ability to accurately predict properties like molecular activity or toxicity. *Second*, the utilization of unlabeled graphs remains underexploited, despite their containing valuable structural information. The unlabeled graphs can act as a regularizer, facilitating the exploration of intrinsic graph semantics, even in the presence of a scarce amount of labeled graphs. For example, in social network analysis, there might be a large amount of unlabeled user interaction graphs, yet these graphs hold insightful community structures and social relationships. As such, we are highly desired to look for an approach that is able to better capture high-order dependencies among nodes and meanwhile sufficiently leverage the unlabeled graphs to overcome the scarcity of labeled graphs.

To address these challenges, in this paper we propose a **H**ypergraph-**E**nhanced Du**AL** framework named HEAL for semi-supervised graph classification. The key idea of HEAL is to capture graph semantics from the perspective of the hypergraph and the line graph, respectively. Specifically, to explore the intricate interdependencies among nodes, we develop a learnable hypergraph structure learning, which possesses the remarkable ability to adaptively acquire higher-order node relationships beyond pairwise connections, and is more flexible to model complex data structures than predefined hypergraph construction. Moreover, due to the presence of higher-order semantic interactions in complex data structures, we hence leverage the learned hypergraph to introduce a line graph, effectively capturing the interactions between hyperedges, thus unlocking deeper insights into the underlying semantic structures of graphs. Finally, since the hypergraph and the line graph explore graph semantics at different levels of higher-order structures, it is crucial to jointly train these two branches to enable mutual knowledge transfer between them. We thus present relational consistency learning, in which two branches are required to produce consistent similarity scores for each unlabeled graph. By encouraging the consistency between two similarity distributions, our method effectively enhances the potential of the model by fully using unlabeled graphs, thereby better serving the semi-supervised graph classification. Experiments validate the effectiveness of our proposed model HEAL.

2. Problem Definition & Preliminaries

Definition 1. *A **graph** can be defined as $G = (V, E, \mathbf{X}, \mathbf{y})$, where V is a set of nodes, and E is a set of edges. $\mathbf{X} \in R^{|V|\times d_f}$ denotes the feature matrix of nodes, where d_f is the dimension of features. $\mathbf{y}$ is the class label of graph G and $\mathbf{A} \in \{0,1\}^{|V|\times|V|}$ represents the adjacency matrix.*

Definition 2. *A **hypergraph** is a generalization of a graph where edges are allowed to connect more than two nodes. Formally, a hypergraph is represented as $H = (V, E_H)$, where V is the node set same as in graph G, and E_H is the set of hyperedges, which can contain any number of nodes. Each hyperedge $e_h \in E_H$ is a subset of the node set V.*

Definition 3. *A **line graph** of the hypergraph is defined as a graph $L(H) = (V_L, E_L)$, where each node $v_l \in V_L$ corresponds to an edge in H, and two nodes in V_L are adjacent in $L(H)$ if and only if the corresponding edges in H share a common node (Whitney, 1992). Formally, $V_L = \{v_l : v_l \in E_H\}$, and $E_L = \{(v_{l_i}, v_{l_j}) : v_{l_i}, v_{l_j} \in E_H, |v_{l_i} \cap v_{l_j}| \geq 1\}$. The weight of each edge $W_{i,j}$ is assigned to $W_{i,j} = |v_{l_i} \cap v_{l_j}| \,/\, |v_{l_i} \cup v_{l_j}|$.*

Semi-supervised Graph Classification. Given a set of graphs $\mathcal{G} = \{\mathcal{G}^L, \mathcal{G}^U\}$, in which $\mathcal{G}^L = \{G_1, \cdots, G_{|\mathcal{G}^L|}\}$ are labeled graphs and $\mathcal{G}^U = \{G_{|\mathcal{G}^L|+1}, \cdots, G_{|\mathcal{G}^L|+|\mathcal{G}^U|}\}$ are unlabeled graphs. the problem of semi-supervised graph classification can be defined as learning a mapping function from graphs to class labels $f : \mathcal{G} \to \mathcal{Y}$ to predict the labels of $\mathcal{G}^U$, where $\mathcal{Y}$ represents the labels corresponding to $\mathcal{G}$.

GNN-based Encoder. The general mechanism of GNNs is to iteratively update node embeddings by aggregating the information of its neighbor nodes via message-passing (Gilmer et al., 2017). Formally, the node embeddings $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{|V|}]^\top \in \mathbb{R}^{|V|\times d}$ can be updated as:

$$\mathbf{H} = \sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}), \quad \hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}, \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$, $\mathbf{W}$ is the trainable weight matrix, and $\sigma(\cdot)$ is the activation function. Then the whole graph representation $\mathbf{h}_G$ can be computed based on all node embeddings as:

$$\mathbf{h}_G = \sum\nolimits_{i=1}^{|V|} \mathbf{h}_i. \tag{2}$$

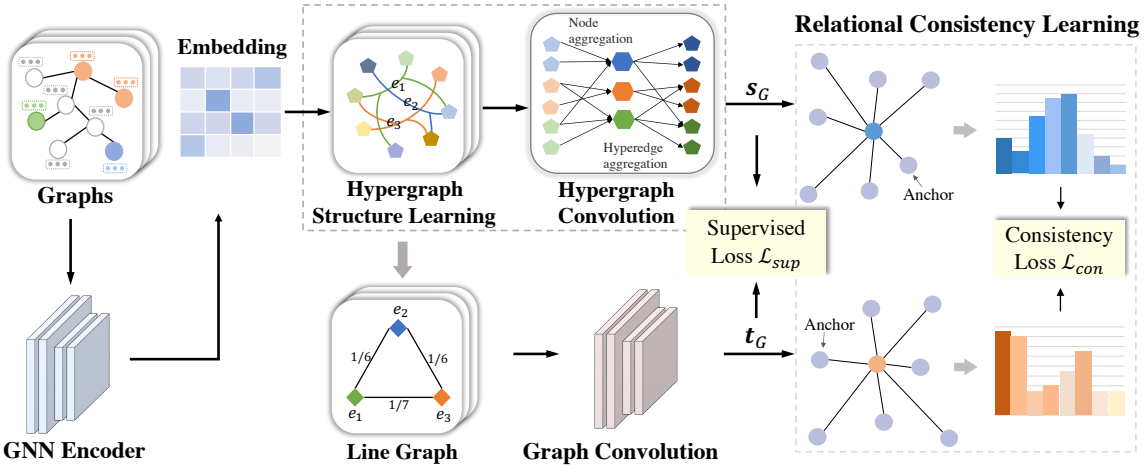


Figure 1: Illustration of the proposed framework HEAL.

3. Methodology

In this section, we introduce our HEAL framework for semi-supervised graph classification, which captures graph semantics from both the hypergraph and line graph perspectives. HEAL consists of three modules: hypergraph high-order dependency learning, graph convolution on the line graph, and relational consistency learning. Figure 1 provides an overview of the whole framework.

3.1. Hypergraph High-order Dependency Learning

GNNs have achieved significant success in learning expressive representations. However, they are inherently limited to capturing only local neighborhood information via message-passing mechanisms (Gilmer et al., 2017) and cannot effectively capture higher-order substructures. This limitation is crucial since many real-world graph data exhibit complex hierarchical relationships that extend beyond immediate neighbors. To address this issue, we propose using hypergraphs to overcome the aforementioned limitation of GNNs. Hypergraphs (Feng et al., 2019) provide a more powerful framework for modeling higher-order dependencies and interactions among nodes, enabling us to better capture the rich structural information present in the graphs.

Hypergraph Structure Learning. Existing approaches typically construct hypergraphs using predefined criteria based on distances (Yu et al., 2012), representations (Wang et al., 2015), or attributes (Huang et al., 2015). However, these methods may suffer from sub-optimal performance and high computational costs due to their lack of flexibility. To overcome these limitations, we develop a flexible way to parameterize a learnable hypergraph structure, being optimized jointly with the network parameters. Nevertheless, directly learning a dense adjacency matrix could incur excessive computational overhead, so we instead adopt a low-rank strategy to efficiently model the hypergraph structural matrix $\mathbf{\Lambda} \in \mathbb{R}^{|V| \times k}$, where k represents the number of hyperedges, calculated as:

$$\mathbf{\Lambda} = \mathbf{H} \cdot \mathbf{W}, \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{|V| \times d}$ is the hidden embedding matrix derived from the GNN-based encoder, and d is the dimension of hidden embeddings. We introduce a learnable weight matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$ to model the hyperedges. As a result, learning the hypergraph structural matrix requires only $\mathcal{O}(k \times d)$ time complexity ($d << |V|$), which significantly improves model efficiency without compromising performance.

Hypergraph Convolution. After obtaining a flexible hypergraph, we can effectively capture higher-order dependencies among nodes. To achieve this, we design a hypergraph convolution to learn high-level node representations. First, we learn hyperedge embeddings by aggregating neighbor nodes connected in the hypergraph. Then, the acquired hyperedge embeddings are leveraged to perform higher-order updates on the node representations. Technically, the hyperedge embedding matrix $\mathbf{R} \in \mathbb{R}^{k \times d}$ is computed as follows:

$$\mathbf{R} = \sigma\left(\mathbf{U}\mathbf{\Lambda}^\top \mathbf{H}\right) + \mathbf{\Lambda}^\top \mathbf{H}, \tag{4}$$

where $\mathbf{U} \in \mathbb{R}^{k \times k}$ denotes the additional trainable matrix, and $\sigma(\cdot)$ is the activation function. Afterward, the updated node embeddings $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|V|}]^\top \in \mathbb{R}^{|V| \times d}$ can be calculated as follows:

$$\mathbf{S} = \mathbf{\Lambda} \cdot \mathbf{R} = \mathbf{\Lambda}\left(\sigma\left(\mathbf{U}\mathbf{\Lambda}^\top \mathbf{H}\right) + \mathbf{\Lambda}^\top \mathbf{H}\right). \tag{5}$$

In this way, the updated node representations can effectively capture high-order semantic features. Finally, for each graph, the graph-level representation $\mathbf{s}_G$ can be obtained by summing all the refined node representations as:

$$\mathbf{s}_G = \sum\nolimits_{i=1}^{|V|} \mathbf{s}_i. \tag{6}$$

3.2. Graph Convolution on the Line Graph

Hypergraph modeling empowers our model to capture high-order semantic features and long-range dependencies. However, we argue that real-world graphs may involve interactions among higher-order substructures, whose interactions often reveal underlying meaningful semantic patterns. For instance, in biology, certain graph motifs may interact and collectively determine the properties of the graph (Borgwardt et al., 2005; Chen et al., 2006).

To this end, we leverage the learned hypergraph to introduce the line graph, effectively capturing interactions among hyperedges and providing a more profound exploration of the underlying semantic structure of the graph. Specifically, based on the hypergraph adjacency matrix $\mathbf{\Lambda} \in \mathbb{R}^{|V|\times k}$, we can construct the line graph $L(G) = (V_L, E_L)$ according to the Definition 3, where each node feature of the line graph is represented by the previously mentioned hyperedge embeddings $\mathbf{R} \in \mathbb{R}^{k\times d}$, and the adjacency matrix is denoted by $\mathbf{A}_L \in \mathbb{R}^{k\times k}$. Then, we can treat the line graph as a regular graph and adopt a GNN-based encoder to obtain its graph-level representation $\mathbf{t}_G$, which is calculated as:

$$\begin{aligned} \mathbf{T} = \sigma(\hat{\mathbf{A}}_L \mathbf{R}\mathbf{W}), \quad \hat{\mathbf{A}}_L = \tilde{\mathbf{D}}_L^{-\frac{1}{2}} \tilde{\mathbf{A}}_L \tilde{\mathbf{D}}_L^{-\frac{1}{2}}, \\ \mathbf{t}_G = \sum\nolimits_{i=1}^{k} \mathbf{t}_i, \end{aligned} \tag{7}$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_k]^\top \in \mathbb{R}^{k\times d}$, $\tilde{\mathbf{A}}_L = \mathbf{A}_L + \mathbf{I}$, $\tilde{\mathbf{D}}_L$ is the degree matrix of $\tilde{\mathbf{A}}_L$, $\mathbf{W}$ is the trainable weight matrix. In this way, the representation $\mathbf{t}_G$ can be viewed as another perspective of the original graph, considering interactions among high-order substructures and better capturing the underlying semantic structure.

3.3. Relational Consistency Learning

Having acquired the graph representations from the two perspectives, i.e., $\mathbf{s}_G$ from the hypergraph view and $\mathbf{t}_G$ from the line graph view, they capture semantic knowledge of the graph from different levels. More specifically, when we regard nodes within the same hyperedge as a substructure, the hypergraph convolution and line graph convolution channels within our network can be viewed as distinct perspectives describing intra-substructure and inter-substructure information. Thus, it is natural to consider how to integrate these two representations, enabling them to mutually supervise and reinforce each other.

Furthermore, in semi-supervised scenarios, the availability of limited label annotations often leads to unreliable and biased pseudo labels. As a result, striving to align the graph representations or pseudo labels of the same instance directly might not prove to be the most effective strategy, especially for unlabeled graphs. To address this challenge, we suggest enhancing the representation of each instance by transfering knowledge among instances. This is achieved by comparing the similarities of the instance to other labeled graphs in the embedding spaces of the two branches.

Technically, we begin by randomly selecting a subset of labeled graphs $\{G_1, \ldots, G_M\} \in \mathcal{G}^L$ as anchor graphs, which are stored in a memory bank. Then, we employ two branches to embed these anchor graphs, yielding the respective representations $\{\mathbf{s}_m\}_{m=1}^M$ and $\{\mathbf{t}_m\}_{m=1}^M$. To ensure that the anchor graphs sufficiently cover the neighborhoods of any unlabeled graph in the embedding space and facilitate the transfer of knowledge from labeled to unlabeled graphs, a large number of anchor graphs is required. However, processing excessive anchor graphs in a single iteration can become computationally expensive due to limitations in computation and memory resources. To address this challenge, we maintain a memory bank as a queue, which dynamically stores a set of anchor graphs selected from the most recent iterations of the two branches.

Specifically, take the hypergraph branch as an example, for an unlabeled graph G_u, we can calculate the relational similarity distribution between its graph representation s_u with the representations $\{s_m\}_{m=1}^M$ of anchor graphs as:

$$\mathcal{P}_u^m = \frac{\exp\left(\cos\left(\mathbf{s}_u, \mathbf{s}_m\right)/\tau\right)}{\sum_{m'=1}^{M} \exp\left(\cos\left(\mathbf{s}_u, \mathbf{s}_{m'}\right)/\tau\right)}. \tag{8}$$

In accordance with You et al. (2020), τ is the temperature parameter set to 0.5, $\cos(a, b)$ denotes the cosine similarity defined as $\frac{a\cdot b}{\|a\|_2 \|b\|_2}$. Analogously, the relational similarity distribution in the line graph branch can be obtained as:

$$\mathcal{Q}_u^m = \frac{\exp\left(\cos\left(\mathbf{t}_u, \mathbf{t}_m\right)/\tau\right)}{\sum_{m'=1}^{M} \exp\left(\cos\left(\mathbf{t}_u, \mathbf{t}_{m'}\right)/\tau\right)}. \tag{9}$$

In this way, we propose the relational consistency learning to encourage the consistency between distributions $\mathcal{P}^u = [\mathcal{P}_1^u, \ldots, \mathcal{P}_M^u]$ and $\mathcal{Q}^u = [\mathcal{Q}_1^u, \ldots, \mathcal{Q}_M^u]$ by minimizing the Kullback-Leibler (KL) Divergence as:

$$\mathcal{L}_{con} = \frac{1}{|\mathcal{G}^U|} \sum_{u\in\mathcal{G}^U} \frac{1}{2} \left(D_{\mathrm{KL}}\left(\mathcal{P}^u \| \mathcal{Q}^u\right) + D_{\mathrm{KL}}\left(\mathcal{Q}^u \| \mathcal{P}^u\right)\right). \tag{10}$$

Optimization Framework. To introduce the supervision signals to guide the model, for each labeled graph G_l, we concatenate the graph representations $\mathbf{s}_l$ and $\mathbf{t}_l$ from the two branches and feed the fused representation to a classifier (multi-layer perception) for label prediction $\hat{\mathbf{y}}$. We then adopt cross-entropy loss to compute the supervised loss:

$$\mathcal{L}_{sup} = -\frac{1}{|\mathcal{G}^L|} \sum_{i\in\mathcal{G}^L} \mathbf{y}_i \log\left(\hat{\mathbf{y}}_i\right), \tag{11}$$

where $\mathbf{y}_i$ denote the ground-truth label for labeled graph G_i. Finally, we integrate the supervised loss $\mathcal{L}_{sup}$ with

Algorithm 1 Optimization Framework of the HEAL

Input: Labeled graphs $\mathcal{G}^L$, unlabeled graphs $\mathcal{G}^U$, the dimension of hidden embeddings d, the number of hyperedges k, the number of anchor graphs M, the hyper-parameter β
Output: Trained classifier

1: Initialize the parameters of the GNN-based encoder, hypergraph structure Learning, and classifier.
2: Select M anchor graphs from labeled set $\mathcal{G}^L$ to construct the memory bank.
3: **while** *not convergence* **do**
4: Sample a minibatch $\mathcal{B}^L$ and $\mathcal{B}^U$.
5: Forward propagation $\mathcal{B}^L$ and $\mathcal{B}^U$ via twin branches.
6: Compute consistency loss $\mathcal{L}_{con}$ by Eq. (10).
7: Compute supervised loss $\mathcal{L}_{sup}$ by Eq. (11).
8: Update the parameters by gradient descent to minimize $\mathcal{L}$ by Eq. (12).
9: Update the memory bank of the two branches following the first-in-first-out principle.
10: **end while**

relational consistency loss $\mathcal{L}_{con}$ in our combined loss:

$$\mathcal{L} = \mathcal{L}_{sup} + \beta \cdot \mathcal{L}_{con}, \tag{12}$$

where β is a balance hyper-parameter. Our training algorithm is detailed in Algorithm 1.

3.4. Computational Complexity Analysis

With $|V|$ as the average number of nodes in input graphs, d is the hidden dimensions, and k is the hyperedge number, the total time complexity of obtaining hyperedge structure and performing hyperedge convolution is $O(kd(|V|+k))$. And the time complexity of line graph convolution is $O(kd(k+d))$. Moreover, the time complexity of computing relational consistency loss for a graph is $O(Md)$, where M is the number of anchor graphs. Therefore, the total computational complexity of HEAL is $O(Md + kd(|V|+k+d))$.

4. Experiment

4.1. Experimental Setups

Datasets. We assess our HEAL on six publicly available datasets, comprising two bioinformatics datasets PROTEINS (Neumann et al., 2016) and DD (Dobson & Doig, 2003); three datasets derived from social networks, specifically IMDB-B, IMDB-M, and REDDIT-M-5k (Yanardag & Vishwanathan, 2015); and one dataset from scientific collaborations, COLLAB (Yanardag & Vishwanathan, 2015). We employ the same data split with DualGraph (Luo et al., 2022), where the labeled training set, unlabeled training set, validation set, and test set are proportioned in a 2:5:1:2 ratio. Unless explicitly stated, we use 50% of the labeled data (corresponding to 10% of all samples) for training.

Baseline Methods. We conduct thorough comparisons with diverse methods categorized into three groups: traditional graph algorithms, conventional semi-supervised learning methods, and graph-specific semi-supervised learning approaches. Traditional graph methods include WL (Shervashidze et al., 2011), Graph2Vec (Narayanan et al., 2017), and Sub2Vec (Adhikari et al., 2018). Conventional semi-supervised learning approaches include EntMin (Grandvalet & Bengio, 2004), Mean-Teacher (Tarvainen & Valpola, 2017) and VAT (Miyato et al., 2018)). The category of graph-specific semi-supervised learning methods encompasses InfoGraph (Sun et al., 2020), GraphCL (You et al., 2020), ASGN (Hao et al., 2020), JOAO (You et al., 2021)), DualGraph (Luo et al., 2022), KGNN (Ju et al., 2022), and TGNN (Ju et al., 2023b).

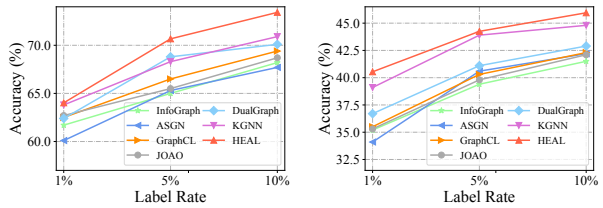
Implementation Details. For the implementation of HEAL, we employ the GIN (Xu et al., 2019) to configure the GNN-based encoder. We empirically set the embedding dimension to 32, the batch size to 64, and the training epochs to 300. For our hypergraph structure learning module, we empirically set the number of hyperedge k to 32. Moreover, we set the weight balance hyper-parameter β for $\mathcal{L}_{con}$ to 0.01. The model HEAL is optimized using the Adam optimizer with an initial learning rate of 0.01, and the weight decay is set to 0.0005. Results are reported as the average classification accuracy (in %) and the standard deviation over five runs.

4.2. Results and Analysis

The quantitative outcomes of semi-supervised graph classification are presented in Table 1, and the following observations can be made from the results. (i) Traditional graph methods generally underperform compared to other methods, highlighting the superior capability of graph neural networks in harnessing valuable semantic information through advanced representation learning from graph-structured data. (ii) Graph-specific semi-supervised learning methods show enhanced performance over conventional semi-supervised learning techniques, demonstrating the suitability of recent graph semi-supervised learning for challenging graph classification tasks. Notably, KGNN and TGNN achieve nearly the best performance on most datasets, outperforming previous state-of-the-art approaches. The success of these approaches can be attributed to their proficient use of unlabeled samples, which boosts consistency across different modules in processing unlabeled graphs. (iii) Our proposed HEAL outperforms other baseline methods across the majority of benchmarks, demonstrating the robustness of our approach. The enhancement in performance can be attributed to the utilization of hypergraph and line graph convolution branches, which enable the capture of higher-order relationships among nodes. Additionally, the relational consistency learning module facilitates knowledge transfer between the two branches, leading to enhanced mutual guidance and

Table 1: Overview of performance (in %) across six benchmark graph classification datasets, with standard deviations calculated over five runs. The highest scores are marked in bold, and the second-highest scores are underlined.

Methods	PROTEINS	DD	IMDB-B	IMDB-M	REDDIT-M-5k	COLLAB
WL	63.5 ± 1.6	57.3 ± 1.2	58.1 ± 2.3	33.3 ± 1.4	37.0 ± 0.9	62.9 ± 0.7
Sub2Vec	52.7 ± 4.5	46.4 ± 3.2	44.9 ± 3.5	31.8 ± 2.7	35.1 ± 1.5	60.8 ± 1.4
Graph2Vec	63.1 ± 1.8	53.7 ± 1.6	61.2 ± 2.6	38.1 ± 2.2	38.1 ± 1.4	63.6 ± 0.9
EntMin	62.7 ± 2.7	59.8 ± 1.3	67.1 ± 3.7	37.4 ± 1.2	38.7 ± 2.8	63.8 ± 1.6
Mean-Teacher	64.3 ± 2.1	60.6 ± 1.8	66.4 ± 2.7	38.8 ± 3.6	39.2 ± 2.1	63.6 ± 1.4
VAT	64.1 ± 1.2	59.9 ± 2.6	67.2 ± 2.9	39.6 ± 1.4	38.9 ± 3.2	64.1 ± 1.1
InfoGraph	68.2 ± 0.7	67.5 ± 1.4	71.8 ± 2.3	42.3 ± 1.8	41.5 ± 1.7	65.7 ± 0.4
ASGN	67.7 ± 1.2	68.5 ± 0.6	70.6 ± 1.4	41.2 ± 1.4	42.2 ± 0.8	65.3 ± 0.8
GraphCL	69.4 ± 0.8	68.7 ± 1.2	71.2 ± 2.5	43.7 ± 1.3	42.3 ± 0.9	66.4 ± 0.6
JOAO	68.7 ± 0.9	67.9 ± 1.3	71.0 ± 1.9	42.6 ± 1.5	42.1 ± 1.2	65.8 ± 0.4
DualGraph	70.1 ± 1.2	69.8 ± 0.8	72.1 ± 0.7	$\mathbf{44.8 \pm 0.4}$	42.9 ± 1.4	67.2 ± 0.6
KGNN	70.9 ± 0.5	70.5 ± 0.6	72.5 ± 1.6	43.3 ± 0.7	$\underline{44.8 \pm 0.6}$	67.4 ± 0.5
TGNN	$\underline{71.0 \pm 0.7}$	$\underline{70.8 \pm 0.9}$	$\underline{72.8 \pm 1.7}$	42.9 ± 0.8	43.8 ± 1.0	$\underline{67.7 \pm 0.4}$
HEAL	$\mathbf{73.4 \pm 0.8}$	$\mathbf{72.1 \pm 0.9}$	$\mathbf{73.5 \pm 1.5}$	$\underline{44.3 \pm 0.6}$	$\mathbf{45.9 \pm 1.0}$	$\mathbf{68.3 \pm 0.5}$



(a) PROTEINS (b) REDDIT-M-5k

Figure 2: Results of HEAL and baselines with different labeling ratios on PROTEINS and REDDIT-M-5k datasets.

improved performance. As for IMDB-M dataset, this discrepancy in performance can be attributed to the relatively smaller size of nodes (13.00) and edges (65.94) in the IMDB-M dataset compared to the others. In such cases, the use of hypergraph convolution may not be as effective, since the construction of a hypergraph might be unnecessary when dealing with a small-scale graph.

Influence of Labeling Ratio. We evaluate our model HEAL and baselines on the PROTEINS and REDDIT-M-5k datasets by varying the labeling ratio of the training data, as shown in Figure 2. The findings illustrate that as the number of labeled instances rises, the performance of both HEAL and the baselines enhances, suggesting that adding more labeled data effectively boosts performance. Notably, our proposed HEAL demonstrates the best performance among all methods in most scenarios, underscoring the advantages of effectively incorporating graph semantics across different levels of higher-order structure.

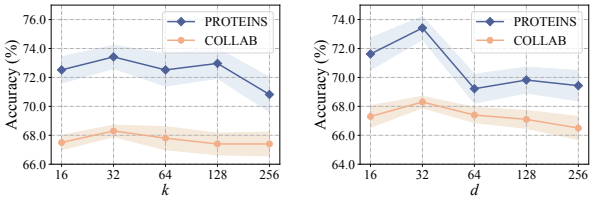
Table 2: Ablation study of HEAL with several variants.

Methods	PROTEINS	REDDIT-M-5K	COLLAB
Hyper-Sup	69.1 ± 1.0	41.2 ± 1.2	64.4 ± 0.7
Line-Sup	66.7 ± 1.2	40.7 ± 1.4	64.2 ± 0.9
Dual-Sup	70.1 ± 0.9	42.4 ± 1.2	65.3 ± 0.8
Hyper-Ensemble	71.8 ± 1.1	43.6 ± 1.3	66.7 ± 0.7
Line-Ensemble	71.5 ± 1.1	43.9 ± 1.4	66.0 ± 0.8
HEAL	$\mathbf{73.4 \pm 0.8}$	$\mathbf{45.9 \pm 1.0}$	$\mathbf{68.3 \pm 0.5}$

4.3. Ablation Study

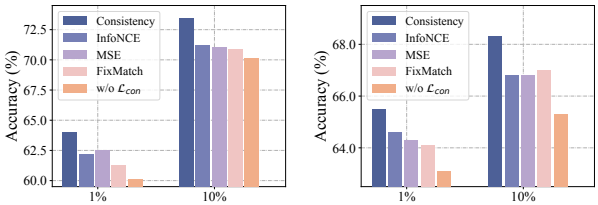
We carry out ablation studies to evaluate the impact of each component within our model. We test several model variants as outlined below: (i) **Hyper-Sup** trains a single hypergraph convolution network using only supervised signals. (ii) **Line-Sup** trains a single line graph convolution network in a supervised manner. (iii) **Dual-Sup** trains a dual branch of the hypergraph and line graph convolution network in a supervised manner. (iv) **Hyper-Ensemble** replaces the line graph convolution branch with another hypergraph learning module, using a different initialization. (v) **Line-Ensemble** replaces hypergraph convolution branch with another line graph convolution module, also with different initialization.

The outcomes for various variants are displayed in Table 2. Firstly, it is evident that Hyper-Sup generally outperforms Line-Sup. Moreover, the combination of both hypergraph and line graph convolution (Dual-Sup) leads to improved performance, validating the joint effectiveness of both branches. Secondly, Hyper-Ensemble (Line-Ensemble) outperforms Hyper-Sup (Line-Sup), indicating that our relational consistency learning module effectively leverages



(a) Hyperedge Number (b) Embedding Dimensions

Figure 3: Hyper-parameter sensitivity study of HEAL on PROTEINS and COLLAB datasets.



(a) PROTEINS (b) COLLAB

Figure 4: Performance comparison with different labeling ratios w.r.t. different types of $\mathcal{L}_{con}$.

unlabeled samples and improve the model via ensemble techniques. Lastly, our full model outperforms both ensemble versions, underscoring its enhanced effectiveness in extracting similarities by simultaneously considering both hypergraph and line graph perspectives.

4.4. Hyper-parameter Study

We analyze how the performance of HEAL changes with different hyper-parameter configurations. In particular, we assess the influence of the number of embedding dimensions d and the number of hyperedges k used within the hypergraph structure learning module.

Influence of Hyperedge Numbers. We first conducted the influence of the number of hyperedges, varying k in $\{16, 32, 64, 128, 256\}$ while keeping all other hyperparameters fixed. The results are shown in Figure 3(a), which reveal that the accuracy initially increases as the hyperedge number rises from 16 to 32. However, after reaching a peak, the accuracy starts to decrease with further increases in the number of hyperedges. This trend can be attributed to the potential capture of noise or aggregation of redundant information from hyperedges as the number of hyperedges grows.

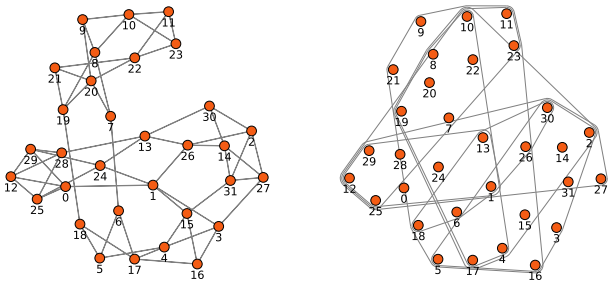
Influence of Embedding Dimensions. We further evaluated the impact of embedding dimensions by varying d in $\{16, 32, 64, 128, 256\}$ while maintaining all other hyperparameters constant. The results depicted in Figure 3(b) indicate that performance reaches the peak when the embedding dimensions approach 32. This trend suggests that while increasing d initially enhances the model's representation ability, it may result in overfitting if d continues to rise.

4.5. Analysis of Consistency Loss

The proposed relational consistency loss $\mathcal{L}_{con}$ aims to enhance each instance representation by exchanging instance knowledge from two correlated views. To highlight the advantages of the consistency learning module, we carry out experiments that compare $\mathcal{L}_{con}$ against other commonly employed contrastive losses (i.e. InfoNCE loss and mean squared error (MSE) loss) and consistency learning approaches (i.e., FixMatch loss). The results are presented in Figure 4, from which we can draw the following conclusions. Firstly, we observe a significant performance decline in HEAL when $\mathcal{L}_{con}$ is omitted (w/o $\mathcal{L}_{con}$), compared to its inclusion with various forms of contrastive loss. This observation highlights the importance of inter-branch knowledge communication in enhancing semi-supervised classification. Secondly, contrastive losses (Consistency, InfoNCE, MSE) generally surpass the pseudo-labeling consistency loss (FixMatch), which may be due to the biased pseudo-labels determined by unreliable prediction probabilities. Finally, our proposed consistency loss achieves better results than both InfoNCE and MSE. These methods typically concentrate on strictly enforcing similarities between two graph representations. Our findings suggest that a more flexible alignment of similarity distributions between hypergraph and line graph views enhances the effectiveness of consistency learning.

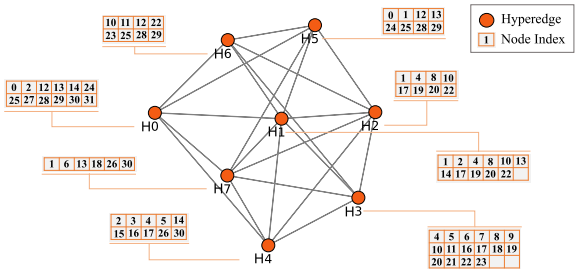
4.6. Visualization Analysis

We conducted a case study on the PROTEINS dataset to visualize the learned hypergraph structure and corresponding line graph, thus demonstrating the effectiveness of the hypergraph structure learning module and line graph module, respectively. Here the thresholds for visualizing the learned hypergraph structure and corresponding line graph are both set to 0. In the PROTEINS dataset, each node represents secondary structure elements (helices, sheets, and turns), and the edges represent sequential or structural connections between nodes. Figure 5(a) reveals that elements in the protein are only connected to their nearest spatial neighbors, making it difficult to model higher-order interactions. However, in Figure 5(b), we showcase part of the hyperedges learned by our hypergraph structure learning module. The hypergraph structure allows elements in the protein to interact in a high-order manner, facilitating the capturing of more complex and intricate relationships within protein structures. The results demonstrate that our hypergraph structure learning module exhibits remarkable adaptability in acquiring higher-order node relationships beyond pairwise connections, enabling enhanced flexibility in modeling complex



(a) Original Graph

(b) Learned Hypergraph



(c) Learned Line Graph

Figure 5: Visualization of the original graph, hypergraph structure and line graph learned by HEAL (only demonstrating the most significant hyperedges for simplicity).

data structures. Analogously, figure 5(c) also depicts the effectiveness of the learned line graph, potentially involving interactions among hypergraphs, whose interactions often reveal underlying meaningful semantic patterns.

5. Related Work

Graph Neural Networks (GNNs) have risen as a powerful tool for handling graph-structured data, facilitating effective node and graph-level representation learning (Ju et al., 2024b). The essence of GNNs lies in their iterative process of enhancing node representations by aggregating information from neighboring nodes, allowing nodes to propagate and exchange information throughout the graph (Gilmer et al., 2017). This message passing enables GNNs to capture local neighborhood information and learn expressive node representations, which are beneficial for a wide range of tasks such as node classification (Yuan et al., 2023; Luo et al., 2023a), node clustering (Yi et al., 2023a), link prediction (Zhang & Chen, 2018; Qin et al., 2024), and graph classification (Ju et al., 2022; Luo et al., 2023b). Compared with existing methods for supervised graph classification, our work goes further and studies a promising yet challenging semi-supervised graph classification.

Hypergraph Learning have gained increasing attention for their ability to model complex relationships beyond pairwise interactions in traditional graphs. The underlying idea behind hypergraphs is to extend the concept of edges in graphs to hyperedges, which can connect multiple nodes simultaneously. This flexibility allows hypergraphs to capture higher-order dependencies and interactions among nodes. Various techniques have been proposed to leverage hypergraphs for diverse applications, including clustering (Takai et al., 2020), classification (Sun et al., 2021), link prediction (Yadati et al., 2020), traffic flow prediction (Zhao et al., 2023), knowledge graphs (Fatemi et al., 2019), and recommender systems (Xia et al., 2021). Recently, hypergraph convolutional networks have been proposed as a generalization of GCNs to handle hypergraph-structured data, enabling effective feature aggregation and representation learning in hypergraphs (Feng et al., 2019; Jiang et al., 2019; Zhang et al., 2022; Cai et al., 2022). Our HEAL also inherits the advantages of hypergraphs in modeling higher-order node relationships and additionally introduces a line graph to capture the semantic interactions between hyperedges.

Semi-supervised Learning has been proven to be a prominent approach to address the limitations of traditional supervised learning, especially when labeled data is scarce or expensive to obtain. Early works in semi-supervised learning focus on spreading label knowledge from labeled data to neighboring unlabeled data points, effectively expanding the labeled set and providing more informative data points for training (Subramanya & Talukdar, 2022; Wan et al., 2021). Another class involves consistency regularization (Laine & Aila, 2017; Tarvainen & Valpola, 2017; Lucas et al., 2022), which encourages the model to maintain stability in its predictions for perturbed versions of the same input, whether labeled or unlabeled. Compared with existing methods, our approach leverages the idea of hypergraphs for both labeled or unlabeled graphs to explore the inherent structure and relationships within the data.

6. Conclusion

In this work, we present a hypergraph-enhanced dual framework HEAL for semi-supervised graph classification, and our HEAL effectively captures graph semantics from the perspectives of hypergraph and line graph. It incorporates hypergraph structure learning to explore higher-order node dependencies and introduces a line graph to capture hyperedge interactions. Then, relational consistency learning is developed to facilitate knowledge transfer between the two branches. Experiments reveal superior performance compared to baseline methods in real-world graph datasets.

Acknowledge

This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Numbers 62306014 and 62276002) as well as the China Postdoctoral Science Foundation with Grant No. 2023M730057.

Impact Statement

The proposed HEAL framework advances semi-supervised graph classification by incorporating hypergraph and line graph perspectives, addressing the limitations of traditional pairwise node relationships. By learning higher-order node dependencies through hypergraph structure learning and capturing hyperedge interactions via a line graph, HEAL enhances the extraction of underlying semantic structures. This dual approach facilitates improved knowledge transfer and mutual guidance between the two graph representations, contributing to more accurate and insightful graph classification. This work holds potential for broad applications in domains requiring effective and efficient graph analysis, such as social network analysis, biological networks, knowledge graphs, and recommender systems.

References

Adhikari, B., Zhang, Y., Ramakrishnan, N., and Prakash, B. A. Sub2vec: Feature learning for subgraphs. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II 22*, pp. 170–182. Springer, 2018.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1): i47–i56, 2005.

Cai, D., Song, M., Sun, C., Zhang, B., Hong, S., and Li, H. Hypergraph structure learning for hypergraph neural networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 1923–1929, 2022.

Chen, J., Hsu, W., Lee, M. L., and Ng, S.-K. Nemofinder: Dissecting genome-wide protein-protein interactions with meso-scale network motifs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 106–115, 2006.

Dobson, P. D. and Doig, A. J. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.

Fatemi, B., Taslakian, P., Vazquez, D., and Poole, D. Knowledge hypergraphs: Prediction beyond binary relations. *arXiv preprint arXiv:1906.00137*, 2019.

Feng, Y., You, H., Zhang, Z., Ji, R., and Gao, Y. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3558–3565, 2019.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. *Advances in Neural Information Processing Systems*, 17:529–536, 2004.

Hao, Z., Lu, C., Huang, Z., Wang, H., Hu, Z., Liu, Q., Chen, E., and Lee, C. Asgn: An active semi-supervised graph neural network for molecular property prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 731–752, 2020.

Huang, S., Elhoseiny, M., Elgammal, A., and Yang, D. Learning hypergraph-regularized attribute predictors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 409–417, 2015.

Jiang, J., Wei, Y., Feng, Y., Cao, J., and Gao, Y. Dynamic hypergraph neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 2635–2641, 2019.

Ju, W., Yang, J., Qu, M., Song, W., Shen, J., and Zhang, M. Kgnn: Harnessing kernel-based networks for semi-supervised graph classification. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 421–429, 2022.

Ju, W., Liu, Z., Qin, Y., Feng, B., Wang, C., Guo, Z., Luo, X., and Zhang, M. Few-shot molecular property prediction via hierarchically structured learning on relation graphs. *Neural Networks*, 163:122–131, 2023a.

Ju, W., Luo, X., Qu, M., Wang, Y., Chen, C., Deng, M., Hua, X.-S., and Zhang, M. Tgnn: A joint semi-supervised framework for graph-level classification. *arXiv preprint arXiv:2304.11688*, 2023b.

Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., Qin, Y., Shen, J., Sun, F., Xiao, Z., et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, pp. 106207, 2024a.

Ju, W., Yi, S., Wang, Y., Long, Q., Luo, J., Xiao, Z., and Zhang, M. A survey of data-efficient graph learning. *arXiv preprint arXiv:2402.00447*, 2024b.

Kashima, H., Tsuda, K., and Inokuchi, A. Marginalized kernels between labeled graphs. In *Proceedings of International Conference on Machine Learning*, pp. 321–328, 2003.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *Proceedings of International Conference on Learning Representations*, 2017.

Li, J., Rong, Y., Cheng, H., Meng, H., Huang, W., and Huang, J. Semi-supervised graph classification: A hierarchical graph perspective. In *The World Wide Web Conference*, pp. 972–982, 2019.

Lucas, T., Weinzaepfel, P., and Rogez, G. Barely-supervised learning: Semi-supervised learning with very few labeled images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1881–1889, 2022.

Luo, X., Ju, W., Qu, M., Chen, C., Deng, M., Hua, X.-S., and Zhang, M. Dualgraph: Improving semi-supervised graph classification via dual contrastive learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 699–712. IEEE, 2022.

Luo, X., Ju, W., Gu, Y., Qin, Y., Yi, S., Wu, D., Liu, L., and Zhang, M. Towards effective semi-supervised node classification with hybrid curriculum pseudo-labeling. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20, 2023a.

Luo, X., Zhao, Y., Mao, Z., Qin, Y., Ju, W., Zhang, M., and Sun, Y. Rignn: A rationale perspective for semi-supervised open-world graph classification. *Transactions on Machine Learning Research*, 2023b.

Luo, X., Zhao, Y., Qin, Y., Ju, W., and Zhang, M. Towards semi-supervised universal graph classification. *IEEE Transactions on Knowledge and Data Engineering*, 36: 416–428, 2023c.

Mao, Z., Ju, W., Qin, Y., Luo, X., and Zhang, M. Rahnet: Retrieval augmented hybrid network for long-tailed graph classification. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3817–3826, 2023.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (8):1979–1993, 2018.

Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., and Jaiswal, S. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.

Neumann, M., Garnett, R., Bauckhage, C., and Kersting, K. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2):209–245, 2016.

Qin, Y., Ju, W., Wu, H., Luo, X., and Zhang, M. Learning graph ode for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., and Borgwardt, K. Efficient graphlet kernels for large graph comparison. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 488–495, 2009.

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

Subramanya, A. and Talukdar, P. P. *Graph-based semi-supervised learning*. Springer Nature, 2022.

Sun, F.-Y., Hoffmann, J., Verma, V., and Tang, J. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *Proceedings of International Conference on Learning Representations*, 2020.

Sun, X., Yin, H., Liu, B., Chen, H., Cao, J., Shao, Y., and Viet Hung, N. Q. Heterogeneous hypergraph embedding for graph classification. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 725–733, 2021.

Takai, Y., Miyauchi, A., Ikeda, M., and Yoshida, Y. Hypergraph clustering based on pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1970–1978, 2020.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pp. 1195–1204, 2017.

Wan, S., Pan, S., Yang, J., and Gong, C. Contrastive and generative graph convolutional networks for graph-based semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10049–10057, 2021.

Wang, M., Liu, X., and Wu, X. Visual classification by ℓ_1-hypergraph modeling. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2564–2574, 2015.

Whitney, H. Congruent graphs and the connectivity of graphs. *Hassler Whitney Collected Papers*, pp. 61–79, 1992.

Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., and Zhang, X. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4503–4511, 2021.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *Proceedings of International Conference on Learning Representations*, 2019.

Yadati, N., Nitin, V., Nimishakavi, M., Yadav, P., Louis, A., and Talukdar, P. Nhp: Neural hypergraph link prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1705–1714, 2020.

Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

Yi, S., Ju, W., Qin, Y., Luo, X., Liu, L., Zhou, Y., and Zhang, M. Redundancy-free self-supervised relational learning for graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 2023a.

Yi, S.-Y., Mao, Z., Ju, W., Zhou, Y.-D., Liu, L., Luo, X., and Zhang, M. Towards long-tailed recognition for graph classification via collaborative experts. *IEEE Transactions on Big Data*, pp. 1683–1696, 2023b.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823, 2020.

You, Y., Chen, T., Shen, Y., and Wang, Z. Graph contrastive learning automated. In *International Conference on Machine Learning*, pp. 12121–12132. PMLR, 2021.

Yu, J., Tao, D., and Wang, M. Adaptive hypergraph learning and its application in image classification. *IEEE Transactions on Image Processing*, 21(7):3262–3272, 2012.

Yuan, J., Luo, X., Qin, Y., Mao, Z., Ju, W., and Zhang, M. Alex: Towards effective graph transfer learning with noisy labels. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3647–3656, 2023.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31:5171–5181, 2018.

Zhang, Z., Feng, Y., Ying, S., and Gao, Y. Deep hypergraph structure learning. *arXiv preprint arXiv:2208.12547*, 2022.

Zhao, Y., Luo, X., Ju, W., Chen, C., Hua, X.-S., and Zhang, M. Dynamic hypergraph structure learning for traffic flow forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 2303–2316. IEEE, 2023.