Hypergraph Consistency Learning With Relational Distillation

Siyu Yi, *Member, IEEE*, Zhengyang Mao, Yifan Wang, Yiyang Gu, Zhiping Xiao, Chong Chen, Xian-Sheng Hua, *Fellow, IEEE*, Ming Zhang, *Member, IEEE*, and Wei Ju, *Member, IEEE*

***Abstract*—This paper studies the problem of semi-supervised learning on graphs, which has recently aroused widespread interest in relational data mininThe focal point of exploration in this area has been the utilization of graph neural networks (GNNs), which stand out for excellent performance. Previous methods, however, typically rely on the limited labeled data while ignoring the abundant structural information in unlabeled nodes inherently on graphs, easily resulting in overfitting, especially in scenarios where only a few label nodes are available. Even worse, GNNs, despite their success, are constrained by their ability to solely capture local neighborhood information through message-passing mechanisms, thereby falling short in modeling higher-order dependencies among nodes. To circumvent the above drawbacks, we propose a simple yet effective framework called Hypergraph COnsistency LeArning (HOLA). Specifically, we employ a collaborative distillation framework consisting of a teacher network and a student network. To achieve effective interaction, we propose momentum distillation, a self-training method that enables the student network to learn from pseudo-targets generated by a momentum teacher network. Further, a novel hypergraph structure learning network is developed to model complex high-order relations among nodes with relational consistency learning, thereby transferring the knowledge to the student network. Extensive experiments conducted on a variety of benchmark datasets demonstrate the superior performance of the HOLA over various state-of-the-art methods.**

Received 7 February 2024; revised 13 September 2024 and 5 November 2024; accepted 23 November 2024. Date of publication 17 February 2025; date of current version 21 October 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFC3341203, in part by the Postdoctoral Fellowship Program (Grade A) of CPSF under Grant BX20240239, in part by the National Natural Science Foundation of China under Grant 62306014 and Grant 62276002, in part by China Postdoctoral Science Foundation under Grant 2024M762201, and in part by Sichuan University Interdisciplinary Innovation Fund. The associate editor coordinating the review of this article and approving it for publication was Prof. Ngai-Man Cheung. *(Corresponding author: Wei Ju.)*

Siyu Yi is with the College of Mathematics, Sichuan University, Chengdu 610065, China (e-mail: siyuyi@scu.edu.cn).

Zhengyang Mao, Yiyang Gu, and Ming Zhang are with the School of Computer Science, National Key Laboratory for Multimedia Information Processing, Peking University-Anker Embodied AI Lab, Peking University, Beijing 100871, China (e-mail: zhengyang.mao@stu.pku.edu.cn; yiyanggu@pku.edu.cn; mzhang_cs@pku.edu.cn).

Yifan Wang is with the School of Information Technology and Management, University of International Business and Economics, Beijing 100029, China (e-mail: yifanwang@uibe.edu.cn).

Zhiping Xiao is with the Department of Computer Science, University of Washington, Seattle, WA 98195 USA (e-mail: patxiao@uw.edu).

Chong Chen and Xian-Sheng Hua are with Terminus Group, Beijing 100027, China (e-mail: chenchong.cz@gmail.com; huaxiansheng@gmail.com).

Wei Ju is with College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: juwei@scu.edu.cn).

Digital Object Identifier 10.1109/TMM.2025.3543068

***Index Terms*—Graph neural networks, semi-supervised learning, consistency learning, hypergraph learning.**

I. Introduction

Graphs serve as highly effective and natural representations for modeling structured and relational data across a diverse range of domains and applications [1], [2]. This versatility is particularly promising in areas such as multimedia, where the intricate relationships and dependencies between different elements can be effectively captured and analyzed through graph structures. In multimedia applications, graphs prove instrumental in representing connections between various multimedia components, facilitating tasks such as image or video categorization [3], [4], content recommendation [5], [6], [7], and multimedia data retrieval [8], [9], [10]. Beyond multimedia, graphs find utility in social networks [11], [12], biology [13], [14], and transportation systems [15], [16], showcasing their versatility in representing connections and dependencies within different datasets. The inherent flexibility of graphs makes them foundational in various data-driven applications and analyses.

In recent years, there has been a notable surge in interest and exploration of graph neural networks (GNNs) to analyze and understand graph-structured data. At the core, GNNs leverage a message passing mechanism [17], effectively unifying vertex attributes and graph topology. By harnessing the message-passing paradigm, GNNs excel in learning expressive node representations, enabling them to capture intricate relationships and dependencies within graphs. The popularity of GNNs can be attributed to their outstanding performance across a myriad of downstream tasks. These tasks include node classification [18], [19], [20], graph classification [21], [22], [23], [24] and graph clustering [25], [26], [27]. Among these, we investigate semi-supervised node classification in this paper, with the goal of predicting the categories of unlabeled nodes in a given graph using only a small number of labeled nodes.

The landscape of semi-supervised node classification has witnessed the emergence of several remarkable methods [19], [28], [29], [30], [31]. For example, MVGRL [29] introduces a self-supervised method for node representation learning by contrasting encodings from two structural views—first-order neighbors and graph diffusion. GRACE [31] maximizes node-level agreement through contrastive representation learning, employing graph views generated by edge removal and node feature masking for effective node embedding alignment. CG^3 [31] combines a contrastive loss for enhanced node representations with labeled and unlabeled data, along with a graph generative loss for additional supervision by extracting relationships between data features and graph topology. CLNode [19] develops a novel framework using a multi-perspective difficulty measurer and a continuous training scheduler to address challenging nodes, progressing from easy to difficult nodes. These methods have contributed substantially to the progress and breakthroughs in the field, paving the way for more sophisticated approaches in this domain.

Despite the widespread success in semi-supervised node classification, previous methods suffer from two major limitations. *On the one hand*, they usually concentrate on fitting the labeled data using GNNs but ignore the unlabeled data inherently on graphs. This issue may lead to easy overfitting, especially when annotated labels are scarce. For example, in social network analysis, focusing solely on users with known preferences or attributes might lead to an incomplete understanding of the network structure. Neglecting users with latent characteristics, which provide crucial information about community structures and interconnections, could result in a biased representation of social relationships. *On the other hand*, GNNs typically follow the neighbor aggregation in the message passing mechanism [17], resulting in each node relying only on neighbors within a few hops, thus capturing limited local information. Besides, modeling high-order dependencies between nodes is crucial for exploring global information in the graph, while existing methods fail to address this effectively, leading to sub-optimal performance. For instance, in biochemistry networks modeling protein interactions, a GNN constrained to nearby neighbors may overlook critical interactions that occur through intermediary proteins. Proteins with high-order dependencies, forming complex biological pathways, may be inadequately represented [13].

In this paper, we attempt to address these limitations by developing a simple yet powerful approach called **H**ypergraph **CO**nsistency **L**e**A**rning (HOLA) for semi-supervised node classification on graphs. Technically, our HOLA first introduces a collaborative distillation framework, consisting of a teacher network and a student network. To cooperatively supervise and deeply interact with each other, we develop momentum distillation, which can be interpreted as a form of online self-distillation, where the student network learns from confident pseudo-targets generated by the momentum teacher network, while the teacher network serves as the ensemble of exponential-moving-average versions of the student network. Note that the collaborative distillation framework can only capture local neighborhood information through the GNN network. To better explore the global semantic structure within the graph, we develop a novel hypergraph structure learning network to encode high-order connectivity among nodes and high-level interactions of hyperedge features, thus better characterizing the global data correlations beyond pairwise relationships. Further, relational consistency learning is proposed to distill the high-order semantics from the hypergraph and transfer this knowledge to the student network, guiding its optimization process. To summarize, this work makes the following contributions:

- We propose a novel approach for semi-supervised node classification on graphs, which contains a collaborative distillation framework coupled with the updation strategy of momentum distillation, thereby producing confident pseudo-targets to sufficiently explore the unlabeled data.
- To explore the global semantics within the graph, we introduce hypergraph structure learning combined with relational consistency learning to guide the student network by distilling high-order semantics from the hypergraph.
- Comprehensive experiments on a variety of benchmark datasets show that HOLA achieves superior performance compared with state-of-the-art approaches.

II. Related Work

A. Graph Neural Networks

Graph Neural Networks (GNNs) have garnered significant attention in recent years due to their remarkable success in modeling complex relationships within graph-structured data [1], [32]. The widespread adoption of GNNs can be attributed to their ability to capture intricate dependencies and patterns, making them a cornerstone in various applications [33], [34], [35]. Existing GNN methods in the literature typically fall into two main categories: those grounded in spectral graph theory and those based on spatial approaches. Spectral graph theory-driven approaches leverage the eigenvalues and eigenvectors of the graph Laplacian matrix to uncover hidden structures within the data. Notable methods such as Graph Convolutional Networks (GCNs) [36] and ChebNet [37] have demonstrated state-of-the-art performance by effectively leveraging graph Laplacian eigen-decomposition. On the other hand, spatial approaches focus on the local neighborhood relationships between nodes and emphasize the local connectivity patterns of nodes. GraphSAGE [38] and SGC [39] exemplify this category, employing node sampling and aggregation mechanisms to capture spatial dependencies. Despite these advancements, existing approaches may encounter challenges in capturing higher-order dependencies or effectively handling diverse graph structures. In contrast, our model HOLA stands out by leveraging hypergraph structure learning to capture high-order semantics, followed by relational consistency learning to allow effective knowledge transfer.

B. Semi-Supervised Learning

Semi-supervised learning (SSL) has gained prominence in machine learning due to its ability to leverage both labeled and unlabeled data, offering a cost-effective solution for training models in scenarios where obtaining labeled data is expensive or impractical [18]. The primary objective of SSL is to improve model generalization by utilizing the additional information embedded in unlabeled samples. Current SSL methods can be broadly categorized into three main classes: those based on self-training, consistency regularization, and knowledge distillation. Specifically, self-training methods hinge on iteratively expanding the labeled dataset by confidently predicting labels for unlabeled samples. A representative technique like pseudo-labeling effectively leverages the model's own predictions to iteratively refine its learning [40]. Consistency regularization introduces the notion of encouraging model predictions to be consistent under various perturbations of the input data. Methods like Virtual Adversarial Training (VAT) [41] and MixMatch [42] enforce the model to produce stable predictions across different augmentations or perturbations. Knowledge distillation involves transferring knowledge from a teacher model to a student model, where the teacher model is typically a well-trained model with high accuracy. This process encourages the student model to mimic the soft labels or intermediate representations produced by the teacher [43]. Recent approaches such as RKD [44] showcase the potential of knowledge distillation in SSL, offering improved generalization and robustness. Our proposed method HOLA is akin to the framework of knowledge distillation, where we develop a collaborative distillation framework composed of a teacher network and a student network. This framework encourages mutual enhancement between the two networks, utilizes relational consistency learning to more effectively transfer high-order relational semantic knowledge from the graph and to guide the optimization of the student network.

C. Hypergraph Learning

Hypergraph learning has emerged as a prominent field within machine learning, demonstrating remarkable success in capturing and modeling complex relationships in data [45], [46]. The primary aim of hypergraph learning is to extend traditional graph-based models by accommodating higher-order interactions and dependencies present in real-world datasets. This approach provides a more expressive representation of data, enabling improved performance in various applications [47], [48]. Existing hypergraph learning methods can be categorized into three key components, each addressing specific aspects of hypergraph modeling: hypergraph construction, hypergraph-based representation learning, and hypergraph convolution operations. The first component involves the creation of hypergraphs from raw data. Notable methods [49], [50] focus on constructing hyperedges that capture higher-order relationships among data points. There are also learning methods that directly target hypergraph-structured data [51], [52]. LE [51] proposes a hypergraph expansion method, which effectively transforms hypergraphs into simple graphs while preserving higher-order relationships. WHATsNet [52] develops a hypergraph neural network designed to address the problem of classifying edge-dependent node labels in hypergraphs by capturing node relationships within each hyperedge through attention mechanisms and positional encodings. Hypergraph-based representation learning aims to derive informative node embeddings from the hypergraph structure. CHGNN [53] employs self-supervised contrastive learning for knowledge transfer, utilizing an adaptive hypergraph view generator, an improved encoder, and a joint loss function to enhance view generation and node classification. The third component involves the development of hypergraph convolution operations. Techniques such as [54] employs attention mechanisms and neural network architectures to effectively capture and propagate information through hypergraph nodes. Different from these methods, our HOLA introduces learnable hypergraph structure learning, reducing complexity while enhancing the flexibility and effectiveness of learned node representations for semi-supervised node classification.

III. Methodology

A. Problem Definition

A graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents a set of N nodes in the graph and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set of the graph. $\mathbf{x}_i \in \mathbb{R}^F$ is the attribute feature of node v_i, where F is the dimension of attributes. Besides, each node v_i corresponds to an one-hot label vector $\mathbf{y}_i \in \{0, 1\}^K$ where K is the class number. $M(M < N)$ nodes have labels $\mathcal{Y}^L$ in semi-supervised scenarios, while the labels of the remaining $N - M$ nodes are unavailable. The objective is to estimate the missing labels $\mathcal{Y}^U$ for unlabeled nodes on graphs. Fig. 1 presents a whole depiction of our HOLA.

B. Graph Neural Networks (GNNs)

In this section, we describe the GNN as the core component of our proposed HOLA. Recently, GNNs have become a go-to solution for encapsulating both node features and graph topology. For a node v_i in the vertex set $\mathcal{V}$, its embedding at layer k is represented by $\mathbf{h}_i^{(k)}$. The neighborhood aggregation in GNNs [17] involves a two-step process: aggregating the embeddings from v_i's neighbors at layer $k-1$ and then combining these with the node's own embedding from the previous layer to form a cohesive representation at layer k. Formally, the neighborhood aggregation process of GNNs can be formulated as:

$$\begin{aligned} \mathbf{h}_{\mathcal{N}(v_i)}^{(k)} &= AGG_\theta^{(k)}(\{\mathbf{h}_j^{(k-1)}\}_{v_j \in \mathcal{N}(v_i)}), \\ \mathbf{h}_i^{(k)} &= COM_\theta^{(k)}(\mathbf{h}_i^{(k-1)}, \mathbf{h}_{\mathcal{N}(v_i)}^{(k)}), \end{aligned} \tag{1}$$

in which $\mathcal{N}(v_i)$ represents the neighbors of v_i. $AGG_\theta^{(k)}$ and $COM_\theta^{(k)}$ denote the aggregation and combination operators at the k-th layer, respectively. After K GNN layers, the output embedding vector $\mathbf{h}_i^K$ (denoted as $\mathbf{h}_i$ for simplicity in the following sections) can be used for prediction in various downstream tasks.

Nevertheless, neighborhood propagation schemes are usually fixed in GNNs, resulting in each node being heavily dependent on its attributes and neighbors. When it comes to noise attacks on node attributes and connection patterns, the network may be misled during message-passing schemes. As a result, we propose two augmentations on graphs to facilitate the generation of the disturb-invariant representations.

- *Attribute Masking:* We randomly select a subset of nodes and mask a portion of their attributes based on the assumption that introducing controlled randomness during training fosters a more robust learning process.
- *Edge Dropping:* We randomly drop certain edges from the graph following an i.i.d uniform distribution, motivated by the hypothesis that inducing controlled sparsity in the graph, through random edge removal, can lead to improved generalization and robustness.

We denote the augmented version of $\mathcal{G}$ as $\tilde{\mathcal{G}}$. After the graph neural network, we fed the node representation $\mathbf{h}_i$ into a Multi-Layer Perception (MLP) to obtain the corresponding prediction

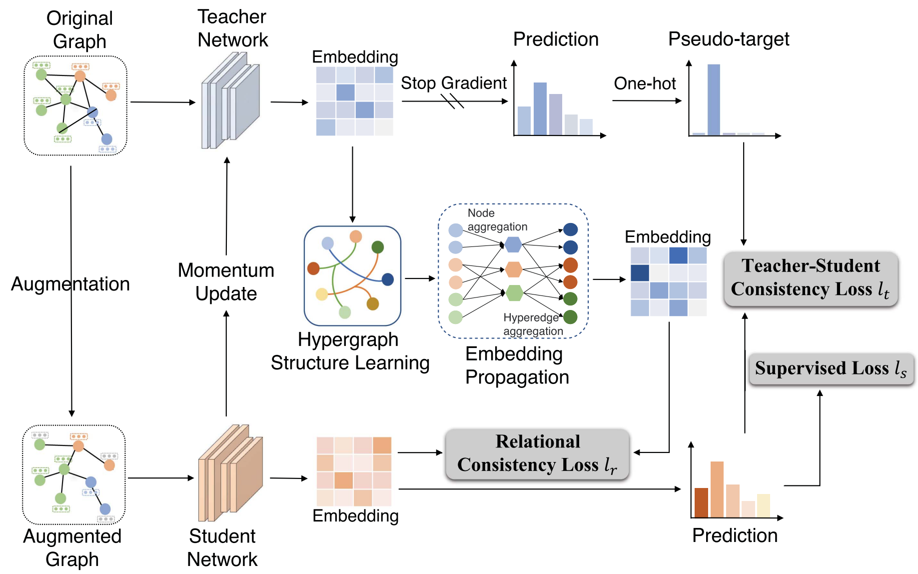


Fig. 1. The illustration of our proposed framework HOLA.

vector $\mathbf{p}_i \in \mathbb{R}^K$. For conciseness, we stack the prediction vectors into a matrix $\mathbf{P} \in \mathbb{R}^{|\mathcal{V}| \times K}$ as:

$$\mathbf{P} = \Phi_\theta(\mathcal{G}), \tag{2}$$

where θ is the parameter of the GNN network.

C. Graph Collaborative Distillation

In this section, we introduce a collaborative distillation framework, which consists of two graph neural networks with the same architecture, i.e., student network Φ_θ and teacher network Φ_ϕ. A slight difference from knowledge distillation [43] is that we only optimize the student network with a standard gradient update. To produce accurate prediction as guidance, the original graph $\mathcal{G}$ is fed into the teacher network while the augmented graph $\tilde{\mathcal{G}}$ is fed into the student network.

To increase the robustness of our model, we randomly drop some edges or attributes of $\mathcal{G}$, denoted as $\tilde{\mathcal{G}}$ before being fed into the student network while the original graph is fed into the teacher network. Let $\mathbf{P}$ and $\mathbf{Q} \in \mathbb{R}^{|\mathcal{V}| \times K}$ denote the matrix of predicted class distribution produced by the student network and teacher network, respectively. Formally, $\mathbf{P} = \Phi_\theta(\tilde{\mathcal{G}})$, $\mathbf{Q} = \Phi_\phi(\mathcal{G})$, where the row vectors $\mathbf{p}_i$ and $\mathbf{q}_i$ denote the predictions of two networks for v_i. Then, we illustrate our detailed learning objectives in our collaborative distillation framework for semi-supervised scenarios.

Supervised Loss: In semi-supervised node classification, ground-truth labels are available for M nodes on graphs. We utilize the conventional cross-entropy loss function to train the labeled nodes within the augmented graphs of the student network. Formally,

$$\ell_s = -\frac{1}{|\mathcal{Y}^L|} \sum_{i \in \mathcal{Y}^L} \mathbf{y}_i^\top \log \mathbf{p}_i. \tag{3}$$

Teacher-Student Consistency Loss: In semi-supervised settings, we propose a novel consistency learning to further explore a large number of unlabeled nodes on graphs. Inspired by recent techniques, i.e., pseudo-labeling [55] and consistency learning [42], we first generate a pseudo-target for each unlabeled node through the teacher network, and then enforce the student network to produce similar predictions. Specifically, we only retain "hard" labels (i.e., the arg max of the prediction distribution) based on the output of the teacher network. Formally, the pseudo-target is defined as:

$$\hat{q}_i = \arg\max(\mathbf{q}_i). \tag{4}$$

Note that we only preserve pseudo-targets whose largest class probability falls above a predefined threshold τ. Then we leverage pseudo-targets to guide the learning of the student network. Formally, the teacher-student consistency loss is formulated as:

$$\ell_t = -\frac{1}{|\mathcal{Y}^U|} \sum_{i \in \mathcal{Y}^U} \mathbf{1}_{(\max(\mathbf{q}_i) \geq \tau)} \hat{\mathbf{q}}_i^\top \log \mathbf{p}_i, \tag{5}$$

where $\hat{\mathbf{q}}_i \in \mathbb{R}^K$ is one-hot pseudo-target $\hat{q}_i$ and $\mathbf{1}_{(\cdot)}$ is an indicator function that returns 1 if the condition is satisfied and 0 otherwise.

Remark: Our consistency loss can be interpreted as a hybrid of two major semi-supervised learning strategies, i.e., pseudo-labeling and consistency regularization. On the one hand, previous pseudo-labeling approaches [55] retain labels with the largest class probability over a predefined threshold. On the other hand, consistency learning approaches [42] explore the unlabeled data with the assumption that the network should produce similar predictions under random data transformations, while our novel consistency loss involves one-hot pseudo-target as well as confidence measurement to output confident and disturb-invariant predictions.

Connection between Consistency Learning and Contrastive Learning: Contrastive learning (CL) [56], [57] shares a similar idea with our consistency learning that leverages the availability of pairs of semantically "similar" data points under different data augmentations, while the difference lies in CL additionally incorporates negative samples and forces the inner product of representations of similar pairs with each other to be higher on average than with negative samples.

D. Hypergraph Consistency Learning

Through the collaborative distillation framework on graphs, we can effectively leverage the information from unlabeled nodes using pseudo-labeling and consistency regularization techniques, thereby alleviating the issue of overfitting. Nevertheless, GNNs commonly employ the message-passing mechanism [17] to capture local neighborhood information, which restricts each node to depend solely on neighbors within a few hops, thereby failing to explore high-order dependencies among nodes. To tackle this, we resort to hypergraphs to model complex higher-order dependencies in the graph.

Hypergraph Structure Learning: Previous methods typically construct predefined hypergraphs based on distances [58], representations [59], or attributes [60], which often lead to suboptimal performance and high computational costs due to their inflexibility. To address this, we parameterize a learnable hypergraph structure and optimize it jointly with the network parameters. To efficiently model the hypergraph structural matrix instead of learning the dense adjacency matrix with high computational cost, we employ a low-rank strategy to flexibly learn the hypergraph structural matrix $\mathbf{\Lambda} \in \mathbb{R}^{|\mathcal{V}| \times c}$ (c denote the number of hyperedges) as follows:

$$\mathbf{\Lambda} = \boldsymbol{H} \cdot \boldsymbol{W}, \tag{6}$$

where $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the hidden embedding matrix of the original graph derived from the teacher network. $\boldsymbol{W} \in \mathbb{R}^{d \times c}$ is learnable weight matrix of hyperedges. In this way, learning the hypergraph structural matrix only takes $\mathcal{O}(c \times d)$ time complexity ($d << |\mathcal{V}|$), largely achieving model efficiency.

To effectively capture complex feature interactions and high-order dependencies among nodes, we design a hypergraph convolution to extract high-level feature information. First, we learn hyperedge embeddings by aggregating connected neighbors. Afterward, the learned hyperedge embeddings are used to globally update the node embeddings. Specifically, the hyperedge embedding matrix $\boldsymbol{R} \in \mathbb{R}^{c \times d}$ can be calculated as:

$$\boldsymbol{R} = \sigma\left(\boldsymbol{U}\mathbf{\Lambda}^\top \boldsymbol{H}\right) + \mathbf{\Lambda}^\top \boldsymbol{H}, \tag{7}$$

where extra trainable matrix $\boldsymbol{U} \in \mathbb{R}^{c \times c}$ implicitly characterizes the correlation among hyperedges. $\sigma(\cdot)$ denotes the activation function. Then, the updated node embeddings $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times d}$ can be refined as:

$$\mathbf{Z} = \mathbf{\Lambda} \cdot \boldsymbol{R} = \mathbf{\Lambda}\left(\sigma\left(\boldsymbol{U}\mathbf{\Lambda}^\top \boldsymbol{H}\right) + \mathbf{\Lambda}^\top \boldsymbol{H}\right). \tag{8}$$

Relational Consistency Loss: We have obtained node embeddings through hypergraph structure learning, which globally models the high-order interaction information among nodes. How to inject this knowledge into the student network is an urgent problem to be solved. To address this, we propose a novel relational consistency learning that effectively combines the interaction information among nodes from both global and local perspectives.

Algorithm 1: Optimization Framework of Our HOLA.

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, attribute feature set $\{\mathbf{x}_i\}_{v_i \in \mathcal{V}}$, label set $\mathcal{Y}^L$, parameters θ and ϕ in student network and teacher network respectively, training epochs T;
Ensure: Predicted lables $\mathcal{Y}^U$ for the unlabeled nodes.
1: Train the student network only using $\mathcal{Y}^L$ via (3);
2: **for** t = 1 to T **do**
3: Generate pseudo-targets for unlabeled nodes through the teacher network via (4);
4: Compute teacher-student consistency loss via (5);
5: Compute relational consistency loss via (11);
6: Compute overall learning objective ℓ via (12).
7: Update parameters θ in student network through standard gradient descent via (13);
8: Update parameters ϕ in teacher network through momentum distillation via (13);
9: Re-compute supervised Loss via (3) for next epoch;
10: **end for**

Specifically, let $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times d}$ denote the embedding matrix derived from the student network, and $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the node embedding matrix from the hypergraph learning described above. We first randomly select a subset of labeled nodes as anchor nodes to store in the memory bank and update them through a queue mechanism to reduce memory costs. For a given unlabeled node, we calculate the relational similarity distribution between its embedding representation s_i with the embedding representations $\{s_t\}_{t=1}^T$ of anchor nodes via the student network branch, which can be calculated as:

$$\mathcal{P}_t^i = \frac{\exp\left(\cos\left(s_i, s_t\right)/\tau\right)}{\sum_{t'=1}^T \exp\left(\cos\left(s_i, s_{t'}\right)/\tau\right)}, \tag{9}$$

where τ is the temperature parameter set to 0.5 following [56]. $\cos(a, b) = \frac{a \cdot b}{\|a\|_2 \|b\|_2}$ is the cosine similarity.

Similarly, the relational similarity distribution in the hypergraph learning branch can be generated in an analogous way:

$$\mathcal{Q}_t^i = \frac{\exp\left(\cos\left(z_i, z_t\right)/\tau\right)}{\sum_{t'=1}^T \exp\left(\cos\left(z_i, z_{t'}\right)/\tau\right)}. \tag{10}$$

In this way, we propose a relational consistency loss to encourage the consistency between distributions $\mathcal{P}^i = [\mathcal{P}_1^i, \ldots, \mathcal{P}_T^i]$ and $\mathcal{Q}^i = [\mathcal{Q}_1^i, \ldots, \mathcal{Q}_T^i]$ by minimizing the Kullback-Leibler (KL) Divergence between them, which can be defined as follows:

$$\ell_r = \frac{1}{|\mathcal{Y}^U|} \sum_{i \in \mathcal{Y}^U} \frac{1}{2}\left(D_{\mathrm{KL}}\left(\mathcal{P}^i \| \mathcal{Q}^i\right) + D_{\mathrm{KL}}\left(\mathcal{Q}^i \| \mathcal{P}^i\right)\right). \tag{11}$$

E. Optimization and Inference

In a nutshell, our overall learning objective is a combination version of supervised loss, teacher-student consistency loss and relational consistency loss. Formally, the final loss of our proposed HOLA is defined as:

$$\ell = \ell_s + \alpha\ell_t + \beta\ell_r, \tag{12}$$

where α, β are weight coefficients used to control their respective contributions. In the experiments, we set $\alpha = \beta = 0.1$.

During optimization, the student network is optimized with standard gradient descent with *relational distillation* while the teacher network is optimized through the updating strategy of *momentum distillation* as follows:

$$\begin{cases} \theta \leftarrow \theta - \eta\frac{\partial \ell}{\partial \theta} \\ \phi \leftarrow \epsilon\phi + (1-\epsilon)\theta, \end{cases} \tag{13}$$

where η denotes the learning rate and ϵ is a momentum coefficient. In this way, parameters in the teacher network evolve smoothly. When it comes to inference, we feed the original graph into the teacher network followed by an MLP classifier, and output prediction distribution for each node. The whole optimization procedure is depicted in Algorithm 1.

IV. Experiments

In this section, we demonstrate the efficacy of our HOLA by conducting comprehensive experiments on six real-world datasets. The key highlights of our findings include:

- Our HOLA consistently demonstrates significantly better performance than all competing baselines across various experimental settings.
- We conduct ablation studies to dissect the impact and efficiency of the different components incorporated within our HOLA, providing insights into how each contributes to its overall effectiveness.
- Our method exhibits stable performance across a range of key hyper-parameters, demonstrating robustness and reliability in practical applications.
- We conduct a case study to effectively showcase the high-order dependencies among nodes through the discerned hypergraph structure, thereby highlighting the efficacy of the hypergraph structure learning module.

A. Experimental Settings

Datasets: Our HOLA is evaluated across six widely adopted benchmark datasets, encompassing various domains. These datasets consist of three paper citation networks, i.e., Cora, CiteSeer, and PubMed [61], [62], two purchasing network datasets sourced from Amazon, namely Amazon Computers and Amazon Photo [63], and one co-author network dataset named Coauthor CS [63]. In the paper citation datasets, nodes represent publications, and edges signify citation relationships, with the primary goal being the classification of these nodes into distinct subject areas. In the Amazon-derived purchasing networks, nodes are products, and edges connect frequently co-purchased items. The Coauthor CS dataset represents a co-authorship network, with nodes as authors and edges signifying collaborative authorship. An overview of the datasets' characteristics is presented in Table I.

TABLE I
Statistics of Datasets Used in Experiments

Dataset	#Nodes	#Edges	#Features	#Classes
Cora	2,708	5,278	1,433	7
CiteSeer	3,327	4,552	3,703	6
PubMed	19,717	44,324	500	3
Amazon Computers	13,752	245,861	767	10
Amazon Photo	7,650	119,081	745	8
Coauthor CS	18,333	81,894	6,805	15

For three citation datasets, we adopt the same splits with [31] to create train/validation/test datasets. For the other three datasets, the training set and validation set both contain 20 labeled nodes per class, and the rest make up the test set.

Compared Baselines: To assess the merits and efficacy of our developed framework HOLA, we benchmark it against state-of-the-art baseline models which are widely recognized for their proficiency in semi-supervised node classification on graphs. These models include Chebyshev [37], GCN [36], GAT [64], SGC [39], DGI [28], MVGRL [29], AM-GCN [65], GRACE [30], CG^3 [31], CLNode [19], SuperGAT [66], Gapformer [67] and RCL [68].

Implementation Details: In all baseline methods and our own approach, we employ a two-layer GCN [36] as the standard GNN backbone for a fair comparison. The GNN backbone consists of two GCN layers with hidden dimensions 64 for three citation network datasets and 256 for the other three datasets. The momentum coefficient ϵ is set to 0.99 following [69] and the threshold τ for defining the largest class of pseudo-labels is set to 0.9. We utilize the Adam optimizer, setting the initial learning rate to 0.01 and employing a decay rate of 0.0005. Throughout our experiments, we present the average accuracy results and their standard deviations, calculated from five separate trials. Hyperparameters are tuned using the validation dataset, while the test dataset is employed to determine the final performance. The parameters for baseline methods are adopted from their respective original papers, following their recommended tuning strategies for optimal performance.

B. Experimental Results

Table II presents the comparative results across all six datasets, revealing the following insights:

- GCN-based approaches (GCN, GAT and SGC) consistently outperform the traditional Chebyshev method. This superiority underscores the advanced representation-learning capabilities inherent in GCN, playing a pivotal role in significantly enhancing the performance of semi-supervised node classification tasks.
- Among the various methods considered, those leveraging unlabeled data (DGI, MVGRL, AM-GCN, GRACE, CG^3, CLNode, SuperGAT and our HOLA) consistently demonstrate superior performance. This underscores the critical role of exploring additional unlabeled data via self-supervised and semi-supervised techniques as an essential supplement for enhancing overall performance.

TABLE II
CLASSIFICATION ACCURACY RESULTS (IN %) FROM TEN ITERATIONS ON SIX BENCHMARK DATASETS

Methods	Cora	CiteSeer	PubMed	Amazon Computers	Amazon Photo	Coauthor CS
Chebyshev [37]	80.7±0.2	70.2±0.6	77.4±0.1	72.5±0.0	88.4±0.1	90.4±0.2
GCN [36]	81.3±0.4	71.5±0.2	78.8±0.6	77.7±0.7	88.1±0.8	91.6±0.7
GAT [64]	82.7±0.1	70.7±0.4	78.5±0.2	79.5±0.2	88.0±0.6	91.2±0.5
SGC [39]	77.7±0.0	72.6±0.0	76.4±0.0	74.8±0.1	87.9±0.1	90.2±0.2
DGI [28]	80.9±0.3	71.4±0.2	76.3±1.1	77.7±0.8	85.3±0.9	90.6±0.5
MVGRL [29]	81.3±0.4	71.9±0.1	79.3±0.1	79.5±0.8	88.1±0.2	91.7±0.1
AM-GCN [65]	81.0±0.3	72.8±0.4	OOM	80.9±0.7	91.3±0.2	OOM
GRACE [30]	82.8±0.3	71.3±0.7	79.0±0.2	75.1±0.1	83.2±0.1	91.2±0.2
CG^3 [31]	83.5±0.3	73.7±0.2	79.2±0.6	80.5±0.1	90.0±0.2	92.4±0.1
CLNode [19]	82.5±0.6	73.3±0.6	80.3±0.9	80.1±0.8	90.5±1.0	92.5±0.6
SuperGAT [66]	**84.3±0.6**	72.6±0.7	**81.7±0.4**	81.6±0.4	91.8±0.7	90.2±0.5
Gapformer [67]	83.4±0.3	72.3±0.4	80.1±0.3	81.2±0.6	91.3±0.5	91.8±0.5
RCL [68]	81.7±0.5	71.9±0.5	79.0±0.4	81.4±0.4	89.1±0.6	91.2±0.4
HOLA (Ours)	84.2±0.5	**73.9±0.6**	80.6±0.4	**81.8±0.6**	**92.2±0.7**	**93.4±0.4**

The top-performing results are highlighted in bold, while the runner-up results are underlined. 'OOM' indicates results of memory overflow.

TABLE III
CLASSIFICATION ACCURACY RESULTS (IN %) ON THE CORA DATASET FOR VARYING LABEL RATES

Label Rate	0.5%	1%	2%	3%	5%	10%	20%	50%
Chebyshev	37.9	59.4	73.5	76.1	80.7	82.6	82.4	82.9
GCN	47.8	63.9	72.7	76.4	81.3	82.1	85.0	86.5
GAT	57.1	70.9	74.3	78.2	82.7	83.4	85.3	87.2
SGC	48.4	66.5	69.7	73.9	77.7	78.9	81.2	79.9
DGI	68.0	73.4	76.7	78.3	80.9	81.2	81.3	81.6
MVGRL	57.6	67.6	76.2	77.8	81.3	83.8	84.5	84.9
GRACE	63.8	73.5	75.2	76.2	82.8	83.6	84.4	85.9
CG^3	68.1	74.2	77.3	79.1	83.5	84.3	85.1	86.6
CLNode	63.1	68.1	75.0	76.0	82.5	83.2	84.3	85.9
SuperGAT	64.0	72.3	77.3	**81.3**	**84.3**	**85.1**	85.5	86.6
Gapformer	65.3	72.7	76.9	79.6	83.1	83.9	85.2	86.9
RCL	62.9	71.5	69.2	76.2	81.7	82.8	83.6	87.8
HOLA(Ours)	**70.0**	**76.9**	**77.8**	80.6	84.2	84.6	**87.5**	**88.3**

TABLE IV
CLASSIFICATION ACCURACY RESULTS (IN %) ON THE CITESEER DATASET FOR VARYING LABEL RATES

Label Rate	0.5%	1%	2%	3%	5%	10%	20%	50%
Chebyshev	34.0	58.3	64.6	67.2	70.2	71.7	72.2	75.7
GCN	47.6	55.8	65.3	69.2	71.5	72.6	73.4	77.6
GAT	53.2	63.9	68.3	69.5	71.2	72.1	75.1	79.0
SGC	46.8	59.3	67.1	68.6	72.7	73.0	74.5	78.8
DGI	61.0	65.8	67.5	68.8	71.6	72.3	73.1	76.5
MVGRL	61.3	65.1	68.5	70.3	71.2	72.8	73.1	74.8
GRACE	61.8	62.5	70.7	71.4	71.9	73.0	74.2	76.6
CG^3	62.9	70.1	70.9	71.7	73.9	74.5	74.8	77.2
CLNode	61.3	67.2	68.4	70.8	73.9	74.4	75.0	78.3
SuperGAT	59.8	66.1	68.6	72.3	73.9	74.2	75.1	78.7
Gapformer	61.5	67.3	69.4	72.0	73.6	73.8	74.7	78.8
RCL	59.9	65.8	67.5	71.8	72.6	73.0	74.2	78.5
HOLA	**63.5**	**70.6**	**71.3**	**72.8**	**74.1**	**74.9**	**75.2**	**79.3**

- Our proposed method HOLA shows competitive performance across most datasets. Compared to the leading baseline, SuperGAT, our HOLA outperforms it on 4 out of 6 datasets. This advantage may stem from our approach's focus on enhancing semantic learning from a different angle. By integrating relational consistency learning within the collaborative distillation framework, our method effectively leverages both global and local information and explores unlabeled data. In contrast, SuperGAT improves performance by incorporating edge self-supervision within the graph attention design.

C. Impact of Label Rates

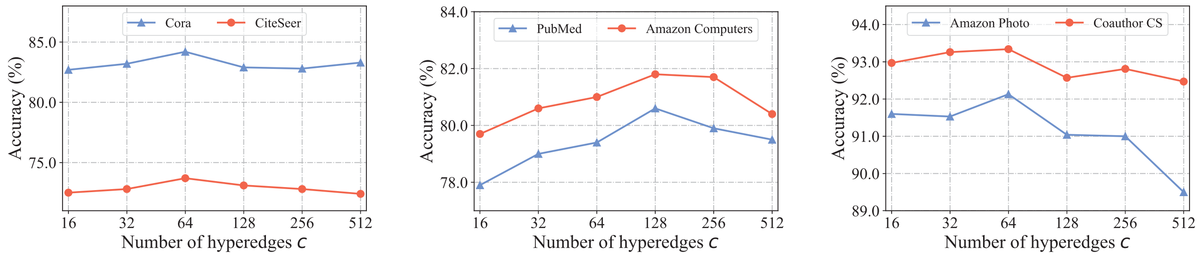
To gain a deeper understanding of our HOLA's performance under varying levels of supervision, we conduct experiments with different proportions of labeled samples to assess its adaptability. Following the approach outlined in [31], we systematically varied the label rates on the Cora and CiteSeer datasets in 0.5%, 1%, 2%, 3%, 5%, 10%, 20%, 50%. The results are presented in Tables III and IV.

Across the diverse label rate settings, our proposed framework HOLA outperforms the baseline methods in most settings. This robust performance demonstrates the remarkable versatility of our HOLA in handling datasets with scarce supervision. In situations where labeled samples are severely limited, our approach HOLA exhibits a substantial performance advantage over the baseline methods. This observation underscores the efficacy of our consistency learning module, which plays a significant role in enhancing learning when confronted with minimal supervision.

D. Sensitivity Analysis

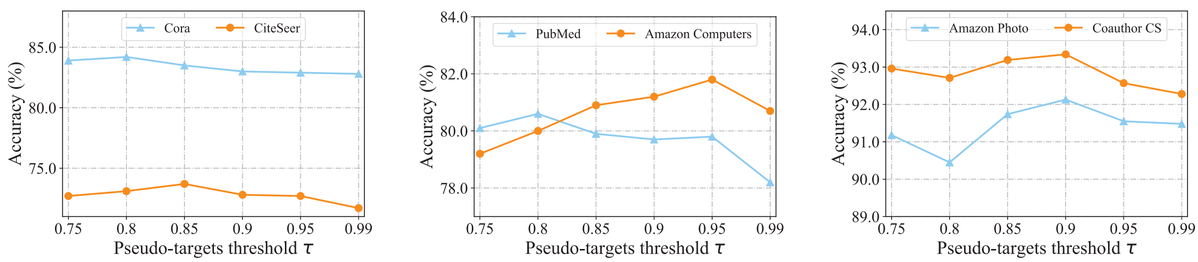
In this section, we delve deeper into the impact of hyperparameters within the HOLA framework, specifically focusing on three crucial aspects: the number of hyperedges, the pseudo-target threshold, and the embedding dimension within the hidden layer.

To begin, we explore the impact of numbers of hyperedges c, considering a range of values from 16 to 512. The results, as depicted in Fig. 2, uncover intriguing trends. Initially, increasing the value of c is associated with a notable enhancement in



(a) Cora and CiteSeer (b) Pubmed and Amazon Computers (c) Amazon Photo and Coauthor CS

Fig. 2. Sensitivity analysis w.r.t. different settings of hyperedge number c on all six datasets.



(a) Cora and CiteSeer (b) Pubmed and Amazon Computers (c) Amazon Photo and Coauthor CS

Fig. 3. Sensitivity analysis w.r.t. different settings of pseudo labeling threshold τ on all six datasets.

performance. This observation suggests that a higher number of hyperedges allows the model to capture more complex relationships and dependencies among nodes, thereby improving its representation power. However, it is crucial to note that pushing the value of c to excessively high levels can result in a decline in performance. This phenomenon might be attributed to the generation of overly intricate hyperedge-specific cross-node structures when using a large number of hyperedges. These intricate structures could introduce noise and unnecessary complexity into the model, ultimately impairing its ability to generalize effectively. We also determine an optimal configuration for the hyperparameter c that results in peak performance. This peak performance is achieved with c set to 32 for smaller datasets like Cora and CiteSeer, while larger values of c (e.g., 64 or 128) are necessary for larger-scale datasets.

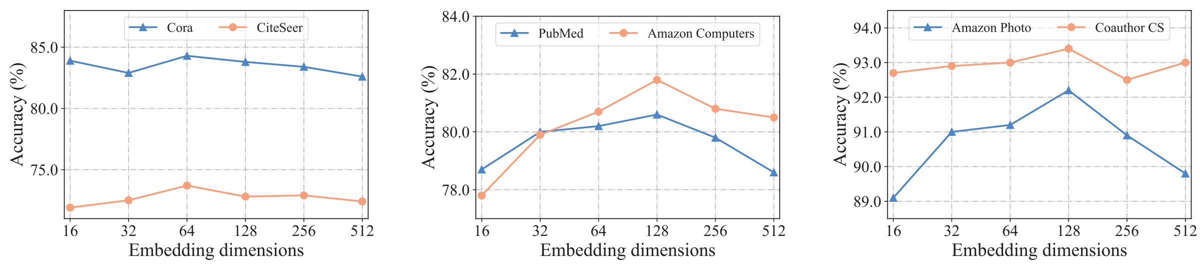
We further investigate the impact of the pseudo-targets threshold parameter τ, which is varied across values of $\{0.75, 0.8, 0.9, 0.95, 0.99\}$ to assess its influence on the model's performance. The experimental results are demonstrated in Fig. 3. From the results, we observe an initial improvement in performance as the τ value increased, followed by a subsequent decline when the threshold grew too large. This behavior can be attributed to that as τ increases, it imposes a stricter criterion for the inclusion of pseudo-labels during the training process, and the pseudo-labels are more reliable for the robust training of the model. However, when the threshold is raised over large, a considerable portion of the training data falls short of meeting the rigorous confidence criteria for pseudo-labels, leading to a reduction in the available pool of training data. In light of these observations, we identify optimal τ values that strike a balance between leveraging sufficiently reliable training samples and avoiding the incorporation of potentially mislabeled or noisy data. Specifically, our experiments indicate that τ values of 0.8 were optimal for datasets such as Cora and PubMed, while a value of 0.85 yielded the best results for CiteSeer. For the remaining three datasets, a τ value of 0.9 is proved to be most effective.

Finally, we explore the impact of varying embedding dimensions within the hidden layer, considering a range of values in $\{16, 32, 64, 128, 256, 512\}$, while keeping other settings constant. The results are depicted in Fig. 4, which reveal that as the embedding dimension increases initially, we observe a corresponding improvement in performance across all datasets. This outcome can be attributed to the fact that a larger embedding dimension allows the model to capture more intricate features, thereby enhancing the quality of representations. However, beyond a certain point, increasing the embedding dimension ceases to yield substantial benefits, and the performance levels off. This behavior suggests that there is an optimal range for the embedding dimension, where it strikes a balance between capturing complex features and preventing overfitting.

E. Ablation Study

In this experimental section, we embark on an in-depth analysis of the core components that constitute our proposed HOLA. We systematically evaluate the impact of five model variants by comparing them with the full model, with each variant involving the removal of a specific aspect of our framework while keeping the other components intact:

- HOLA w/o aug: We exclude the augmentation strategies applied to the input of the student network.



(a) Cora and CiteSeer (b) Pubmed and Amazon Computers (c) Amazon Photo and Coauthor CS

Fig. 4. Sensitivity analysis w.r.t. different settings of embedding dimensions on all six datasets.

TABLE V
PERFORMANCE COMPARISON WITH VARIANTS IN ABLATION STUDY (IN %)

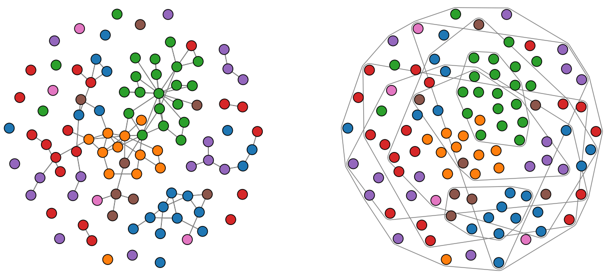
Methods	Cora	CiteSeer	PubMed	Amazon Computers	Amazon Photo	Coauthor CS
HOLA w/o aug	83.2±0.6	72.0±0.7	80.0±0.6	80.8±0.5	91.6±0.5	92.7±0.7
HOLA w both_aug	83.0±0.6	72.6±0.7	80.1±0.7	81.1±0.7	92.0±0.7	93.0±0.5
HOLA w reverse_aug	82.8±0.5	72.3±0.6	79.7±0.8	80.7±0.6	91.5±0.5	92.4±0.7
HOLA w/o tscl	82.1±0.7	71.0±0.8	79.8±0.8	80.1±0.7	91.1±0.9	92.7±0.6
HOLA w/o rcl	81.7±0.8	70.3±0.9	78.9±0.8	77.6±0.9	90.6±0.8	91.9±0.7
HOLA w/o mom	83.2±0.7	70.6±0.8	79.6±0.8	81.1±0.7	91.2±0.7	92.4±0.6
HOLA (Ours)	**84.2±0.5**	**73.9±0.6**	**80.6±0.4**	**81.8±0.6**	**92.2±0.7**	**93.4±0.4**

- HOLA w both_aug: We deploy augmentation strategies to both the input of the student and teacher networks.
- HOLA w reverse_aug: We implement a reverse augmentation operation for the two networks, using the original graph for the student network and applying the augmentation strategies to the teacher network.
- HOLA w/o tscl: We eliminate the teacher-student learning mechanism, relying solely on hypergraph consistency learning to enhance dual branch learning.
- HOLA w/o hcl: We discard hypergraph consistency learning, relying exclusively on teacher-student consistency learning to harness the information from unlabeled data.
- HOLA w/o mom: We replace the momentum update of the teacher network with supervised loss.

The ablation study results, presented in Table V, provide valuable insights into the individual contributions of the core components within our HOLA framework. Firstly, when we examine the performance of HOLA w/o aug, we observe a noticeable decline in its performance. This outcome underscores the significance of our data augmentation strategies, which not only enhance the robustness of our method but also play a crucial role in maintaining its overall effectiveness. Additionally, using the same augmentation for both networks has a negative impact on the performance, as it reduces the diversity of the data. When we apply the reverse augmentation strategy to obtain HOLA w reverse_aug, we observed a performance drop in testing compared to HOLA w both_aug. A possible reason is that the student network, trained on the original graph, is unable to transfer parameters with semantic perturbation invariance to the teacher network through momentum updates. This causes the embeddings generated by the teacher network, using the augmented graph, to potentially contain noise, leading to less accurate pseudo-targets and hypergraph embeddings, resulting in sub-optimal performance. Secondly, a comparison between HOLA and HOLA w/o tscl reveals that our full model outperforms the variant lacking the teacher-student consistency learning component. This result validates the importance of the teacher-student learning mechanism in our framework. By leveraging the reliable pseudo-labeling mechanism, our model benefits from the knowledge transfer between the teacher and student networks, leading to improved performance. Thirdly, the removal of the hypergraph consistency learning module results in the most noticeable performance decline. This decline highlights the role of hypergraph consistency learning in our framework, which captures the complex higher-order dependencies between various sub-structures and enhances the model's effectiveness. Moreover, we observe a performance drop when replacing the momentum update with supervised learning for the teacher network. This change appears to make the teacher network less consistent, resulting in unreliable guidance for the student. Finally, while the model variants excluding specific components can still perform reasonably well due to the effectiveness of the remaining components, it is essential to note that they consistently exhibit a decline in performance when compared to the full model. This consistent performance drop in the variants reaffirms the effectiveness of each component within our framework.

F. Visualization Analysis

We carry out a case study using the Cora dataset to illustrate the hypergraph structure that was discerned by the HOLA,



(a) Original Graph (b) Learned Hypergraph

Fig. 5. Visualization of the original graph and hypergraph structure learned by HOLA (only demonstrating the subgraph of the Cora dataset and 8 hyperedges for simplicity).

thus demonstrating the effectiveness of the hypergraph structure learning module. Within the Cora dataset, each node is symbolic of scientific papers, which are sorted into one of seven distinct categories, with the edges indicating the citations between them. To facilitate a more clear demonstration, we select a subgraph of the entire citation network, focusing on only 8 hyperedges.

Fig. 5(a) shows that each paper in the citation network links to merely a small number of neighboring papers, posing a challenge in modeling complex interactions. Additionally, the sparsity of the network, where many nodes are not interconnected, hampers the flow of information between them. In Fig. 5(b), we present a portion of the hyperedges derived from our hypergraph structure learning module. As can be seen from the figure, many nodes that are initially unconnected in the original graph, are now engaged in information propagation within the hypergraph. The hypergraph structure enables nodes within the network to engage in higher-order interactions, effectively capturing more complex and intricate relationships within the complete network. The outcomes indicate that our module is exceptionally skilled at discerning complex node relationships beyond pairwise interactions, thereby offering substantial flexibility in modeling complex data structures.

V. Conclusion

In this paper, we propose a simple yet effective model HOLA for semi-supervised node classification on the graph. Our HOLA possesses a collaborative distillation framework where the teacher network produces confident pseudo-targets to guide the learning of the student network and the teacher network is momentum updated from the knowledge distilled by the student network. Further, a novel relational consistency learning with hypergraph structure learning is developed to model complex high-order correlations among nodes, transferring the knowledge to the student network. Comprehensive experimental evaluations across six benchmark datasets substantiate the efficacy of our HOLA. For future research endeavors, we aim to delve deeper into the intrinsic exploration of higher-order semantics within graphs, gaining a fundamental understanding of the operational mechanisms of graphs. We expect to adapt our technology to more intricate scenarios, such as few-shot and zero-shot learning. Additionally, we plan to enhance the generalization capabilities of graph-based models by incorporating promising large language models.

References

[1] Z. Wu et al., "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[2] W. Ju et al., "A comprehensive survey on deep graph representation learning," *Neural Netw.*, vol. 173, 2024, Art. no. 106207.

[3] X. Zhong, C. Gu, M. Ye, W. Huang, and C.-W. Lin, "Graph complemented latent representation for few-shot image classification," *IEEE Trans. Multimedia*, vol. 25, pp. 1979–1990, 2023.

[4] Y. Feng, J. Gao, and C. Xu, "Learning dual-routing capsule graph neural network for few-shot video classification," *IEEE Trans. Multimedia*, vol. 25, pp. 3204–3216, 2023.

[5] K. Liu et al., "Multimodal graph contrastive learning for multimedia-based recommendation," *IEEE Trans. Multimedia*, vol. 25, pp. 9343–9355, 2023.

[6] W. Ju et al., "Kernel-based substructure exploration for next POI recommendation," in *Proc. 2022 IEEE Int. Conf. Data Mining*, 2022, pp. 221–230.

[7] Y. Qin, W. Ju, H. Wu, X. Luo, and M. Zhang, "Learning graph ODE for continuous-time sequential recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3224–3236, Jul. 2024.

[8] Z. Ma et al., "Improved deep unsupervised hashing via prototypical learning," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 659–667.

[9] H. Zhang, Y. Li, and X. Li, "Constrained bipartite graph learning for imbalanced multi-modal retrieval," *IEEE Trans. Multimedia*, vol. 26, pp. 4502–4514, 2024.

[10] J. Guo et al., "HGAN: Hierarchical graph alignment network for image-text retrieval," *IEEE Trans. Multimedia*, vol. 25, pp. 9189–9202, 2023.

[11] H. Zhang, C. Yi, B. Zhu, H. Ren, and Q. Li, "Multimodal topic modeling by exploring characteristics of short text social media," *IEEE Trans. Multimedia*, vol. 25, pp. 2430–2445, 2023.

[12] Y. Gu et al., "DEER: Distribution divergence-based graph contrast for partial label learning on graphs," *IEEE Trans. Multimedia*, early access, May 31, 2024, doi: 10.1109/TMM.2024.3408038.

[13] W. Ju et al., "Few-shot molecular property prediction via hierarchically structured learning on relation graphs," *Neural Netw.*, vol. 163, pp. 122–131, 2023.

[14] J. Yang et al., "Poisoning scientific knowledge using large language models," *Nature Mach. Intell.*, Nature Publishing Group, London, U.K., pp. 1–13, 2024.

[15] H. Li et al., "A survey on graph neural networks in intelligent transportation systems," 2024, *arXiv:2401.00713*.

[16] Y. Zhao et al., "Dynamic hypergraph structure learning for traffic flow forecasting," in *Proc. IEEE 39th Int. Conf. Data Eng.*, 2023, pp. 2303–2316.

[17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[18] X. Luo et al., "Toward effective semi-supervised node classification with hybrid curriculum pseudo-labeling," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 20, no. 3, pp. 1–19, 2023.

[19] X. Wei et al., "CLNode: Curriculum learning for node classification," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, 2023, pp. 670–678.

[20] J. Yuan et al., "Alex: Towards effective graph transfer learning with noisy labels," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 3647–3656.

[21] W. Ju et al., "TGNN: A joint semi-supervised framework for graph-level classification," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 2122–2128.

[22] Z. Mao, W. Ju, Y. Qin, X. Luo, and M. Zhang, "Rahnet: Retrieval augmented hybrid network for long-tailed graph classification," in *Proc. 31st ACM Int. Conf. Multimedia*, 2023, pp. 3817–3826.

[23] X. Luo, Y. Zhao, Y. Qin, W. Ju, and M. Zhang, "Towards semi-supervised universal graph classification," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 416–428, Jan. 2024.

[24] W. Ju et al., "Hypergraph-enhanced dual semi-supervised graph classification," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 22594–22604.

[25] W. Xia, Q. Wang, Q. Gao, M. Yang, and X. Gao, "Self-consistent contrastive attributed graph clustering with pseudo-label prompt," *IEEE Trans. Multimedia*, vol. 25, pp. 6665–6677, 2023.

[26] W. Ju et al., "GLCC: A general framework for graph-level clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4391–4399.

[27] S. Yi et al., "Redundancy-free self-supervised relational learning for graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 12, pp. 18313–18327, Dec. 2024.

[28] P. Velickovic et al., “Deep graph infomax,” in *Proc. Int. Conf. Learn. Representations*, 2019.

[29] K. Hassani and A. H. Khasahmadi, “Contrastive multi-view representation learning on graphs,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.

[30] Y. Zhu et al., “Deep graph contrastive representation learning,” 2020, *arXiv:2006.04131*.

[31] S. Wan, S. Pan, J. Yang, and C. Gong, “Contrastive and generative graph convolutional networks for graph-based semi-supervised learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10049–10057.

[32] W. Ju et al., “A survey of graph neural networks in real world: Imbalance, noise, privacy and OOD challenges,” 2024, *arXiv:2403.04468*.

[33] Y. Wang et al., “Disencite: Graph-based disentangled representation learning for context-specific citation generation,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11449–11458.

[34] J. Luo et al., “GALA: Graph diffusion-based alignment with jigsaw for source-free domain adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9038–9051, Dec. 2024.

[35] J. Luo et al., “Rank and align: Towards effective source-free graph domain adaptation,” in *Proc. 33rd Int. Joint Conf. Artif. Intell.*, 2024, pp. 4706–4714.

[36] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Representations*, 2017.

[37] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[38] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.

[39] F. Wu et al., “Simplifying graph convolutional networks,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.

[40] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning,” *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 542–542, Feb. 2009.

[41] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: A regularization method for supervised and semi-supervised learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[42] D. Berthelot et al., “Mixmatch: A holistic approach to semi-supervised learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5049–5059.

[43] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, *arXiv:1503.02531*.

[44] W. Park, D. Kim, Y. Lu, and M. Cho, “Relational knowledge distillation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3967–3976.

[45] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, “Hypergraph neural networks,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3558–3565.

[46] Y. Gao et al., “Hypergraph learning: Methods and practices,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.

[47] X. Su, J. Yang, J. Wu, and Z. Qiu, “Hy-defake: Hypergraph neural networks for detecting fake news in online social networks,” 2023, *arXiv:2309.02692*.

[48] Z. Lin et al., “Automatic hypergraph generation for enhancing recommendation with sparse optimization,” *IEEE Trans. Multimedia*, vol. 26, pp. 5680–5693, 2024.

[49] Z. Zhang, Y. Feng, S. Ying, and Y. Gao, “Deep hypergraph structure learning,” 2022, *arXiv:2208.12547*.

[50] D. Cai et al., “Hypergraph structure learning for hypergraph neural networks,” in *Proc. 31st Int. Joint Conf. Artif. Intell.*, 2022, pp. 1923–1929.

[51] C. Yang, R. Wang, S. Yao, and T. Abdelzaher, “Semi-supervised hypergraph node classification on hypergraph line expansion,” in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 2352–2361.

[52] M. Choe, S. Kim, J. Yoo, and K. Shin, “Classification of edge-dependent labels of nodes in hypergraphs,” in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2023, pp. 298–309.

[53] Y. Song et al., “CHGNN: A semi-supervised contrastive hypergraph learning network,” *IEEE Trans. Knowl. Data Eng.*, 2024.

[54] S. Bai, F. Zhang, and P. H. Torr, “Hypergraph convolution and hypergraph attention,” *Pattern Recognit.*, vol. 110, 2021, Art. no. 107637.

[55] D.-H. Lee et al., “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Proc. Workshop Challenges Representation Learn.*, 2013, Art. no. 896.

[56] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[57] W. Ju et al., “Towards graph contrastive learning: A survey and beyond,” 2024, *arXiv:2405.11868*.

[58] J. Yu, D. Tao, and M. Wang, “Adaptive hypergraph learning and its application in image classification,” *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3262–3272, Jul. 2012.

[59] M. Wang, X. Liu, and X. Wu, “Visual classification by ℓ_1-hypergraph modeling,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2564–2574, Sep. 2015.

[60] S. Huang, M. Elhoseiny, A. Elgammal, and D. Yang, “Learning hypergraph-regularized attribute predictors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 409–417.

[61] P. Sen et al., “Collective classification in network data,” *AI Mag.*, vol. 29, no. 3, pp. 93–93, 2008.

[62] A. Bojchevski and S. Günnemann, “Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[63] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, “Pitfalls of graph neural network evaluation,” 2018, *arXiv:1811.05868*.

[64] P. Veličković et al., “Graph attention networks,” in *Proc. Int. Conf. Learn. Representations*, 2018.

[65] X. Wang et al., “Am-GCN: Adaptive multi-channel graph convolutional networks,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1243–1253.

[66] D. Kim and A. Oh, “How to find your friendly neighborhood: Graph attention design with self-supervision,” in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=Wi5KUNlqWty

[67] C. Liu et al., “Gapformer: Graph transformer with graph pooling for node classification,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2023, pp. 2196–2205.

[68] Z. Zhang, J. Wang, and L. Zhao, “Curriculum learning for graph neural networks: Which edges should we learn first,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 51113–51132.

[69] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

Siyu Yi (Member, IEEE) received the B.S. and M.S. degrees in statistics from Sichuan University, Sichuan, China, in 2017 and 2020, respectively, and the Ph.D. degree in statistics from Nankai University, Tianjin, China, in 2024. She is currently a Postdoctoral Researcher in Mathematics with Sichuan University, Chengdu, China. She has authored or coauthored more than 20 papers. Her research interests include graph machine learning, statistical learning, and subsampling in Big Data.

Zhengyang Mao is currently working toward the master’s degree with the School of Computer Science, Peking University, Beijing, China. His research interests include graph representation learning and long-tailed learning.

Yifan Wang received the B.S. and M.S. degrees in software engineering from Northeastern University, Liaoning, China, in 2014 and 2017, respectively, and the Ph.D. degree in computer science from Peking University, Beijing, China, in 2023. He is currently an Assistant Professor with the School of Information Technology and Management, University of International Business and Economics, Beijing. His research interests include graph representation learning, graph neural networks, disentangled representation learning, and corresponding applications such as drug discovery and recommender systems.

Yiyang Gu received the B.S. degree in computer science in 2021 from Peking University, Beijing, China, where he is currently working toward the Ph.D. degree. His research interests include graph representation learning, knowledge graph and bioinformatics.

Zhiping Xiao received the Ph.D. degree in computer science from the University of California, Los Angeles, Los Angeles, CA, USA, in 2024. Her major is artificial intelligence, minor is data mining, did research in the area of multi-modality social-media data analysis, and her current research focuses on AI for pathology. She is also interested in other interdisciplinary applications.

Chong Chen received the B.S. degree in mathematics from Peking University, Beijing, China, in 2013, and the Ph.D. degree in statistics from Peking University, in 2019, under the supervision of Prof. Ruibin Xi. He is currently a Research Scientist with Terminus Group. His research interests include image understanding, self-supervised learning, and data mining.

Xian-Sheng Hua (Fellow, IEEE) received the B.S. and Ph.D. degrees in applied mathematics from Peking University, Beijing, China, in 1996 and 2001, respectively. In 2001, he joined Microsoft Research Asia, as a Researcher, and has been a Senior Researcher with Microsoft Research Redmond since 2013. He became a Researcher and the Senior Director of Alibaba Group, in 2015. He has authored or coauthored more than 250 research articles and has filed more than 90 patents. His research interests include multimedia search, advertising, understanding, and mining, pattern recognition, and machine learning. He was honored as one of the recipients of MIT35. He was a Program Co-Chair for the IEEE ICME 2013, ACM Multimedia 2012, and IEEE ICME 2012, and on the Technical Directions Board for the IEEE Signal Processing Society. He is an ACM Distinguished Scientist.

Ming Zhang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from Peking University, Beijing, China. She is currently a Full Professor with the School of Computer Science, Peking University. She has authored or coauthored more than 200 research papers on text mining and machine learning in the top journals and conferences. Prof. Zhang is a Member of Advisory Committee of Ministry of Education in China and the Chair of ACM SIGCSE China. She is one of the 15 members of ACM/IEEE CC2020 Steering Committee. She won the best paper of ICML 2014 and best paper nominee of WWW 2016. Prof. Zhang is the leading author of several textbooks on Data Structures and Algorithms in Chinese, and the corresponding course is awarded as the National Elaborate Course, National Boutique Resource Sharing Course, National Fine-designed Online Course, and National First-Class Undergraduate Course by MOE China.

Wei Ju (Member, IEEE) received the B.S. degree in mathematics from Sichuan University, Sichuan, China, in 2017, and the Ph.D. degree from the School of Computer Science, Peking University, Beijing, China, in 2022. He was a Postdoc Research Fellow with Peking University. He is currently an Associate Professor with the College of Computer Science, Sichuan University, Chengdu, China. His research interests include the area of machine learning on graphs including graph representation learning and graph neural networks, and interdisciplinary applications such as recommender systems, bioinformatics, drug discovery and knowledge graphs. He has authored or coauthored more than 50 papers in top-tier venues and has won the best paper finalist in IEEE ICDM 2022.