Graph Learning with Consistency and Diversity against Label Noise

1st Jie Zhang
College of Computer Science and Engineering
Shandong University of Science and Technology
Qingdao, China
zj_9996@163.com

***Abstract*—Graph Neural Networks (GNNs) have shown remarkable effectiveness in learning graph representations across diverse applications. However, in many real-world scenarios, graph-structured data often contains noisy labels, which substantially compromise the performance of GNNs. This degradation occurs because noisy labels weaken effective supervision during training, causing GNN-based models to memorize mislabeled samples and consequently impair their performance on downstream tasks. Existing approaches for handling label noise in images and text typically rely on large-scale labeled datasets and assume data independence, making them unsuitable for addressing the label sparsity and structural dependencies inherent in graphs. To overcome these challenges, we propose a multi-network learning paradigm in which models supervise each other, thereby reducing the bias accumulated when learning from noisy samples. We introduce a novel multi-network framework, Joint Training with Consistency and Diversity (JoCaD), which is designed to maximize prediction consistency across networks while maintaining sufficient diversity in their representation learning. Specifically, we incorporate graph augmentation through link addition to enrich the aggregated information, and employ a contrastie loss to ensure alignment between the node representations learned by both networks. Extensive experiments on four real-world datasets demonstrate the effectiveness of the proposed JoCaD in addressing the node classification task under varying levels of label noise.**

***Index Terms*—Graph Neural Networks, Noisy Label, Multi-network Method, Robust Learning**

I. Introduction

A vast amount of real-world data can be naturally represented as graphs, spanning diverse application domains such as social networks [1], recommender systems [2], and traffic flow prediction [3]. Consequently, the processing and analysis of graph-structured data have become a critical research direction in machine learning, with graph representation learning providing an effective paradigm for tackling such challenges. In this context, Graph Neural Networks (GNNs) have emerged as a powerful class of models, demonstrating remarkable success in modeling and analyzing graph data. Among various downstream tasks, node classification has become a fundamental benchmark, as it requires models to jointly capture discriminative feature information and rich topological dependencies. By leveraging these mechanisms, GNNs have achieved successful performance in node classification tasks.

Despite their success, the performance of GNNs in node classification tasks heavily depends on the availability of accurately labeled training samples [4]. Reliable labels provide strong supervision, enabling GNNs to effectively capture both explicit node attributes and latent structural information within the graph. However, in real-world applications, acquiring high-quality labels is often costly and time-consuming, and the annotation process is inherently susceptible to human errors and inconsistencies. Consequently, label noise remains a critical challenge that hinders the practical deployment of GNNs [4]. Prior studies have shown that GNNs tend to memorize noisy samples, leading to overfitting on incorrectly labeled data and impairing their generalization ability [5]. Moreover, due to the message-passing mechanism, noise from mislabeled nodes can propagate to their neighbors, thereby contaminating the representation learning of other nodes.

In recent years, numerous approaches have been proposed to address the challenge of label noise, including multi-network method [6], loss correction [4], and label correction [7], enhancing the robustness of models from different perspectives. To address the adverse effects of neural networks memorizing noisy samples in image classification, researchers have proposed multi-network approaches. These methods enable networks to supervise each other, ensuring that the error flow resulting from biased sample selection is not accumulated within a single network. Existing methods for handling label noise in image classification typically assume that samples are independently and identically distributed (i.i.d.) and that sufficient labels are available. In contrast, nodes in graphs are inherently connected and often suffer from label sparsity. To mitigate this issue, several studies have explored incorporating additional information to reduce the impact of label noise. A representative example is NRGNN, which enhances supervision for graph learning by generating pseudo-links and pseudo-labels to regularize the loss function [4]. Label correction methods, in contrast, directly measure the reliability of training samples through specific metrics and modify the labels of suspected noisy samples. Inspired by multi-network approaches in image classification and considering the tendency of GNNs to memorize noisy samples, we propose a multi-network method specifically designed for graph data. Furthermore, current multi-network methods often fail to properly balance consistency (i.e., agreement among models or samples) and diversity (i.e., maintaining complementary learning signals), both of which are crucial for achieving robust learning under

noisy labels [8].

To address the aforementioned challenges, we propose a novel joint training framework, Joint Training with Consistency and Diversity (JoCaD). The key idea of JoCaD is to maximize the consistency of node representations learned by multiple networks while simultaneously ensuring diversity in node representation learning. This is achieved by incorporating view learning with supplementary edge information, which enables the model to balance consistency and diversity and thus capture both rich node features and structural information. Moreover, JoCaD leverages the memory capacity of deep neural networks positively by exchanging samples between networks, thereby mitigating the influence of noisy labels on the learning of other nodes. Specifically, a pseudo-link miner is designed to add pseudo-links to the adjacency matrix of the original graph, facilitating graph enhancement and constructing an additional view enriched by edge information. To further strengthen robustness, JoCaD integrates three types of loss functions: a reconstruction loss, a contrastive loss, and a dynamic prediction loss. Together, these components encourage the model to learn more informative node representations while effectively reducing the negative impact of label noise during training. The main contributions of this paper are summarized as follows:

(1) We propose a new co-training framework, JoCaD, which jointly enforces both consistency and diversity in node representations. This design relieves the tendency of models to overfit mislabeled samples and improves the robustness of graph representation learning under noisy labels.

(2) To enhance the robustness of our model, we design graph augmentation and the contrastive loss to guarantee the quality of node representations while adaptively adjusting the weights of different samples within the joint training paradigm. These strategies effectively strengthen the model's robustness against label noise.

(3) We provide an intuitive analysis to demonstrate the rationality and feasibility of our approach. Extensive experiments on multiple benchmark datasets further validate the effectiveness of JoCaD in graph node classification and confirm its superiority in robust graph representation learning under noisy labels.

II. Related works

Graph neural networks have been widely applied in various domains. Graph-based data naturally models entities as nodes with features and edges capturing complex dependencies. Leveraging both structure and attributes, Graph Neural Networks (GNNs) have emerged as a powerful framework for representation learning across diverse tasks. Several effective strategies have been proposed to address label noise in graph representation learning, including loss correction [4], [9], multi-network methods [10], and label correction [7]. Current methods often overlook the combined effects of neural network memorization for noisy samples and the topological information of graphs. Motivated by this limitation, we propose a novel approach that integrates a multi-network framework with graph enhancement, explicitly designed to improve robustness against label noise while fully leveraging structural information.

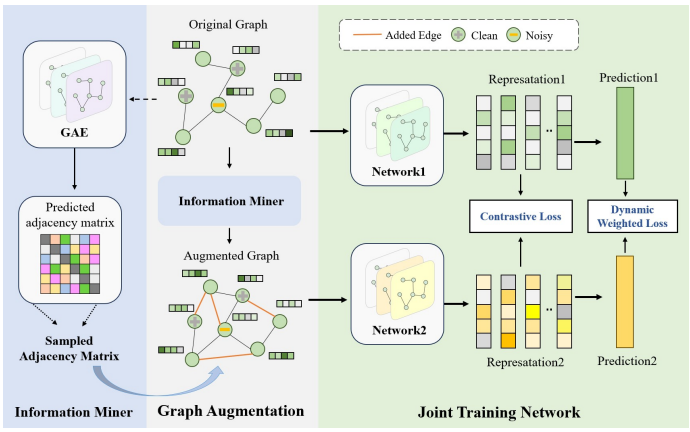

Fig. 1. The overall framework of our method.

III. Preliminaries

An undirected graph can be formally defined as $\mathcal{G} = (\mathcal{V}, E)$, where $\mathcal{V}$ denotes the set of nodes , and $\mathcal{E}$ represents the set of edges. A node is denoted by $v_i \in \mathcal{V}$, and $e_{ij} = (v_i, v_j) \in \mathcal{E}$ to denote an edge pointing from v_j to v_i. The neighborhood of a node v is defined as $\mathcal{N}(v) = \{u \in \mathcal{V} | (v, u) \in \mathcal{E}\}$. The graph structure can be equivalently represented by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $A_{ij} = 1$ if $e_{ij} \in \mathcal{E}$, and $A_{ij} = 0$ otherwise. Each node is further associated with an attribute vector, collected in the feature matrix $\mathbf{X} \in \mathbf{R}^{n \times d}$, where $\mathbf{x}_v \in \mathbf{R}^d$ corresponds to the feature representation of node v. Let $\mathcal{V}_L = \{v_1, ..., v_l\}$ denote the set of labeled nodes, and $\mathcal{V}_U = \mathcal{V} - \mathcal{V}_L$ is the set of unlabeled nodes. In the presence of label noise, the noisy labeled set is denoted as $\mathcal{V}_N = \{v_1^N, ..., v_l^N\}$. Given the feature matrix $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, and a small set of labels Y_L, the objective is to learn a robust model that can accurately predict the labels y_i for nodes $v_i \in \mathcal{V}_U$.

IV. Methods

The overall framework of our model is illustrated in Fig.1. The core components of JoCaD are summarized in the following two points: (1) graph augmentation, (2) joint training.

A. Graph Augmentation

Due to the existence of graph topology, digging deeper into graph latent structure information works well in various graph learning tasks [11]. Inspired by this, we utilize latent graph structures to perform graph augmentation, thereby enhancing supervised propagation and alleviating the challenge of label scarcity. In many real-world networks, connected nodes often share the same label or exhibit similar functionality [12]. Link prediction algorithms leverage attribute graphs to infer missing links, thereby providing directional guidance for node connections. Our motivation is to identify which

edges and nodes in the training graph data optimally preserve the graph's semantic information and to implement edge-level and node-level data augmentation tailored to distinct graph attributes. By conducting data augmentation on edges and nodes separately, JoCaD can more effectively capture semantic information—thereby enhancing the robustness and generalization of graph contrastive learning while suppressing the representation of irrelevant information.

In real-world graph data, the original topological structure of the graph may be incomplete because graph data is always extracted from complex interaction networks. This requires the view generator to remove incorrect edges as accurately as possible and add missing edges based on the original graph data. In previous works, operations that randomly add/remove information may not effectively adaptively enhance the graph data to retain the original semantic information. To solve the above problems, we designed a learnable edge-level information miner. This miner performs edge-level enhancements. By adaptively interpolating the predicted matrix with the original adjacency matrix, effective graph contrastive learning samples are generated.

Applying the node feature matrix for edge prediction can help us capture potential connectivity, thereby optimizing the original adjacency matrix. Using the graph autoencoder GAE based on the variational autoencoder as the input, it generates an edge probability matrix N, where N_{ij} represents the probability that there is an edge between nodes v_i and v_j. The A is a binary adjacency matrix. It can effectively learn the latent representation of the graph and can be used to reconstruct the adjacency matrix of the graph. The predicted connection information can be integrated into the original adjacency matrix through edge interpolation, thereby enhancing the edges. We use an interpolation coefficient α to perform linear interpolation between the predicted edge probability matrix N and the original adjacency matrix A. This process generates a new adjacency matrix E, which balances the predicted and original topological information. α is not a hyperparameter but is initialized as a parameter with a specific value. During the training stage, α will be updated according to the Adam optimizer

$$\mathbf{E}_{ij} = \alpha \mathbf{N}_{ij} + (1-\alpha)\mathbf{A}_{ij}. \tag{1}$$

By sampling the interpolation adjacency matrix, we can update and optimize the enhanced adjacency matrix. To prevent the interpolation from affecting the performance of the model, we first apply Bernoulli sampling to sample a binary adjacency matrix $\mathbf{A}'$ from the adjacency matrix E as the sampled topological structure. To make the sampling process differentiable, we use Gumbel-Softmax as an approximation for Bernoulli sampling and utilize the Direct Propagation (ST) gradient estimator so that the sampling results can be 0 or 1 during forward propagation, and the gradients can be directly propagated during backward propagation. The sampling process is shown in the following formula:

$$\mathbf{A}' = \frac{\tau \mathbf{E}_{ij}}{\tau \mathbf{E}_{ij} + e^{G}} + \frac{\tau \mathbf{E}_{ij}}{2} \tag{2}$$

Among them, the matrix $\mathbf{A}'$ represents the sampled adjacency matrix, while τ represents the temperature of the Gumbel-Softmax distribution. Additionally, $G \sim \text{Gumbel}(0,1)$ indicates a Gumbel random variable. Secondly, by combining $\mathbf{A}'$ with the original node features, we can obtain an enhanced graph. We can optimize the parameters of the view generator through backpropagation, laying the foundation for the end-to-end characteristics. Moreover, this differentiability enables the interpolation parameter α to adapt to various graphs.

B. Joint Training

In JoCaD, each network is capable of independently predicting labels. However, during training, the two networks are optimized under a pseudo-Siamese paradigm, where their parameters remain distinct but are updated simultaneously through a joint loss function [10]. For Network 1, the input consists of the node features $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$ of the original graph, from which the node representations $\mathbf{Z}^1$ are derived. For Network 2, the input includes the node features $\mathbf{X}$ and the adjacency matrix $\hat{\mathbf{A}}$ of the graph with the newly added links, yielding the node representations $\mathbf{Z}^2$. A softmax activation function is applied to obtain the prediction probabilities $\mathbf{P}^1$, $\mathbf{P}^2$ for node classification in both networks, defined as

$$\mathbf{P}^i = softmax(\mathbf{Z}^i). \tag{1}$$

Intuitively, different classifiers yield distinct decision boundaries, which in turn reflect their varying learning capacities. Therefore, when trained on noisy labels, they are expected to exhibit different abilities in filtering out label noise. Given that the two networks possess different learning capacities due to their distinct initial conditions, an instance with a small loss is more likely to have a clean label compared to instances selected by a single model. This process is analogous to co-training, as even if the selected instances are not entirely clean, both networks can adaptively correct training errors by leveraging mutual guidance. Owing to the inherent sparsity of graph labels, the identified small-loss samples are assigned greater importance in the prediction loss function. we adopt the cross-entropy loss as the supervised objective to minimize the discrepancy between predictions and ground-truth labels.

$$\mathcal{L} = -\frac{1}{N}\sum_{i} y_i \log\left(p_i\right) \tag{2}$$

where $\hat{y}^p_{j,i}$ is the prediction of node v_i from f_θ under two networks, and $I^p_i = 1$ if $\hat{y}^p_{1,i} = \hat{y}^p_{2,i}$; otherwise $I^p_i = 0$. It can reduce the dependence of model training on label information in the presence of label noise. The prediction results of training nodes across different views are pushed close to each other. The nodes across these views offer valuable assistance in maintaining label consistency.

We introduce the Kullback–Leibler (KL) divergence as a regularization term to constrain the information enhancement, thereby ensuring that the enriched information does not compromise accuracy. Accordingly, the consistency loss is defined as

$$\mathcal{L}_{\text{con}} = D_{KL}\left(\mathbf{P}^1 \| \mathbf{P}^2\right) + D_{KL}\left(\mathbf{P}^2 \| \mathbf{P}^1\right) \tag{6}$$

where

$$D_{KL}\left(\mathbf{P}^1 \| \mathbf{P}^2\right) = \sum_{i=1}^{N} \sum_{m=1}^{M} p_1^m\left(\mathbf{x}_i\right) \log \frac{p_2^m\left(\mathbf{x}_i\right)}{p_1^m\left(\mathbf{x}_i\right)} \tag{7}$$

$$D_{KL}\left(\mathbf{P}^2 \| \mathbf{P}^1\right) = \sum_{i=1}^{N} \sum_{m=1}^{M} p_2^m\left(\mathbf{x}_i\right) \log \frac{p_1^m\left(\mathbf{x}_i\right)}{p_2^m\left(\mathbf{x}_i\right)} \tag{8}$$

The final overall loss of the proposed model can be expressed as follows:

$$\mathcal{L}_{obj} = r_1 \mathcal{L}_{\text{con}} + r_2 \mathcal{L}_{\text{dyn}}, \tag{9}$$

where r_1 and r_2 denote hyperparameters that balance the contributions of the consistency loss and the dynamic prediction loss, respectively.

V. Experiments

Datasets. To assess the effectiveness of the proposed approach in handling label noise, we conduct experiments on four benchmark datasets: three citation networks (Cora, CiteSeer, and PubMed) [13] and one co-purchase network (Amazon-Photo) [14].

Noise Types. In this paper, we introduce label noise by manually corrupting the datasets with two commonly used noise types for node classification: uniform noise and pair noise [4]. Since the original datasets are noise-free, we apply corruption using a noise transition matrix. Under uniform noise, the true label is flipped to a randomly selected incorrect class with probability $\epsilon \in (0, 1)$. Under pair noise, the true label is flipped to its designated paired class with probability ϵ. The corresponding noise transition matrix is defined as follows, where n denotes the number of classes.

Baselines. The baseline methods considered in this study consist of three categories. (1) GCN [15] employs graph convolutional operations to aggregate information from neighboring nodes and iteratively propagate it throughout the graph. (2) Forward [19] adopts a loss correction strategy that incorporates the noise transition matrix into the training objective. (3) Co-teaching [6] trains two networks simultaneously, where each network selects small-loss samples identified by its peer to update its own parameters. (4) JoCoR [10] trains two networks in parallel to jointly predict node labels. (5) DGNN [16] incorporates a noise estimator to approximate the noise transition matrix and leverages it to compute an auxiliary loss that guides model training. (6) NRGNN [4] is a loss-correction method that addresses node label sparsity and edge sparsity. (7) RTGNN [17] improves robustness to label noise by employing two GNN models to perform labeled node partitioning and

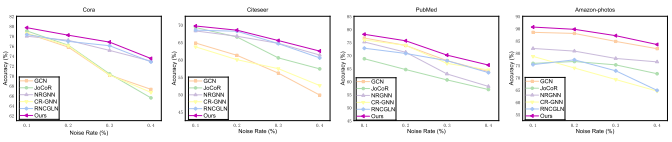

Fig. 2. Results with different label rates under uniform noise.

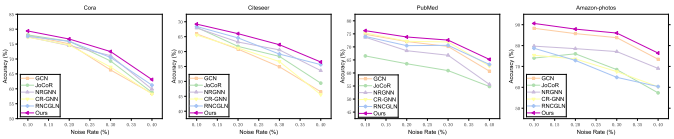

Fig. 3. Results with different label rates under pair noise.

by incorporating a consistency loss. (8) CRGNN [9] integrates contrastive learning into a loss-correction framework to enhance robustness against label noise. (9) RNCGLN [18] enhances graph representation learning by replacing traditional graph convolutional networks (GCNs) with an efficient self-attention mechanism to aggregate global structural information.

Comparison with Baselines. The effectiveness of the proposed method is assessed by comparing it with nine baseline methods. We report the node classification accuracy of all methods across four datasets under various noise types and label noise ratios. Table I presents the accuracy and its standard deviation for all methods on four benchmark datasets under label noise ratios of 20%. The best-performing result for each dataset is highlighted in bold, and the second-best is underlined.

The accuracy of our method and baselines—GCN, JoCoR, NRGNN, CR-GNN, and RNCGLN—under uniform noise and pair noise is illustrated in Fig.2 and Fig.3.

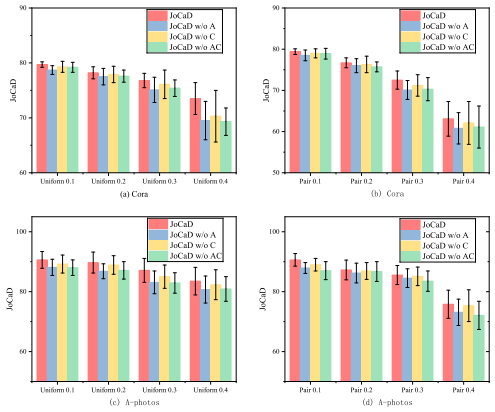

Fig. 4. Comparisons between RNRGCL and its variants on Cora and Amazon-Photo.

Ablation Experiments. To investigate the contributions of graph augmentation and consistency regularization, we conduct ablation experiments with three variants of JoCaD:

TABLE I
NODE CLASSIFICATION PERFORMANCE UNDER VARIOUS TYPES OF NOISE (ACCURACY (%) ± STANDARD DEVIATION).

Datasets		Uniform				Pair			
		0.1	0.2	0.3	0.4	0.1	0.2	0.3	0.4
Cora	GCN	78.56±0.88	75.80±1.09	70.18±1.57	67.30±1.25	78.00±0.76	75.08±1.05	66.32±3.09	58.48±1.79
	Forward	78.72±0.91	75.70±1.44	71.12±2.65	67.60±2.16	78.54±0.96	75.88±1.45	67.84±2.99	59.80±4.75
	Co-teaching	72.46±1.06	67.02±1.99	61.08±1.64	57.58±2.58	72.72±2.28	68.74±2.53	61.40±3.98	53.10±3.76
	JoCoR	79.05±0. 69	76.18±1.00	70.40±1.66	65.58±2.67	78.08±1.29	75.94±1.18	69.34±2.78	58.78±5.33
	DGNN	69.26±0.99	64.46±1.96	57.26±1.64	49.82±1.38	68.30±1.42	67.38±1.87	61.76±2.67	52.14±9.24
	NRGNN	77.98±0.42	77.17±0.84	75.14±0.81	72.97±1.63	77.42±1.11	74.50±0.97	71.12±2.55	59.87±3.06
	RTGNN	73.81±1.33	71.25±1.06	69.67±2.16	68.01±1.24	72.05±1.28	68.32±1.81	66.83±2.82	55.87±4.87
	CRGNN	78.58±1.25	76.08±2.02	70.18±1.82	66.68±2.20	77.32±0.92	74.66±1.38	67.30±2.40	58.26±4.74
	RNCGLN	78.32±0.59	76.96±1.20	76.08±1.01	72.83±1.81	77.62±0.89	75.92±1.06	70.52±1.72	61.24±3.66
	JoCaD	**79.72±0.54**	**78.22±1.13**	**76.79±1.28**	**73.47±2.87**	**79.38±0.74**	**76.62±1.86**	**72.52±2.22**	**62.86±4.19**
Citeseer	GCN	64.80±0.90	61.28±0.95	56.16±3.68	49.86±4.03	65.92±1.99	60.80±3.59	54.88±6.00	46.52±5.25
	Forward	66.28±0.88	62.08±2.01	57.66±3.80	51.80±4.14	67.30±1.35	61.84±3.27	56.84±2.28	48.22±5.42
	Co-teaching	60.98±1.51	53.84±2.46	47.68±5.59	42.66±9.54	58.36±1.93	54.20±3.81	52.48±3.74	41.30±5.30
	JoCoR	69.28±1.19	66.54±1.49	60.54±3.74	57.38±5.63	68.12±2.05	61.73±4.39	58.38±4.34	49.40±7.72
	DGNN	58.18±1.31	52.58±2.39	48.72±4.23	45.52±5.52	60.48±2.29	55.98±2.88	53.14±2.80	40.78±5.20
	NRGNN	68.28±0.87	66.76±0.85	64.72±2.03	61.36±2.33	67.89±2.03	63.18±3.23	60.56±3.64	53.60±4.74
	RTGNN	65.36±1.16	61.93±2.75	58.39±3.57	55.44±7.52	57.85±2.76	53.17±5.70	52.31±5.61	41.36±7.48
	CRGNN	63.72±0.85	60.02±1.10	57.44±2.87	52.60±3.35	65.52±1.87	61.18±2.15	56.64±4.00	45.56±2.55
	RNCGLN	68.52±0.91	68.20±0.73	64.64±3.12	60.54±5.54	68.40±1.76	64.58±2.81	59.16 ± 4.20	55.62±6.47
	JoCaD	**69.72±0.71**	**68.86±1.07**	**65.52±2.54**	**62.53±3.80**	**69.22±1.82**	**66.03±2.88**	**62.34±3.67**	**56.52±4.66**
PubMed	GCN	76.66±2.04	73.80±2.82	67.98±7.43	63.82±6.97	74.82±0.26	72.04±3.01	70.16±4.21	60.70±4.52
	Forward	77.58±1.24	74.66±3.14	67.64±8.79	64.82±6.91	75.46±1.42	72.80±1.06	70.90±5.21	60.40±3.87
	Co-teaching	73.78±2.66	70.48±4.38	65.46±6.93	61.02±10.07	73.60±1.91	69.44±2.96	68.78±3.36	59.54±3.92
	JoCoR	68.71±4.45	64.57±5.65	60.61±5.43	56.93±5.57	66.53±4.32	63.54±5.05	60.92±6.57	54.82±5.14
	DGNN	76.58±1.54	74.52±2.67	69.19±3.27	64.82±5.73	72.45±3.37	69.23±3.97	64.35±4.67	58.76±4.37
	NRGNN	75.14±6.02	71.20±5.16	62.88±7.11	58.12±8.24	73.70±3.60	68.54±4.89	66.78±3.68	55.62±8.92
	RTGNN	72.46±4.57	72.56±2.72	67.31±6.16	65.34±6.53	74.53±2.29	72.12±2.64	71.83±4.56	60.93±5.33
	CRGNN	75.94±1.24	73.86±3.78	66.92±7.22	64.46±6.18	75.36±0.80	72.14±2.45	71.88±3.07	62.62±3.74
	RNCGLN	72.76±2.13	70.74±3.58	67.98±8.03	63.42±8.62	74.00±1.33	70.48±1.29	70.60±2.66	63.26±4.13
	JoCaD	**78.11±1.12**	**75.61±2.91**	**70.09±5.11**	**66.33±6.68**	**76.16±1.31**	**73.81±2.02**	**72.62±3.12**	**65.21±4.27**
Photo	GCN	88.47±1.45	88.01±1.99	84.75±3.21	81.80±4.05	88.29±1.29	85.74±2.86	83.8±3.84	73.44±3.74
	Forward	79.45±10.34	64.21±15.18	63.96±15.75	69.90±5.44	68.68±7.75	68.91±9.38	68.60±6.42	64.33±6.52
	Co-teaching	89.14±1.43	88.77±2.27	82.66±7.86	82.10±2.76	88.67±1.80	86.74±3.25	84.63±3.08	74.97±4.67
	JoCoR	75.82±7.95	76.59±9.10	75.22±8.24	71.60±8.55	74.03±7.45	76.04±8.12	68.48±6.62	57.33±8.35
	DGNN	76.22±7.13	62.00±5.65	54.28±8.37	49.35±4.40	70.58±5.88	61.95±3.77	54.09±7.97	50.22±5.86
	NRGNN	81.89±3.18	80.82±3.65	77.82±3.69	76.43±2.72	79.74±3.78	78.53±2.49	77.12±5.04	69.05±5.29
	RTGNN	82.32±1.68	83.00±2.41	82.08±6.02	81.99±1.17	82.15±1.53	82.31±2.61	77.53±5.36	75.32±6.02
	CRGNN	78.67±9.10	73.93±8.56	69.24±8.66	64.55±9.13	75.35±8.82	73.59±5.13	67.53±2.25	60.27±5.85
	RNCGLN	75.31±3.78	77.28±5.43	72.80±6.60	64.81±4.72	78.70±4.12	72.87±4.39	64.79±2.12	60.37±5.49
	JoCaD	**90.62±2.79**	**89.68±3.47**	**87.14±4.03**	**83.52±4.63**	**90.62±2.13**	**87.87±3.31**	**86.04±3.22**	**76.42±4.67**

JoCaD w/o A, JoCaD w/o C, and JoCaD w/o AC. Each variant is constructed by systematically removing specific components from the proposed framework. Specifically, to show the importance of the graph augmentation by adding links, we train the variant JoCaD w/o A, which does not use the graph augmentation to get node representations. JoCaD w/o C represents the variant in which the consistency regularization and dynamic prediction loss are replaced by cross-entropy loss. Furthermore, JoCaD w/o AC denotes the variant without graph augmentation, consistency regularization, and dynamic prediction loss. We report the classification accuracy of all variants on two benchmark datasets, Cora and Amazon-Photo, as shown in Fig.4.

VI. CONCLUSION

We propose a novel method for graph representation learning under label noise, in which multiple networks are jointly trained for the node classification task. To address edge sparsity, the graph is augmented through link generation, thereby enriching the diversity of node representations. The mutually supervised joint training strategy alleviates the tendency of neural networks to memorize mislabeled samples and mitigates bias accumulation during iterative training. We believe that the construction of real-world datasets with realistic noise distributions would enable more thorough and practical empirical evaluations, providing stronger guidance for the development of robust and noise-resilient methods.

REFERENCES

[1] L. Peng, Y. Mo, J. Xu, J. Shen, X. Shi and X. Li, et al., GRLC: Graph Representation Learning With Constraints. in IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 6, pp. 8609-8622, June 2024.

[2] X. Li, X. Wang and Y. Lin, Graph Neural Network Enhanced Sequential Recommendation Method for Cross-Platform Ad Campaigns. 2025 5th International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), Beijing, China, 2025, pp. 502-506.

[3] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li. DSTAGNN: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting. International Conference on Machine Learning, 2022.

[4] Dai, Enyan, C. Aggarwal, and S. Wang. NRGNN: Learning a Label Noise Resistant Graph Neural Network on Sparsely and Noisily Labeled Graphs. ACM, 2021.

[5] K. Kong, L. Lee, Y. Kwak, Y. R. Cho, S. E. Kim, and W. J. Song, Penalty based robust learning with noisy labels. Neurocomputing, 2020.

[6] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, and W. Hu, et al., Co-teaching: robust training of deep neural networks with extremely noisy labels. Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018.

[7] J. Yuan, X. Luo, Y. Qin, Y. Zhao, W. Ju and M. Zhang, Learning on Graphs under Label Noise. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, Rhodes Island, Greece, 2023, pp. 1-5.

[8] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularized approach to semi-supervised learning with multiple views. proceedings of the icml workshop on learning with multiple views, 2005.

[9] X. Li, Q. Li, D. Li, H. Qian, and J. Wang, Contrastive learning of graphs under label noise. Neural Networks, 2024.

[10] H. Wei, L. Feng, X. Chen and B. An, Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020.

[11] B. Fatemi, L. E. Asri, and S. M. Kazemi, Slaps: self-supervision improves structure learning for graph neural networks. 35th Conference on Neural Information Processing Systems (NeurIPS), 2021.

[12] M. Mcpherson, and S. L. M. Cook, Birds of a feather: homophily in social networks. Annual Review of Sociology, 2001.

[13] Z. Yang, W. W. Cohen, and R. Salakhutdinov, Revisiting semi-supervised learning with graph embeddings, JMLR.org, 2016.

[14] O. Shchur, M. Mumme, A. Bojchevski, and Stephan Günnemann, Pitfalls of graph neural network evaluation. unpublished, 2018.

[15] T. N. Kipf, and M. Welling, Semi-supervised classification with graph convolutional networks. The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR) 2016.

[16] H. Nt, C. J. Jin, and T. Murata, Learning graph neural networks with noisy labels. 2019.

[17] S. Qian, H. Ying, R. Hu, J. Zhou, J. Chen, D. Z. Chen, and J. Wu, Robust Training of Graph Neural Networks via Noise Governance. Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2022.

[18] Y. Zhu, L. Feng, Z. Deng, Y. Chen, R. Amor, and M, Witbrock, Robust Node Classification on Graph Data with Graph and Label Noise. AAAI Technical Track on Machine Learning VI, 2024.

[19] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1944–1952, 2017.