

Robust Graph Neural Networks via Community-Guided Label Refinement and Progressive Contrastive Learning

Zhigang Sun

Guangxi Normal University

Haoran Xu

Guangxi Normal University

Jinke Xu

Guangxi Normal University

Lie Wang

wle3@163.com

Guangxi Normal University

Research Article

Keywords: Graph Neural Networks, Label noise, Community-Guided Label Refinement, Contrastive learning

Posted Date: June 26th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9900086/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Robust Graph Neural Networks via Community-Guided Label Refinement and Progressive Contrastive Learning

Zhigang Sun^{1,2,3}, Haoran Xu^{1,2,3}, Jinke Xu^{1,2,3}, Lie Wang^{1,2,3*}

¹Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin, 541004, China.

²Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin, 541004, China.

³Guangxi Engineering Research Center of Educational Intelligent Technology, Guangxi Normal University, Guilin, 541004, China.

*Corresponding author(s). E-mail(s): wle3@163.com;
Contributing authors: zung@163.com; xuhrkie@163.com;
xujk@stu.gxnu.edu.cn;

Abstract

Graph Neural Networks (GNNs) are highly sensitive to label noise because their message-passing mechanism can propagate erroneous supervision across graph structures. Existing robust GNN methods rely heavily on local homophily but overlook global structural information. Existing robust GNN methods rely heavily on local homophily but overlook global structural information, while in graph contrastive learning, noisy supervision signals can distort representation alignment and limit the exploitation of label semantics. To address these challenges, we propose CG-GNN, a robust graph neural network framework that integrates community-guided label refinement and progressive contrastive learning. Specifically, we leverage community-level structural information to identify and refine noisy labels through an entropy-guided adaptive correction strategy; beyond label correction, we further prune noisy edges at the structural level to suppress noise propagation and optimize information aggregation. We further design a progressive contrastive learning scheme that transitions from unsupervised structural learning to supervised semantic learning using high-confidence refined labels. Extensive experiments on benchmark datasets demonstrate that CG-GNN consistently outperforms state-of-the-art methods under various label-noise settings

and exhibits strong robustness in scenarios with high noise rates and limited labeled data.

Keywords: Graph Neural Networks, Label noise, Community-Guided Label Refinement, Contrastive learning

1 Introduction

Graph-structured data provide a fundamental non-Euclidean representation for modeling entities and their interrelationships, and are prevalent across diverse real-world applications, including social networks [1, 2], recommender systems [3, 4], biological networks [5, 6], and traffic networks [7, 8]. In recent years, Graph Neural Networks (GNNs) have attracted substantial attention from both academia and industry as powerful tools for learning on such data. Through the message-passing paradigm, GNNs enable information propagation and aggregation among nodes, effectively capturing complex dependencies inherent in graph structures and facilitating downstream tasks. In particular, they have demonstrated remarkable performance in semi-supervised node classification [9–11].

Despite the success of GNNs in node classification, their performance heavily depends on the quality and quantity of labeled data. In practice, obtaining large-scale, high-quality annotations is costly and labor-intensive, leading to label sparsity. More critically, the limited labeled data are often contaminated by incorrect annotations due to human subjectivity or adversarial interference, resulting in label noise. These two challenges severely impair GNNs: label sparsity limits the model’s capacity to learn informative representations [12, 13], while noisy labels are easily memorized during training and further propagated through message passing [14, 15], contaminating neighboring node representations and ultimately degrading generalization performance.

To address label noise, the computer vision community has developed several representative approaches. For instance, Co-teaching [16] employs two networks that mutually teach each other by selecting small-loss samples, while DivideMix [17] fits a Gaussian Mixture Model to the loss distribution to dynamically separate clean and noisy samples. However, due to the unique structural dependencies of graph data and the message-passing mechanism of GNNs, these methods are not directly transferable to graph-based learning scenarios. Existing GNN-oriented approaches for handling label noise can be broadly categorized into four types: neighborhood-based label correction, contrastive learning, graph structure augmentation, and sample selection. Specifically, methods such as UnionNET [18], GNNCleaner [19], PIGNN [20], and ERASE [21] leverage neighboring node information for label correction or smoothing; BRGNN [22] and CGNN [23] adopt contrastive learning to enhance robustness, while SSCL [24] proposes a semi-supervised contrastive learning framework guided by label information, and CR-GNN [25] introduces cross-space consistency for mutual learning between contrastive and classification spaces. For graph structure augmentation, NRGNN [26] and RTGNN [27] connect unlabeled nodes with labeled ones to reduce

noise propagation, GNN-CFGD [28] further links unlabeled nodes to clean nodes, and GTA [29] adaptively refines graph topology during training. Finally, sample selection methods like TSS [30] incorporate topological characteristics for clean node selection, RASH [31] disentangles homophilous and heterophilous relations to mitigate interference, and M-RGNN [32] identifies clean and noisy nodes via three-view prediction consistency.

Although the aforementioned methods have made notable progress in handling label noise, several limitations remain. First, existing label correction strategies exploit graph topology primarily from a local perspective, overlooking global structural consistency. This makes them susceptible to interference from local noisy signals in complex graph structures, thereby limiting further improvements in robustness. Second, current graph contrastive learning frameworks neither fully leverage the semantic information embedded in labels nor provide effective control over noise propagation paths. Directly incorporating potentially noisy labels can cause the model to learn erroneous supervision signals in early training stages, while the message-passing mechanism of GNNs further facilitates the spread of noise throughout the graph.

In many graph datasets, communities identified from topological structure often exhibit strong semantic consistency [33], where nodes within the same community tend to share identical or similar class labels. This observation suggests that structural partitioning captures not only connectivity patterns but also an inherent semantic clustering effect at the community level. Motivated by this property, it is structurally justified to elevate label noise handling from the node level to the community level, performing noise mitigation over higher-level structural units.

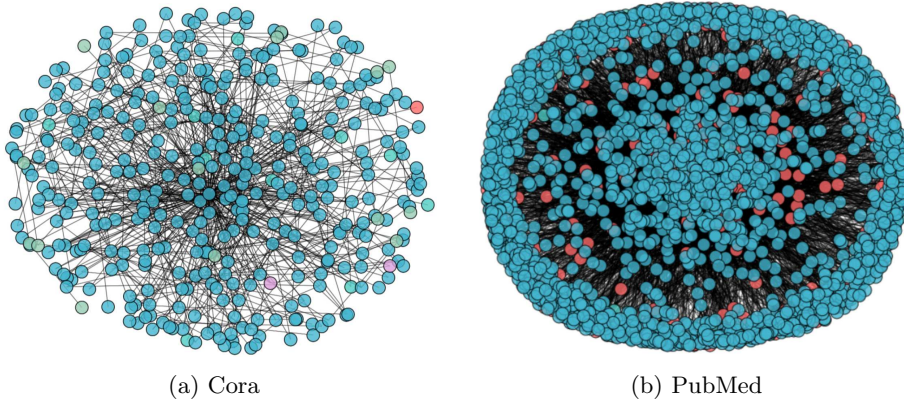
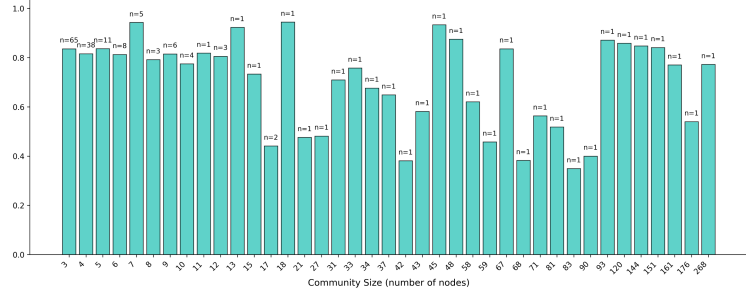
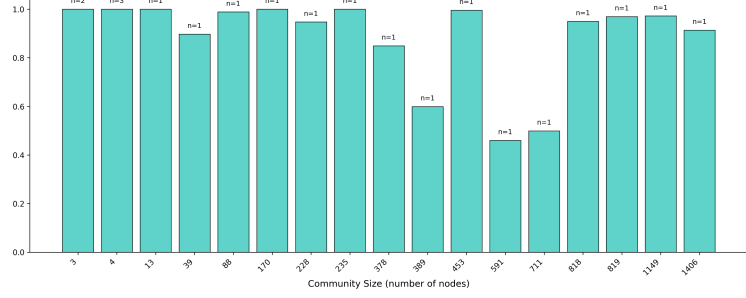


Fig. 1: Visualizing the largest Louvain communities on the Cora and PubMed datasets, where (a) and (b) correspond to Cora and PubMed, respectively. Node colors denote ground-truth class labels.

To validate whether graph data exhibit stable and semantically coherent community structures, we apply the Louvain algorithm [34] to the Cora and PubMed datasets and visualize the largest detected communities in Figure 1. The results show that nodes



(a) Citeseer



(b) Photo

Fig. 2: Statistical analysis of community partitioning on the Citeseer and Photo datasets, where (a) and (b) correspond to Citeseer and Photo, respectively. The horizontal axis denotes community size, while the vertical axis indicates intra-community consistency strength.

within the same community tend to share identical or similar ground-truth labels, a pattern particularly evident in the Cora dataset, where communities form densely connected and semantically consistent clusters. Figure 2 further presents statistical results on the Citeseer and Photo datasets, where the horizontal axis represents community size and the vertical axis indicates intra-community consistency, the notation n above each bar indicates the number of communities of that specific size, measured by the proportion of nodes belonging to the majority class. These observations suggest that handling label noise solely at the node level may be insufficient, as it overlooks higher-level structural dependencies among nodes. Therefore, it is more desirable to elevate noise mitigation from the node level to the community level, where structural consistency and semantic coherence at the community level can be leveraged to achieve more robust label governance.

Motivated by the above challenges and observations, we propose CG-GNN, a robust graph neural network framework that integrates community-guided label refinement with progressive contrastive learning. The proposed framework mitigates the adverse effects of label noise from a global structural perspective while effectively leveraging reliable labels to support model training. To this end, global structural information

is introduced at the community level to assist noise handling and improve robustness. A two-stage contrastive learning strategy is designed to progressively incorporate trustworthy semantic supervision, reducing the impact of noisy labels in the early optimization stages. Additionally, an edge pruning mechanism is employed to restrain noise propagation across the graph. The main contributions of this work are summarized as follows:

- We propose a community-level adaptive label correction method that treats each community as the fundamental unit for refinement. By dynamically adjusting the correction strategy based on the consistency of label distributions within each community, the method enables differentiated noise handling across different structural regions of the graph.
- We develop a progressive contrastive learning framework that gradually incorporates semantic supervision during training, thereby mitigating the disruption caused by noisy labels in the early training stage and facilitating more stable topological representation learning.
- We introduce an edge pruning strategy to suppress potential noise propagation paths at the structural level, thereby limiting the spread of erroneous information across the graph.
- Extensive experiments on multiple benchmark datasets demonstrate that CG-GNN consistently achieves competitive performance and exhibits strong robustness under various noisy-label settings.

2 Related work

2.1 Graph Neural Networks

Graph Neural Networks (GNNs) have achieved substantial progress as powerful tools for modeling graph-structured data in non-Euclidean spaces. Among existing architectures, spatial-based methods, which perform convolution directly on the graph topological domain by aggregating information from neighboring nodes, have become the prevailing paradigm. GCN[35] performs graph convolution through normalized aggregation of neighboring node features, while GraphSAGE[36] extends GNNs to large-scale graphs through neighborhood sampling and feature aggregation. GAT[37] further enhances representation learning by introducing an attention mechanism to assign different importance weights to different neighbors. From a theoretical perspective, GIN[38] analyzes the expressive power of GNNs, identifies limitations of earlier models in distinguishing graph structures, and proposes a more expressive architecture. In addition, S2GNN[39] combines spatial message passing with spectral graph filtering, allowing broader information interaction through spectral parameterization and thereby improving long-range information propagation and model expressiveness. Despite these advances, most existing GNNs still depend strongly on high-quality labeled data, whereas real-world graph datasets often contain sparse and noisy labels, which poses additional challenges to their practical effectiveness.

2.2 Graph Neural Networks under Label Noise

In graph data, label noise may propagate and accumulate through the message-passing mechanism of GNNs, thereby undermining representation learning and degrading generalization performance. To reduce the adverse effects of noisy labels on GNN training, existing studies have primarily investigated four main directions: sample selection, graph structure enhancement, contrastive learning, and noise correction.

Sample Selection: Sample selection methods aim to identify relatively clean nodes from noisy labeled data and use these high-confidence samples for supervised training, thereby reducing the direct influence of noisy labels. RTGNN[27] and GNN-CFGD[28] adopt the small-loss criterion within the GNN training process to identify potential clean nodes based on training loss. TSS[30] further incorporates node topological characteristics and structural importance into the selection process, while M-RGNN[32] distinguishes clean nodes from noisy ones through multi-view prediction consistency.

Graph Structure Enhancement: Because noisy labels may spread along the graph topology, graph structure enhancement methods aim to suppress noise propagation or improve the graph structure at the structural level. NRGNN[26] reconnects labeled and unlabeled nodes based on latent structural similarity to reduce the influence of noisy labels. After identifying clean nodes, GNN-CFGD[28] reconstructs the graph structure using reliable labels to alleviate noise diffusion. GTA[29] further improves robustness through dynamic and adaptive graph topology optimization via coordinated edge augmentation and pruning. Similarly, RNCGLN[40] proposes a graph self-improvement strategy that progressively refines the original graph structure based on node representations learned during training.

Contrastive Learning: Contrastive learning enhances robustness in noisy environments by constructing positive and negative sample pairs, encouraging similar samples to stay close in the representation space while separating dissimilar ones. Owing to this property, it has been widely used to improve the robustness of GNNs under label noise. In this line of research, view generation and the construction of positive and negative node pairs are two key factors that directly affect the quality of contrastive learning. DGI[41] learns node representations by maximizing mutual information between local node embeddings and a global graph summary. MVGRL[42] adopts a multi-view contrastive strategy by generating different views through diffusion matrices and maximizing the consistency between representations from different views. GRACE[43] further constructs augmented views through random perturbation strategies, such as node feature masking and edge dropping, and treats representations of the same node across different views as positive pairs while treating those of different nodes as negative pairs. Building on this idea, GCA[44] refines the view generation process by considering the unequal importance of nodes in a graph and proposes adaptive augmentation based on node centrality and edge weights, making the augmentation strategy more compatible with graph structural characteristics. In addition, SUGRL[45] introduces multiple contrastive loss functions to capture relationships at different levels and thereby enrich supervisory signals. CR-GNN[25] further incorporates a neighborhood contrastive loss that exploits the homophily property of graph structure by regarding adjacent nodes as positive pairs, which helps alleviate the impact of label noise.

Noise Correction: Noise correction methods seek to explicitly revise erroneous labels in order to reduce the adverse effects of noisy supervision. D-GNN[46], grounded in loss correction theory, introduces a noise transition matrix to address symmetric label noise. UnionNET[18] incorporates graph structural information into the label correction process. PIGNN[20] reformulates node-level label correction as the learning and refinement of pairwise node relationships. GNNCleaner[19] replaces noisy labels with a convex combination of pseudo-labels and initial labels, thereby retaining information from the original annotations while incorporating pseudo-label signals propagated through the graph structure. In addition, ERASE[21] progressively smooths target nodes with information from neighboring nodes during training to obtain more robust node representations.

Overall, existing studies on label noise in Graph Neural Networks have made progress in node-level modeling and local structural optimization. Nevertheless, they remain limited to exploit global structural information, restrain noise propagation across different structural regions, and maintain the stability of contrastive learning under high-noise settings. These limitations point to the need for noise mitigation mechanisms defined at a higher structural level and provide further motivation for exploring this direction.

3 Preliminaries

3.1 Notations and problem definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ represents the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. Each node v_i is associated with a feature vector $x_i \in \mathbb{R}^d$, and the feature vectors of all nodes collectively form the feature matrix $X \in \mathbb{R}^{N \times d}$. The graph structure is characterized by an adjacency matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$, where $\mathbf{A}_{ij} = 1$ indicates that an edge exists between nodes v_i and v_j . Let $\mathcal{V}_L \subset \mathcal{V}$ denote the set of labeled nodes and $\mathcal{V}_U = \mathcal{V} \setminus \mathcal{V}_L$ denote the set of unlabeled nodes, where the number of labeled nodes is much smaller than that of unlabeled nodes. In the presence of label noise, the objective of semi-supervised node classification is to develop a robust graph neural network model that can learn reliable node representations from limited labeled data despite partial label corruption, and accurately predict the classes of unlabeled nodes. Table 1 presents the principal notations used in this paper and their definitions.

3.2 Louvain community detection algorithm

Community detection seeks to partition a graph into subsets of nodes with dense intra-community connections and relatively sparse inter-community connections. Among existing methods, the Louvain algorithm[34] adopts modularity as the optimization objective to assess the quality of a given community partition. Modularity measures the degree to which edges are more densely distributed within communities than would be expected under a random baseline. In general, a higher modularity value suggests a clearer community structure, characterized by stronger intra-community connectivity

Table 1: Principal notations and their definitions

Symbol	Definition
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	The original graph $\mathcal{G}$ where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set.
$\mathbf{X} \in R^{N \times d}$	Original feature matrix of nodes.
$\mathbf{A} \in R^{N \times N}$	Original adjacency matrix.
h_i, z_i	Feature representations of node v_i .
$\mathcal{V}_L \subset \mathcal{V}$	Set of labeled nodes.
$\mathcal{V}_U = \mathcal{V} \setminus \mathcal{V}_L$	Set of unlabeled nodes.
W_{ij}	Cosine similarity weight between nodes v_i and v_j .
$\hat{\mathbf{A}}$	Augmented adjacency matrix after graph enhancement and edge pruning.
$f_{\theta_1}, f_{\theta_2}$	GCN encoders with independent parameters θ_1 and θ_2 .
$P_{\theta_1}^i, P_{\theta_2}^i$	Predicted probability distribution over classes for node v_i .
$C = \{C_1, C_2, \dots, C_m\}$	Set of communities obtained by the Louvain algorithm.
d_k	Number of nodes of class k in community C_m .
$H(C_m)$	Community information entropy of C_m .
N_{c_m}	Number of labeled nodes in community C_m .
τ	Temperature hyperparameter in contrastive loss.
α, β, λ	Hyperparameters balancing the total loss function.

and weaker inter-community connectivity. The modularity is defined as follows:

$$Q = \frac{1}{2|\mathcal{E}|} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2|\mathcal{E}|} \right) \Pi(c_i = c_j) \quad (1)$$

Here, A_{ij} denotes the adjacency relationship between nodes i and j ; k_i and k_j represent the degrees of the corresponding nodes; $|\mathcal{E}|$ is the total number of edges in the graph; and c_i denotes the community assignment of node v_i . The function $\Pi(\cdot)$ is an indicator function. Specifically, the algorithm initially treats each node as an individual community and iteratively moves nodes to neighboring communities in order to maximize the gain in modularity. The resulting communities are then aggregated into super-nodes to construct a new weighted graph, and the above procedure is repeated until no further substantial increase in modularity is observed.

By applying the Louvain algorithm, the graph can be partitioned into a set of structurally cohesive communities, $C = \{C_1, C_2, \dots, C_m\}$, characterized by dense intra-community connectivity and relatively strong internal consistency. This process relies solely on graph topology and does not involve node label information, thereby reducing the influence of noisy labels on community partitioning. Even when label noise is present, although some individual nodes may be incorrectly annotated, the overall structural properties and label distribution within their respective communities may still exhibit a relatively consistent pattern. Based on this observation, we treat Louvain communities as the basic units for noise mitigation. Building on this

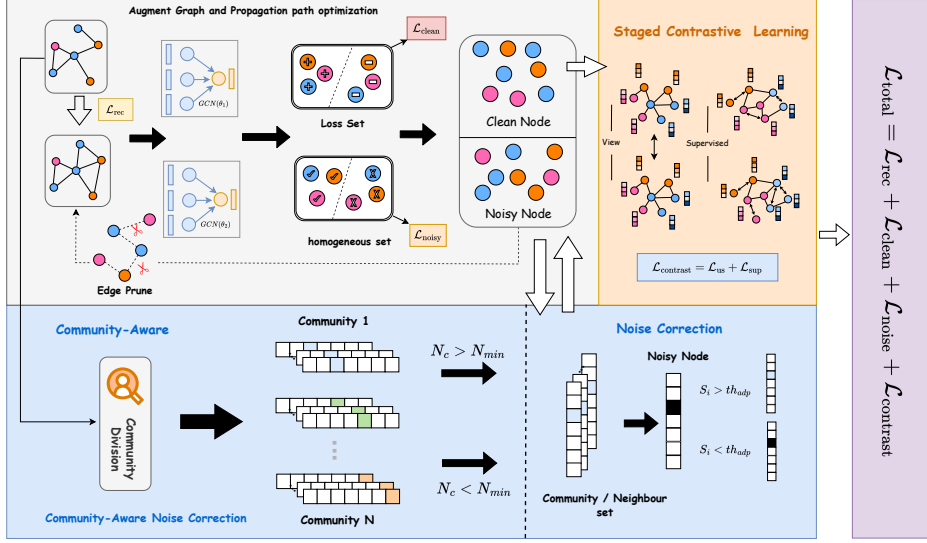


Fig. 3: The framework of our CG-GNN approach, consisting of three components: augmented graph and propagation path optimization, community-aware noise correction, and staged contrastive learning.

foundation, we further develop a community-aware adaptive noise correction strategy according to the label distribution characteristics of each community.

4 Methodology

This section proposes a community-guided robust graph neural network framework, termed CG-GNN, to address performance degradation caused by label noise in graph data. CG-GNN incorporates community-level global structural information to guide noise mitigation and develops a staged contrastive learning strategy to make effective use of label semantics in a more robust manner. In addition, the framework integrates a noisy-edge pruning mechanism to proactively suppress noise propagation from a topological perspective. As illustrated in Figure 3, CG-GNN consists of three core components: (1) augmented graph and propagation path optimization; (2) community-aware noise correction; and (3) staged contrastive learning. The following sections describe these components in detail.

4.1 Graph Augmentation and Propagation Path Optimization

4.1.1 Graph Augmentation

Since the number of labeled nodes in a graph is limited, we enhance the graph structure by connecting unlabeled nodes to labeled nodes so as to facilitate the propagation of supervisory signals. Specifically, we first employ a GCN encoder to obtain node

representations, denoted as $Z = \text{GCN}(X, A)$. Then, for each unlabeled node v_i , we compute its structural similarity to each labeled node v_j based on their feature representations, and use this value as the latent association weight W_{ij} between the two nodes. A larger value of W_{ij} indicates a higher feature similarity between v_i and v_j , and thus implies a greater likelihood that an edge should be established between nodes v_i and v_j .

$$W_{ij} = \max\left(\frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}, 0\right) \quad (2)$$

Furthermore, the weight estimation is optimized via an edge reconstruction loss defined as:

$$\mathcal{L}_{rec} = \sum_{v_i \in V} \left(\sum_{v_j \in N(i)} (W_{ij} - A_{ij})^2 + N_{neg} \cdot \mathbb{E}_{v_j \sim P_n} [(W_{ij'} - A_{ij'})^2] \right) \quad (3)$$

Here, the positive sample set for node v_i consists of its true neighbors in the original graph, namely node pairs (v_i, v_j) satisfying $A_{ij} = 1$. The negative samples correspond to node pairs without direct connections in the original graph, that is, pairs satisfying $A_{ij} = 0$. To avoid the high computational cost incurred by enumerating all non-neighboring nodes, we randomly sample N_{neg} nodes from the non-neighbor set of v_i according to a uniform distribution as negative samples, where the sampling distribution is denoted by P_n . Given a threshold th_{ag} for filtering edges with small weights, the augmented adjacency matrix $\hat{A}$ is constructed as:

$$\hat{A}_{ij} = \begin{cases} 1 & \text{if } A_{ij} = 1; \\ W_{ij} & \text{if } A_{ij} = 0, \text{ and } W_{ij} > th_{ag}; \\ 0 & \text{others.} \end{cases} \quad (4)$$

Here, $j \in \mathcal{V}_L$ indicates that node v_j belongs to the labeled node set.

4.1.2 Noise Detection

To effectively identify and mitigate the impact of label noise on graph neural network training, we adopt both loss-based estimation and neighborhood-consistency-based detection to identify noisy nodes. Specifically, two GCN encoders f_{θ_1} and f_{θ_2} with identical architectures but independent parameters, are employed to learn node representations. Based on the learned representations, we combine loss-driven sample selection with a neighborhood homophily-based voting mechanism to achieve high-confidence identification of noisy nodes.

We utilize two independent GCN encoders to learn node representations on the enhanced graph $\hat{A}$, and obtain the predicted class probability distribution for each node as:

$$P_{\theta_1} = \text{softmax}\left(f_{\theta_1}(\hat{A}, X, \theta_1)\right), \quad P_{\theta_2} = \text{softmax}\left(f_{\theta_2}(\hat{A}, X, \theta_2)\right) \quad (5)$$

The joint cross-entropy loss for node v_i is defined as:

$$\begin{aligned}\mathcal{L}_{pre}^i &= -(y^i \log P_{\theta_1}^i + y^i \log P_{\theta_2}^i) \\ &= -y^i \log(P_{\theta_1}^i \cdot P_{\theta_2}^i)\end{aligned}\tag{6}$$

where $P_{\theta_1}^i$ and $P_{\theta_2}^i$ denote the predicted label distributions for node v_i produced by encoders f_{θ_1} and f_{θ_2} , respectively. This loss quantifies the consistency and confidence of the two GCN models with respect to the node’s label prediction. A smaller loss value suggests that the node label is more likely to be clean.

Loss-based noisy node detection: The memorization effect of deep neural networks suggests that models tend to first learn simple patterns, namely clean labels [47], during the early stages of training. As a result, clean nodes generally incur smaller training losses than noisy nodes. Motivated by this observation, we partition nodes according to their loss values. Specifically, in each training epoch, a subset of nodes with relatively small prediction losses, as estimated by the two GCN encoders, is selected as the candidate clean node set, while the remaining nodes are regarded as potentially corrupted by label noise. During the early training phases, a relatively relaxed selection threshold is adopted so that more nodes are included in the clean set, ensuring sufficient supervisory signals and stable convergence. As training progresses, the criterion is gradually tightened to reduce the size of the clean node set, thereby filtering out high-loss nodes and suppressing overfitting to noisy labels. To enhance robustness, we introduce r_{epo} and r_{avg} to evaluate node reliability from two perspectives: training stability across epochs and the overall loss magnitude. This design alleviates fluctuations and biases caused by single-epoch loss estimation, leading to more stable node partitioning results. Finally, based on the maximum value $r^* = \max(r_{epo}, r_{avg})$, the labeled node set $\mathcal{V}_L$ is divided into a clean subset $\mathcal{V}_{sc}$ identified by small-loss selection and a noisy subset $\mathcal{V}_{sn}$ identified accordingly.

$$r_{epo} = 1 - \left(0.5 \cdot \frac{epoch}{T}\right), r_{avg} = \frac{\text{Count}(\mathcal{L}_{pre} < \text{mean}(\mathcal{L}_{pre}))}{|\mathcal{L}_{pre}|}\tag{7}$$

$$\mathcal{V}_{sc} = \{i \mid \mathcal{L}_{pre}^i < r^*\}, \quad \mathcal{V}_{sn} = \mathcal{V}_L \setminus \mathcal{V}_{sc}\tag{8}$$

Here, $epoch$ denotes the current training iteration, T represents the total number of training epochs, $\mathcal{L}_{pre}$ denotes the loss set, and $\mathcal{L}_{pre}^i$ represents the loss of node v_i .

Homogeneity-based noisy node detection: To further improve the reliability of noisy node identification, we introduce a node partitioning strategy based on a neighborhood homophily voting mechanism, which partitions nodes from the perspective of local consistency. Graph data typically satisfy the homophily assumption, meaning that neighboring nodes tend to share similar labels. Based on this property, we aggregate the predicted information of neighboring nodes to evaluate the consistency between a target node and its neighborhood in terms of label prediction. Specifically, for any labeled node v_i , we first obtain the predicted labels of its neighboring nodes $v_j \in \mathcal{N}(i)$ under encoders f_{θ_1} and f_{θ_2} , $\mathbf{y}_j^1 = \arg \max_k \mathbf{P}_{\theta_1}^j(k)$, $\mathbf{y}_j^2 = \arg \max_k \mathbf{P}_{\theta_2}^j(k)$. Subsequently, the predicted labels produced by the two encoders over all neighboring nodes are aggregated, and a majority voting strategy is applied to determine the

neighborhood consensus label of node v_i :

$$y_{nbr}^i = \arg \max_k \sum_{v_j \in \mathcal{N}(i)} (\Pi(y_j^1 = k) + \Pi(y_j^2 = k)) \quad (9)$$

Here, $k \in \{1, 2, \dots, K\}$ denotes the class label, and $\Pi(\cdot)$ is the indicator function.

Furthermore, we define the consistency ratio of node v_i as ρ_i , which measures the degree of agreement among the predicted labels of its neighbors:

$$\rho_i = \frac{1}{2|\mathcal{N}(i)|} \sum_{v_j \in \mathcal{N}(i)} (\Pi(y_j^1 = y_{nbr}^i) + \Pi(y_j^2 = y_{nbr}^i)) \quad (10)$$

If the original label y_i of node v_i is inconsistent with the neighborhood consensus label y_{nbr}^i , while the neighborhood homophily ratio ρ_i remains high, the node label is considered likely to be noisy. Given a consistency threshold th_{hm} , the labeled node set $\mathcal{V}_L$ can be partitioned into a homophily-based clean subset $\mathcal{V}_{hc}$ and a noisy subset $\mathcal{V}_{hn}$:

$$\mathcal{V}_{hn} = \{i \mid i \in \mathcal{V}_L \wedge y_i \neq y_{nbr}^i \wedge \rho_i > th_{hm}\}, \mathcal{V}_{hc} = \mathcal{V}_L \setminus \mathcal{V}_{hn} \quad (11)$$

Finally, we fuse the small-loss-based partition results $\mathcal{V}_{sc}$, $\mathcal{V}_{sn}$ with the homophily-based partition results $\mathcal{V}_{hc}$, $\mathcal{V}_{hn}$ to obtain the final clean sets $\mathcal{V}_c$ and noisy node sets $\mathcal{V}_n$:

$$\mathcal{V}_c = \mathcal{V}_{sc} \cap \mathcal{V}_{hc}, \quad \mathcal{V}_n = \mathcal{V}_{sn} \cap \mathcal{V}_{hn} \quad (12)$$

4.1.3 Propagation path optimization

To prevent noisy nodes from propagating erroneous information through the graph structure, we perform edge pruning on the noisy node set $\mathcal{V}_n$. This strategy jointly considers the predictive uncertainty of nodes and local structural inconsistency, dynamically assessing the noise risk of each node and selectively removing its connections accordingly.

We first compute the predictive uncertainty score s_{loss}^i and the structural inconsistency score s_{struct}^i for each noisy node $v_i \in \mathcal{V}_n$. These two components are then combined to obtain the overall noise risk score η_i for node v_i . By jointly characterizing prediction confidence deviation and structural inconsistency, this score enhances the stability and robustness of the noise identification process:

$$s_{loss}^i = \frac{\exp(\mathcal{L}_{pre}^i)}{\sum_{j \in \mathcal{V}_{sn}} \exp(\mathcal{L}_{pre}^j)}, \quad s_{struct}^i = 1 - \rho_i \quad (13)$$

$$\eta_i = \frac{s_{loss}^i + s_{struct}^i}{2} \quad (14)$$

Since the identification of noisy nodes may itself be uncertain, directly removing all their connections could mistakenly discard potentially useful structural information. To address this issue, we adopt a randomized edge pruning strategy. When the noise

risk of node v_i exceeds a predefined threshold th_{prune} , its adjacent edges are pruned with probability p_{cut} , thereby enabling progressive noise suppression:

$$A_{ij}^{prune} = \begin{cases} 0, & \eta_i > th_{prune} \wedge \text{rand}() < p_{cut} \\ A_{ij}, & \text{otherwise} \end{cases} \quad (15)$$

4.2 Community-aware noise correction

In this section, we propose an adaptive noise correction strategy based on community information entropy. This approach treats each community as a fundamental unit of noise governance, measuring its purity by evaluating the consistency of the internal label distribution. Based on the estimated purity level, differentiated correction measures are applied accordingly. Compared with node-level processing methods, this strategy elevates noise mitigation to the community level, thereby enhancing the global consistency and structural adaptability of the correction process.

4.2.1 Community-aware

We first apply the Louvain algorithm to partition the original graph $\mathcal{G}$ into communities, resulting in a community set $C = \{C_1, C_2, \dots, C_m\}$. Each community consists of a group of densely connected nodes, and there is no overlap between different communities. In this work, communities are used solely as structural auxiliary information during the noise mitigation stage and do not participate in node representation learning or message passing. Specifically, the community partition is computed once prior to model training and remains fixed throughout the entire training process, providing stable structural grouping information for the subsequent label correction module.

Furthermore, the community partition serves as an important criterion for selecting the correction strategy for noisy nodes. For a noisy node, if the number of nodes in its belonging community is sufficiently large, a community-based correction strategy is adopted, since the community can provide relatively stable and informative structural context. In contrast, if the size of its community is small, a neighbor-based correction strategy is used instead. The detailed correction procedures will be introduced in Section 4.2.2. For nodes assigned to neighbor-based correction, if the number of available neighbors is too small, the correction process will be skipped to avoid unreliable updates caused by insufficient local information.

4.2.2 Noise correction

We adopt community information entropy as a metric to quantify label consistency within each community. For a community C_m containing N_{c_m} labeled nodes, assume that the labeled nodes in C_m are distributed across K classes. Let the number of nodes in C_m with label k be $d_k = \sum_{i=1}^{N_{c_m}} \Pi(y_i = k)$, and define the empirical label distribution within the community as $P(k) = \frac{d_k}{N_{c_m}}$. The information entropy of community C_m is then defined as:

$$H(C_m) = - \sum_{k=1}^K P(k) \cdot \log P(k) \quad (16)$$

A smaller value of $H(C_m)$ indicates stronger label consistency within the community, whereas a larger entropy value suggests a more heterogeneous label distribution, implying that the community may be more severely affected by label noise.

To enable differentiated treatment of communities with varying purity levels, we design an adaptive threshold th_{adp} that is negatively correlated with community information entropy $th_{adp} = \sigma \left(1 - \frac{H(C_m)}{\log K} \right)$, where $\sigma(x)$ denotes the Sigmoid function. The entropy-aware threshold th_{adp} is inversely proportional to the information entropy of the community and is determined based on the following considerations. 1) For low-entropy communities with highly consistent label distributions, the community exhibits strong internal structural homophily, and the existing labeled nodes possess high prior reliability. In this case, a relatively high correction threshold is adopted to preserve reliable label information within the community and prevent additional errors caused by excessive correction. 2) For high-entropy communities with heterogeneous label distributions, the community may be significantly affected by noise contamination or class boundary ambiguity, resulting in lower confidence in supervisory signals. In this scenario, a lower correction threshold is applied to facilitate the formation of consistent semantic representations and suppress the diffusion of noise across the graph.

Next, we design a community-based correction strategy and a neighborhood-based correction strategy to handle communities of different scales. For a given community, if the number of nodes satisfies $N_{c_m} \geq N_{min}$, meaning that the community size is sufficiently large, the community-based correction strategy is applied. Specifically, we first determine the majority consensus label of community C_m as $y_{cons}^c = \arg \max_k d_k$ and construct the consensus node set $\mathcal{V}_{cons}^c = \{v_i \in C_m \mid y_i = y_{cons}^c\}$. Then, for any noisy node v_i within the community, we compute the average similarity between its representation and those of the consensus nodes. This value serves as the semantic consistency score between the node and the community majority $S_i = \frac{1}{|\mathcal{V}_{cons}^c|} \sum_{u \in \mathcal{V}_{cons}^c} \text{sim}(h_v, h_u)$. If $S_i \geq th_{adp}$, the node is considered to exhibit strong semantic consistency with the majority nodes in the community within the representation space, indicating high confidence in label correction. In this case, the label of the noisy node v_i is corrected to y_{cons}^c .

If the number of nodes in community C_m is smaller than the threshold N_{min} , that is $N_{c_m} < N_{min}$, the community scale is considered insufficient. In this case, the internal label distribution and consistency statistics may lack stability and representativeness, and the community-based correction strategy becomes more susceptible to noise interference. Therefore, a neighborhood-based correction strategy is adopted as an alternative. For a noisy node v_i within such a community, local consistency modeling is performed based on the label distribution of its one-hop neighbors $\mathcal{N}_v$. Specifically, we first determine the neighborhood consensus label y_{cons}^n and the corresponding consensus node set $\mathcal{V}_{cons}^n$. The neighborhood consistency score is then computed in the same manner as the community-level consistency score. If the number

of neighbors supporting the consensus label is fewer than two, no correction is applied to avoid unreliable updates. In addition, the adaptive threshold mechanism used in neighborhood correction remains consistent with that of community correction. It is also dynamically adjusted based on entropy guidance, with the only difference being that the entropy is computed from the neighborhood label distribution rather than the community label distribution. Formally, based on the adaptive threshold and the consistency score, the label correction decision for node v_i can be expressed as:

$$y_i = \begin{cases} y_{cons}^c & \text{if } S_i \geq th_{adp} \text{ and } N_{c_m} > N_{min} \\ y_{cons}^n & \text{if } S_i \geq th_{adp} \text{ and } N_{c_m} < N_{min} \\ y_i & \text{otherwise} \end{cases} \quad (17)$$

4.3 Staged contrastive learning

To leverage label semantic information while mitigating noise interference, we design a staged contrastive learning framework. In the early stage of training, unsupervised contrastive learning is employed to fully explore the topological structure of the graph, thereby avoiding contamination from noisy labels at the initial phase. In the later stage, high-confidence labels obtained after community-based correction are incorporated to construct supervised semantic contrastive learning, enabling deeper integration of structural topology and semantic information.

4.3.1 Constructing of augmented views

Unlike conventional approaches that rely on random data augmentation, we directly construct contrastive views using encoders with independent parameters. Based on the enhanced graph structure $\hat{A}$, two graph encoders f_{θ_1} and f_{θ_2} are employed. Through random parameter initialization, two augmented views are generated that share the same topological structure but yield different node representations. The node feature matrices of the two views are defined as $Z_1 = f_{\theta_1}(\hat{A}, X) \in R^{N \times d}$, $Z_2 = f_{\theta_2}(\hat{A}, X) \in R^{N \times d}$. The construction of augmented views does not depend on additional random graph perturbations or data augmentation operations. Instead, under the shared enhanced graph structure and node features, representation diversity is introduced through parameter-space independence, providing stable and effective sample pairs for contrastive learning. On the one hand, sharing the same structure ensures that node representations across different views maintain consistent structural and semantic foundations. On the other hand, parameter-level differences offer complementary perspectives, thereby enhancing the effectiveness of contrastive learning signals.

4.3.2 Unsupervised contrastive learning

To enhance robustness against noisy labels, an unsupervised contrastive learning module is introduced during the early training stage. The unsupervised contrastive objective encourages the model to learn generalizable and robust node representations from the graph topology, thereby reducing interference from noisy labels at the beginning of training. For each node v_i , its representations under the two views, Z_1^i and

Z_2^i , are treated as a positive pair, while representations of different nodes form negative pairs. The InfoNCE loss is adopted as the optimization objective for unsupervised contrastive learning:

$$D_i^{\text{us}} = \sum_{j=1}^N \exp(\text{sim}(Z_1^i, Z_2^j)/\tau) + \sum_{j=1, j \neq i}^N \exp(\text{sim}(Z_1^i, Z_1^j)/\tau) \quad (18)$$

$$\ell_{\text{us}}(Z_1^i, Z_2^i) = -\log \frac{\exp(\text{sim}(Z_1^i, Z_2^i)/\tau)}{D_i^{\text{us}}} \quad (19)$$

$$\mathcal{L}_{\text{us}} = \frac{1}{2N} \sum_{i=1}^N [\ell_{\text{us}}(Z_1^i, Z_2^i) + \ell_{\text{us}}(Z_2^i, Z_1^i)] \quad (20)$$

Here, τ denotes the temperature parameter, which controls the smoothness of the similarity distribution. This loss encourages representations of the same node across different views to be close in the embedding space, while pushing representations of different nodes apart, thereby learning node embeddings that effectively capture structural similarity in the graph.

4.3.3 Supervised contrastive learning

After completing the unsupervised contrastive learning stage, the model proceeds to supervised semantic contrastive learning. At this stage, high-confidence labels produced by the community-aware noise correction module are used as supervisory signals to construct a semantic-aware contrastive objective. For each labeled node v_i , we define its semantic positive sample set as $\text{Pos}(i) = \{v_j \in \mathcal{V} \mid y_j = y_i, j \neq i\}$, and its semantic negative sample set as $\text{Neg}(i) = \{v_j \in \mathcal{V} \mid y_j \neq y_i\}$. Here, $\text{Pos}(i)$ consists of nodes sharing the same class label as v_i , while $\text{Neg}(i)$ contains nodes belonging to different classes. Based on these definitions, the supervised semantic contrastive loss is formulated as follows:

$$\ell_{\text{sup}}(Z_i, Z_p) = -\log \frac{\exp(\text{sim}(Z_i, Z_p)/\tau)}{\sum_{a \in \text{Pos}(i) \cup \text{Neg}(i)} \exp(\text{sim}(Z_i, Z_a)/\tau)} \quad (21)$$

$$\mathcal{L}_{\text{sup}} = \frac{1}{N} \sum_{i \in V} \frac{1}{|\text{Pos}(i)|} \sum_{p \in \text{Pos}(i)} \ell_{\text{sup}}(Z_i, Z_p) \quad (22)$$

This loss function serves a dual optimization objective. On the one hand, it pulls representations of nodes belonging to the same class closer in the embedding space, thereby enhancing intra-class compactness. On the other hand, it pushes representations of nodes from different classes apart, improving inter-class separability. In practical implementation, for a fixed anchor node v_i , the similarity between the anchor and nodes in the candidate set is computed only once, and the result is shared across all positive pairs. By using high-confidence labels obtained from the community-aware correction module as supervisory signals, the model is able to learn semantically enriched node representations while mitigating the adverse influence of noisy labels during the early training stage.

4.3.4 Total loss

To jointly optimize noise detection, graph structure enhancement, and contrastive representation learning, we design a multi-task objective function. This overall loss consists of three components: the prediction loss, the edge reconstruction loss, and the contrastive loss. Trainable weighting coefficients are introduced to balance these modules and enable coordinated optimization.

To effectively distinguish clean nodes from noisy nodes and assign appropriate supervisory weights to different node subsets, we adopt a node-partition-aware weighted prediction loss. This loss leverages the node partition results obtained from the noise detection module and assigns different weights to different node subsets, thereby suppressing noisy nodes while reinforcing high-confidence nodes during training. For clean nodes in $\mathcal{V}_c$, whose labels are considered reliable, the original labels are directly used for supervision. For noisy nodes in $\mathcal{V}_n$, whose label confidence is lower, a smaller supervision weight β is assigned to mitigate the negative impact of noisy labels and prevent overfitting. This weighting strategy reduces the contribution of noisy nodes in the loss computation and protects the model from misleading supervision. The weight function $\omega(i)$ is defined as:

$$\omega(i) = \begin{cases} 1, & \text{if } v_i \in \mathcal{V}_c \\ \beta, & \text{if } v_i \in \mathcal{V}_n \end{cases} \quad (23)$$

The overall prediction loss is then formulated as:

$$\mathcal{L}_{pred} = \frac{1}{|\mathcal{V}_L|} \sum_{v_i \in \mathcal{V}_L} \omega(i) \cdot \mathcal{L}_{pre}^i \quad (24)$$

The contrastive loss is defined as:

$$\mathcal{L}_{contrast} = \mathcal{L}_{us} + \mathcal{L}_{sup} \quad (25)$$

The overall training objective is formulated as:

$$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_{rec} + \mathcal{L}_{pred} + \lambda \cdot \mathcal{L}_{contrast} \quad (26)$$

The pseudocode of the proposed framework is presented in Algorithm 1. The model takes as input the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, node features $\mathbf{X}$, noisy training labels Y_L , and related hyperparameters, and outputs the predicted labels for $\mathcal{V}_U$. First, the encoder parameters are initialized independently, and the original graph is partitioned into communities using the Louvain algorithm to obtain community-level structural information. During each training epoch, the model first constructs the enhanced graph and updates the adjacency matrix, and then employs the two encoders to learn node representations and prediction probabilities. Based on the prediction loss and neighborhood consistency, nodes are further divided into clean and noisy subsets, followed by dynamic edge pruning on noisy nodes to optimize the information propagation paths. Next, label correction is performed for nodes in the noisy set to alleviate the adverse effect of noisy labels. After that, a staged contrastive learning strategy

is adopted: the unsupervised contrastive loss is used in the warm-up stage, while the supervised contrastive loss is introduced after the warm-up stage to further enhance the semantic discriminability and robustness of node representations. Finally, the total loss is computed and the model parameters are updated via backpropagation. After all training epochs, the trained encoders f_{θ_1} and f_{θ_2} are returned.

Algorithm 1 CG-GNN Training Procedure

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; node features $\mathbf{X}$; noisy training labels Y_L ; hyperparameters α, β, λ ; total epochs T ; warm-up epochs T_w

Ensure: Predicted labels for $\mathcal{V}_U$

- 1: Initialize encoder parameters θ_1 and θ_2 independently
 - 2: Partition graph $\mathcal{G}$ into communities $C = \{C_1, C_2, \dots, C_m\}$ using the Louvain algorithm
 - 3: **for** $epoch = 1$ to T **do**
 - 4: Construct enhanced graph and update adjacency matrix $\hat{A}$ (Eq. 4)
 - 5: Compute node representations and prediction probabilities using f_{θ_1} and f_{θ_2} (Eq. 5)
 - 6: Detect noisy nodes based on prediction loss and neighborhood consistency (Eq. 8, Eq. 11)
 - 7: Partition nodes into clean set $\mathcal{V}_c$ and noisy set $\mathcal{V}_n$ (Eq. 12)
 - 8: Perform dynamic edge pruning on noisy nodes and update adjacency matrix (Eq. 15)
 - 9: **for** each $v \in \mathcal{V}_n$ **do**
 - 10: Perform label correction (Eq. 17)
 - 11: **end for**
 - 12: **if** $epoch < T_w$ **then**
 - 13: Compute unsupervised contrastive loss (Eq. 20)
 - 14: **else**
 - 15: Compute supervised contrastive loss (Eq. 22)
 - 16: **end if**
 - 17: Compute total loss (Eq. 26) and update parameters via backpropagation
 - 18: **end for**
 - 19: **return** Predicted labels for $\mathcal{V}_U$
-

5 Experiments

5.1 Datasets

In this section, we evaluate the robustness of CG-GNN under different label noise settings. Experiments are conducted on four widely used graph node classification datasets, including three citation network benchmarks—Cora, Citeseer, and PubMed, as well as the Amazon product co-purchase dataset Amazon Photo. The Cora dataset contains 2,708 nodes and 5,429 edges, with nodes categorized into 7 classes. Citeseer

Table 2: Statistics of Experimental Datasets

Dataset	Nodes	Edges	Features	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
PubMed	19717	44338	500	3
Photo	7650	119081	745	8

consists of 3,327 nodes and 4,732 edges across 6 classes. PubMed includes 19,717 nodes and 44,338 edges, divided into 3 classes. The Photo dataset contains 7,650 nodes and 119,081 edges, with nodes grouped into 8 classes. Detailed statistics are summarized in Table 2.

To simulate real-world label noise, two commonly adopted noise types are introduced for each dataset: Uniform Noise and Pair Noise. Under uniform noise, the label of each node is randomly flipped to any other class with probability p , and the transition probability is identical across all classes. Under pair noise, each node label is flipped with probability p to a specific semantically or structurally similar class.

5.2 Baselines

To evaluate the effectiveness of CG-GNN, we compare it with several representative baseline methods, including the standard GCN model and 9 robust approaches: Self-Training, Co-teaching, GRACE, NRGNN, RTGNN, ERASE, RNCGLN, GTA, and DREAM.

- **GCN** [35] utilizes graph convolution operations to iteratively aggregate feature information from local neighborhoods.
- **Self-Training** [48] employs a pseudo-labeling strategy that iteratively incorporates high-confidence predictions to enhance supervision.
- **Co-teaching** [16] adopts a dual-network collaborative learning paradigm originally proposed in computer vision and later adapted to graph data.
- **GRACE** [43] implements graph contrastive learning to improve robustness through node-level representation alignment.
- **NRGNN** [26] suppresses noise propagation by reconstructing graph structure.
- **RTGNN** [27] combines noisy-label correction with graph structure refinement to improve robust node classification.
- **ERASE** [21] improves robustness through error-resilient representation learning and encoding consistency maximization.
- **RNCGLN** [40] simultaneously addresses structural noise and label noise while stabilizing node representations with attention mechanisms.
- **GTA** [29] improves graph robustness through adaptive topology optimization with edge augmentation and pruning.
- **DREAM** [49] estimates label reliability via dynamic weighting based on semantic homophily.

Table 3: Classification accuracy (%) under 20% noise rate for Uniform and Pair noise (mean $\pm$ std).

Method	Cora		Citeseer		PubMed		Photo	
	Uniform	Pair	Uniform	Pair	Uniform	Pair	Uniform	Pair
GCN	72.8 $\pm$ 1.0	74.5 $\pm$ 0.2	64.9 $\pm$ 1.7	60.3 $\pm$ 1.0	73.8 $\pm$ 1.2	74.5 $\pm$ 2.1	86.7 $\pm$ 0.5	85.4 $\pm$ 2.0
Self-Training	75.6 $\pm$ 0.8	75.5 $\pm$ 2.2	66.3 $\pm$ 0.5	65.9 $\pm$ 1.1	77.4 $\pm$ 1.2	76.6 $\pm$ 1.5	87.2 $\pm$ 0.7	86.2 $\pm$ 1.0
Co-teaching+	77.0 $\pm$ 0.3	76.7 $\pm$ 0.8	70.6 $\pm$ 1.0	69.3 $\pm$ 1.3	78.6 $\pm$ 0.4	78.5 $\pm$ 0.1	88.2 $\pm$ 1.9	87.1 $\pm$ 0.4
GRACE	71.6 $\pm$ 1.8	72.2 $\pm$ 2.4	63.6 $\pm$ 1.6	65.6 $\pm$ 2.3	63.6 $\pm$ 1.6	65.6 $\pm$ 2.3	88.3 $\pm$ 0.9	86.5 $\pm$ 1.3
NRGNN	80.4 $\pm$ 1.2	79.5 $\pm$ 0.8	68.1 $\pm$ 0.3	68.2 $\pm$ 0.6	79.1 $\pm$ 0.4	78.5 $\pm$ 1.7	88.4 $\pm$ 1.2	86.8 $\pm$ 1.1
RTGNN	80.8 $\pm$ 0.2	80.9 $\pm$ 0.3	70.1 $\pm$ 2.1	69.8 $\pm$ 1.2	<u>80.9$\pm$1.0</u>	79.6 $\pm$ 1.1	89.3 $\pm$ 0.9	87.9 $\pm$ 2.7
ERASE	80.7 $\pm$ 1.0	81.9$\pm$0.8	69.9 $\pm$ 1.7	69.7 $\pm$ 2.8	80.2 $\pm$ 0.4	<u>80.2$\pm$0.6</u>	88.5 $\pm$ 1.1	88.3 $\pm$ 1.9
RNCGLN	<u>81.0$\pm$0.7</u>	80.4 $\pm$ 0.3	70.3 $\pm$ 0.4	69.2 $\pm$ 0.1	78.9 $\pm$ 0.3	78.5 $\pm$ 0.3	<u>89.4$\pm$0.4</u>	<u>88.7$\pm$0.5</u>
GTA	79.9 $\pm$ 2.0	80.8 $\pm$ 2.5	68.5 $\pm$ 3.7	69.1 $\pm$ 3.5	79.7 $\pm$ 4.6	77.3 $\pm$ 4.2	88.4 $\pm$ 3.5	87.6 $\pm$ 4.0
DREAM	80.9 $\pm$ 1.5	80.4 $\pm$ 0.4	<u>71.5$\pm$0.6</u>	<u>69.9$\pm$0.6</u>	80.7 $\pm$ 0.5	79.4 $\pm$ 0.5	89.1 $\pm$ 0.9	87.8 $\pm$ 0.8
CG-GNN	81.6$\pm$1.0	<u>81.7$\pm$0.9</u>	72.3$\pm$0.4	71.9$\pm$0.5	82.1$\pm$1.0	81.6$\pm$0.6	90.1$\pm$0.4	90.3$\pm$0.3

5.3 Experimental settings

All experiments are conducted on an NVIDIA A100-PCIE-40GB GPU with 40GB memory. A two-layer GCN is adopted as the backbone GNN model for f_{θ_1} and f_{θ_2} , with a hidden dimension of 128. The learning rate is set to 0.001, and the weight decay is fixed at 5×10^{-4} . The total number of training epochs is 200, and the warm-up stage lasts for $T_w = 50$ epochs. The edge reconstruction coefficient α is selected from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, while β and λ are chosen from $\{0.01, 0.05, 0.1, 0.2, 0.3\}$. The temperature parameter τ for contrastive learning is tuned within $\{0.001, 0.01, 0.1, 1, 10, 100\}$. The homophily consistency threshold th_{hm} is set to 0.5, and the edge pruning threshold th_{prune} is set to 0.4. Under the semi-supervised setting, the label rates for Cora and Citeseer are fixed at 5%, while those for PubMed and Photo are set to 1%. Unless otherwise specified, the default label noise rate is 20%. The community size threshold N_m is set to 5 for Cora and Citeseer, and 10 for PubMed and Photo. All experiments are repeated five times under identical parameter settings, and the average performance along with the standard deviation is reported.

5.4 Results analysis

5.4.1 Overall performance comparison

To comprehensively evaluate the overall performance of CG-GNN on noisy graph datasets, we conduct comparative experiments on Cora, Citeseer, PubMed, and Photo, against 9 representative methods. These baselines include both conventional graph neural network models and recently proposed robust graph learning approaches. All experiments are performed under identical data splits and noise configurations to ensure a fair comparison. As shown in Table 3, the performance of different methods is compared on four datasets under different noise settings with the noise rate fixed at 20%.

Table 3 show that CG-GNN consistently achieves the best or second-best classification performance across all datasets and noise settings. In most configurations,

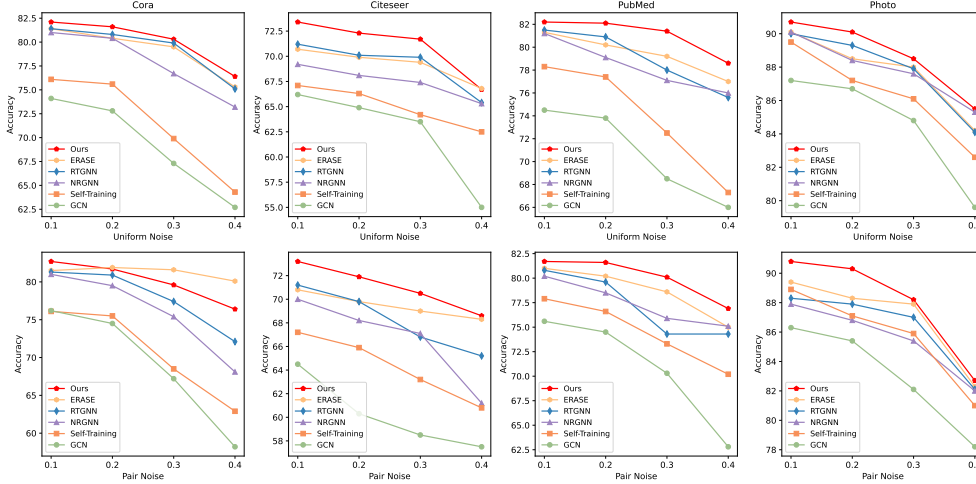


Fig. 4: Evaluating the performance of different methods under varying levels of label noise, where the subplots report results under Uniform and Pair noise settings across different datasets.

it significantly outperforms the competing methods. Notably, on datasets with limited labeled nodes, such as PubMed and Photo, CG-GNN exhibits more pronounced advantages, demonstrating strong stability and generalization ability.

A closer examination of different methods reveals that the standard GCN model suffers substantial performance degradation in the presence of label noise, indicating its sensitivity to noise propagation. Although several robust approaches partially alleviate the negative impact of noisy labels, many of them still rely heavily on homophily assumptions or explicit node-level noise modeling. In contrast, CG-GNN integrates community-level noise governance with a staged contrastive learning strategy, effectively suppressing noise diffusion while jointly exploiting structural and semantic information. This design leads to more stable node representation learning. Overall, the performance comparison results demonstrate that CG-GNN maintains strong adaptability across datasets of varying scales and structural characteristics, validating its effectiveness in realistic noisy graph learning scenarios.

5.4.2 Impact of different noise rates

To systematically evaluate the robustness of CG-GNN under varying noise conditions, we conduct comparative experiments under both uniform and pair noise settings, while progressively increasing the noise ratio to examine model performance in different scenarios.

As shown in Figure 4, the experimental results indicate that as the noise rate increases, the classification accuracy of all methods exhibits varying degrees of performance degradation. However, compared with competing approaches, CG-GNN consistently demonstrates a slower decline in performance across different noise levels. This advantage becomes more pronounced under moderate to high noise rates. These

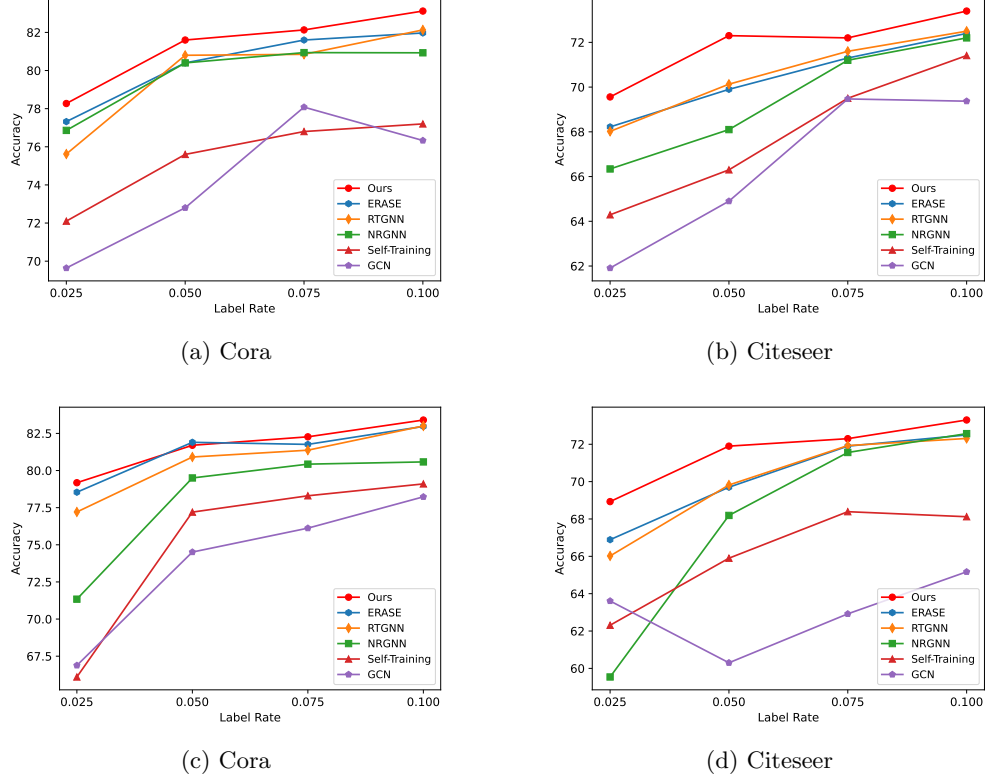


Fig. 5: Performance comparison of different methods under varying label rates on Cora and Citeseer with Uniform and Pair noise settings at a fixed noise rate of 20%. where (a) and (b) present the results under uniform noise, (c) and (d) present the results under pair noise.

findings suggest that CG-GNN effectively mitigates the propagation of noisy labels through graph structures and reduces the risk of overfitting to corrupted supervision during training. In contrast, several baseline methods suffer substantial performance deterioration at higher noise levels, highlighting their limited robustness in challenging noisy environments. Overall, the results confirm that CG-GNN maintains stable and reliable performance across a wide range of noise intensities.

5.4.3 Impact of different label rates

To evaluate model performance under varying supervision levels, we conduct experiments on Cora and Citeseer with different label rates. Specifically, the proportion of labeled nodes is set to 2.5%, 5%, 7.5%, and 10%, and comparisons are performed under both Uniform and Pair noise settings against multiple robust baselines. As shown in Figure 5, CG-GNN consistently maintains stable and competitive classification performance across almost all label rate configurations. The advantage is particularly evident

Table 4: Ablation study under 20% noise rate (mean $\pm$ std).

Dataset	Variant	Uniform	Pair
Cora	w/o AP	78.4 $\pm$ 2.6	77.7 $\pm$ 1.9
	w/o NC	80.1 $\pm$ 1.7	80.4 $\pm$ 0.9
	w/o CL	80.5 $\pm$ 0.6	80.3 $\pm$ 1.2
	All	81.6$\pm$1.0	81.7$\pm$0.9
Citeseer	w/o AP	68.1 $\pm$ 3.8	67.3 $\pm$ 2.1
	w/o NC	71.4 $\pm$ 0.5	71.3 $\pm$ 0.6
	w/o CL	71.7 $\pm$ 0.5	71.5 $\pm$ 1.1
	All	72.3$\pm$0.4	71.9$\pm$0.5
PubMed	w/o AP	80.1 $\pm$ 1.6	79.5 $\pm$ 0.7
	w/o NC	80.7 $\pm$ 0.5	80.1 $\pm$ 0.9
	w/o CL	80.0 $\pm$ 0.5	79.1 $\pm$ 0.4
	All	82.1$\pm$1.0	81.6$\pm$0.6
Photo	w/o AP	87.7 $\pm$ 2.7	87.5 $\pm$ 2.2
	w/o NC	88.2 $\pm$ 1.0	89.1 $\pm$ 0.5
	w/o CL	88.6 $\pm$ 1.0	87.5 $\pm$ 0.6
	All	90.1$\pm$0.4	90.3$\pm$0.3

in low-label scenarios, such as 2.5% and 5%, where supervision is severely limited. Under such conditions, many competing methods experience substantial performance degradation, and some even struggle to converge effectively. In contrast, CG-GNN preserves relatively high classification accuracy. These observations indicate that CG-GNN does not overly rely on abundant clean labels. Instead, it effectively leverages structural topology and semantic information, even when labeled supervision is sparse and potentially corrupted. Consequently, the proposed framework mitigates the performance degradation typically caused by insufficient supervision and demonstrates strong adaptability under limited-label settings.

5.4.4 Ablation study

To systematically evaluate the contribution of each key component in CG-GNN, we conduct ablation experiments on Cora, Citeseer, PubMed, and Amazon Photo. By selectively removing or modifying core modules of the framework, we analyze the role of each component in improving robustness and representation learning. As shown in Table 4, the following variants are considered:

- **w/o AP** removes the augmented graph construction and noisy edge pruning module, training the model solely on the original graph structure.
- **w/o NC** removes the noise detection module, eliminating the process of noisy node identification and partitioning.
- **w/o CL** removes the unsupervised structural contrastive learning stage, retaining only the supervised semantic contrastive learning component.

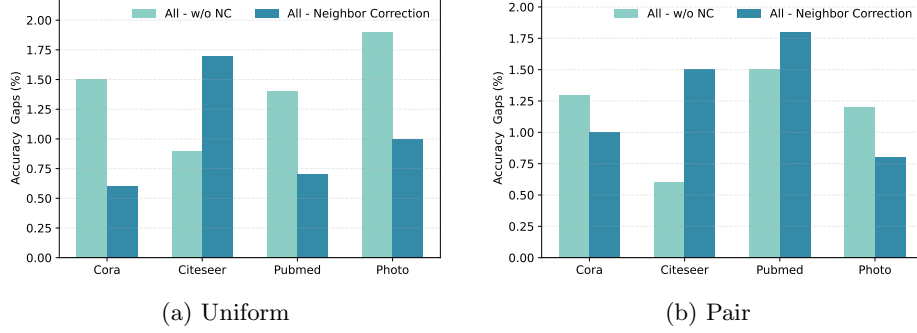


Fig. 6: The accuracy gaps between the full model and the variant without noise correction, as well as between the full model and the variant using only neighbor-based correction, on the Cora, Citeseer, PubMed, and Photo datasets.

Across all datasets and noise types, the full model consistently achieves the best performance. Removing any single component leads to noticeable performance degradation, confirming that each module contributes to the overall robustness of the framework. In particular, eliminating augmented graph construction (w/o AP) results in substantial accuracy drops under both Uniform and Pair noise, highlighting the importance of topology refinement in suppressing noise propagation. Similarly, removing noise detection (w/o NC) weakens the model’s ability to distinguish clean from noisy nodes, while removing structural contrastive learning (w/o CL) reduces representation stability under noisy supervision. Overall, the ablation results demonstrate that the three key components—graph enhancement and pruning, noise detection, and staged contrastive learning—work collaboratively to improve performance under noisy conditions.

To further investigate the effectiveness of the noise correction module, we present a more detailed ablation analysis under uniform and pair noise settings, as shown in Figure 6. Specifically, we compare the full model with two variants: one without noise correction (All - w/o NC) and another using only neighbor-based correction (All - Neighbor Correction). The former evaluates the overall contribution of the noise correction module, while the latter measures the additional gain brought by community-based correction.

The full model consistently achieves higher accuracy than the variant without noise correction on all datasets, confirming the necessity of label correction under noisy supervision. In addition, the full model also outperforms the variant with only neighbor-based correction, indicating that community-based correction is able to provide more reliable structural guidance than purely local correction in challenging noisy environments.

5.4.5 Visualization analysis

To visually assess the effectiveness of CG-GNN in node representation learning, we employ t-SNE [50] to project the learned node embeddings into a two-dimensional

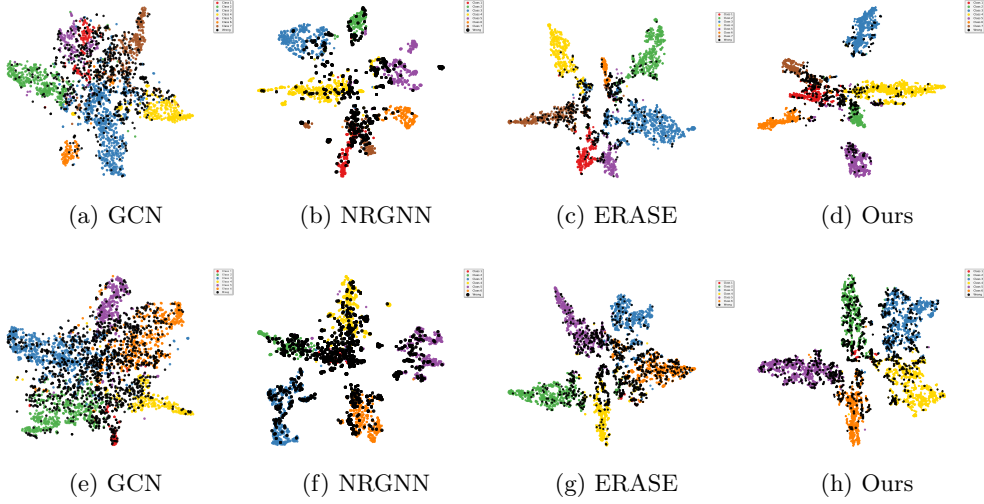


Fig. 7: Visualizing the node embeddings learned by different methods on the Cora and Citeseer datasets using t-SNE. Nodes of different classes are shown in different colors, while misclassified nodes are marked in black.

space on the Cora and Citeseer datasets. The visualization results are compared with representative baseline methods, including GCN, NRGNN, and ERASE. In the plots, different colors correspond to different classes, while misclassified nodes are marked in black to highlight the model’s discriminative capability under noisy conditions.

The visualization results are shown in Figure 7. From the overall distribution, the node representations learned by CG-GNN exhibit a clearer clustering structure. Nodes belonging to the same class are more tightly grouped in the embedding space, indicating enhanced intra-class compactness, while the separation between different classes becomes more distinct, resulting in sharper class boundaries. A comparison with baseline methods further highlights these differences. When label noise is present, the embeddings produced by the standard GCN model appear more scattered, with substantial overlap across multiple classes. Misclassified nodes are often concentrated near class boundaries, reflecting its sensitivity to noise propagation. Although NRGNN and ERASE alleviate this issue to some extent, their embeddings still show fragmented clusters and discontinuities within certain communities affected by noisy nodes. In contrast, CG-GNN demonstrates stronger internal consistency within communities. Benefiting from the community-aware noise correction strategy, the model preserves semantic structure in high-purity communities while actively restoring semantic consistency in high-entropy communities corrupted by noise. As a result, nodes within the same community form more stable and coherent clusters in the representation space. Furthermore, the staged contrastive learning mechanism encourages the model to first capture topological consistency in the early training phase and subsequently incorporate high-confidence semantic supervision. This progressive learning strategy

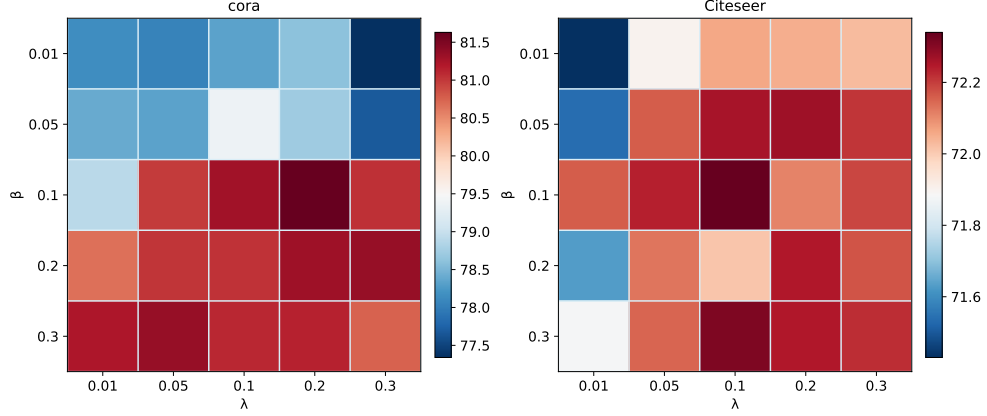


Fig. 8: The performance of CG-GNN to different combinations of λ and β on the Cora and Citeseer datasets.

enables the final embeddings to achieve both strong structural discrimination and clear semantic separability.

5.4.6 Hyperparameter analysis

To further investigate the impact of key hyperparameters in CG-GNN, we conduct systematic sensitivity analysis on the Cora, Citeseer, PubMed, and Amazon Photo datasets. The analysis focuses on the reconstruction loss weight α , the noisy-node supervision weight β , the contrastive loss coefficient λ , and the temperature parameter τ . The results are illustrated in Figures 8, Figures 9 and Figure 10.

The parameter β controls the weighting of noisy nodes in the prediction loss, while λ balances the contribution of the contrastive objective in the overall loss. As shown in the heatmaps for Cora and Citeseer, model performance varies noticeably across different combinations of β and λ , indicating a certain degree of sensitivity to these parameters. When β is too small, the suppression of noisy nodes is insufficient; conversely, excessively large values may overly weaken potentially useful information from mislabeled nodes. Empirical results suggest that moderate values in the range of 0.1–0.2 generally yield stable and competitive performance. Similarly, inappropriate choices of λ may either lead to underemphasis of contrastive learning or cause the model to overfocus on representation alignment at the expense of classification objectives. Overall, the results demonstrate that CG-GNN maintains robust performance within a relatively broad range of hyperparameter settings, indicating stable optimization behavior and practical applicability.

The reconstruction loss weight α controls the contribution of $\mathcal{L}_{rec}$ to the overall objective. As shown in Figure 9, the performance on both Citeseer and PubMed first improves and then declines as α increases, indicating that the reconstruction loss should be properly balanced with the classification objective. Compared with PubMed, Citeseer is more sensitive to α , with more noticeable performance fluctuations under both Uniform and Pair noise settings. In contrast, PubMed shows relatively smaller

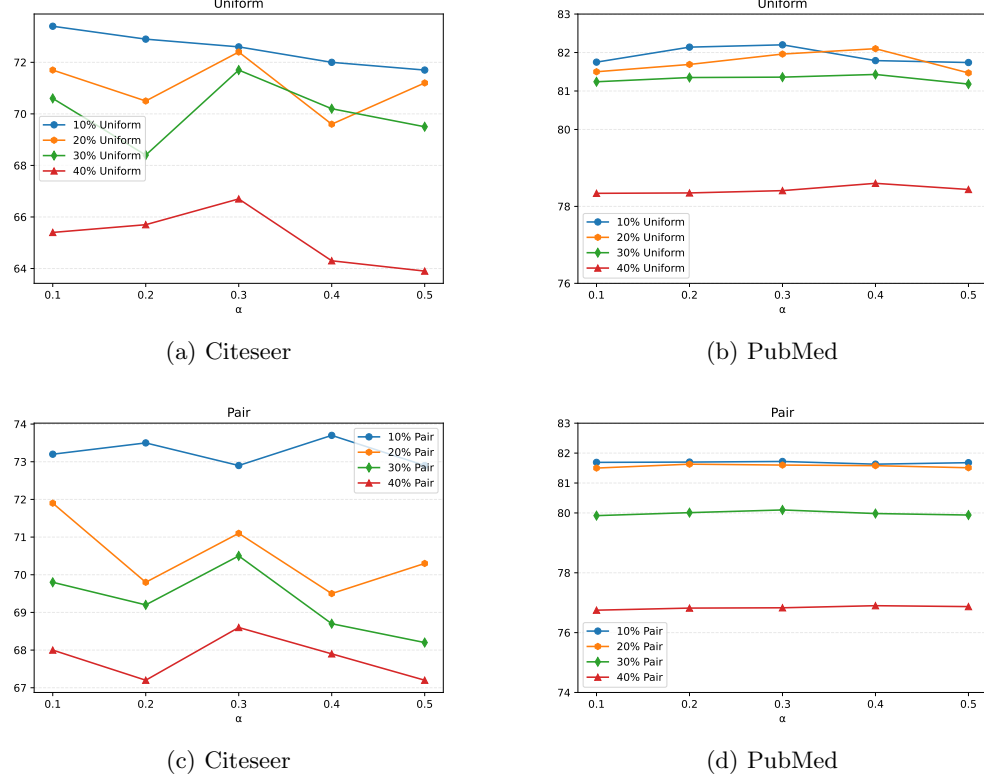


Fig. 9: The performance of CG-GNN under different values of α on Citeseer and PubMed datasets, where the curves report results under Uniform and Pair noise settings.

variations, suggesting that large-scale datasets are less sensitive to the reconstruction loss weight and can maintain more stable performance over a wider range of α .

Figure 10 illustrates the performance variation on the Cora and Amazon Photo datasets as the temperature parameter τ ranges from 0.001 to 100. The results show that the two datasets exhibit different sensitivities to τ under both Uniform and Pair noise settings. When τ is very small, Cora achieves relatively better performance, whereas Amazon Photo performs poorly. As τ increases to a moderate range, the performance gap between different settings becomes smaller. However, when τ becomes excessively large, the performance on Cora drops sharply, while the variation on Amazon Photo remains relatively mild. These observations suggest that Cora is more sensitive to the temperature parameter, whereas Amazon Photo is comparatively more stable over a wider range of τ .

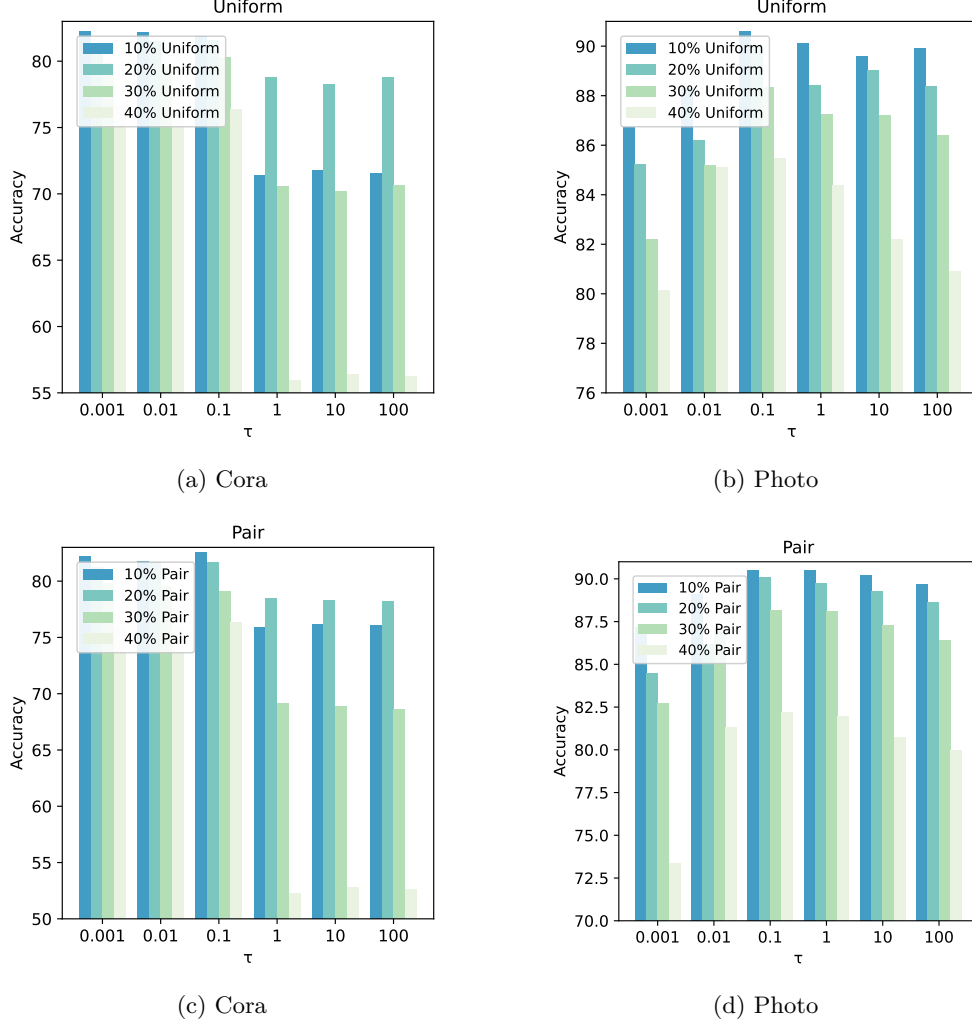


Fig. 10: Evaluating the effect of the temperature parameter τ on Cora and Amazon Photo datasets under Uniform and Pair noise settings.

6 Conclusion

In this paper, we propose a community-governed robust graph neural network framework, termed CG-GNN, to alleviate the performance degradation of GNNs under label noise and sparse supervision signals by incorporating community-level structural information from a global perspective to enhance representation stability and robustness in noisy environments. Extensive experiments demonstrate that CG-GNN consistently outperforms SOTA across multiple real-world benchmark datasets, various noise types, and different noise ratios. The results validate the effectiveness and generalization

capability of the proposed framework in complex noisy scenarios. In future work, we plan to extend CG-GNN to more challenging settings, such as graph data simultaneously affected by both label noise and structural noise, in order to further assess its applicability in realistic and large-scale graph learning applications.

Acknowledgement

This work is supported in part by the Guangxi Science and Technology Project (2025GXNSFFA069015 and GuikeAA22068070), the National Natural Science Foundation of China (No. U21A20474 and No. 62262003), and the Innovation Project of GUET Graduate Education, China (2025YCXS044).

References

- [1] Yan, J., Pu, P., Jiang, L.: Emotion-rgc net: A novel approach for emotion recognition in social media using roberta and graph neural networks. *Plos one* **20**(3), 0318524 (2025)
- [2] Sharma, K., Lee, Y.-C., Nambi, S., Salian, A., Shah, S., Kim, S.-W., Kumar, S.: A survey of graph neural networks for social recommender systems. *ACM Computing Surveys* **56**(10), 1–34 (2024)
- [3] Li, X., Wang, X., Lin, Y.: Graph neural network enhanced sequential recommendation method for cross-platform ad campaigns. In: 2025 5th International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), pp. 502–506 (2025). IEEE
- [4] Wang, B., Chen, J., Li, C., Zhou, S., Shi, Q., Gao, Y., Feng, Y., Chen, C., Wang, C.: Distributionally robust graph-based recommendation system. In: Proceedings of the ACM Web Conference 2024, pp. 3777–3788 (2024)
- [5] Sim, J., Kim, D., Kim, B., Choi, J., Lee, J.: Recent advances in ai-driven protein-ligand interaction predictions. *Current opinion in structural biology* **92**, 103020 (2025)
- [6] La Rosa, M., Fiannaca, A., La Paglia, L., Urso, A.: A graph neural network approach for the analysis of sirna-target biological networks. *International Journal of Molecular Sciences* **23**(22), 14211 (2022)
- [7] Ghanbari, M., Karimi, M., Claramunt, C., Lagesse, C.: Enhancing urban transportation flow modeling through a graph neural network-based spatially weighted interaction model: a case study of chicago taxi data. *Journal of Geovisualization and Spatial Analysis* **9**(1), 15 (2025)
- [8] Altaf, T., Wang, X., Ni, W., Yu, G., Liu, R.P., Braun, R.: Gnn-based network traffic analysis for the detection of sequential attacks in iot. *Electronics* **13**(12), 2274 (2024)

- [9] Lu, W., Guan, Z., Zhao, W., Yang, Y., Zhan, Y., Lu, Y., Tao, D.: Agmixup: Adaptive graph mixup for semi-supervised node classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, pp. 19143–19151 (2025)
- [10] Chen, X., Li, J., Huang, Y., Zhuang, J., Shi, C., Xu, Z., Fu, Y.-G.: Unitrain: A universal iterative semi-supervised training framework for graph representation learning. *Neural Networks*, 108623 (2026)
- [11] Duan, L., Chen, X., Liu, W., Liu, D., Yue, K., Li, A.: Structural entropy based graph structure learning for node classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 8372–8379 (2024)
- [12] Wang, Z., Sun, D., Zhou, S., Wang, H., Fan, J., Huang, L., Bu, J.: Noisygl: A comprehensive benchmark for graph neural networks under label noise. *Advances in neural information processing systems* **37**, 38142–38170 (2024)
- [13] Ju, W., Yi, S., Wang, Y., Xiao, Z., Mao, Z., Li, H., Gu, Y., Qin, Y., Yin, N., Wang, S., et al.: A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025)
- [14] Dai, E., Jin, W., Liu, H., Wang, S.: Towards robust graph neural networks for noisy graphs with sparse labels. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 181–191 (2022)
- [15] Yuan, J., Luo, X., Qin, Y., Mao, Z., Ju, W., Zhang, M.: Alex: Towards effective graph transfer learning with noisy labels. In: Proceedings of the 31st ACM International Conference on Multimedia, pp. 3647–3656 (2023)
- [16] Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems* **31** (2018)
- [17] Li, J., Socher, R., Hoi, S.C.H.: Dividemix: Learning with noisy labels as semi-supervised learning. In: International Conference on Learning Representations (2020)
- [18] Li, Y., Yin, J., Chen, L.: Unified robust training for graph neural networks against label noise. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 528–540 (2021). Springer
- [19] Xia, J., Lin, H., Xu, Y., Tan, C., Wu, L., Li, S., Li, S.Z.: Gnn cleaner: Label cleaner for graph structured data. *IEEE Transactions on Knowledge and Data Engineering* **36**(2), 640–651 (2023)
- [20] Du, X., Bian, T., Rong, Y., Han, B., Liu, T., Xu, T., Huang, W., Li, Y., Huang, J.: Noise-robust graph learning by estimating and leveraging pairwise interactions.

- [21] Chen, L.-H., Zhang, Y., Huang, T., Su, L., Lin, Z., Xiao, X., Xia, X., Liu, T.: Erase: Error-resilient representation learning on graphs for label noise tolerance. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pp. 270–280 (2024)
- [22] Wang, Y., Yang, Y.: Bayesian robust graph contrastive learning. arXiv preprint arXiv:2205.14109 (2022)
- [23] Yuan, J., Luo, X., Qin, Y., Zhao, Y., Ju, W., Zhang, M.: Learning on graphs under label noise. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2023). IEEE
- [24] Lu, Z., Ma, J., Wu, Z., Zhou, B., Zhu, X.: A noise-resistant graph neural network by semi-supervised contrastive learning. Information Sciences **658**, 120001 (2024)
- [25] Li, X., Li, Q., Qian, H., Wang, J., *et al.*: Contrastive learning of graphs under label noise. Neural networks **172**, 106113 (2024)
- [26] Dai, E., Aggarwal, C., Wang, S.: Nrgnn: Learning a label noise resistant graph neural network on sparsely and noisily labeled graphs. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 227–236 (2021)
- [27] Qian, S., Ying, H., Hu, R., Zhou, J., Chen, J., Chen, D.Z., Wu, J.: Robust training of graph neural networks via noise governance. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 607–615 (2023)
- [28] Li, S., Zhang, B., Song, J., Ruan, G., Wang, C., Xie, J.: Graph neural networks with coarse-and fine-grained division for mitigating label noise and sparsity. Neural Networks **187**, 107338 (2025)
- [29] Zhou, M., Han, X., Wei, W., Yan, G.: Graph topology adaptive judgment against node label noise. Knowledge-Based Systems, 114162 (2025)
- [30] Wu, Y., Yao, J., Xia, X., Yu, J., Wang, R., Han, B., Liu, T.: Mitigating label noise on graphs via topological sample selection. In: Forty-first International Conference on Machine Learning (2024)
- [31] Zheng, Z., Yang, Y., Guan, Z., Zhao, W., Lu, W.: Enhancing homophily-heterophily separation: Relation-aware learning in heterogeneous graphs. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, pp. 4050–4061 (2025)
- [32] Su, M., Wang, J., Li, Y., Cai, J., Yang, H., He, Y., Wang, W.: M-rngn: A robust

gnn based on m-estimators for semi-supervised node classification. Available at SSRN 5385769 (2025)

- [33] Luo, L., Fang, Y., Cao, X., Zhang, X., Zhang, W.: Detecting communities from heterogeneous graphs: A context path-based graph neural network model. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 1170–1180 (2021)
- [34] Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* **2008**(10), 10008 (2008)
- [35] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
- [36] Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. *Advances in neural information processing systems* **30** (2017)
- [37] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
- [38] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019)
- [39] Geisler, S.M., Kosmala, A., Herbst, D., Günnemann, S.: Spatio-spectral graph neural networks. *Advances in Neural Information Processing Systems* **37**, 49022–49080 (2024)
- [40] Zhu, Y., Feng, L., Deng, Z., Chen, Y., Amor, R., Witbrock, M.: Robust node classification on graph data with graph and label noise. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 17220–17227 (2024)
- [41] Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: International Conference on Learning Representations (2019)
- [42] Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: International Conference on Machine Learning, pp. 4116–4126 (2020). PMLR
- [43] Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Deep Graph Contrastive Representation Learning. In: ICML Workshop on Graph Representation Learning and Beyond (2020)
- [44] Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021, pp.

2069–2080 (2021)

- [45] Mo, Y., Peng, L., Xu, J., Shi, X., Zhu, X.: Simple unsupervised graph representation learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 7797–7805 (2022)
- [46] NT, H., Jin, C.J., Murata, T.: Learning graph neural networks with noisy labels. arXiv preprint arXiv:1905.01591 (2019)
- [47] Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representations (2017)
- [48] Li, Q., Han, Z., Wu, X.-M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
- [49] Zhao, Y., Xie, J., Zhang, Q., Zhang, W., Luo, X., Xiao, Z., Yu, P.S., Zhang, M.: Dream: Dual-standard semantic homogeneity with dynamic optimization for graph learning with label noise. ArXiv **abs/2601.17449** (2026)
- [50] Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)