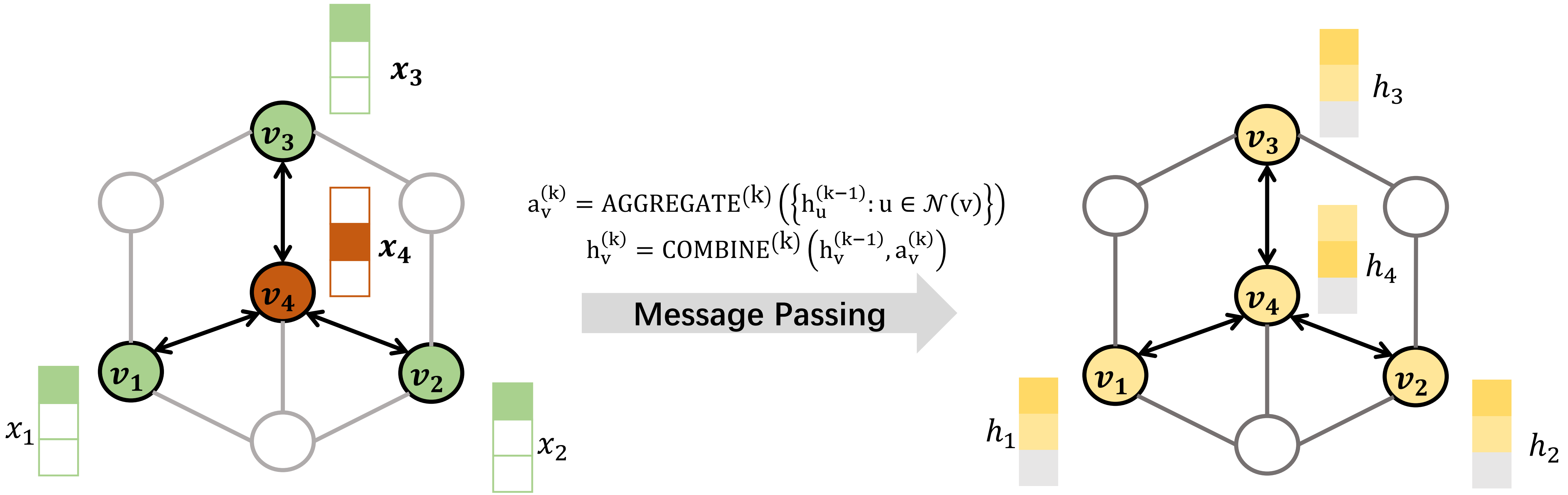
# NoisyGL: A Comprehensive Benchmark for Graph Neural Networks under Label Noise

**Zhonghao Wang** [1], Danyu Sun [1], Sheng Zhou [1] *, Haobo Wang [1], Jiapei Fan [2], Longtao Huang [2] and Jiajun Bu [2]
[1] Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems,
Collage of Computer Science, Zhejiang University
[2] Alibaba Group

# Graph Neural Networks

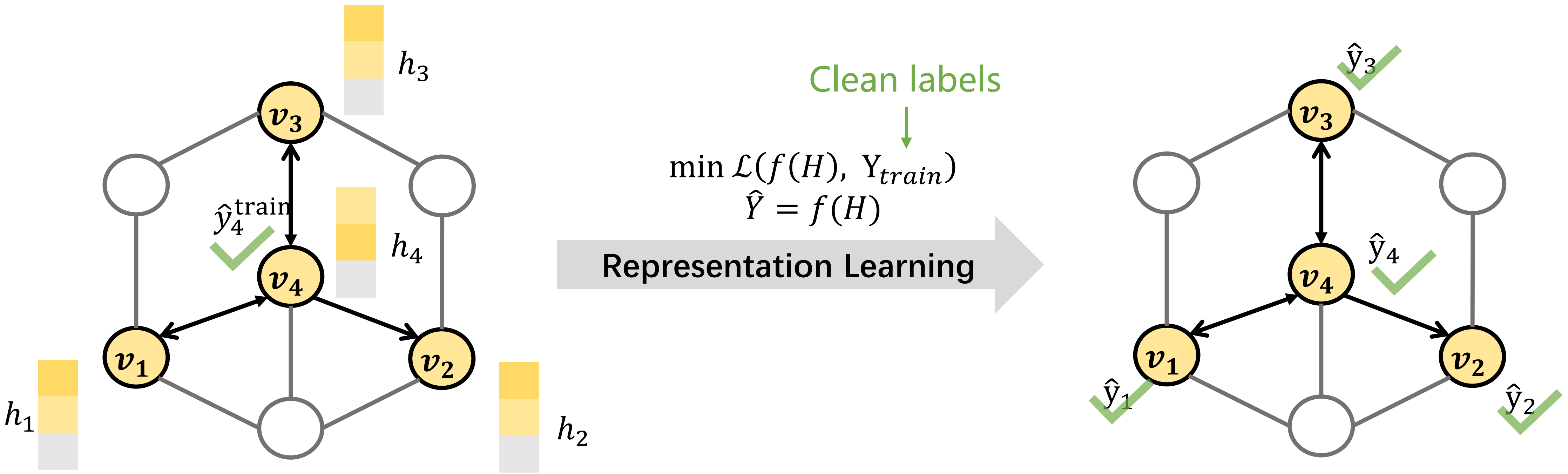
- Graph Neural Networks (GNNs) demonstrate strong potential in node classification tasks through the **message-passing** mechanism.
- The **message-passing** mechanism aggregates information from neighboring nodes, resulting in **similar (homophily) representations** for adjacent nodes.



# Background

## Graph Neural Networks

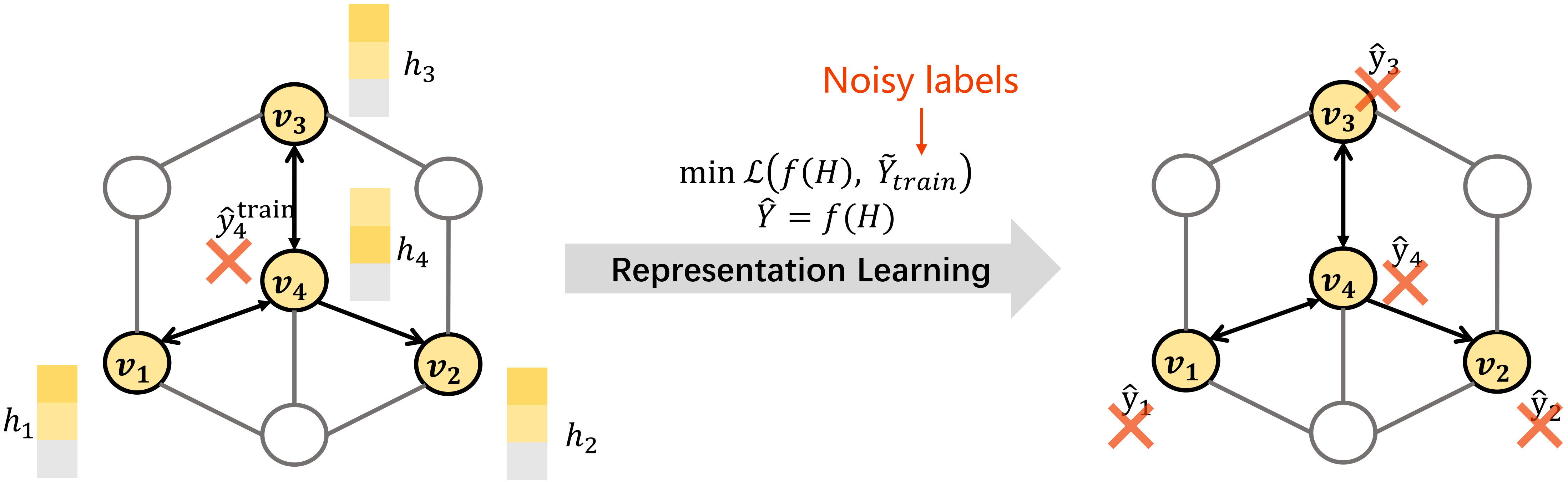
- Using **semi-supervised** learning to develop homophily node representations, GNNs can effectively generalize patterns from labeled training nodes to neighboring unlabeled nodes.
- Thus, GNNs can **perform well** on node-level tasks **with relatively few labels**.



# Background

## Graph Neural Networks

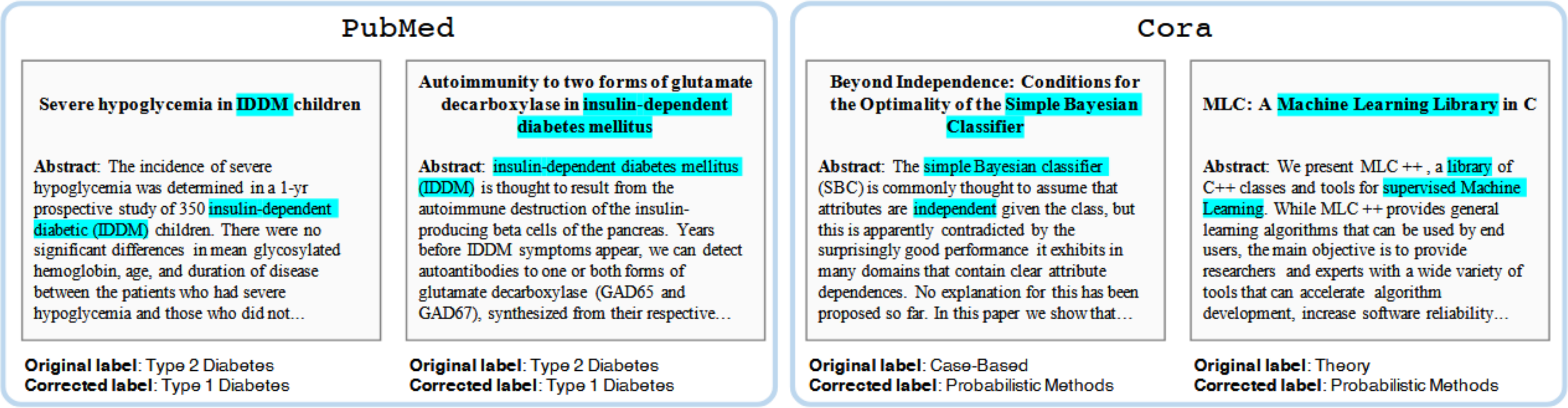
- However, if the **training labels are incorrect**, it can cause the unlabeled nodes with similar representations in the neighborhood to **be incorrectly learned together**.
- Therefore, node classification tasks using **GNNs rely on high-quality node labels**.



# Background

## Graph Label Noise

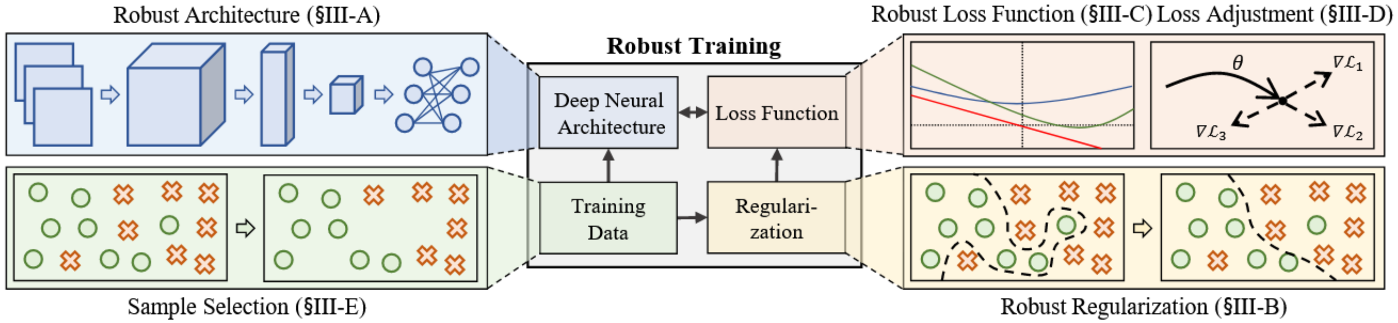
- Graphs are inherently abstract high-dimensional data that are **difficult for humans to understand**, and labeling methods cannot be universally applied to different graph data.
- According to Li et al.[1], at least 6.91% of labels in the PubMed dataset are incorrect.




[1] Li, Yuwen, Miao Xiong, and Bryan Hooi. "Graphcleaner: Detecting mislabelled samples in popular graph learning benchmarks." International Conference on Machine Learning. PMLR, 2023.

# Challenges when dealing with graph label noise

➢ To address this challenge, an intuitive solution is to draw on the success of previous Learning with Label Noise (LLN)[2] strategies and apply them to GNNs.



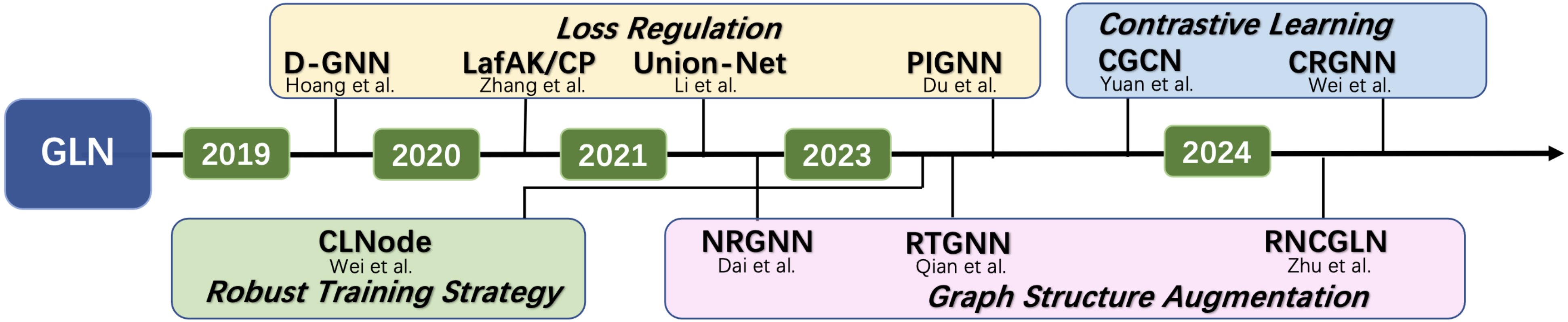

➢ However, these approaches are not always applicable to graph learning tasks due to:

1. Non-i.i.d nature of graph data.
2. Sparse labeling of graph data
3. Message-passing mechanism of GNNs.

[2] Song H, Kim M, Park D, et al. Learning from noisy labels with deep neural networks: A survey[J]. IEEE transactions on neural networks and learning systems, 2022, 34(11): 8135-8153.

## GNN with Label Noise

- To achieve robust graph learning, researchers have proposed a series of GNN with Label Noise (GLN) methods.
- However, these works use different experimental settings in their benchmarks (Dataset selection, Data splitting strategy, Noise type, Noise rate, etc.)



# NoisyGL-Introduction

## NoisyGL: A Comprehensive Benchmark for Graph Neural Networks under Label Noise

- We introduce NoisyGL: A comprehensive benchmark for graph neural networks under label noise.
- NoisyGL enables **fair comparisons** and **detailed analyses** of GLN methods on noisy labeled graph data across **various datasets**, with **unified experimental settings** and interface.
- Our benchmark has uncovered several important **insights**, and we believe these findings will be **highly beneficial for future studies**.

NoisyGL

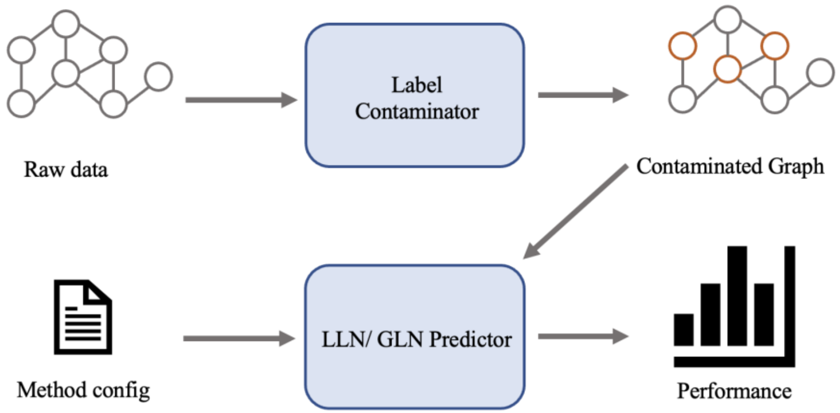



Table A7: Overview of the datasets used in this study.

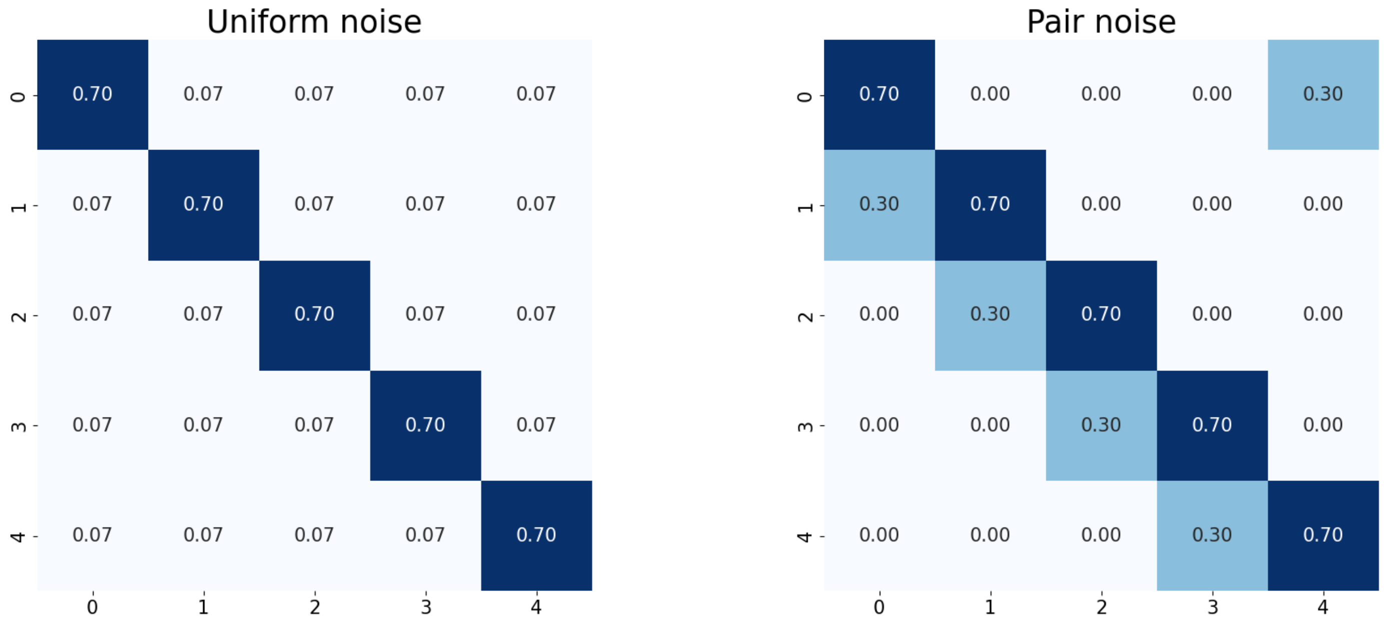
| Dataset | # Nodes | # Edges | # Feat. | # Classes | # Homophily | Avg. # degree |
|---|---|---|---|---|---|---|
| Cora | 2,708 | 5,278 | 1,433 | 7 | 0.81 | 3.90 |
| Citeseer | 3,327 | 4,552 | 3,703 | 6 | 0.74 | 2.74 |
| Pubmed | 19,717 | 44,324 | 500 | 3 | 0.80 | 4.50 |
| Amazon-Computers | 13,752 | 491,722 | 767 | 10 | 0.78 | 35.76 |
| Amazon-Photos | 7,650 | 238,162 | 745 | 8 | 0.83 | 31.13 |
| DBLP | 17,716 | 105,734 | 1,639 | 4 | 0.83 | 5.97 |
| BlogCatalog | 5,196 | 343,486 | 8,189 | 6 | 0.40 | 66.11 |
| Flickr | 7575 | 239738 | 12047 | 9 | 0.24 | 63.30 |

# NoisyGL-Introduction

## NoisyGL: A Comprehensive Benchmark for Graph Neural Networks under Label Noise

We mainly consider two types of label noise that were most commonly used:

- **Pair Noise:** Labels only flip to their corresponding pair class.
- **Uniform Noise:** Labels flip to any class with equal probability.



# NoisyGL-Research Questions

## Research Questions:

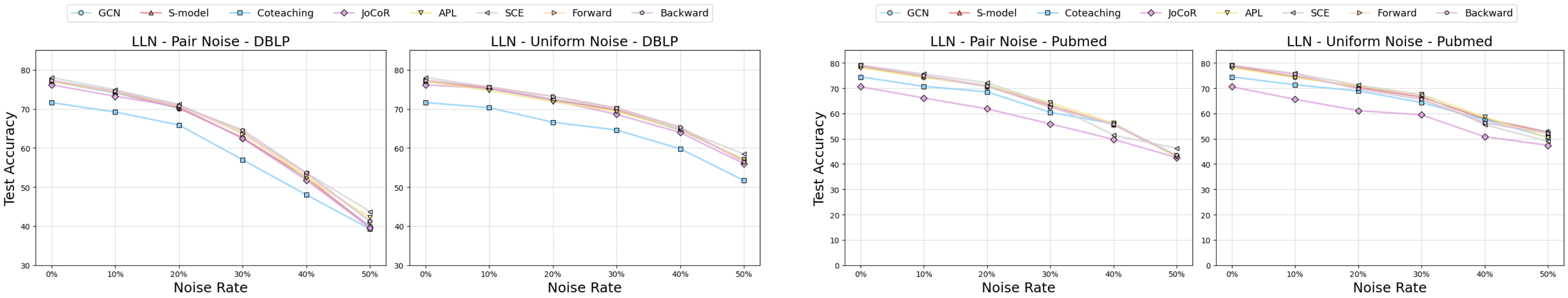
In this study, we aim to answer the following research questions:

- RQ1: Can LLN methods be applied directly to graph learning tasks?
- RQ2: How much progress has been made by existing GLN methods?
- RQ3: Are existing GLN methods computationally efficient?
- RQ4: Are existing GLN methods sensitive to noise rate?
- RQ5: Are existing GLN methods robust to different types of label noise?
- RQ6: Good or bad? Revisiting the role of graph structure in label noise.

# NoisyGL-Findings

## Finding 1: Most LLN methods do not significantly improve GNN robustness to label noise.

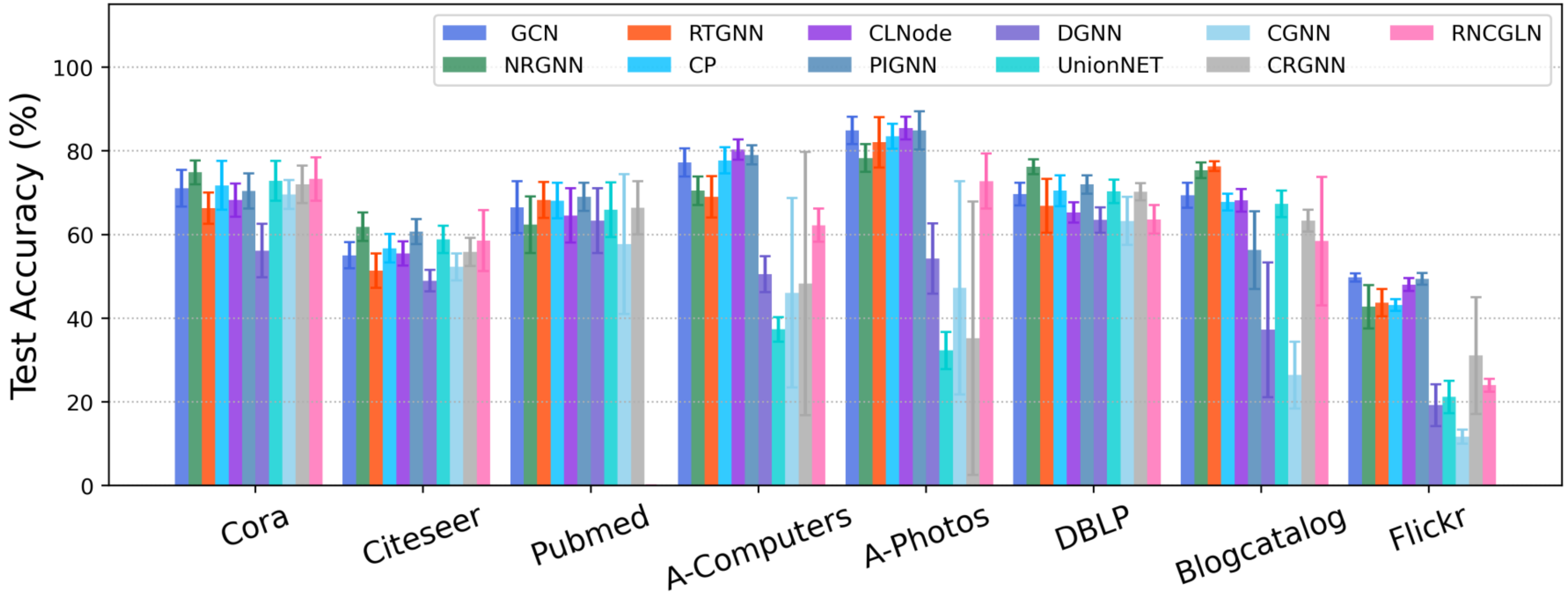
- Most of the selected **LLN methods do not substantially improve the performance** of the GNN backbone under label noise.



# NoisyGL-Findings

## Finding 2: Existing GLN methods can alleviate the negative impact of label noise in specific applicable scenarios.

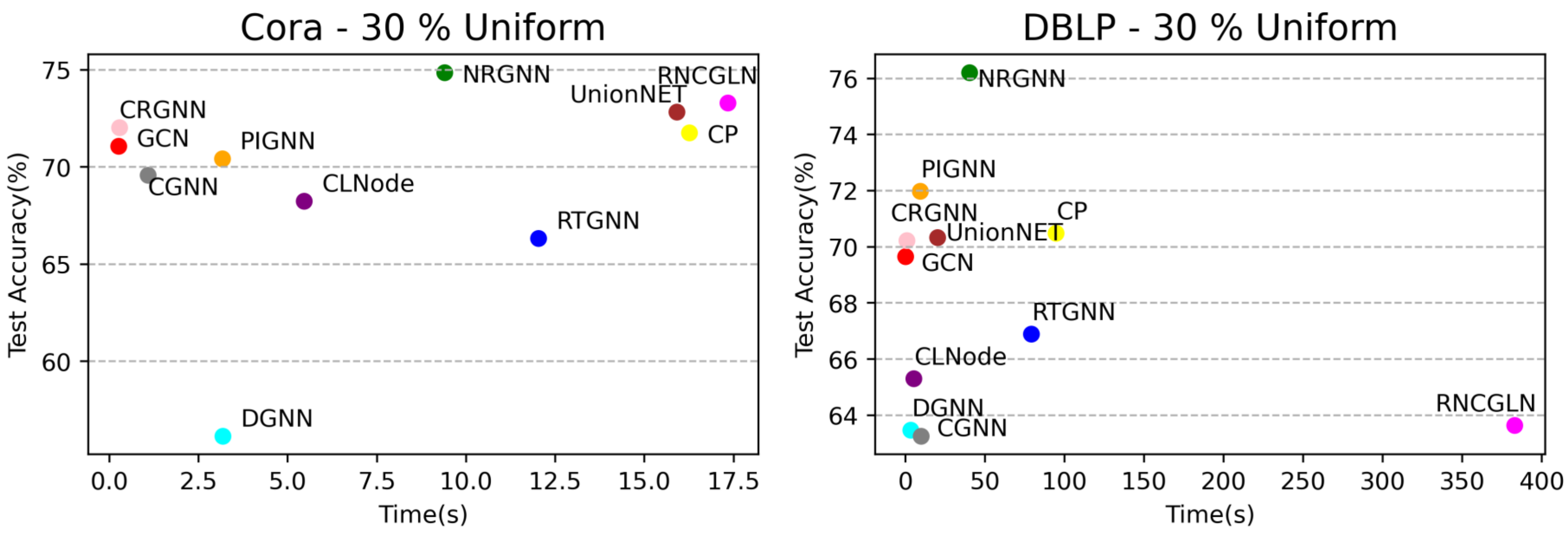
- In most circumstances, **GLN methods are more robust to label noise** than the baseline method. However, **none of them consistently perform well** across all datasets, especially on highly **heterophilous graph.**



# NoisyGL-Findings

## Finding 3: Some GLN methods are computationally inefficient.

- Some GLN methods take more time to converge.
- For instance, RNCGLN is the slowest, taking 66.8 times longer than GCN on the Cora dataset and an astounding 2945.8 times longer on the DBLP dataset.



# NoisyGL-Findings

## Finding 4:  Pair noise is more harmful to graph learning.

- In our experiments, we consistently observed that **pair noise poses the most significant threat** to the generalization ability of models.
- Our empirical study shows that pair noise has the greatest impact, leading the model to overfit the mislabeled classes.

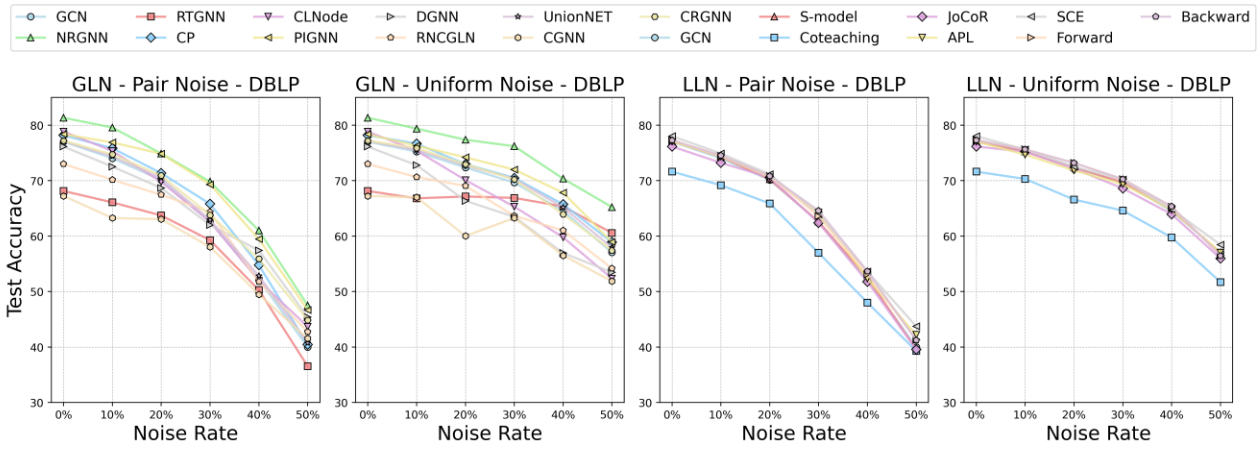



Table 1: Misleading train accuracy of GCN under pair and uniform noise (10 Runs)

| Noise type / Dataset (Avg. # Degree) | Cora (3.90) | Citeseer (2.74) | Pubmed (4.50) | A-Computers (35.76) | Blogcatalog (66.11) |
|---|---|---|---|---|---|
| 50% **Pair noise** | **92.86 ± 6.63** | **99.00 ± 2.23** | **91.23 ± 9.38** | **69.63 ± 12.93** | **71.67 ± 8.84** |
| 50% **Uniform noise** | 76.41 ± 6.52 | 96.36 ± 4.02 | 90.24 ± 18.34 | 22.05 ± 7.28 | 32.53 ± 12.61 |

# NoisyGL-Findings

## Finding 4: Pair noise is more harmful to graph learning.

- **Intuitively:** Pair noise flips labels to the pair class, which can be more misleading than uniform noise.
- **Formally:** A simplified GCN with ridge regression has the following closed-form solution[3]:

$$\hat{Y} = A^k X \left( \left( A^k X \right)^T \left( A^k X \right) + \lambda I \right)^{-1} \left( A^k X \right)^T Y_t = P Y_t$$

- Introducing label noise is equivalent to multiplying the prediction by a transition matrix: $\hat{Y}_{ptb} = P Y_t \mathrm{Q}$
- The difference in prediction results trained under clean and noisy labels:

$$\Delta Y = \| \hat{Y} - \hat{Y}_{ptb} \|_2 = \| P Y_t - P Y_t Q \|_2 = \| P Y_t (I - Q) \|_2$$

- The upper bound of prediction error can be expressed as:

$$\Delta Y \leq \| P \|_2 \cdot \| Y_t \|_2 \cdot \| I - Q \|_2$$

[3] Zhang, Mengmei, et al. "Adversarial label-flipping attack and defense for graph neural networks." 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020.

# NoisyGL-Findings

## Finding 4: Pair noise is more harmful to graph learning.

- The upper bound of prediction error can be expressed as:

$$\Delta Y \leq \|P\|_2 \cdot \|Y_t\|_2 \cdot \|I - Q\|_2,$$

- where $\|P\|_2 \cdot \|Y_t\|_2$ are fixed values and are independent of the noise model.
- Let the noise rate be $\epsilon$, for Uniform noise and Pair noise, $\|I - Q\|_2$ has different values:

$$\left\|I - Q_{uniform}\right\| = \sqrt{c\epsilon^2 + c(c-1)\left(-\frac{\epsilon}{c-1}\right)^2} = \epsilon\sqrt{c + \frac{c}{c-1}}$$

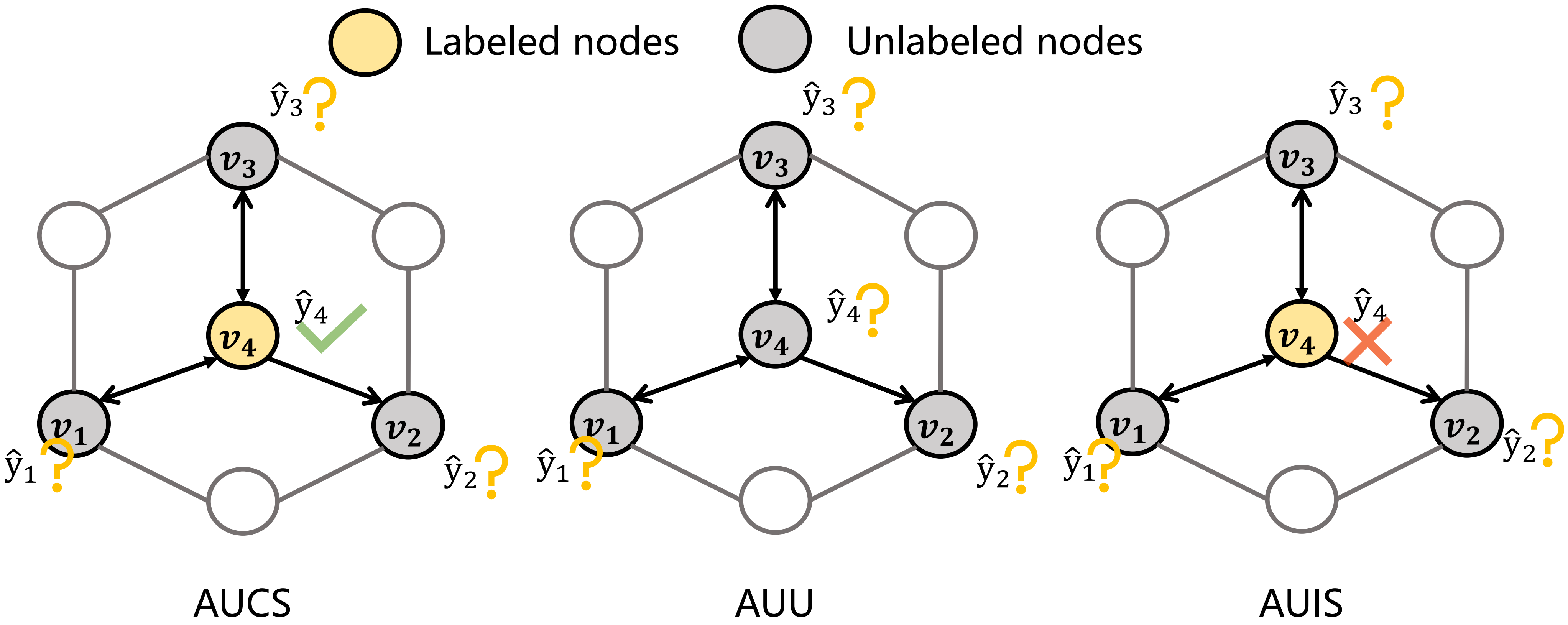
$$\left\|I - Q_{pair}\right\| = \sqrt{c\epsilon^2 + c\epsilon^2} = \sqrt{2c\epsilon^2} = \epsilon\sqrt{2c}$$

- In multi-class node classification tasks, where $c \geq 2$, we have $\left\|I - Q_{uniform}\right\| \leq \left\|I - Q_{pair}\right\|$。
- This proves that **the upper bound of the prediction error caused by pair noise is greater** than uniform noise, indicating that **pair noise is more disruptive**.

# NoisyGL-Findings

## Finding 5: Label noise can Propagate through graph topology.

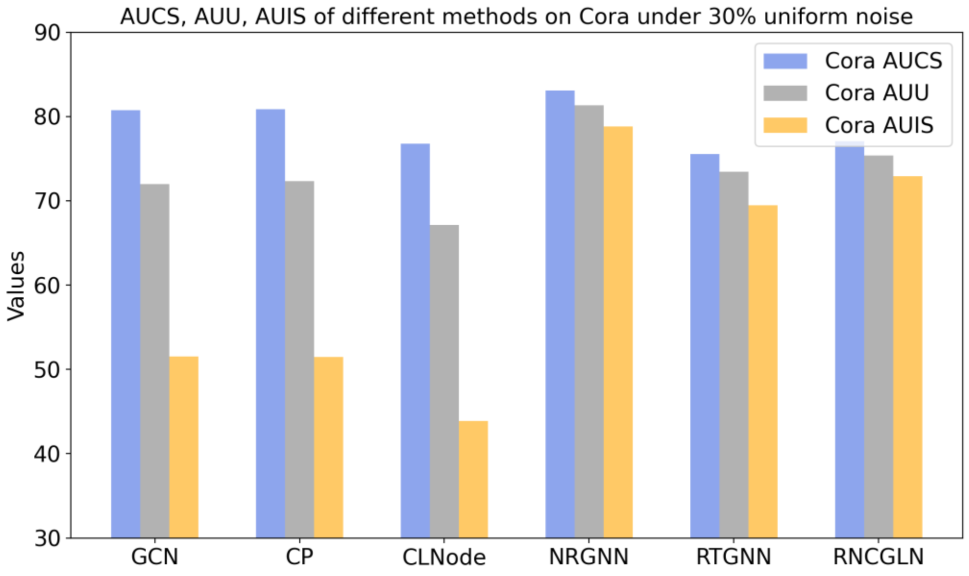
- **AUCS**, **AUU**, and **AUIS** represent the probability of **correctly classifying an unlabeled node** (test node) under three conditions: the presence of **correctly labeled neighboring nodes**, **no labeled neighboring nodes**, and the presence of **incorrectly labeled neighboring nodes**, respectively.

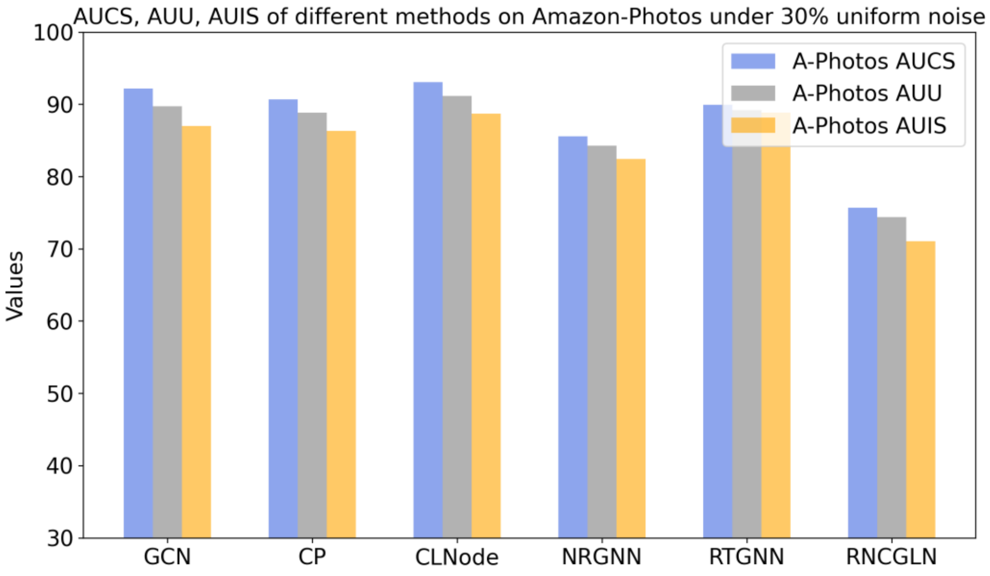


# NoisyGL-Findings

## Finding 5：Label noise can Propagate through graph topology.

- Experimental results show that the classification accuracy of unlabeled data significantly decreases **when there are incorrectly labeled nodes in the neighborhood**.

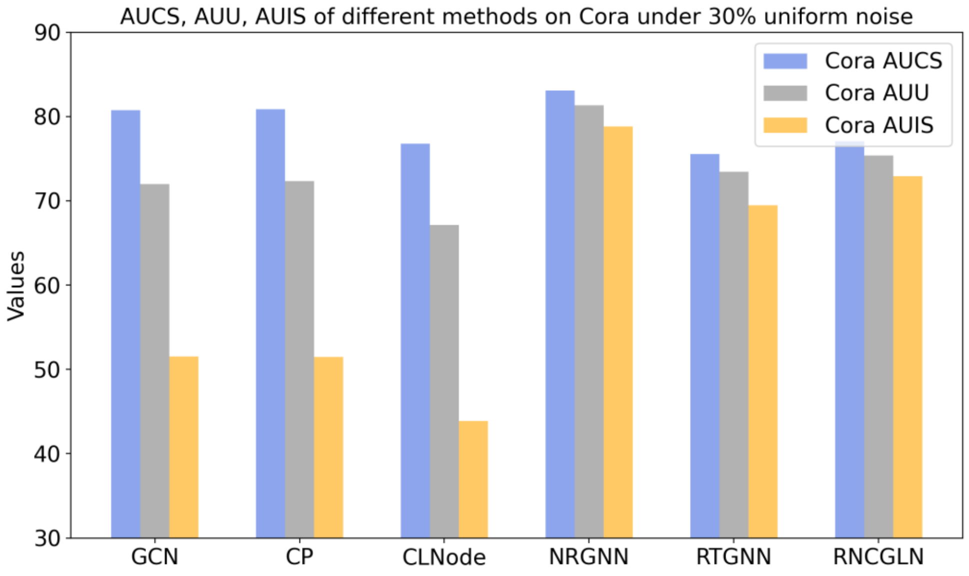


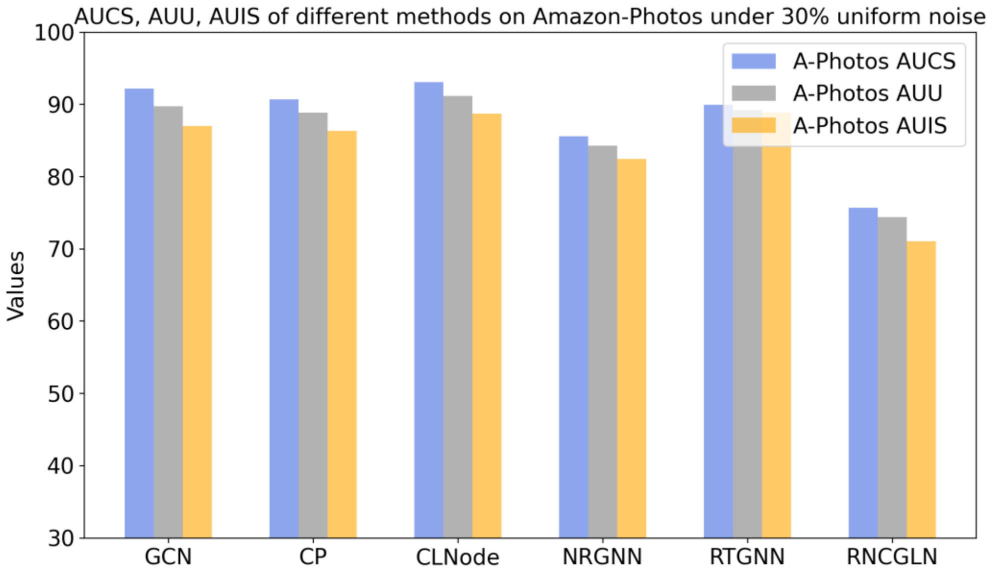



# NoisyGL-Findings

## Finding 6: Sparse graphs are more vulnerable to the propagation effect of label noise.

- **The propagation of label noise is more apparent in sparse graphs** (e.g. Cora) and less noticeable in dense graphs (e.g. A-Photos). This probably because **nodes in dense graphs can receive more supervision** from the neighboring nodes.

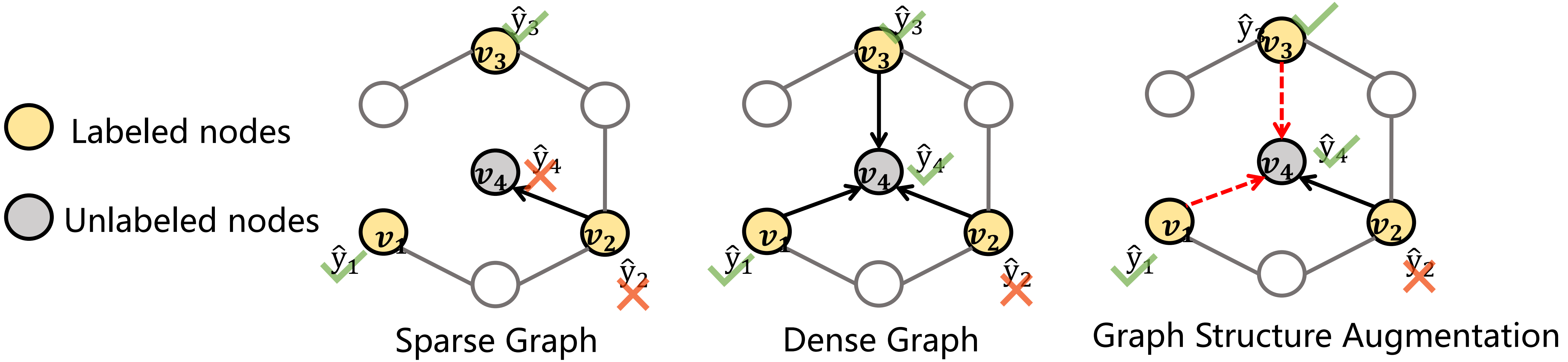


# NoisyGL-Findings

## Finding 6: Sparse graphs are more vulnerable to the propagation effect of label noise.

- **Graph Structure Augmentation** methods (NRGNN, RTGNN, RNCGLN) can mitigate the propagation of label noise in sparse graphs.
- The **up-sampling process** in these methods introduces more edges, providing additional supervision for unlabeled nodes.



# NoisyGL-Future directions

## Designing widely applicable GLN approaches

- There is no existing GLN method applicable for all scenarios, **especially for highly heterophilous graphs**. Therefore, it is necessary to develop GLN methods that are broadly applicable to various types of data.

## Designing GLN approaches for various graph learning tasks

- Previous studies on GLN have predominantly focused on node classification tasks. Hence, designing GLN methods for more graph learning tasks (e.g. link prediction, graph classification) is crucial .

## Considering other types of label noise in graph learning

- Current GLN methods assume that labels are affected by i.i.d. label noise. However, the non-i.i.d. Graph data, may also have non-i.i.d. label noise. Therefore, it is necessary to redesign noise models and GLN schemes under new assumptions.

# NoisyGL

## Paper

## Code

**Thank you for listening!**