Graph Label Denoising via Neighborhood Agreement–Guided Expectation Maximization

Dezhi Liu[1], **Richong Zhang**[1*], **Junfan Chen**[2], **Fengbo Tian**[3] and **Si Chen**[1]
[1] School of Computer Science and Engineering, Beihang University
[2] School of Software, Beihang University
[3] University of Electronic Science and Technology of China
{dezhi.liu, zhangrichong,chenjf91,chen.si}@buaa.edu.cn, 2022090912003@std.uestc.edu.cn

Abstract

Graph Neural Networks are susceptible to label noise, in which message-passing mechanisms serve as conduits for propagating erroneous supervision. Current mitigation techniques typically recover clean labels via heuristics that lack theoretical grounding, which often leads to ineffective denoising. To tackle the issue, we propose NAEM, a latent label estimation framework that models clean labels as latent variables by integrating neighborhood agreement into the Expectation-Maximization (EM) paradigm. To overcome the posterior collapse problem in standard EM under severe noise, we design a structure-aware E-step that leverages neighborhood agreement as a structural prior. This mechanism acts as a dynamic confidence gate based on local consensus to prevent the model from overfitting to noise. Simultaneously, by explicitly modeling the noise transition matrix in the M-step, NAEM decouples noise dynamics from semantic representation learning. Extensive experiments on five benchmark datasets demonstrate that NAEM consistently outperforms SOTA methods under varying noise conditions, validating the effectiveness of our framework.

1 Introduction

Graph Neural Networks (GNNs) have emerged as the dominant paradigm for representation learning on graph-structured data, demonstrating exceptional performance in node classification [Kipf and Welling, 2016; Velikovi *et al.*, 2018]. Despite their success, GNNs often struggle in real-world settings due to unavoidable label noise from unreliable annotations or malicious attacks. While label noise is a well-studied topic in computer vision, its impact on graphs is more severe and complex due to the non-i.i.d. nature of the data. This message-passing paradigm transforms GNNs into effective spreaders of label noise. Unlike i.i.d. (independent and identically distributed) data, where noise is localized, graph edges facilitate the propagation of erroneous signals. Thus, a single noisy node can significantly compromise the learned representations of its clean neighbors through topological aggregation [NT *et al.*, 2019a; Dai *et al.*, 2021]. This topological dependency causes a cascading effect, where a few noisy labels can pollute large areas of the graph, significantly reducing model generalization.

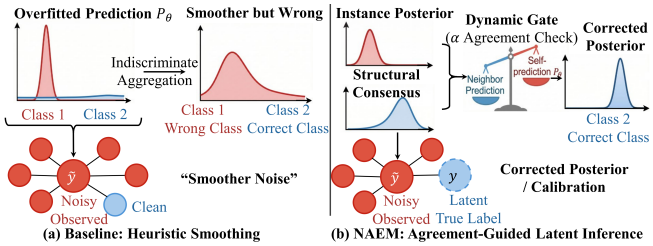

Figure 1: (a) Existing heuristic methods only rely on naive model predictions, which may already be overfitting to noise. Indiscriminately aggregating these biased predictions leads to "smoother noise" and reinforces bias. (b) NAEM (Ours) treats the true label as a latent variable y. We introduce Neighborhood Agreement as a dynamic gate mechanism. When the model's instance-level prediction conflicts with neighborhood consensus, structural priors refine the posterior distribution, effectively estimating the clean latent label.

To mitigate this issue, recent research has adapted strategies from the i.i.d. domain, predominantly categorizing into *sample selection* and *label correction*. Sample selection methods, such as TSS [Wu *et al.*, 2024] and CLNode [Wei *et al.*, 2023], employ heuristics like the small-loss criterion to segregate clean nodes from noisy ones. While effective in low-noise regimes, these methods suffer from data starvation and confirmation bias as noise rates increase, leading the model to overfit early while discarding informative hard samples. Conversely, label correction approaches, such as NRGNN [Dai *et al.*, 2021] and RTGNN [Qian *et al.*, 2023], attempt to rectify noisy labels via neighborhood smoothing. However, these methods often apply graph smoothing indiscriminately based on the naive homophily assumption. As illustrated in Figure 1(a), we contend that employing noise-dominated neighborhoods to refine node labels does not eliminate the noise within the graph; rather, it merely produces smoother noise.

A critical oversight in current methods is the formulation of the *Clean Label*as a deterministic target to be recovered via a selection or smoothing paradigm. However, since ground

*Corresponding author

truth is intrinsically unobservable under noisy conditions, we propose that it should be rigorously modeled as a *Latent Variable* within a probabilistic generative paradigm. We redefine the graph noisy learning task as maximum likelihood estimation via the EM algorithm [Dempster *et al.*, 1977]. However, a major limitation of standard EM in noisy environments is its susceptibility to *posterior collapse*, which is particularly pronounced in graph learning since the topology-dependent noise propagation pattern. As GNNs recursively encode noisy signals from neighbors into node representations, the model learns to interpret this noise as a valid structural pattern, resulting in self-reinforcing bias. The posterior converges towards the noisy observations $\tilde{Y}$ with misleadingly high confidence. To alleviate posterior collapse on graph data, we integrate Neighborhood Agreement (NA) into the EM framework. NA acts as an informative structural prior to guide and refine the posterior estimation rather than serving as a hard constraint.

We propose NAEM, a structure-aware Neighborhood Agreement EM framework that provides a principled solution for denoising graph labels, as shown in Figure 1(b). Unlike heuristic approaches, NAEM maximizes the Expected Complete-Data Log-Likelihood of the observed noisy graph data. First, we design a structure-aware E-step that produces a refined posterior explicitly incorporating the neighborhood agreement mechanism. In detail, we leverage an adaptive *agreement gate* to refine inference based on structural consistency. It retains instance-level predictions for nodes aligned with their local neighbors, while adjusting inconsistent nodes towards the direction of neighborhood consensus, thereby mitigating overfitting to noisy labels. Moreover, during the corresponding adaptive M-step, we simultaneously optimize the GNN parameters and the noise transition matrix through iterative refinement. NAEM effectively disentangles corruption patterns from semantic features by explicitly modeling the noise transition process, facilitating robust representation learning across varied noise environments. Extensive experiments on five benchmark datasets show our method consistently outperforms baselines and enjoys great scalability under various noise conditions. Supplementary material is available at Zenodo [1].

Our main contributions are in the following three-fold:

- We propose a probabilistic generative framework, named NAEM, that reformulates graph label noise learning as a latent variable estimation problem. We derive a unified optimization target with theoretical convergence guarantees.
- We design a structure-aware E-step that integrates Neighborhood Agreement as a structural prior. Through an adaptive gating mechanism, we dynamically refine the posterior to curb noisy signal propagation.
- Extensive experiments on five benchmark datasets demonstrate that NAEM consistently outperforms baselines under various noise conditions, and comprehensive analysis confirms the effectiveness and robustness of our framework.

[1] https://doi.org/10.5281/zenodo.20098669

2 Related Work

2.1 Label Noise Learning on I.I.D. Data

Traditional independent and identically distributed label noise learning follows three paradigms: estimating the noise transition matrix, such as the S-model [Goldberger and Ben-Reuven, 2017]; selecting clean samples based on the small-loss assumption, such as Co-teaching [Han *et al.*, 2018] and JoCoR [Wei *et al.*, 2020]; and employing robust objective functions, such as SCE [Wang *et al.*, 2019]. However, these methods struggle with graph data where noise spreads through edges, as they overlook properties that depend on topology.

2.2 Label Noise Learning on Graphs

Graph noise learning in research is mainly categorized into sample selection, label correction, and representation learning.

Sample Selection Strategies

Similar to i.i.d. methods, these strategies train GNNs using only the estimated clean label set. CLNode [Wei *et al.*, 2023] employs a curriculum learning strategy, while TSS [Wu *et al.*, 2024] and RTGNN [Qian *et al.*, 2023] utilize topology identification criteria and collaborative learning mechanisms, respectively. However, sample selection often leads to data insufficiency and degraded generalization performance when facing a severe noise setting.

Label Correction and Propagation

Conversely, these methods correct noise through smoothing or constraints. NRGNN [Dai *et al.*, 2021], Union-NET [Li *et al.*, 2021], and CGNN [Yuan *et al.*, 2023] use naive neighborhood aggregation, while other methods incorporate structural guidelines such as community clustering [Zhang *et al.*, 2020], pairwise interactions [Du *et al.*, 2021], and backward loss correction [NT *et al.*, 2019b]. However, random aggregation of noisy neighborhoods tends to produce smoother noise rather than clear signals, thereby enhancing confirmation bias.

Robust Representation and Contrastive Learning

Recent studies enhance GNN robustness by utilizing graph contrastive learning [Zhu *et al.*, 2024; Li *et al.*, 2024] or energy-based detection and prototype alignment [Chen *et al.*, 2024; Li *et al.*, 2025]. Although these methods enhance the robustness of features, they do not provide a unified probabilistic framework for inferring labels. Unlike existing works treating noisy labels as deterministic targets, our NAEM framework models the true label as a *latent variable*.

3 Methodology

In this section, we present the proposed NAEM framework to mitigate graph label noise, as illustrated in Figure 2. Specifically, we formulate the noisy node classification problem as a likelihood maximization task involving a latent variable. Then, we incorporate a structure-aware prior during the inference stage to rectify the posterior estimation within the EM paradigm, followed by an adaptive maximization stage to decouple noise patterns from semantic representations.

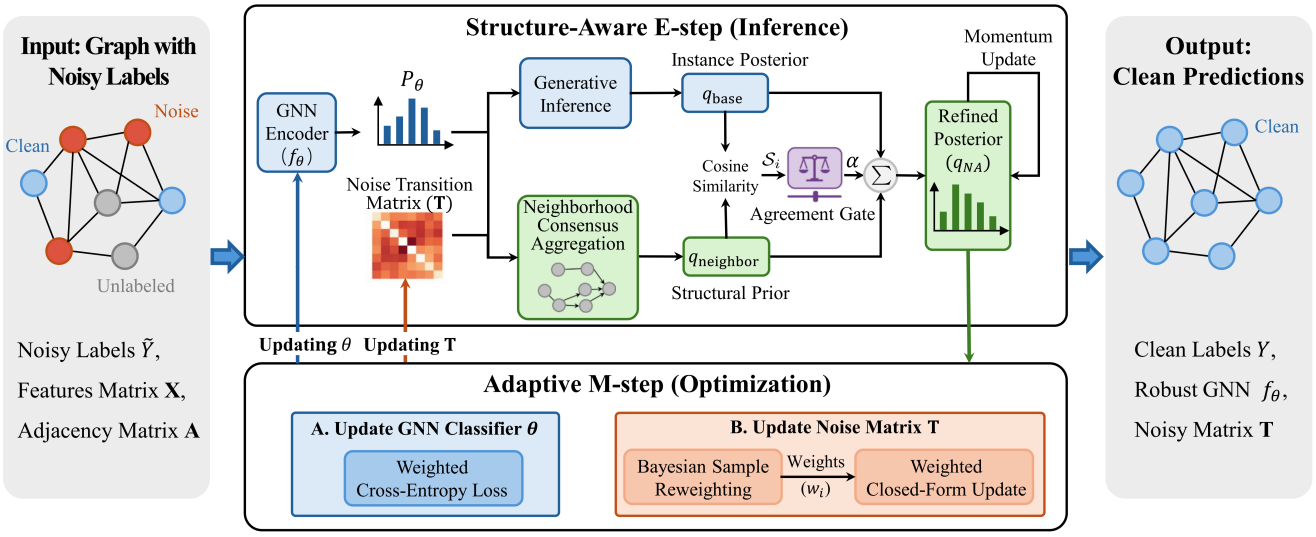

Figure 2: Overview of NAEM. The model iterates between a structure-aware E-Step and an adaptive M-Step. The E-Step infers latent labels by fusing the instance posterior (q_{base}) and structural prior (q_{neighbor}) via an adaptive agreement gate (α). The M-Step utilizes the refined posterior to jointly optimize GNN parameters θ and the noise transition matrix $\mathbf{T}$.

3.1 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected graph where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of N nodes and $\mathcal{E}$ is the set of edges. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ be the node feature matrix and $\mathbf{A} \in \{0,1\}^{N \times N}$ be the adjacency matrix. The node set $\mathcal{V}$ can be split into labeled set $\mathcal{V}_L$ and unlabeled set $\mathcal{V}_U$, where usually $|\mathcal{V}_L| \ll |\mathcal{V}_U|$ in graph node classification task. The class set is $\mathcal{C} = \{1, \ldots, K\}$.

The observed labeled set $\mathcal{V}_L$ may be corrupted in the real-world settings. We represent the observed noisy labels as $\tilde{Y}_L = \{\tilde{y}_i \mid v_i \in \mathcal{V}_L\}$, where $\tilde{y}_i \in \mathcal{C}$. We also assume that the latent clean labels, represented as $Y = \{y_i \mid v_i \in \mathcal{V}\}$, are unobservable. The goal is to learn a robust GNN classifier f_θ that can accurately predict the clean labels Y for both $\mathcal{V}_L$ and $\mathcal{V}_U$ under noisy supervision from $\tilde{Y}_L$.

3.2 Probabilistic Generative Framework

In contrast to sample selection paradigms, we model the label correction as a probabilistic inference problem. To estimate clean label, we establish a generative process connecting the latent clean Y and the noisy observation $\tilde{Y}$ conditioned on graph data $\mathbf{X}$ and $\mathbf{A}$.

Latent Label Generation. The clean label y_i is generated based on node features $\mathbf{X}$ and graph structure $\mathbf{A}$, and is implemented by the GNN parameters θ:

$$P(y_i = k \mid \mathbf{X}, \mathbf{A}; \theta) = f_\theta(\mathbf{x}_i, \mathbf{A})_k. \tag{1}$$

where $f_\theta(\cdot)_k$ denotes the predicted probability for class k.

Noisy Label Generation. Given the latent clean label y_i, the observed noisy label $\tilde{y}_i$ is produced via a noise transition process. We adopt a class-dependent noise model defined by a transition matrix $\mathbf{T} \in [0,1]^{K \times K}$, following standard assumptions for graph label noise learning [Wang *et al.*, 2024]:

$$P(\tilde{y}_i = k \mid y_i = j; \mathbf{T}) = \mathbf{T}_{jk}, \quad \text{s.t.} \sum_{k=1}^{K} \mathbf{T}_{jk} = 1, \tag{2}$$

where $\mathbf{T}_{jk}$ represents the probability that clean label j is corrupted into noisy label k. Our goal is to jointly infer the posterior distribution of the latent true labels Y and learn the robust classifier θ by maximizing the marginal likelihood:

$$\begin{aligned}\mathcal{L}(\theta, \mathbf{T}) &= \log P(\tilde{Y}_L \mid \mathbf{X}, \mathbf{A}; \theta, \mathbf{T}) \\ &= \log \sum_{Y} P(\tilde{Y}_L \mid Y; \mathbf{T}) \cdot P(Y \mid \mathbf{X}, \mathbf{A}; \theta).\end{aligned} \tag{3}$$

As direct maximization is intractable, we employ the EM algorithm to maximize the Expected Complete-Data Log-Likelihood ($\mathcal{Q}$-function):

$$\begin{aligned}&\mathcal{Q}(\Theta \mid \Theta^{(t)}) = \mathbb{E}_{Y \sim P(Y \mid \tilde{Y}_L, \mathbf{X}, \mathbf{A}; \Theta^{(t)})} \left[\log P(\tilde{Y}_L, Y \mid \mathbf{X}, \mathbf{A}; \Theta)\right] \\ &= \sum_{i \in \mathcal{V}_L} \sum_{j=1}^{K} q_i^{(t)}(j) \left(\log P(\tilde{y}_i \mid y_i = j; \mathbf{T}) + \log P(y_i = j \mid \mathbf{X}, \mathbf{A}; \theta)\right),\end{aligned} \tag{4}$$

where $\Theta = \{\theta, \mathbf{T}\}$ and $q_i^{(t)}(j)$ is the posterior estimate at iteration t. Detailed derivation in Supplementary Sec. A.1.

Standard EM estimates the posterior $q(Y)$ relying solely on the current model prediction P_θ and transition matrix $\mathbf{T}$. Nevertheless, this phenomenon is especially harmful in graph learning because of the so-called topology-dependent noise propagation. Compared to i.i.d. data, where noise is typically associated with specific samples, the recursive messages that flow through GNNs may unintentionally carry misleading signal information from noisy neighbors into the node representations. This creates a self-reinforcing cycle where the model is led to believe that label noise is a valid structural pattern. As a result, the graph classifier f_θ is more likely to overfit noisy labels with misleadingly high confidence, and the posterior collapses to the noisy observations, and hence it loses its ability to correct errors. To mitigate this, we introduce Neighborhood Agreement as a structural prior to calibrate the posterior inference in the E-step.

3.3 Structure-Aware E-Step

In the E-step, our objective is to estimate the posterior distribution of the clean labels $q^{(t)}(y_i = j) \approx P(y_i = j \mid \tilde{y}_i, \mathbf{x}_i, \mathbf{A})$. To prevent the posterior from being dominated by noisy supervision, we inject the graph structure as an inductive bias.

Initialization and Evidence Decomposition

Before the iterative process, we initialize the model parameters $\theta^{(0)}$ via a warm-up phase using the standard cross-entropy loss on $\tilde{Y}_L$ for a few epochs. The transition matrix $\mathbf{T}^{(0)}$ employs a diagonally dominant random initialization.

At iteration t, we decompose the evidence for the latent label y_i into the instance-level posterior and the neighborhood consensus.

First, the **instance-level posterior** q_{base} represents the standard Bayesian update for node i, conditioning on its observed noisy label $\tilde{y}_i$. It integrates the predictive distribution from the GNN classifier (acting as the latent prior) and the noise transition probability (acting as the observation likelihood):

$$q^{(t)}_{\text{base}}(y_i = j) \propto \underbrace{P(y_i = j \mid \mathbf{X}, \mathbf{A}; \theta^{(t)})}_{\text{GNN Prediction}} \cdot \underbrace{P(\tilde{y}_i \mid y_i = j; \mathbf{T}^{(t)})}_{\text{Noise Transition}}. \tag{5}$$

The posterior remains heavily anchored to the current parameters $\theta^{(t)}$, rendering it susceptible to the bias of noisy labels during early training.

Second, we introduce the **neighborhood consensus** q_{neighbor} to rectify potential biases in q_{base}. The proposed mechanism diverges from feature aggregation by enforcing *explicit posterior smoothing* within the label space, grounded in the graph homophily assumption. Pooling instance-level posteriors from neighboring nodes establishes a structural prior:

$$q^{(t)}_{\text{neighbor}}(y_i = j) = \frac{1}{|\mathcal{N}(i)|} \sum_{v \in \mathcal{N}(i)} q^{(t)}_{\text{base}}(y^*_v = j). \tag{6}$$

where $\mathcal{N}(i) = \{v_j \in \mathcal{V} \mid \mathbf{A}_{ij} = 1\}$ denotes the set of immediate neighbors of node v_i.

Neighborhood Consistency and Posterior Refinement

To evaluate the reliability of the instance-level posterior, we compute the Neighborhood Agreement (NA) score $\mathcal{S}_i \in [0, 1]$ for node i via the normalized cosine similarity between the two evidence views:

$$\mathcal{S}_i = \frac{1}{2}\left(\frac{q_{\text{base},i} \cdot q_{\text{neighbor},i}}{\|q_{\text{base},i}\|\|q_{\text{neighbor},i}\|} + 1\right), \tag{7}$$

where $q_{\text{base},i}$ and $q_{\text{neighbor},i}$ denote the probability vectors of $q^{(t)}_{\text{base}}(y_i)$ and $q^{(t)}_{\text{neighbor}}(y_i)$, respectively. A high $\mathcal{S}_i$ signifies that the GNN prediction aligns with the local topology, typical of reliable samples, whereas a low agreement suggests a conflict indicative of label noise. Dynamic regulation of the neighborhood consensus influence relies on the computation of an adaptive agreement gate α_i. Mapping $\mathcal{S}_i$ to a confidence weight w_i precedes the derivation of α_i:

$$w_i = \text{clip}\left(\frac{\mathcal{S}_i - \tau_{\text{low}}}{\tau_{\text{high}} - \tau_{\text{low}}}, 0, 1\right), \quad \alpha_i = \lambda_{\text{NA}}(1 - w_i). \tag{8}$$

Here, τ_{high} and τ_{low} are agreement thresholds determining the trust interval, and λ_{NA} is agreement strength controlling the maximum strength of the structural prior. The gating mechanism α_i dynamically adjusts the mixing ratio by retaining the instance-level posterior for high-agreement nodes ($w_i \approx 1$) or enforcing neighborhood consensus correction as the coefficient approaches λ_{NA} under low-agreement conditions ($w_i \approx 0$). We derive the refined posterior $q^{(t)}_{\text{NA}}(y_i)$ via convex combination:

$$q^{(t)}_{\text{NA}}(y_i) = (1 - \alpha_i) \cdot q^{(t)}_{\text{base}}(y_i) + \alpha_i \cdot q^{(t)}_{\text{neighbor}}(y_i). \tag{9}$$

Finally, we apply a momentum coefficient hyperparameter, denoted as γ, to update the model to ensure stability:

$$q^{(t)} = \gamma q^{(t-1)} + (1 - \gamma) q^{(t)}_{\text{NA}}, \tag{10}$$

3.4 Adaptive M-Step for Joint Optimization

In the M-step, we maximize the $\mathcal{Q}$-function with respect to GNN parameters θ and transition matrix $\mathbf{T}$ given $q^{(t)}$, decoupling semantic learning from noise modeling, as θ and $\mathbf{T}$ appear in separate terms in Eq. (4).

Robust Representation Learning (Updating θ)

The optimization for θ corresponds to minimizing the expected risk with respect to the latent labels. This is equivalent to minimizing a posterior-weighted cross-entropy loss:

$$\min_{\theta} \mathcal{L}_\theta = -\sum_{i \in \mathcal{V}_L} \sum_{j=1}^{K} q^{(t)}(y_i = j) \log P_\theta(y_i = j \mid \mathbf{x}_i, \mathbf{A}). \tag{11}$$

Supervision of the GNN relies on the corrected latent distribution through the substitution of noisy one-hot targets $\tilde{y}_i$ with refined soft labels $q^{(t)}$.

Noise Transition Estimation (Updating T)

NAEM explicitly learns the class-dependent noise pattern $\mathbf{T}$ to decouple noise patterns from semantic features. Maximizing the $\mathcal{Q}$-function with respect to $\mathbf{T}$ under the constraint $\sum_k \mathbf{T}_{jk} = 1$ yields a weighted closed-form solution. To mitigate random outliers, Bayesian Sample Reweighting assigns each sample a transition-model reliability score r_i, which estimates the posterior probability that the sample is generated by the structured transition process rather than a uniform outlier component; its closed-form definition and derivation are provided in the supplementary material, Sec. A.2. This leads to the Weighted Closed-Form Solution for the update rule at iteration $t + 1$:

$$\mathbf{T}^{(t+1)}_{jk} = \frac{\sum_{i \in \mathcal{V}_L} r_i \cdot q^{(t)}(y_i = j) \cdot \mathbb{I}(\tilde{y}_i = k)}{\sum_{k'=1}^{K} \left(\sum_{i \in \mathcal{V}_L} r_i \cdot q^{(t)}(y_i = j) \cdot \mathbb{I}(\tilde{y}_i = k')\right)}, \tag{12}$$

Here $\mathbb{I}(\cdot)$ denotes the indicator function. The symbol r_i is distinct from the neighborhood-agreement confidence w_i in Eq. (8). The estimation mechanism aggregates the fractional counts of estimated true labels j flipping to observed labels k weighted by sample reliability to approximate the noise transition probability. Algorithm details and complexity analysis are provided in Supplementary Sec. A.3; NAEM preserves the $\mathcal{O}(|\mathcal{E}|d + Nd^2)$ complexity inherent to the backbone GNN.

Dataset	Cora		Citeseer		Pubmed		A-Computers		A-Photos	
Noise type	30% PN	30% UN	30% PN	30% UN	30% PN	30% UN	30% PN	30% UN	30% PN	30% UN
GCN	65.36±5.54	71.06±4.39	53.66±5.03	55.05±3.11	62.91±5.49	66.53±6.23	70.95±4.21	77.26±3.34	79.26±4.79	84.86±3.27
GAT	62.93±4.04	65.53±4.01	50.90±4.78	52.88±3.07	59.77±4.19	60.64±6.43	68.78±4.58	76.11±2.75	79.46±4.23	85.64±2.61
S-model	65.59±5.28	71.05±4.88	53.41±5.26	54.68±3.26	63.16±5.46	66.52±6.88	73.81±5.91	79.42±3.17	80.06±5.25	85.32±4.17
Coteaching	56.87±4.40	61.22±5.88	48.34±3.48	46.75±4.80	60.45±7.23	64.38±7.27	67.89±9.09	76.25±5.65	74.56±7.56	82.78±5.75
JoCoR	64.52±5.80	71.16±6.53	56.30±5.14	57.14±3.22	55.88±7.20	59.44±5.23	66.82±4.70	70.01±11.03	68.48±8.62	75.22±8.24
SCE	66.97±4.29	71.15±5.04	54.61±4.53	57.00±2.87	63.78±4.86	67.38±6.86	72.85±2.19	77.01±2.82	81.57±4.96	85.85±4.42
NRGNN	69.73±4.82	74.86±2.82	59.28±6.21	61.87±3.44	57.22±5.93	62.37±6.79	67.00±2.65	70.47±3.45	71.92±6.84	78.30±3.31
RTGNN	63.44±5.93	66.33±3.72	48.39±5.15	51.38±4.11	59.04±6.65	68.30±4.31	61.40±10.02	69.04±4.96	70.33±8.11	82.08±6.02
CP	66.12±5.63	71.76±5.85	54.15±5.44	56.72±3.42	62.67±4.62	68.11±4.27	70.49±4.59	77.74±3.13	77.45±5.26	83.52±3.00
CLNode	64.55±5.04	68.24±3.95	53.19±3.19	55.48±2.92	62.25±5.23	64.56±6.50	73.96±4.61	80.31±2.41	79.12±4.90	85.46±2.70
PIGNN	65.04±5.83	70.43±4.19	55.97±4.79	60.71±2.98	66.75±3.88	69.02±3.34	72.22±5.80	79.04±2.28	79.43±4.63	84.90±4.59
DGNN	58.49±4.41	56.15±6.40	46.76±6.57	48.99±2.57	58.36±4.87	63.29±7.74	48.11±7.24	50.53±4.26	51.09±13.97	54.28±8.37
RNCGLN	68.48±6.37	73.29±5.17	54.16±4.56	58.56±7.25	62.88±5.80	68.42±5.31	63.62±3.93	62.22±3.97	64.79±9.92	72.80±6.60
UnionNET	66.29±5.23	72.83±4.74	58.15±6.09	58.87±3.25	62.77±4.97	65.93±6.56	37.33±2.18	37.34±2.94	30.89±7.39	32.29±4.47
CGNN	63.27±5.71	69.57±3.47	51.38±4.19	52.31±3.22	56.01±14.88	57.73±16.75	40.55±21.80	46.07±22.63	45.70±25.46	47.26±25.51
CR-GNN	65.37±5.48	72.03±4.48	52.45±5.07	55.87±3.38	62.91±5.47	66.42±6.35	39.18±31.03	48.29±31.43	37.53±35.25	35.24±32.66
R2LP	58.42±1.45	57.08±1.62	32.51±9.06	37.80±11.22	50.73±10.75	48.80±17.57	47.97±11.85	45.13±13.25	55.14±10.81	45.18±13.77
TSS	60.77±4.90	62.61±4.97	51.88±5.25	53.52±3.02	62.95±5.12	64.14±5.92	66.69±7.98	73.08±3.73	77.00±5.80	83.04±2.66
ERASE	73.38±4.47	77.40±2.43	53.63±4.48	58.78±4.29	63.35±4.78	67.52±5.05	64.68±5.64	69.39±3.53	81.62±6.66	87.00±1.80
ProCon	74.66±3.80	76.61±4.55	55.16±8.02	59.68±5.53	67.10±3.56	71.77±3.90	70.47±4.58	73.48±4.11	82.02±3.38	81.81±6.42
NAEM	**78.50±3.99**	**81.70±0.93**	**61.54±4.08**	**68.29±1.74**	**68.13±6.55**	**72.90±4.88**	**82.41±1.75**	**84.96±1.00**	**89.06±1.56**	**90.75±0.78**

Table 1: Test accuracy on five datasets under 30% Pair Noise (PN) and Uniform Noise (UN) (10 Runs). The best results are in bold.

4 Experiment

We evaluate the effectiveness of the proposed NAEM framework by addressing the following research questions:

RQ1: Does NAEM outperform state-of-the-art baselines on node classification tasks under various noise settings?

RQ2: Can NAEM mitigate noise propagation and maintain accuracy under incorrect supervision?

RQ3: How does each component contribute to the overall performance of the framework?

RQ4: Does the Neighborhood Agreement strategy effectively prevent posterior collapse and enhance latent label inference?

4.1 Experimental Setup

Datasets

We evaluate NAEM on five benchmarks: Cora, Citeseer, Pubmed [Sen *et al.*, 2008], Amazon-Computers, and Amazon-Photos [Shchur *et al.*, 2018], with OGBN-Arxiv [Hu *et al.*, 2020] and Flickr used for scalability and heterophily evaluation. Dataset details are available in Supplementary Sec. B.1.

Noise Injection

We follow standard protocols [Wang *et al.*, 2024; Li *et al.*, 2025] to inject synthetic label noise into the training set:

Uniform Noise (UN): Uniformly flips the true label to another class with probability η. Formally, for $\forall j \neq i$, $p(\widetilde{y} = j \mid Y = i) = \frac{\eta}{c-1}$, where c represents the number of classes.

Pair Noise (PN): Flips the true label to its corresponding pair class with probability η. We test noise rates $\eta \in \{10\%, 20\%, 30\%, 40\%, 50\%\}$ for both types.

Baselines

We compare NAEM against baselines from three categories:

Standard GNNs: GCN [Kipf and Welling, 2016], GAT [Velikovi *et al.*, 2018].

Label Noise Learning (I.I.D.): S-model [Goldberger and Ben-Reuven, 2017], Co-teaching [Han *et al.*, 2018], JoCoR [Wei *et al.*, 2020], SCE [Wang *et al.*, 2019].

Graph Label Noise: NRGNN [Dai *et al.*, 2021], RTGNN [Qian *et al.*, 2023], CP [Zhang *et al.*, 2020], DGNN [NT *et al.*, 2019b], Union-NET [Li *et al.*, 2021], PIGNN [Du *et al.*, 2021], CLNode [Wei *et al.*, 2023], RNCGLN [Zhu *et al.*, 2024], CGNN [Yuan *et al.*, 2023], CR-GNN [Li *et al.*, 2024], TSS [Wu *et al.*, 2024], ERASE [Chen *et al.*, 2024], R2LN [Cheng *et al.*, 2024], ProCon [Li *et al.*, 2025].

To ensure a fair and rigorous comparison, for baselines included in the NoisyGL benchmark [Wang *et al.*, 2024], we report the results directly from the original paper. For recent methods not covered by NoisyGL, we reproduce their results using their official implementations, strictly adhering to the experimental settings of NoisyGL. Implementation specifics of our method, including hyperparameter grid search ranges and optimization setups, are detailed in Supplementary Sec. B.2.

4.2 Performance Comparison (RQ1)

Main Result

Table 1 reports the test accuracy (mean ± std over 10 runs). NAEM consistently outperforms all baselines on five datasets under 30% noise. For instance, on Amazon-Computers (30%

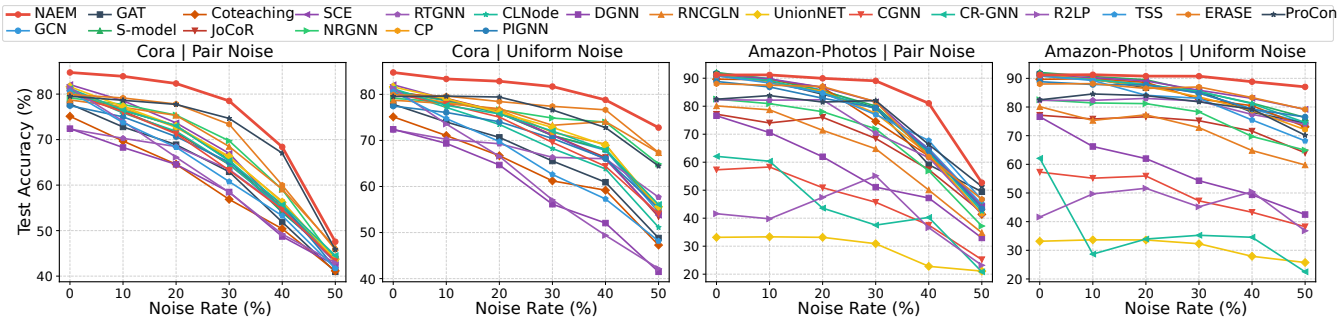

Figure 3: Test accuracy on Cora and Amazon-Photos datasets under different rates of pair and uniform noise (10 Runs).

Dataset	Cora					Amazon-Computers				
Metric	ACLT	AILT	AUCS	AUU	AUIS	ACLT	AILT	AUCS	AUU	AUIS
NRGNN	84.53±4.07	75.64±6.68	83.11±3.16	81.33±2.25	78.81±5.94	61.82±5.95	52.34±7.43	72.48±5.35	71.15±5.00	69.03±6.67
RTGNN	82.29±3.33	71.57±8.33	75.53±4.80	73.44±4.95	69.50±8.06	73.58±7.57	65.89±6.97	72.57±5.67	71.37±5.33	67.42±6.79
CP	98.77±1.49	32.04±14.17	80.85±3.46	72.32±4.45	51.46±12.99	88.51±4.38	70.81±4.68	82.29±3.76	80.65±4.00	77.52±4.44
CLNode	94.14±3.44	34.88±9.96	76.77±3.30	67.11±5.11	43.86±7.48	92.01±2.61	75.59±4.39	84.09±2.50	82.77±2.99	80.19±5.43
PIGNN	98.43±1.55	29.19±20.67	80.30±4.69	70.28±8.69	49.25±16.24	**93.43±2.98**	70.94±7.66	83.08±2.38	81.69±3.08	79.78±5.93
DGNN	81.06±10.80	37.58±15.48	70.89±7.28	63.60±6.94	45.72±10.64	37.44±10.38	37.92±9.60	50.31±8.42	49.28±7.37	46.11±7.94
RNCGLN	**99.00±1.18**	13.44±8.24	77.06±3.30	75.36±3.33	72.92±4.66	84.67±10.92	26.18±13.75	58.66±4.12	57.63±3.75	54.96±6.97
UnionNET	98.60±1.18	33.89±15.53	82.45±2.69	73.46±3.70	52.59±9.00	13.17±4.37	13.20±5.35	26.42±5.21	26.26±5.46	21.99±7.48
CGNN	97.16±2.79	31.37±10.72	80.41±3.62	70.38±4.04	46.81±10.04	44.71±32.57	38.55±29.40	47.06±24.61	46.04±24.70	42.20±26.69
CR-GNN	96.39±1.92	34.93±8.11	80.09±2.90	71.75±4.73	51.88±6.64	53.11±30.63	43.38±24.06	52.15±30.77	51.12±30.06	48.21±29.15
R2LP	99.14±0.92	1.71±2.26	71.68±3.39	55.45±1.98	44.48±4.78	17.15±7.48	20.79±8.86	32.54±13.66	45.35±19.41	31.88±12.50
TSS	88.20±9.53	14.94±10.08	74.12±3.68	62.28±5.48	38.71±9.03	85.16±7.87	49.93±14.11	73.47±3.94	73.62±4.46	66.16±6.69
ERASE	91.30±2.54	75.17±8.41	82.31±2.58	76.49±2.75	78.94±5.30	85.66±3.52	66.10±6.72	72.70±5.50	68.45±3.85	70.73±4.79
ProCon	94.46±3.71	64.49±11.99	82.13±3.43	75.99±4.54	73.86±10.43	84.80±2.68	74.96±3.14	75.62±5.21	72.58±4.33	76.82±4.49
NAEM	92.58±2.17	**77.13±6.93**	**85.49±2.45**	**81.04±0.96**	**81.59±4.68**	90.25±1.67	**84.40±2.66**	**84.45±1.84**	**85.17±1.19**	**83.98±3.42**

Table 2: ACLT, AILT, AUCS, AUU, and AUIS of different methods on Cora and Amazon-Computers under 30% uniform noise (10 Runs).

uniform noise), NAEM surpasses the runner-up CLNode by 4.65% (84.96% vs. 80.31%). Furthermore, NAEM demonstrates superior robustness against pair noise; on Citeseer (30% pair noise), it achieves 61.54%, significantly exceeding GCN (53.66%) and RTGNN (48.39%). Additionally, lower standard deviations confirm the stability of our principled probabilistic framework in modeling latent labels. Additional scalability results on the large-scale OGBN-Arxiv benchmark are provided in Supplementary Sec. B.4. The comparison with baselines demonstrates good scalability of our method on large-scale datasets.

Robustness against Varying Noise Rates

To evaluate model robustness, we varied noise rates from 0% to 50% as shown in Figure 3. NAEM consistently outperforms baselines in both settings. Under Uniform Noise, NAEM exhibits remarkable stability, widening the performance gap as noise intensity increases. In the more challenging Pair Noise scenarios, while all methods degrade due to structured label flipping, NAEM maintains a distinct lead over SOTA competitors. Competitive performance on clean data (0% noise) confirms that our method effectively handles noise without compromising the clean signal. To further test robustness beyond homophilic graphs, we report supplementary results on the heterophilic Flickr benchmark in Supplementary Sec. B.3. Comprehensive results on five datasets are detailed in Supplementary Sec. B.8.

4.3 Analysis of Noise Propagation (RQ2)

To verify noise mitigation, we analyze performance across five subgroups [Wang *et al.*, 2024]: Accuracy of Correct (ACLT) and Incorrect (AILT) Labeled Training nodes, as well as Accuracy of Unlabeled Correct Supervised (AUCS), Incorrect Supervised (AUIS), and Unsupervised (AUU) nodes.

Table 2 presents results on Cora and Amazon-Computers under 30% uniform noise. NAEM demonstrates superior accuracy maintenance under incorrect supervision, evidenced by significantly higher scores in AILT and AUIS compared to baselines. High AILT indicates that our framework effectively corrects noisy labels during EM rather than overfitting. Crucially, the substantial margin on AUIS confirms that the Neighborhood Agreement mechanism successfully blocks erroneous signal propagation from incorrect neighbors. Furthermore, robust AUU performance demonstrates effective global representation learning without direct supervision.

Dataset	Cora		Citeseer		Pubmed		Amazon-Computers		Amazon-Photos	
Noise type	30 % PN	30 % UN	30 % PN	30 % UN	30 % PN	30 % UN	30 % PN	30 % UN	30 % PN	30 % UN
w/o Agreement	66.41±5.06	72.42±4.38	51.96±4.69	55.19±2.78	63.71±5.47	66.82±4.81	68.85±5.54	75.02±3.26	79.13±6.25	84.41±2.75
w/o T-Estimation	71.80±8.09	78.72±2.32	59.97±5.01	67.35±1.76	64.42±5.00	68.84±6.00	82.49±1.54	83.06±1.27	88.14±1.85	89.82±1.80
w/o Latent Variable	70.12±4.46	75.07±3.78	56.96±6.03	59.15±2.98	67.79±6.58	69.16±6.27	**83.25±0.55**	81.99±2.00	86.51±3.12	88.73±1.53
NAEM	**78.50±3.99**	**81.70 ±0.93**	**61.54±4.08**	**68.29±1.74**	**68.13±6.55**	**72.90±4.88**	82.41±1.75	**84.96±1.00**	**89.06±1.56**	**90.75±0.78**

Table 3: Ablation study on five datasets under 30% Pair Noise (PN) and Uniform Noise (UN) (10 Runs)

.

4.4 Ablation Studies (RQ3)

Table 3 compares NAEM against three variants to assess module contributions:

w/o Agreement (Standard EM) removes the structural prior q_{neighbor}, reverting to a standard EM algorithm. Its significant performance drop, particularly on Cora (12.09% decrease under Pair Noise), confirms that standard EM suffers from posterior collapse in a graph without structural guidance.

w/o T-Estimation fixes the transition matrix $\mathbf{T}$ to the initial matrix. While effective on uniform noise, it fails on pair noise (2.38% drop on Pubmed), underscoring the need for adaptive noise modeling.

w/o Latent Variable replaces the soft posterior $q(Y)$ with hard pseudo-labels. Consistent degradation verifies that modeling uncertainty via soft assignments is crucial.

The full NAEM framework consistently achieves the best performance. Additional ablation studies on GNN backbones (GCN, GAT, GraphSAGE) are in Supplementary Sec. B.5, confirming that our framework consistently enhances performance across various architectures and datasets.

4.5 Quantitative Analysis (RQ4)

Quality of Latent Posterior

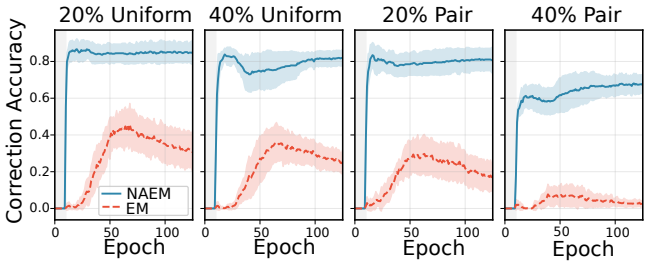

Figure 4: Evolution of Correction Accuracy on Amazon-Photos

To verify NAEM recovers latent ground truth rather than overfitting, we monitor *Correction Accuracy*, which is defined as the accuracy of hard labels inferred from posterior $q(Y)$ against true labels on initially corrupted nodes. Figure 4 compares NAEM with Standard EM on Amazon-Photos. Standard EM displays a characteristic rise-and-fall pattern that peaks mid-training before degrading, signifying *posterior collapse* as the model overfits noisy supervision. Conversely, NAEM sustains high accuracy following a rapid ascent. Under 40% Pair Noise, NAEM maintains accuracy near 0.7, whereas the baseline remains below 0.1. This validates Neighborhood Agreement as a structural regularizer: by calibrating uncertainty via consensus, NAEM prevents posterior degeneration toward noise, enabling iterative refinement toward ground truth. Qualitatively, t-SNE visualization in Supplementary Sec. B.6 demonstrates that NAEM learns more compact, discriminative representations than baselines, aligning noisy nodes with true semantic clusters.

Noise Transition Matrix Estimation

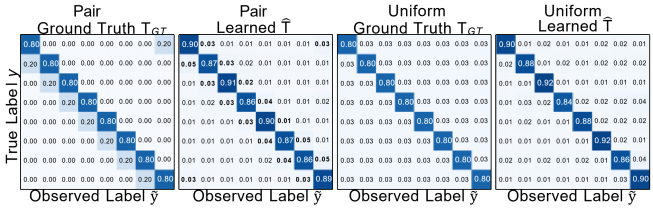

Figure 5: $\mathbf{T}$ estimation on Amazon-Photos under 20% noise.

To validate whether NAEM can capture the underlying noise patterns, we compare the learned transition matrix $\hat{\mathbf{T}}$ with the ground truth $\mathbf{T}_{GT}$ on the Amazon-Photos dataset under 20% Pair and Uniform noise. Figure 5 visualizes these matrices as heatmaps. For Pair noise, while NAEM underestimates the absolute noise magnitude, it correctly captures the structured pairwise flipping pattern via the dominant off-diagonal elements. Similarly, under uniform noise, the model successfully estimates the dispersed noise distribution. This ability to reconstruct the true noise dynamics allows NAEM to decouple corruption from semantic features, thereby explaining away systematic biases in the labels and ensuring robust learning. Sensitivity analyses for E_{warm}, λ_{NA}, γ, τ_{high}, and τ_{low} are provided in Supplementary Sec. B.7, confirming NAEM's robustness across broad hyperparameter settings.

5 Conclusion

We presented NAEM, a probabilistic framework that treats graph label denoising as latent variable inference rather than heuristic sample selection or indiscriminate smoothing. By integrating Neighborhood Agreement into EM, NAEM calibrates posterior uncertainty, mitigates topology-dependent noise propagation, and jointly estimates the noise transition matrix to disentangle corruption from semantic representations. Extensive experiments demonstrate consistent robustness under diverse noise settings. Future work will study open-set noise and broader heterophilic regimes.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. U2433212), in part by the Fundamental Research Funds for the Central Universities, and in part by the State Key Laboratory of Complex & Critical Software Environment.

References

[Chen *et al.*, 2024] Ling-Hao Chen, Yuanshuo Zhang, Taohua Huang, Liangcai Su, Zeyi Lin, Xi Xiao, Xiaobo Xia, and Tongliang Liu. Erase: Error-resilient representation learning on graphs for label noise tolerance. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 270–280, New York, NY, USA, 2024. Association for Computing Machinery.

[Cheng *et al.*, 2024] Yao Cheng, Caihua Shan, Yifei Shen, Xiang Li, Siqiang Luo, and Dongsheng Li. Resurrecting label propagation for graphs with heterophily and label noise. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 433–444, 2024.

[Dai *et al.*, 2021] Enyan Dai, Charu Aggarwal, and Suhang Wang. 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*, KDD '21, page 227–236, New York, NY, USA, 2021. Association for Computing Machinery.

[Dempster *et al.*, 1977] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977.

[Du *et al.*, 2021] Xuefeng Du, Tian Bian, Yu Rong, Bo Han, Tongliang Liu, Tingyang Xu, Wenbing Huang, Yixuan Li, and Junzhou Huang. Noise-robust graph learning by estimating and leveraging pairwise interactions. *arXiv preprint arXiv:2106.07451*, 2021.

[Goldberger and Ben-Reuven, 2017] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*, 2017.

[Han *et al.*, 2018] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: datasets for machine learning on graphs. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Li *et al.*, 2021] Yayong Li, Jie Yin, and Ling Chen. Unified robust training for graph neural networks against label noise. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 528–540. Springer, 2021.

[Li *et al.*, 2024] Xianxian Li, Qiyu Li, Haodong Qian, Jinyan Wang, et al. Contrastive learning of graphs under label noise. *Neural networks*, 172:106113, 2024.

[Li *et al.*, 2025] Kailai Li, Jiawei Sun, Jiong Lou, Zhanbo Feng, Hefeng Zhou, Chentao Wu, Guangtao Xue, Wei Zhao, and Jie Li. Leveraging peer-informed label consistency for robust graph neural networks with noisy labels. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 5598–5606, 2025.

[NT *et al.*, 2019a] Hoang NT, Choong Jun Jin, and Tsuyoshi Murata. Learning graph neural networks with noisy labels. *arXiv preprint arXiv:1905.01591*, 2019.

[NT *et al.*, 2019b] Hoang NT, Choong Jun Jin, and Tsuyoshi Murata. Learning graph neural networks with noisy labels. *arXiv preprint arXiv:1905.01591*, 2019.

[Qian *et al.*, 2023] Siyi Qian, Haochao Ying, Renjun Hu, Jingbo Zhou, Jintai Chen, Danny Z. Chen, and Jian Wu. Robust training of graph neural networks via noise governance. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, page 607–615, New York, NY, USA, 2023. Association for Computing Machinery.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[Velikovi *et al.*, 2018] Petar Velikovi, Arantxa Casanova, Pietro Lio, Guillem Cucurull, Adriana Romero, and Yoshua Bengio. Graph attention networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[Wang *et al.*, 2019] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 322–330, 2019.

[Wang *et al.*, 2024] Zhonghao Wang, Danyu Sun, Sheng Zhou, Haobo Wang, Jiapei Fan, Longtao Huang, and Jiajun Bu. Noisygl: A comprehensive benchmark for graph neural networks under label noise. *Advances in Neural Information Processing Systems*, 37:38142–38170, 2024.

[Wei *et al.*, 2020] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint

training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13726–13735, 2020.

[Wei *et al.*, 2023] Xiaowen Wei, Xiuwen Gong, Yibing Zhan, Bo Du, Yong Luo, and Wenbin Hu. Clnode: Curriculum learning for node classification. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, pages 670–678, 2023.

[Wu *et al.*, 2024] Yuhao Wu, Jiangchao Yao, Xiaobo Xia, Jun Yu, Ruxin Wang, Bo Han, and Tongliang Liu. Mitigating label noise on graphs via topological sample selection. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[Yuan *et al.*, 2023] Jingyang Yuan, Xiao Luo, Yifang Qin, Yusheng Zhao, Wei Ju, and Ming Zhang. Learning on graphs under label noise. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[Zhang *et al.*, 2020] Mengmei Zhang, Linmei Hu, Chuan Shi, and Xiao Wang. Adversarial label-flipping attack and defense for graph neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 791–800, 2020.

[Zhu *et al.*, 2024] Yonghua Zhu, Lei Feng, Zhenyun Deng, Yang Chen, Robert Amor, and Michael Witbrock. Robust node classification on graph data with graph and label noise. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17220–17227, 2024.