NC-ALG: Graph-based Active Learning under Noisy Crowd

Wentao Zhang[1], Zhenbang You[1], Yexin Wang[1], Yang Li[1], Gang Cao[2], Zhi Yang[1], Bin Cui[1,3]
[1]School of CS, Peking University [2]Beijing Academy of Artificial Intelligence
[3]Institute of Computational Social Science, Peking University (Qingdao)
[1]{wentao.zhang, zhenbangyou, yexinwang,liyang.cs, yangzhi, bin.cui}@pku.edu.cn [2]caogang@baai.ac.cn

ABSTRACT

Graph Neural Networks (GNNs) have achieved a stride of success in various tasks but they heavily rely on a large number of annotated training nodes, requiring considerable human efforts. Despite the effectiveness of some previous GNN-based Active Learning (AL) methods, they assume that the labels annotated by oracles are always correct, which is contradictory to the error-prone labeling process in a practical crowdsourcing environment. Besides, due to such an impractical assumption, previous works only focus on optimizing the node selection in AL but neglect optimizing the labeling process under noisy crowd. Therefore, we present NC-ALG, a cost-effective framework that optimizes both the node selection and node labeling process under a noisy crowd. For node selection, NC-ALG introduces a new measurement to model influence reliability and an effective influence maximization objective to select more valuable nodes. For node labeling, NC-ALG significantly reduce the labeling cost by considering the model predicted labels and the labels of mirror nodes. To the best of our knowledge, this is the first attempt to consider GNN-based AL under the practical noisy crowd. Empirical studies on public datasets demonstrate that NC-ALG significantly outperforms the state-of-the-art AL methods in terms of both accuracy and labeling cost. Notably, it only takes NC-ALG one-third of the labeling budget that the competitive baseline GRAIN needs to achieve an accuracy of 70.7 % on PubMed.

CCS CONCEPTS

• **Mathematics of computing** → Graph algorithms; • **Computing methodologies** → **Active learning settings**.

KEYWORDS

Graph Neural Network, Active Learning, Crowdsourcing

ACM Reference Format:
Wentao Zhang, Zhenbang You, Yexin Wang, Yang Li, Gang Cao, Zhi Yang, Bin Cui. 2023. NC-ALG: Graph-based Active Learning under Noisy Crowd. In *Proceedings of Proceedings of the 2023 International Conference on Management of Data (SIGMOD '23).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

SIGMOD '23, June 18–23, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

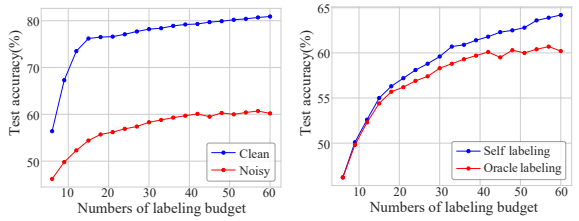
Graph Neural Network (GNNs) have achieved great success in a wide range of fields, including traffic network [21], biology [5], and social network [30], etc. Despite their popularity, GNNs typically require a large amount of labeled data to achieve satisfactory accuracy. However, acquiring these labels is a time-consuming, labor-intensive, and usually costly process. Therefore, how to annotate data more economically and efficiently becomes a critical challenge for graph data mining.

One of the most popular strategies to tackle this challenge is Active Learning (AL) [9]. With carefully designed sample selection strategies, AL can significantly reduce labeling costs by selecting the most valuable nodes to label. For example, AGE [3] is designed for active learning on GNNs, which adopts the uncertainty, density, and node degree to select nodes. ANRMAB [8] improves AGE by introducing a multi-armed bandit mechanism for adaptive decision-making. Considering the long training time of GNN, ALG [37] decouples the GNN model and proposes a new node selection metric that maximizes the effective reception field (RF). In addition, GRAIN [39] further generalizes RF to the number of activated nodes in social influence maximization, and it also introduces the diversity influence maximization for node selection.

All the above AL methods assume that the oracle will assign a correct label to each selected node. However, even human experts are not able to guarantee the correctness of all labels, and a more economical and commonly-used way in data annotation is the crowdsourcing platforms, such as Amazon Mechanical Turk (MTurk) [14] and CrowdFlower [26]. With these crowdsourcing platforms, users can have their samples annotated with a certain fee, which is much cheaper than employing an expert. However, labels obtained from crowdsourcing often include lots of noise (i.e., due to *noisy crowd*), thus posing new challenges to both node selection and node labeling in graph-based AL.

Node Selection. To better understand the impact of the noisy crowd, we randomly select 2 nodes for each class as the initial noisy labeled set, and gradually increase the labeling budget with ALG under the clean or noisy labels with an error rate of 40% on PubMed [20] dataset. We then train a two-layer GCN with a different set of labeled nodes. As shown in Figure 1(a), increasing the number of clean labels indeed improves accuracy (blue line). However, increasing the number of noisy labels does necessarily lead to accuracy improvement (red line). Therefore, *it is unsuitable to simply increasing influence propagation under a noisy crowd.*

Node Labeling. Previous graph-based AL methods pay much attention to the node selection but ignore the potential of model prediction in node labeling. However, under the noisy crowd, the incorrect node labels will possibly degrade the effectiveness of node labeling in AL. Surprisingly, after conducting lots of AL tasks, we observe that, *besides the oracle in a noisy crowd, the trained model*



(a) ALG with and without noise (b) ALG with and without self labeling

Figure 1: The influence of noisy labels in both node selection and node labeling.

itself may provide extra beneficial label information (especially when the model has been trained for a number of epochs). As shown in Figure 1(b), we gradually increase the labeling budget and select new nodes with ALG under a labeling error rate of 40% on PubMed. With the newly annotated nodes, we find that, by using some labels predicted by the model, a higher test accuracy can be achieved without any labeling cost. Therefore, a possible direction for cost-effective AL under a noisy crowd is to *co-design* the node selection and node labeling, i.e., leveraging the GNN model obtained during node selection to guide the node labeling.

In this paper, we propose NC-ALG, a graph-based AL framework that considers both node selection and node labeling to address these issues in the noisy crowd. For node selection, NC-ALG considers the consistency of model predicted labels and labels given by the crowd and then proposes a novel strategy for measuring influence reliability. NC-ALG considers the consistency of labels from the predictive model and the noisy crowd and then proposes a novel strategy for measuring influence reliability. Besides, it combines both the influence reliability and influence magnitude to measure the influence effectiveness, and proposes the effective influence maximization (EIM) objective to select valuable nodes. For node labeling, we co-design it with model prediction. Concretely, We reduce the labeling cost by 1) using the model predicted label and 2) relabeling the most similar node (i.e., mirror node) for the node with a low quality of model predicted label. We further improve the model training process by treating the influence reliability as sample weights in the objective loss function.

Contributions. The main contributions of this work can be summarized as follows: 1) To the best of our knowledge, we are the first to consider the GNN-based AL in the noisy crowd, and we propose a novel AL framework that could address the two aforementioned issues simultaneously. 2) We propose a novel method for measuring influence reliability by considering the consistency of model prediction and oracle labels. The concepts of influence effectiveness and an effective influence maximization objective are also proposed for node selection. 3) We connect model prediction and node labeling so that both the trained model and labeling cost can benefit interactively. 4) Experimental results on five open graph datasets show that NC-ALG outperforms RIM by a margin of 1.2-3.0% in terms of predictive accuracy, and it only requires a third of the labeling cost of GRAIN to achieve an accuracy of 70.7% on PubMed.

2 PRELIMINARY

2.1 Problem Formulation

Given a graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ with $|\mathcal{V}| = N$ nodes and $|\mathcal{E}| = M$ edges, feature matrix $\mathbf{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}..., \boldsymbol{x_N}\}$ in which $\boldsymbol{x_i} \in \mathbb{R}^d$, the node set $\mathcal{V}$ is partitioned into training set $\mathcal{V}_{train}$ (including both the labeled set $\mathcal{V}_l$ and unlabeled set $\mathcal{V}_u$), validation set $\mathcal{V}_{val}$ and test set $\mathcal{V}_{test}$. Suppose k is the number of label classes, the one-hot vector $\boldsymbol{y_i} \in \mathbb{R}^k$ is the ground-truth label for node v_i, $\mathcal{B}$ is the labeling budget, ℓ is the loss function to be optimized, graph-based AL algorithms aim to select a subset of nodes $\mathcal{V}_l \subset \mathcal{V}_{train}$ to label from the noisy crowd with the labeling accuracy α, so that the labels are used to train a model f with the lowest loss in the test set:

$$\underset{\mathcal{V}_l:|\mathcal{V}_l|=\mathcal{B}}{\arg\min}\ \mathbb{E}_{v_i\in\mathcal{V}_{test}}\left[\ell\left(\boldsymbol{y_i}, \hat{\boldsymbol{y}}_i\right)\right], \tag{1}$$

where $\hat{\boldsymbol{y}}_i = P(\hat{\boldsymbol{y}}_i|f)$ is the predicted label distribution of node v_i. Specifically, we focus on f being GNNs in this paper.

2.2 Graph Neural Networks

Each node in GNNs can aggregate the node embedding of its neighbors for embedding enhancement. Concretely, each GNN layer contains two operations: embedding propagation and embedding transformation, and these two processes can be formulated as

$$\mathbf{X}^{(k+1)} = \delta\left(\widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}\mathbf{X}^{(k)}\mathbf{W}^{(k)}\right), \tag{2}$$

where $\mathbf{X}^{(k)}$ and $\mathbf{X}^{(k+1)}$ are the embeddings of layer k and $k+1$ respectively, $\mathbf{X}$ (i.e., $\mathbf{X}^{(0)}$) is the original node feature. $\widetilde{\mathbf{D}}$ is the diagonal node degree matrix for normalization and $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ is the adjacent matrix with self connection, where $\mathbf{I}_N$ is the identity matrix. Specifically, the process of $\widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}\mathbf{X}^{(k)}$ is the embedding propagation and $\delta\left(\mathbf{X}^{(k)}\mathbf{W}^{(k)}\right)$ can be seen as the non-linear transformation, where $\mathbf{W}^{(k)}$ is a trainable weight matrix and $\delta(\cdot)$ is the activation function (e.g., ReLU).

2.3 Social Influence Maximization

Social influence maximization aims to select $\mathcal{B}$ influential individuals (seed nodes) that can maximize the diffusion of information or behaviors through a social network [1]. Concretely, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a seeding budget $\mathcal{B}$, it can be defined as

$$\max_{S}\ |\sigma(S)|,\ \text{s.t.}\ S \subseteq \mathcal{V},\ |S| = \mathcal{B}, \tag{3}$$

where $\sigma(S)$ is a set of nodes activated and influenced by the seed set S under the influence propagation models, such as Linear Threshold (LT) and Independent Cascade (IC) models [17]. This optimization process is known to be NP-hard, and previous work [24] also show that a greedy algorithm can provide an approximation guarantee of $(1-\frac{1}{e})$ if $\sigma(S)$ is nondecreasing and submodular with respect to S. Our proposed NC-ALG is the first work to connect the influence propagation with GNN-based AL under the more practical noisy crowd by defining the effective influence.

2.4 Node Labeling with Crowdsourcing

Most AL methods assume the expert annotates the data without labeling error [27]. However, such a requirement is impractical due to the expensive labeling cost of the expert. Besides, even the

experienced expert cannot guarantee the correctness of all labels for some complex graphs [28]. To improve the labeling efficiency and labeling cost, a more practical and convenient way for data annotation is crowdsourcing [26]. For example, the popular ImageNet dataset is annotated by 49K workers recruited from Amazon Mechanical Turk over three years [34]. Despite its popularity, the labels obtained from crowdsourcing are often imperfect, and the label noise will also be introduced [9]. Therefore, NC-ALG is proposed to train a robust and accurate GNN model while minimizing the data annotation effort in a noisy crowd.

2.5 Active Learning

Common AL strategies. Active learning is proposed to select the most valuable samples to label for the classifier, and they have been widely used in many data mining tasks.

One commonly used node selection strategy in AL is informativeness. Follow this criterion, Uncertainty Sampling [40] selects the nodes which are least certain how to label, e.g., the nodes which have the highest entropy of the predicted softmax outputs. As the single model prediction is not robust to select the most informative nodes, Query-by-Committee [23] trains a committee of models, and the selected node is considered to be the instance about which they most disagree. Since informativeness-based methods are more likely to select the outliers, recent AL algorithms also take the representativeness into consideration, such as density-based [29, 40] and clustering-based [7] strategies. Recently, some other AL algorithms [15, 32] propose to incorporate the diversity for the pooling-based node selection.

GNN-based AL strategies. Although the above AL algorithms are general and effective, directly applying them to GNN will lead to sub-optimal accuracy because the graph characteristics have not been considered. Therefore, many GNN-based methods have been proposed to tackle this issue.

AGE is the first to consider GNN-based AL by introducing the density of node embedding and PageRank centrality. Besides, ANRMAB proposes a multi-armed bandit mechanism for adaptive decision combinations for the query strategies used in ALG. Considering the key characteristic of GCN, ALG further introduces the concept of RF to the query strategies and get better accuracy than AGE and ANRMAB. Recently, GRAIN has connected GNN-based AL with social influence maximization. It also proposes the diversified influence maximization objective to select nodes in a model-free manner. However, the effectiveness of these methods is based on the impractical assumption that all the labels annotated by oracle are correct, and NC-ALG is the first work to consider GNN-based AL under a more practical noisy crowd.

AL with Noise. Existing AL works [2, 36] pertaining to label noise primarily concentrates on the following stages: 1) noise detection and 2) noise handling.

In terms of noise detection, data-driven and model-based approaches have been proposed. Methods belonging to the former class build a graph based on the dataset as the first step and then make use of graph properties, e.g., the homophily assumption [22], that is, labels that corrupt the graph structure are likely to be incorrect [25]. Besides, methods in the latter class [36] evaluates the probability of noise by the soft labels predicted by models.

When it comes to noise handling, current works can generally be divided into three classes: data correction, objective function modification, and optimization policy modification [11]. However, none of them is specifically devised for graphs, thus unable to take the influence quantity caused by the graph structure into account, resulting in non-ideal accuracy.

3 COST-EFFECTIVE NODE LABELING

3.1 Self Labeling

As shown in Figure 1(b), as the model itself becomes more accurate with the increased node labels, we may get a more accurate label than the noisy crowd. Therefore, rather than annotating all the labels by the crowdsourcing platforms, we propose to directly use the label predicted by the trained model according to the influence reliability. Concretely, supposing the label reliability for the unlabeled node v_i given by the trained GNN model is β_i and the labeling accuracy of oracle is α, we set its label as

$$y_i = P(\bar{\boldsymbol{y}}_i|f), \quad \forall \beta_i \geq \alpha, \tag{4}$$

where $P(\bar{\boldsymbol{y}}_i|f)$ is the predicted pseudo label (in one-hot version). Compared with the labels annotated by the noisy crowd, labels given by the trained model is cost-free.

3.2 Mirror Node Labeling

To deal with the label noise introduced by the noisy crowd, a commonly used ground truth inference method is relabeling the uncertain nodes by another oracle who hasn't seen this sample before. Even such a method can get relatively more accurate labels, and they will cost twice the labeling cost to annotate the same number of nodes. Motivated by the idea that nodes share similar graph structures and features are more likely to have the same label, for each uncertain node, we propose to annotate its most similar node rather the relabeling the node itself.

Concretely, we first find the uncertain nodes, which is hard to estimate its label by just the trained model or oracle.

Definition 1 (**Uncertain Nodes**). *Given a batch of selected nodes set $\mathcal{V}_b$ and a node v_i, supposing $\hat{y}_i$ and $\tilde{y}_i$ are the labels given by the trained model and noisy oracle respectively, the uncertain nodes set are defined as*

$$\{v_i | \hat{y}_i \neq \tilde{y}_i, \gamma \leq \beta_i < \alpha\}, \forall v_i \in \mathcal{V}_b, \tag{5}$$

where γ is a threshold that measures the usability of $\hat{y}_i$. We assume $\hat{y}_i$ is useless if its reliability β_i is smaller than γ.

We find the most similar node in both graph structure and feature for each uncertain node and define this node as a mirror node. According to Eq. (2), $\mathbf{X}^{(k)}$ contains the graph information of each node itself and its k-hop neighbourhoods. After k iterations of propagation, we calculate the similarity of node v_i and v_j in GNN by measuring their similarity in the graph embedding $\mathbf{X}^{(k)}$. However, the parameter-based graph embedding $\mathbf{X}^{(k)}$ is unreliable in the several initial batches when the GNN model is under-fitted with only a few available node labels. Fortunately, previous work SGC [31] reveals that the true success of GNN lies in the feature propagation rather than the non-linear transformation. Therefore, we remove the activate function $\delta(\cdot)$ and the trainable weight $\mathbf{W}^{(k)}$ in Eq. 2, and get the parameter-free graph embedding as: $\hat{\mathbf{X}}^{(k+1)} = \widetilde{\mathbf{D}}^{-1}\widetilde{\mathbf{A}}\hat{\mathbf{X}}^{(k)}$.

Definition 2 (**Mirror Node**). *Given an unlabeled node v_i, its mirror node v_j is defined as*

$$v_j = \arg\min_{v_j \in \mathcal{V}_u} \left(\theta_t S(\hat{\mathbf{X}}_i^{(k)}, \hat{\mathbf{X}}_j^{(k)}) + (1-\theta_t) S(\mathbf{X}_i^{(k)}, \mathbf{X}_j^{(k)}) \right), \tag{6}$$

where $\mathbf{X}_i^{(k)}$ and $\mathbf{X}_j^{(k)}$ are the node embedding of v_i and v_j respectively, and $S(\cdot)$ is the cosine similarity. Besides, $\theta_t = \cos\left(\frac{\pi t}{2T}\right)$ controls the weight of similarity value in the t-th iteration of the total T batches, and decreases along with the AL iterations. We gradually move the attention from $\hat{\mathbf{X}}^{(k)}$ to $\mathbf{X}^{(k)}$ in the calculation of node similarity because the parameter-based graph embedding $\mathbf{X}^{(k)}$ becomes more and more accurate as the AL iteration t increases.

With such a mirror node labeling mechanism, we can annotate two similar nodes with high confidence. The labeled nodes in a k-layer GNN can propagate the label supervision to their k-hop neighborhood nodes. Therefore, it can boost the model accuracy by getting more unlabeled nodes involved in GNN training.

3.3 Summary

We annotate the selected node based on 1) the consistency of the label $\hat{y}_i$ predicted by the GNN model and the label $\tilde{y}_i$ given by noisy crowd. 2) the label reliability β_i given by the model, the labeling accuracy α of the noisy crowd, and the threshold γ. Since the label given by the model may be more accurate than the noisy oracle, we first assign the unlabeled node v_i with the label predicted by the model $\hat{y}_i$ when $\beta_i \geq \alpha$. After the self-labeling process, we annotate the node v_i in the remaining unlabeled nodes below.

- *Case 1:* $\hat{y}_i = \tilde{y}_i$. As the label given by the model and crowd is consistent, we annotate the label of node v_i with $\tilde{y}_i$.
- *Case 2:* $\hat{y}_i \neq \tilde{y}_i$, *and* $\beta_i < \gamma < \alpha$. Since the reliability of model predicted label is apparently lower than the noisy crowd, we annotate the label as $\tilde{y}_i$.
- *Case 3:* $\hat{y}_i \neq \tilde{y}_i$, *and* $\gamma \leq \beta_i < \alpha$. In this case, we find the mirror node v_j and label it by another new oracle. If the newly annotated label $\tilde{y}_j$ is same to the previous label $\hat{y}_i$ or $\tilde{y}_i$, we assign $\tilde{y}_i$ to both node v_i and its mirror node v_j. Otherwise, we discard both $\tilde{y}_i$ and $\tilde{y}_j$.

4 EIM-AWARE NODE SELECTION

4.1 Influence Reliability Estimation

We first estimate the reliability of label annotated by the model and oracle, and then the influence reliability can be estimated by considering the following two factors.

Reliability of model annotation. Usually, the model prediction is more reliable with more powerful GNN models, i.e., models with higher predictive accuracy in the validation set. However, even a powerful base model is likely to make mistakes on nodes with low prediction confidence. As a result, we propose to estimate of label reliability of node v_i by considering 1) Acc_{val}: the predictive ability of the model itself and 2) p_i: the model prediction confidence for each specific node.

Suppose Acc_{val} is the GNN model's predictive accuracy in the validation set, and p_i is the maximum value of the predicted softmax outputs in node v_i. We define the label reliability as

$$\beta_i = p_i \cdot \text{Acc}_{val}. \tag{7}$$

A larger β_i means the model prediction is more reliable.

Reliability of oracle annotation. As discussed in Section 3.2, we label the mirror node if $\hat{y}_i \neq \tilde{y}_i$ and $\gamma \leq \beta_i < \alpha$. Therefore, the oracle will give two labels for the selected uncertain node and mirror node, respectively. So we propose to estimate the reliability of the oracle annotated label by considering the consistency of the model prediction along with the previous annotated label.

If there is only one label, we just set the label reliability as α. Otherwise, the newly annotated label $\tilde{y}_i$ is identical to the previous label $\hat{y}_i$ or $\tilde{y}_i$, and we calculate the reliability of it and regard it as the combined reliability.

Theorem 4.1 (**Reliability of oracle annotation**). *Given the number of classes k, and the oracle labeling accuracy α, the combined label reliability α_i' as*

$$\alpha_i' = \begin{cases} \alpha & \text{the oracle gives one label} \\ \frac{\alpha^2}{\alpha^2 + \frac{(1-\alpha)^2}{k-1}} & \text{two oracles give the same label} \\ \frac{\alpha}{1+\alpha} & \text{two oracles give different labels.} \end{cases} \tag{8}$$

Proof of Theorem 4.1 is included in Appendix A.3. Intuitively, a larger α_i' means the label $\tilde{y}_i$ is more reliable. Specifically, the oracle gives one label when 1) $\hat{y}_i = \tilde{y}_i$ and $\beta_i < \alpha$, and 2) $\hat{y}_i \neq \tilde{y}_i$, and $\beta_i < \gamma < \alpha$. These two oracles give the same label means $\hat{y}_i \neq \tilde{y}_i$, $\gamma < \beta_i < \alpha$ and $\tilde{y}_j = \tilde{y}_i$. In addition, the two oracles output different labels means $\tilde{y}_j \neq \hat{y}_i$, and $\tilde{y}_j = \hat{y}_i$.

Influence Reliability. Now we have exactly one label from the oracle with the reliability α_i' and one from the model with the reliability β_i. Further, we can calculate the final influence reliability of the label we pick if applicable (we discard all labels in some cases). As discussed in Section 3.3, we assign the node v_i with the label $\hat{y}_i$ from the model when $\beta_i \geq \alpha$. Otherwise, we use the label annotated by the oracle or discard the label.

Theorem 4.2 (**Influence Reliability**). *Given the reliability of model annotation β_i and oracle annotation α_i', the final influence reliability r_i of node v_i is defined as*

$$r_i = \begin{cases} \beta_i & \beta_i \geq \alpha \\ \frac{\alpha_i'(1-\beta_i)}{1-\alpha_i'\beta_i} & \beta_i < \alpha, \hat{y}_i \neq \tilde{y}_i \\ \frac{\alpha_i'\beta_i}{\alpha_i'\beta_i + \frac{(1-\alpha_i')(1-\beta_i)}{k-1}} & \beta_i < \alpha, \hat{y}_i = \tilde{y}_i. \end{cases} \tag{9}$$

Proof of Theorem 4.2 is provided in Appendix A.4. Specifically, if the oracle annotates the mirror node v_j, we set its influence reliability r_j to r_i due to their high similarity. Based on the prediction from both the model and oracle, r_i combines α_i' and β_i, and thus is more accurate to estimate the label reliability. The larger r_i is, annotating r_i will bring more reliable influence to its neighbors.

4.2 Influence Magnitude Estimation

We measure the influence magnitude of a node v_i on v_j by how much change in the input feature of v_i affects the *aggregated* feature/label of v_j after k iterations propagation [33].

Definition 3 (**Influence Magnitude**). *The influence magnitude score of node v_i on node v_j after k-step propagation is the L1-norm*

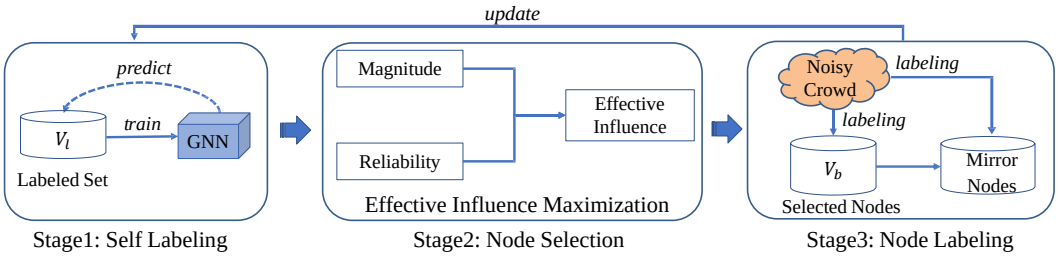


Figure 2: An overview of the NC-ALG framework.

of the expected Jacobian matrix $\hat{I}(v_j, v_i, k) = \left\| \mathbb{E}[\partial \mathbf{X}_j^{(k)} / \partial \mathbf{X}_i^{(0)}] \right\|_1$. *Formally, the normalized influence magnitude score is defined as*

$$I(v_j, v_i, k) = \frac{\hat{I}(v_j, v_i, k)}{\sum_{v_w \in \mathcal{V}} I(v_j, v_w, k)}. \tag{10}$$

After k-step feature propagation as (2) in GNNs, the influence magnitude score $I(v_j, v_i, k)$ captures the sum over probabilities of all possible influential paths from node v_i to v_j. Besides, $I(v_j, v_i, k)$ can be treated as the probability that a random walk starts at v and ends at u after taking k steps. It is harder to get a path from v_i to v_j with a larger step k, and $I(v_j, v_i, k)$ will gradually decay along with the increasing propagation step. Intuitively, a larger $I(v_j, v_i, k)$ requires a higher similarity of node v_i and v_j in their graph structure, and it also means node v_i has sufficient impact on v due to a large number of influence paths to propagate labels.

4.3 Effective Influence

Unlike the previous social influence method, which only considers the influence magnitude [16, 35], we measure the effective influence $E(v_j, v_i, k)$ by taking both the influence magnitude and influence reliability into consideration.

Definition 4 (**Effective Influence Score**). *Given the influence reliability r_i and influence magnitude $I(v_j, v_i, k)$, the effective influence score of node v_i on node v_j after k-step propagation is*

$$E(v_j, v_i, k) = r_i I(v_j, v_i, k). \tag{11}$$

The effective influence score $E(v_j, v_i, k)$ is positively correlated with the influence reliability r_i and influence magnitude $I(v_j, v_i, k)$. Therefore, node v_j can get a highly effective influence from the labeled node v_i if they are close neighbors in the graph structure, and the influence of v_i is more reliable.

As the node class is dominated by the maximum value in the predicted label distribution $P(\hat{\boldsymbol{y}}_i|f)$, we assume an unlabeled node v_j can be activated if

$$E(v_j, \mathcal{V}_l, k) \geq \theta, \tag{12}$$

where $E(v_j, \mathcal{V}_l, k) = \max_{v_i \in \mathcal{V}_l} E(v_j, v_i, k)$ is the maximum effective influence of node $v_i \in \mathcal{V}_l$ on the node v_j, and the threshold θ is a hyper-parameter to be tuned.

Definition 5 (**Activated Nodes**). *Given the effective influence score $E(v, \mathcal{V}_l, k)$, the activated node set $\sigma(\mathcal{V}_l)$ is a subset of nodes in $\mathcal{V}$ that can be activated by the labeled nodes set $\mathcal{V}_l$:*

$$\sigma(\mathcal{V}_l) = \bigcup_{v \in \mathcal{V}, E(v, \mathcal{V}_l, k) \geq \theta} \{v\}. \tag{13}$$

Similar to ALG, the threshold $\theta = 0$ is equal to maximize the receptive field, and the target in such a case is to get more nodes involved in the semi-supervised GNN training. However, some nodes may be weakly supervised and lead to sub-optimal accuracy if we only consider the receptive field. We should increase θ to a larger value to encourage that the activated nodes can get sufficient effective influence when the labeling budget $\mathcal{B}$ is large.

4.4 Effective Influence Maximization

To make more nodes sufficiently influenced in the semi-supervised GNN training, we aim to select and annotate a subset $\mathcal{V}_l$ from $\mathcal{V}$ so that more nodes can be activated according to Eq. (13). Specifically, we optimize this problem by defining the effective influence maximization objective as below.

EIM Objective. Specifically, NC-ALG adopts an effective influence maximization objective (EIM):

$$\max_{\mathcal{V}_l} F(\mathcal{V}_l) = |\sigma(\mathcal{V}_l)|, \text{s.t. } \mathcal{V}_l \subseteq \mathcal{V}, \ |\mathcal{V}_l| = \mathcal{B}. \tag{14}$$

Considering the influence reliability and influence magnitude, the proposed EIM objective can be used to find a subset $\mathcal{V}_l$ that can activate nodes as more as possible.

5 NC-ALG FRAMEWORK

5.1 Framework Overview

As shown in Figure 2, our proposed NC-ALG framework contains three stages: self labeling, node selection, and node labeling. Without losing generality, we consider a batch setting where b nodes are selected in each iteration. For the first batch, when the initial labeled set $\mathcal{V}_0 = \emptyset$, we ignore the influence reliability r_i in Eq. (11) since there is no model prediction for measuring the influence reliability. We first label some nodes with the confident model prediction for other batches and then select nodes that maximize the EIM objective. At last, the noisy crowd is used to annotate the selected nodes set $\mathcal{V}_b$ and then the labeled set $\mathcal{V}_l$ is updated.

5.2 Framework Pipeline

Self Labeling. With the trained GNN, each unlabeled node v_i can be annotated with a label $\hat{y}_i$ predicted from the model and the label reliability β_i (Lines 3-4). For each high confidence node with $\beta_i \geq \alpha$, we directly annotate it with $\hat{y}_i$ because it is more reliable than the annotation of the noisy crowd (Line 5). Next, we update its influence reliability r_i, remove it from the training set $\mathcal{V}_{train}$, and add it to the labeled set $\mathcal{V}_l$ (Lines 6-7). At last, we update the activated nodes set according to Eq. (13) (Line 8).

Node Selection. For better efficiency in each batch node selection, we set the influence quality of each node to the labeling accuracy α during the node selection process (Line 10) and then simultaneously update these values according to Eq. (9) during the next

Algorithm 1: Pipeline of NC-ALG

Input: Initial labeled set $\mathcal{V}_0$, query batch size b, and labeling accuracy α.
Output: Labeled set $\mathcal{V}_l$

1 $\mathcal{V}_l = \mathcal{V}_0, \mathcal{V}_b = \emptyset$;
2 **Stage 1: Self Labeling**
3 **for** $v_i \in \mathcal{V}_{train} \setminus \mathcal{V}_0$ **do**
4 Get the reliability β_i according to Eq. (7);
5 Label v_i with $\hat{y}_i$ if $\beta_i \geq \alpha$;
6 Update the influence reliability according to Eq. (9);
7 $\mathcal{V}_l = \mathcal{V}_l \cup \{v_i\}$, and $\mathcal{V}_{train} = \mathcal{V}_{train} \setminus \{v_i\}$;
8 Update the activated nodes set according to Eq. (13);
9 **Stage 2: Node Selection**
10 $\forall v_i \in \mathcal{V}_{train} \setminus \{v_i\}$, set the influence quality r_i to α;
11 **for** $t = 1, 2, \ldots, b$ **do**
12 Select the most valuable node: $v_i = \arg\max_{v \in \mathcal{V}_{train} \setminus \mathcal{V}_l} F(\mathcal{V}_l \cup \{v\})$;
13 $\mathcal{V}_l = \mathcal{V}_l \cup \{v_i\}, \mathcal{V}_b = \mathcal{V}_b \cup \{v_i\}, \mathcal{V}_{train} = \mathcal{V}_{train} \setminus \{v_i\}$;
14 **Stage 3: Node Labeling**
15 **for** $v_i \in \mathcal{V}_b$ **do**
16 Label v_i with $\tilde{y}_i$;
17 **if** $\tilde{y}_i = \hat{y}_i$ *or* $\tilde{y}_i \neq \hat{y}_i$ & $\beta_i \leq \gamma$ **then**
18 Update r_i according to Eq. (9);
19 Update $\sigma(\mathcal{V}_l)$ according to Eq. (13);
20 **else**
21 Find the mirror node v_j according to Eq. (6);
22 Label v_j with $\tilde{y}_j$, and **if** $\tilde{y}_j = \tilde{y}_i$ *or* $\tilde{y}_j = \hat{y}_i$ **then**
23 label v_i and v_j with $\hat{y}_j$, and update $\mathcal{V}_l$ and $\mathcal{V}_{train}$;
24 Update r_i and r_j according to Eq. (9);
25 Update $\sigma(\mathcal{V}_l)$ according to Eq. (13);
26 **else**
27 Discard both $\hat{y}_i$ and $\hat{y}_j$, and set $\mathcal{V}_l = \mathcal{V}_l \setminus \{v_i\}$;
28 **return** $\mathcal{V}_l$

node labeling stage. Given the training set $\mathcal{V}_{train}$, and query batch size b, we first select the node v_i, which generates the maximum marginal gain (Line 12). We then update the labeled set $\mathcal{V}_l$, the oracle annotated set $\mathcal{V}_b$, and the training set $\mathcal{V}_{train}$ (Line 13).

Theorem 5.1. *The greedily selected batch node set is within a factor of* $(1 - \frac{1}{e})$ *of the optimal set for the objective of effective influence maximization (EIM).*

Proof of Theorem 5.1 is provided in Appendix A.5. For monotone and submodular F, the selected node set $\mathcal{V}_b$ is within a factor of $(1 - \frac{1}{e})$ of the optimal set $\mathcal{V}_b{}^*$: $F(S) \geq (1 - \frac{1}{e})F(\mathcal{V}_b{}^*)$.

Node Labeling. After getting a batch of labeled nodes, we acquire the label from a noisy crowd (Line 16). First, we use the label given by the oracle when 1) the label $\hat{y}_i$ predicted from model is equal to the oracle label $\tilde{y}_i$; 2) $\hat{y}_i \neq \tilde{y}_i$, but the reliability of model prediction β_i is equal or smaller than a defined parameter γ (Line 17). We then update the corresponding influence reliability r_i and activated node set $\sigma(\mathcal{V}_l)$ according to Eq. (9) and Eq. (13) respectively. For node v_i who gets inconsistent label from the model and oracle with higher reliability than γ, we find and label its mirror node v_j with another new oracle. If the newly annotated label $\tilde{y}_j$ equals to the previous label $\hat{y}_i$ or $\tilde{y}_i$, we annotate both v_i and v_j with $\tilde{y}_j$ according to the majority voting mechanism. Correspondingly, $\mathcal{V}_l, \mathcal{V}_{train}, r_i, r_j$ and $\sigma(\mathcal{V}_l)$ will also be updated. Moreover, if the newly annotated label $\tilde{y}_j$ is different to both $\hat{y}_i$ and $\tilde{y}_i$, we discard both $\hat{y}_i$ and $\hat{y}_j$ because the label noise may be introduced to degrade the model accuracy. At last, the labeled set $\mathcal{V}_l$ is returned.

Efficiency Optimization. By leveraging the existing works [18, 19] on scalable and parallelizable social influence maximization, we could enable NC-ALG to effectively deal with large-scale graphs. The key idea is to identify and dismiss uninfluential nodes to reduce the computation cost for evaluating influence propagation. For example, we can use node degree to filter out a vast number of uninfluential nodes in calculating the mirror nodes.

Model Training. Intuitively, a node with larger influence quality in $\mathcal{V}_l$ should contribute more to the training process, so we introduce the influence quality in the model training. For GNN, we use the weighted cross-entropy loss as follows

$$\mathcal{L} = -\sum_{v_i \in \mathcal{V}_l} r_i \boldsymbol{y_i} \log \hat{\boldsymbol{y}}_i, \tag{15}$$

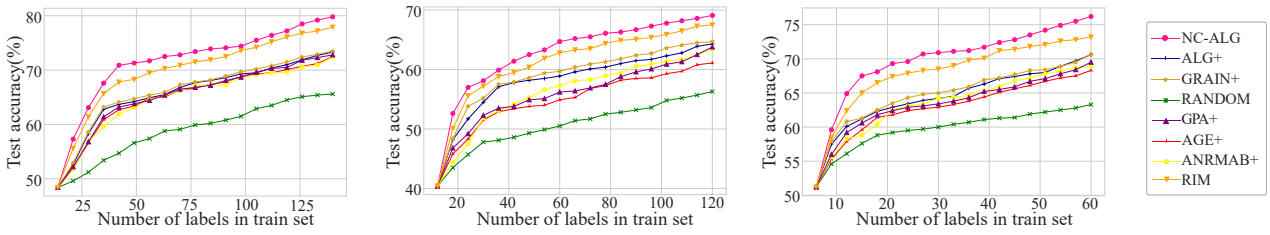
where r_i is the influence quality of node v_i. As the GNN model becomes more accurate, the unlabeled nodes can correspondingly get more reliable model predictions. This brings two benefits: 1) more nodes can be directly annotated by the trained model, thus further reducing the labeling budget; 2) the influence reliability becomes better for the labeled nodes, so each labeled node is more informative for the model training.

6 EXPERIMENTS

We conduct comprehensive experiments of NC-ALG on five real-world graphs to show that: 1) Compared with the SOTA method, NC-ALG can achieve better predictive accuracy. 2) the ablation study about each component in NC-ALG. 3) NC-ALG can generalize well to different GNN models. 4) The reasons for NC-ALG being more effective than the baselines.

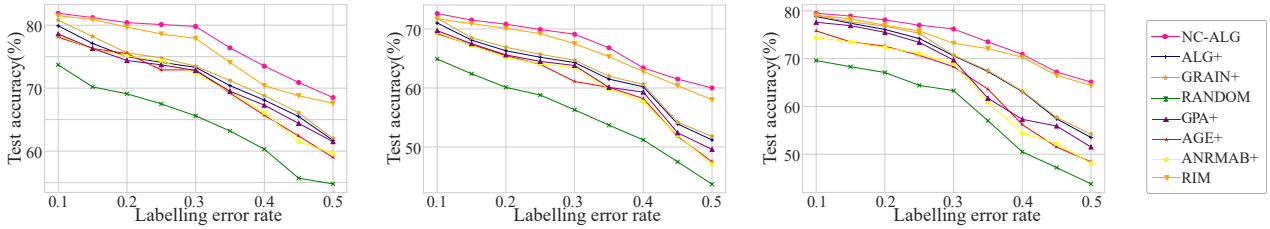
6.1 Experiment Setup

Datasets and Baselines. We use node classification tasks to evaluate NC-ALG in both inductive and transductive settings [10] on three citation networks (Citeseer, Cora, and PubMed) [20], one large social network (Reddit) and an OGB dataset (ogbn-arxiv) [13]. The properties of these datasets are summarized in Appendix B. We compare NC-ALG with the following baselines: (1) **Random**: Randomly select the nodes to query; (2) **AGE** [3]: Combine different query strategies linearly with time-sensitive parameters for GCN; (3) **ANRMAB** [8]: Adopt a multi-armed bandit mechanism for adaptive decision making to select nodes for GCNs; (4) **GPA** [12]: Jointly train on several source graphs and learn a transferable active learning policy which can directly generalize to unlabeled target graphs; (5) **ALG** [37]: Select a node that maximizes the effective RF; (6) **GRAIN** [39]: Select nodes that benefit both influence magnitude and influence diversity. (7) **RIM** [38]: a model-free anti-noise active learning algorithm on graphs.



(a) Cora (b) Citeseer (c) PubMed

Figure 3: The test accuracy (in %) of each method along with the increasing labeling budget for the noisy crowd when the labeling accuracy is 70% on different graphs.



(a) Cora (b) Citeseer (c) PubMed

Figure 4: The test accuracy with different labeling error rate of labeled nodes for GCN.

Table 1: The test accuracy (%) on different datasets when labeling accuracy is 70% using the same labeling cost.

Method	Cora	Citeseer	PubMed	Reddit	ogbn-arxiv
Random	65.6	56.3	63.3	75.2	47.7
AGE+	72.5	61.1	68.3	77.6	53.9
ANRMAB+	72.4	63.4	68.9	77.2	54.1
GPA+	72.8	63.8	69.7	77.9	56.3
ALG+	73.3	64.3	70.6	78.2	57.4
GRAIN+	73.5	64.7	70.7	78.6	57.8
RIM	77.9	67.5	73.2	80.1	60.8
NC-ALG	**79.8**	**69.1**	**76.2**	**81.3**	**62.4**

None of these baselines has considered the label noise in AL. we have equiped these methods with anti-noise machanisms [11, 25] to alleviate the influence of noise in GNN for fairness, and all of them bring improvement on accuracy. Finally, we choose PTA [6] and incorporate it into our baselines since it can get the best accuracy on most methods and datasets. We name AGE enhanced with PTA as AGE+, so do other baselines. Similarly, PTA computes dynamic label reliability based on the graph proximity and the similarity of the prediction and then applies it as weight in the weighted training loss function.

Implementations. To be fair, we either use the grid search for each method to find the optimal hyper-parameters or follow the setting in the original papers. We repeat each method ten times and report the mean test accuracy to eliminate randomness. And the initial node sets are selected with the same error rate. The implementation details are shown the Appendix C.

6.2 Results Comparison

End-to-end Comparison. We choose the labeling budget as 20 labels per class to show the end-to-end accuracy, and then we report the test accuracy when the labeling accuracy is set as 70%. Table 1 shows that RIM, GRAIN+, ALG+, GPA+, AGE+, and ANRMAB+ outperform Random in all the datasets, as they are specially designed for GNNs. GRAIN+ and ALG+ perform better than other baselines except for NC-ALG because they consider the RF. NC-ALG and RIM further boosts the accuracy by a significant margin, and NC-ALG performs better than RIM, because it is model-based and uses self-training. Notably, NC-ALG improves the test accuracy of the best baseline, i.e., RIM, by 1.6-3.0% on the three citation networks, 1.2% on Reddit, and 1.6% on ogbn-arxiv.

Accuracy under Different Labeling Budgets. To show the influence of the labeling budget, we test the accuracy of different AL methods under different labeling budgets on three citation datasets. More concretely, we range the number of oracle labels from $2k$ to $20k$ with the labeling accuracy of 70% and show the test accuracy of GCN in Figure 3. The experimental results demonstrate that NC-ALG consistently outperforms other baselines with the increase of labeling budget.

Accuracy under Different Noisy Rates. To show the influence of oracle noise, we choose the labeling budget as 20 nodes per class with the labeling error rate ranging from 0.1 to 0.5, and then report the corresponding test accuracy of GCN in Figure 4. Compared to other baselines, NC-ALG consistently outperforms the baselines as the labeling error rate grows. Besides, with the labeling error rate increasing, the accuracy gap grows larger, which shows the robust anti-noise ability of NC-ALG.

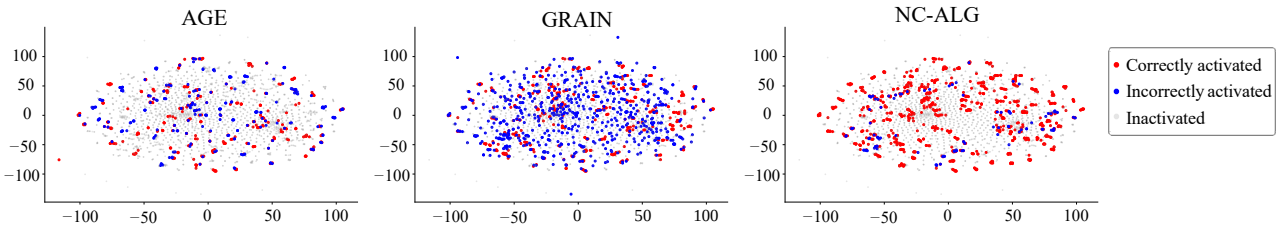


Figure 5: The node distribution of different methods on Citeseer.

Table 2: Test accuracy (%) of different models on PubMed.

Method	SGC	APPNP	GCN	MVGRL
Random	59.9	63.7	63.3	63.9
AGE+	66.2	68.4	68.3	68.5
ANRMAB+	66.3	69.5	68.9	69.7
GPA+	68.1	69.9	69.7	69.9
ALG+	69.4	71.1	70.6	71.3
GRAIN+	69.8	71.3	70.7	71.4
RIM	71.6	73.4	73.2	73.7
NC-ALG	**75.8**	**76.5**	**76.2**	**76.7**

Table 3: Influence of different components in test accuracy(%).

Method	Cora	Δ	Citeseer	Δ	PubMed	Δ
w/o RT	76.8	-3.0	66.0	-3.1	73.5	-2.7
w/o RS	76.2	-3.6	65.3	-3.8	73.2	-3.0
w/o RTS	74.8	-5.0	63.4	-5.7	70.9	-5.3
w/o SL	75.9	-3.9	69.1	–	72.8	-3.4
w/o MN	77.2	-2.6	66.2	-2.9	73.9	-2.3
NC-ALG	**79.8**	–	**69.1**	–	**76.2**	–

6.3 Generalization Evaluation

Besides of GCN, NC-ALG can also be applied to a large variety of GNN variants. GCN, SGC, MVGRL, and APPNP are four representative GNNs [4] which adopt different message passing mechanisms. Unlike the coupled GCN, both SGC and APPNP are decoupled, and their difference is the order of feature propagation and transformation. Besides, MVGRL is a classic self-supervised GNN. We test the generalization ability of NC-ALG by evaluating the four aforementioned GNNs on $20k$ labeled nodes selected by NC-ALG and other baselines in the AL scenario, and the corresponding results are shown in Table 2. The results suggest that NC-ALG consistently outperforms the other baselines. Therefore, we conclude that NC-ALG can generalize to different types of GNNs well.

6.4 Ablation Study

NC-ALG combines reliable selecting, reliable training, and self-labeling in the method. To verify the necessity of each component, we evaluate NC-ALG on GCN while disabling one component at a time when the labeling accuracy is 70%. We evaluate NC-ALG: *(i)* without the influence reliability score serving as the loss weight (called "w/o Reliable Training (RT)"); *(ii)* without the influence reliability when selecting the node (called "w/o Reliable Selection (RS)"); *(iii)* without influence reliability when training model and selecting node (called "w/o Reliable Training and Selecting (RTS)"); *(iv)* without self-labeling(called "w/o Self-labeling (SL)"); *(v)* without mirror node(called "w/o Mirror Node (MN)"). Table 3 shows the results of these different settings.

From Table 3, we have that: 1) If reliable training is ignored, the test accuracy will decrease in all three datasets, e.g., the accuracy gap is as large as 3.0% on Cora. This is because reliable training help avoiding the bad influence from the nodes with low influence reliability. 2) Reliable node selection significantly impacts model accuracy on all datasets, and it is more important than reliable training since removing the reliable node selection will lead to a more substantial accuracy gap. For example, the gap on Citeseer is 3.8%, which is higher than the other gap (3.1%). The more reliable the label is, the more reliably activated nodes we can use to train the GCN. 3) Lacking reliability in both model training and node selection will lead to worse accuracy, proving the former conclusion's correctness. 4) With the removal of self-labeling, the reliable nodes in the training set become fewer, leading to lower accuracy, e.g., the accuracy on Cora decreases by 3.9%. When the oracle accuracy is 70% on PubMed, label reliability given by the model is always lower than the oracle, so self-labeling is not applicable here. 5) The accuracy will decrease by a large margin without the mirror node, showing its benefit in node labeling.

6.5 Interpretability

To show the insight of NC-ALG, we evaluate the distribution of the correctly activated nodes, incorrectly activated nodes, and inactivated nodes for three methods: AGE, GRAIN, and NC-ALG when the labeling accuracy is 70% for Citeseer in GCN. Note that a node is correctly activated only if it is firstly activated by a correctly labeled node. The result in Figure 5 shows that AGE has the fewest activated nodes, and nearly half of them are incorrectly activated. Specifically, NC-ALG positively activates 654 nodes and negatively activates 143 nodes, whereas AGE positively activates 174 nodes and negatively activates 262 nodes. Besides, GRAIN has the most activated nodes but many of them are activated by incorrectly labeled nodes. Compared to them, NC-ALG has enough activated nodes, and most of them are activated by correctly labeled nodes (i.e., the noisy propagation is restrained), which could explain the better performance of NC-ALG in node classification.

7 CONCLUSION

Despite the popularity of GNNs in many practical applications, they usually require a large amount of labeled data, which is time-consuming and laborious. Although some GNN-based AL methods are proposed to tackle this issue, they all assume the absolutely correct oracle and thus fail to deal with the label noise under a more practical and noisy crowd circumstance. This paper proposes NC-ALG, a novel AL method that connects node selection with social influence maximization under a noisy crowd. NC-ALG makes the first step in this direction by showing the feasibility and the potential of such a connection. Concretely, based on the prediction consistency between the model and oracle, we first propose a novel method for measuring the influence reliability. Besides, we combine the influence reliability and influence magnitude and propose to select the node that activates more nodes. Finally, the self-labeling and mirror node labeling mechanisms are designed to reduce the labeling cost. Empirical studies on real-world graphs show that NC-ALG outperforms competitive baselines by a large margin, especially when the noisy rate is high.

ACKNOWLEDGMENTS

To Robert, for the bagels and explaining CMYK and color spaces.

REFERENCES

[1] Sinan Aral and Paramveer S Dhillon. 2018. Social influence maximization under empirical influence models. *Nature human behaviour* 2, 6 (2018), 375–382.

[2] Mohamed-Rafik Bouguelia, Slawomir Nowaczyk, KC Santosh, and Antanas Verikas. 2018. Agreeing to disagree: active learning with noisy labels without crowdsourcing. *International Journal of Machine Learning and Cybernetics* 9, 8 (2018), 1307–1319.

[3] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2017. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085* (2017).

[4] Lei Chen, Zhengdao Chen, and Joan Bruna. 2021. On Graph Neural Networks versus Graph-Augmented MLPs. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.*

[5] Kien Do, Truyen Tran, and Svetha Venkatesh. 2019. Graph Transformation Policy Network for Chemical Reaction Prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 750–760.

[6] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. 2021. On the Equivalence of Decoupled Graph Convolution Network and Label Propagation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021.* 3651–3662.

[7] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen, and Dacheng Tao. 2015. Exploring representativeness and informativeness for active learning. *IEEE transactions on cybernetics* 47, 1 (2015), 14–26.

[8] Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. 2018. Active Discriminative Network Representation Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.* 2142–2148.

[9] R. A. Gilyazev and D. Yu. Turdakov. 2018. Active Learning and Crowdsourcing: A Survey of Optimization Methods for Data Labeling. *Program. Comput. Softw.* 44, 6 (2018), 476–491.

[10] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA.* 1024–1034.

[11] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406* (2020).

[12] Shengding Hu, Zheng Xiong, Meng Qu, Xingdi Yuan, Marc-Alexandre Côté, Zhiyuan Liu, and Jian Tang. 2020. Graph Policy Network for Transferable Active Learning on Graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

[13] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

[14] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. 2010. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation.* 64–67.

[15] Heinrich Jiang and Maya R. Gupta. 2021. Bootstrapping for Batch Active Sampling. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021.* 3086–3096.

[16] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and Robust Influence Maximization in Social Networks. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012.* 918–923.

[17] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* 137–146.

[18] Jinha Kim, Seung-Keol Kim, and Hwanjo Yu. 2013. Scalable and parallelizable processing of influence maximization for large-scale social networks?. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013.* 266–277.

[19] Seung-Keol Kim, Dongeun Kim, Jinoh Oh, Jeong-Hyon Hwang, Wook-Shin Han, Wei Chen, and Hwanjo Yu. 2017. Scalable and parallelizable influence maximization with Random Walk Ranking and Rank Merge Pruning. *Inf. Sci.* 415 (2017), 171–189.

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

[21] Jia Li, Zhichao Han, Hong Cheng, Jiao Su, Pengyun Wang, Jianfeng Zhang, and Lujia Pan. 2019. Predicting Path Failure In Time-Evolving Graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 1279–1289.

[22] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[23] Prem Melville and Raymond J Mooney. 2004. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning.* 74.

[24] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming* 14, 1 (1978), 265–294.

[25] Sudipta Paul, Shivkumar Chandrasekaran, BS Manjunath, and Amit K Roy-Chowdhury. 2020. Exploiting Context for Robustness to Label Noise in Active Learning. *arXiv preprint arXiv:2010.09066* (2020).

[26] Chris Van Pelt and Alex Sorokin. 2012. Designing a scalable crowdsourcing platform. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012.* 765–766.

[27] Burr Settles. 2011. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010.* JMLR Workshop and Conference Proceedings, 1–18.

[28] Victor S. Sheng and Jing Zhang. 2019. Machine Learning with Crowdsourcing: A Brief Summary of the Past Research and Future Directions. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019.* AAAI Press, 9837–9843.

[29] Min Tang, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.* 120–127.

[30] Hao Wang, Tong Xu, Qi Liu, Defu Lian, Enhong Chen, Dongfang Du, Han Wu, and Wen Su. 2019. MCNE: An End-to-End Framework for Learning Multiple Conditional Network Representations of Social Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.* 1064–1072.

[31] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA.* 6861–6871.

[32] Jian Wu, Victor S. Sheng, Jing Zhang, Hua Li, Tetiana Dadakova, Christine Leon Swisher, Zhiming Cui, and Pengpeng Zhao. 2020. Multi-Label Active Learning Algorithms for Image Classification: Overview and Future Promise. *ACM Comput. Surv.* 53, 2 (2020), 28:1–28:35.

[33] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *ICML.* 5449–5458.

[34] Jie Yang, Thomas Drake, Andreas C. Damianou, and Yoelle Maarek. 2018. Leveraging Crowdsourcing Data for Deep Active Learning An Application: Learning Intents in Alexa. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 23–32.

[35] Yu Yang, Enhong Chen, Qi Liu, Biao Xiang, Tong Xu, and Shafqat Ali Shad. 2012. On Approximation of Real-World Influence Spread. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012,*

Bristol, UK, September 24-28, 2012. Proceedings, Part II. 548–564.

[36] Chicheng Zhang and Kamalika Chaudhuri. 2015. Active Learning from Weak and Strong Labelers. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 703–711.

[37] Wentao Zhang, Yu Shen, Yang Li, Lei Chen, Zhi Yang, and Bin Cui. 2021. ALG: Fast and Accurate Active Learning Framework for Graph Convolutional Networks. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*. 2366–2374.

[38] Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. 2021. RIM: Reliable Influence-based Active Learning on Graphs. *Advances in Neural Information Processing Systems* 34 (2021).

[39] Wentao Zhang, Zhi Yang, Yexin Wang, Yu Shen, Yang Li, Liang Wang, and Bin Cui. 2021. Grain: Improving Data Efficiency of Graph Neural Networks via Diversified Influence Maximization. *Proc. VLDB Endow.* 14, 11 (2021), 2473–2482.

[40] Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics*. 1137–1144.

A PROOF OF RELIABILITY CALCULATION

A.1 Problem Description and Assumptions

In the following, the reliability of a label means the probability that the label is correct.

Formally, for every sample we choose to be labeled by the oracle, we will eventually obtain the following related labels:

- One label from the model, corresponding to that sample.
- One or two labels from the oracle(s), the first corresponding to that sample, and the second corresponding to a similar sample, if existing.

The known conditions are:

- For a given sample, the accuracy of the model is denoted by β, which is equal to the accuracy on the validation set times the confidence of the model. Therefore, β changes w.r.t. the sample and the time.
- There are k classes in total.
- For every sample we choose, the accuracy of the oracle who labels this sample is denoted by α_1. If a similar sample is also chosen, the accuracy of the oracle who labels this similar sample is denoted by α_2. For simplicity, we just refer to these two oracles as the first and the second oracle, and the corresponding accuracy is α_1, α_2 respectively.

We have made the following assumptions.

- For each sample and an oracle with accuracy α, the probability that the oracle gives a correct label only depends on α.
- Labels given by different oracles are **independent**, even if different oracles may have different accuracy, i.e., their proficiency may differ.
- Labels given by the oracle and the model are **independent**.
- When we say two samples are similar, we expect that they share the same label.
- When a sample is mislabeled by either the oracle or the model, the wrong label will be chosen **uniformly at random** from the remaining $k-1$ classes.

Goal: Given α, β, k, and labels, calculate the reliability of our final decision.

A.2 Framework

We propose a two-stage solution, and these stages are decoupled:

- Stage 1 (Theorem 4.1): If two labels given by oracles with the reliability α_1 and α_2, calculate the reliability of the latter and regard it as the combined reliability of the oracles α'. If there is just one, just set the combined reliability as α_1.
- Stage 2 (Theorem 4.2): Now we have exactly one label from the oracles with the reliability α' and one from the model with the reliability β. Calculate the final reliability of the label we pick, if applicable (in some cases, we just discard all labels).

A.3 Details of Stage 1 (Theorem 4.1)

Just consider the case where two labels are given by oracles. According to whether these labels are identical, we have two cases.

A.3.1 Case 1: Two oracles give the same label.

Theorem 1. *If two oracles give the same label, the reliability of these two labels are independent and the reliability of them is α_1 and α_2 respectively, then the reliability of the latter (i.e., the combined reliability) is*

$$\frac{\alpha_1\alpha_2}{\alpha_1\alpha_2 + \frac{(1-\alpha_1)(1-\alpha_2)}{k-1}} \tag{16}$$

In this case, these two labels can be both correct or incorrect. The probability that both are correct or incorrect is$P(both\ correct) = \alpha_1\alpha_2$, $P(both\ incorrect) = (k-1) * \frac{1-\alpha_1}{k-1} * \frac{1-\alpha_2}{k-1} = \frac{(1-\alpha_1)(1-\alpha_2)}{k-1}$.

Thus, the reliability of the latter label is

$$\begin{aligned} &P(both\ correct \mid oracle\ gives\ same\ label) \\ &= \frac{P(both\ correct)}{P(both\ correct) + P(both\ incorrect)} = \frac{\alpha_1\alpha_2}{\alpha_1\alpha_2 + \frac{(1-\alpha_1)(1-\alpha_2)}{k-1}} \end{aligned} \tag{17}$$

A.3.2 Case 2: Two oracles give different labels.

Theorem 2. *If two oracle give two different labels, the reliability of these two labels are independent and the reliability of them is α_1 and α_2 respectively, then the reliability of the latter (i.e., the combined reliability) is*

$$\frac{(1-\alpha_1)\alpha_2}{1-\alpha_1\alpha_2}. \tag{18}$$

In this case, the oracle can only be correct at most once. The probability that the oracle is correct at the second time (thus incorrect at the first time) is given by

$$P(first\ incorrect,\ second\ correct) = (1-\alpha_1)\alpha_2. \tag{19}$$

The correctness of this formula is, again, due to the independence of these two trials. The probability that two oracle give two different labels is given by

$$P(oracle\ gives\ diff\ labels) = 1 - P(both\ correct) = 1 - \alpha_1\alpha_2, \tag{20}$$

where $P(both\ correct)$ refers to the same thing as that in Case 1 – the oracle is correct twice. Therefore, the reliability of the latter label can be calculated as follows

$$\begin{aligned} &P(first\ incorrect,\ second\ correct | oracle\ gives\ diff\ labels) \\ &= \frac{(1-\alpha_1)\alpha_2}{1-\alpha_1\alpha_2} \end{aligned} \tag{21}$$

A.4 Details of Stage 2 (Theorem 4.2)

Now we have the combined reliability of the oracle(s) α', and the reliability of the model β.This stage is indeed very similar to the previous one.

As a special note, in the following, when we say "the oracle and the model agree with each other", we mean the label we pick from the one or two labels given by the oracle is the same as the one given by the model, and vice versa.

Once again, divide the problems into two cases according to whether they agree with each other.

A.4.1 *The oracle and the model agree with each other.*

Theorem 3. *If the oracle and the model agree with each other, the reliability of these two labels are independent and are α' and β respectively, then the combined reliability is $\alpha' * \beta = \alpha'\beta$.*

In this case, they can only be both correct or both incorrect. The probability that they are both correct or incorrect is

$$P(oracle\ model\ correct) = \alpha' * \beta = \alpha'\beta \tag{22}$$

,

$$\begin{aligned} P(oracle\ model\ incorrect) &= (k-1) * \frac{1-\alpha'}{k-1} * \frac{1-\beta}{k-1} \\ &= \frac{(1-\alpha')(1-\beta)}{k-1} \end{aligned} \tag{23}$$

.

Thus, the reliability of the combined reliability is

$$\begin{aligned} &P(oracle\ model\ correct \mid oracle\ model\ agree) \\ &= \frac{P(oracle\ model\ correct)}{P(oracle\ model\ correct) + P(oracle\ model\ incorrect)} = \frac{\alpha'\beta}{\alpha'\beta + \frac{(1-\alpha')(1-\beta)}{k-1}} \end{aligned} \tag{24}$$

.

A.4.2 *The oracle and the model disagree with each other.*

Theorem 4. *If the oracle and the model disagree with each other, the reliability of these two labels are independent and are α' and β respectively, then the reliability of the oracle is $\frac{\alpha'(1-\beta)}{1-\alpha'\beta}$*

For simplicity, let the label we pick from the oracle represent the opinion of the oracle.

In this case, the oracle and the model can be either correct or both incorrect. The probability that the oracle or the model is correct is given by

$$P(oracle\ correct,\ model\ incorrect) = \alpha'(1-\beta) \tag{25}$$

$$P(oracle\ incorrect,\ model\ correct) = (1-\alpha')\beta \tag{26}$$

The probability that the oracle and the model disagree with each other is given by

$$\begin{aligned} P(oracle\ model\ disagree) &= 1 - P(oracle\ model\ correct) \\ &= 1 - \alpha'\beta \end{aligned} \tag{27}$$

.

Therefore, the reliability of the oracle in this case is:

$$\begin{aligned} &P(oracle\ correct,\ model\ incorrect \mid oracle\ model\ disagree) \\ &= \frac{P(oracle\ correct,\ model\ incorrect)}{P(oracle\ model\ disagree)} = \frac{\alpha'(1-\beta)}{1-\alpha'\beta} \end{aligned} \tag{28}$$

.

A.5 Proof of Theorem 5.1

Theorem 5. *The greedily selected batch node set is within a factor of $(1-\frac{1}{e})$ of the optimal set for the objective of effective influence maximization (EIM).*

Consider a batch selection setting with $\mathcal{B}/b$ rounds where b nodes are selected in each iteration (see Alg.1) and $\mathcal{B}$ denotes the labeling budget. Formally, EIM objective is equal to:

$$\max_{\mathcal{V}_b} F(\mathcal{V}_b) = |\sigma(\mathcal{V}_l \cup \mathcal{V}_b)|, \mathbf{s.t.} \mathcal{V}_b \subseteq \mathcal{V} \setminus \mathcal{V}_l, |\mathcal{V}_b| = b \tag{29}$$

Table 4: Overview of the Four Datasets

Dataset	#Nodes	#Features	#Edges	#Classes	#Train/Val/Test	Task type	Description
Cora	2,708	1,433	5,429	7	1,208/500/1,000	Transductive	citation network
Citeseer	3,327	3,703	4,732	6	1,827/500/1,000	Transductive	citation network
Pubmed	19,717	500	44,338	3	18,217/500/1,000	Transductive	citation network
ogbn-arxiv	169,343	128	1,166,243	40	90,941/29,799/48,603	Transductive	citation network
Reddit	232,965	602	11,606,919	41	155,310/23,297/54,358	Inductive	social network

where $\sigma\ \sigma(\mathcal{V}_l) = \bigcup_{v\in\mathcal{V}, Q(v,\mathcal{V}_l,k)>\theta}\{v\}$., Q is defined in Definition 4.4, θ is the hyper-parameter defined in eq (12), $\mathcal{V}_l$ is the set of nodes selected in all previous rounds.

For every $A \subseteq B \subseteq S$ and $s \in S \setminus B$, let $Q_A(v) = \max_{v_i\in\mathcal{V}_l\cup A} Q(v, v_i, k)$ and $Q_B(v) = \max_{v_j\in\mathcal{V}_l\cup B} Q(v, v_j, k)$. Since $(\mathcal{V}_l \cup A) \subseteq (\mathcal{V}_l \cup B)$, for any $v \in \mathcal{V}$, $Q_A(v) \leq Q_B(v)$. Thus,

$$\begin{aligned} F(A \cup \{s\}) - F(A) &= |Q(v,s,k) > \theta \geq Q_A(v)| \\ &\geq |Q(v,s,k) > \theta \geq Q_B(v)| = F(B \cup \{s\}) - F(B) \end{aligned} \tag{30}$$

.

B DATASET DESCRIPTION

Cora, **Citeseer**, and **Pubmed**[1] are popular citation network datasets, and we follow the public training/validation/test split in GCN [20]. In these networks, nodes are papers from different topics, and the edges are citations among the papers, node attributes are binary word vectors, and class labels indicate the topic of a certain paper. **Reddit** is a social network dataset derived from the community structure of numerous Reddit posts. The training/validation/test split in our experiment is the same as that in GraphSAGE [10]. The public version provided by GraphSAINT[2] is used in our paper. **ogbn-arxiv** is a directed graph, representing the citation network among all Computer Science (CS) arXiv papers indexed by MAG. The training/validation/test split in our experiment is the same as the public version provided by OGB[3] is used in our paper.

For more specifications about the five datasets, see Table 4.

C IMPLEMENTATION DETAILS

$\theta = 0.05$ for Cora and Citesser, 0.005 otherwise. $\gamma = 0.5$ for Cora, PubMed and Reddit, 0.4 otherwise. For GPA, adopt the pre-trained model released by its authors: Cora: pre-trained on PubMed, Citeseer; PubMed: the model pre-trained on Cora, Citeseer; Citeseer and Reddit: pre-trained on Cora, PubMed. Other hyper-parameters are kept the same. For AGE [3] and ANRMAB [8], GCN is trained for 200 epochs in each node selection iteration. AGE is implemented by its open-source version and ANRMAB in accordance with the paper. For ALG [37] and GRAIN [39], follow the code of the paper. k (e.g., 7 in Cora) labels are given by oracles in each iteration.

Experiments are conducted on Ubuntu 16.04 with Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz, 4 NVIDIA GeForce GTX 1080 Ti GPUs and 256 GB DRAM, Python 3.6, Pytorch 1.7.1, CUDA 10.1.

Implementations are provided in the github repository (https://github.com/zwt233/NC-ALG).

[1] https://github.com/tkipf/gcn/tree/master/gcn/data
[2] https://github.com/GraphSAINT/GraphSAINT
[3] https://ogb.stanford.edu/docs/nodeprop/#ogbn-arxiv