Know Your Neighbors: Subgraph Importance Sampling for Heterophilic Graph Active Learning

Wenjie Yang[1], Shengzhong Zhang[2*], Chen Ye[1], Jiaxing Guo[1], Tongshan Xu[1], Zengfeng Huang[1, 3*]

[1]Fudan University, China
[2]Nanjing University of Aeronautics and Astronautics, China
[3]Shanghai Innovation Institution, China
szzhang@nuaa.edu.cn, huangzf@fudan.edu.cn

Abstract

Graph neural networks (GNNs) have demonstrated strong performance in various graph mining tasks but rely heavily on extensively labeled nodes. To improve training efficiency, graph active learning (GAL) has emerged as a solution for selecting the most informative nodes for labeling. However, existing GAL methods are primarily designed for homophilic graphs, where nodes with the same labels are more likely to be connected. In this work, we systematically study active learning on heterophilic graphs, a setting that has received limited attention. Surprisingly, we observe that *existing GAL methods fail to consistently outperform random sampling on heterophilic graphs*. Through an in-depth investigation, we reveal that these methods implicitly assume homophily even on heterophilic graphs, leading to suboptimal performance. To address this issue, we introduce the principle of *"Know Your Neighbors"* and propose an active learning algorithm KyN specifically for heterophilic graphs. The core idea of KyN is to provide GNNs with accurate estimations of homophily distribution by labeling nodes together with their neighbors. We implement KyN based on subgraph sampling with probabilities proportional to ℓ_1 Lewis weights, which is supported by solid theoretical guarantees. Extensive experiments on diverse real-world datasets, including a large heterophilic graph with over 2 million nodes, demonstrate the effectiveness and scalability of KyN.

Introduction

Graphs are ubiquitous in real-world applications, spanning diverse domains such as recommendation systems (Ma et al. 2024; Ni et al. 2024), misconduct detection (Tao et al. 2024; Wu and Hooi 2023), and AI for science (Gasteiger, Becker, and Günnemann 2021; Lam et al. 2023). Recently, graph neural networks (Kipf and Welling 2017; Wu et al. 2019a; Velickovic et al. 2018; Chen et al. 2020) have become the de facto standard for many graph learning tasks. Like other deep learning methods, the success of GNNs heavily relies on the availability of high-quality training labels. However, data labeling for the node samples is costly, as it typically requires significant human effort. To address this challenge, graph active learning has emerged as an effective approach for improving labeling efficiency (Song, Zhang, and King 2023; Zhang et al. 2022). GAL methods aim to maximize model performance by identifying the most informative nodes for annotation within a given labeling budget. Despite their success, we are surprised to find that existing GAL methods have been predominantly evaluated on homophilic graphs, where nodes with the same labels are more likely to be connected. With the growing interest in heterophilic graph learning, a critical question arises:

Are current graph active learning methods effective on heterophilic graphs?

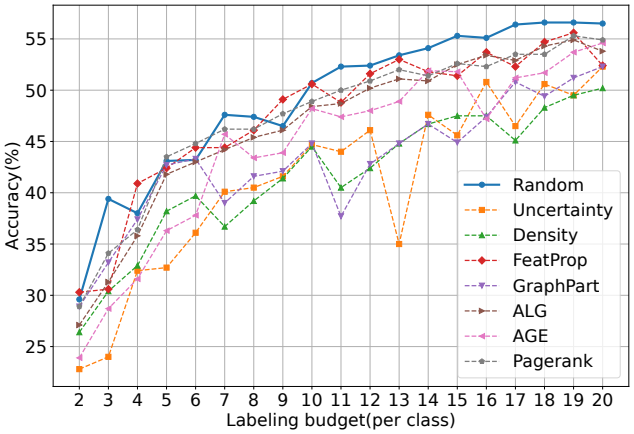


Figure 1: **Current GAL methods struggle to outperform naive random sampling on heterophilic graphs**. Performance of various GAL methods on the heterophilic graph dataset Roman-Empire. The labeling budget ranges from $2C$ to $20C$, where C denotes the number of classes in the dataset.

Unfortunately, the answer is no. We evaluated the performance of representative GAL methods on a heterophilic graph dataset, Roman-Empire (Platonov et al. 2023b). The results, shown in Figure 1, uncover a notable limitation: *none of the existing GAL methods consistently outperform uniform random sampling.* Since GAL methods are typically not faster or simpler than random sampling, they must at least deliver consistently better performance to justify their use. However, on this heterophilic graph, random sampling

*Corresponding authors.

is often the strongest approach. This outcome is particularly unexpected given the strong performance of these GAL methods on homophilic graphs.

The discovery of the limitations of existing GAL methods on heterophilic graphs is not only novel but also significant. Recent surveys (Luan et al. 2024) indicate that heterophilic graphs are prevalent across many real-world applications, such as fraud and anomaly detection (Gao et al. 2023) and graph clustering (Pan and Kang 2023). Applying off-the-shelf GAL methods to these graphs without recognizing their limitations can result in substantial waste of time and resources. Furthermore, it is already well-documented that GNNs tend to perform suboptimally on heterophilic graphs, often being outperformed by graph-agnostic models (Loveland et al. 2023). Introducing GAL methods that fail to surpass random sampling would only exacerbate this problem.

In this paper, we aim to develop an active learning algorithm tailored for heterophilic graphs, addressing the gap left by prior work. Our method is motivated by key limitations observed in existing GAL methods. Specifically, we identify two main issues: (1) existing GALs fail to equip GNNs with adequate information to discern whether a graph is homophilic or heterophilic, because they tend to label isolated nodes that do not reflect the real homophily. And (2) the fusion of ego-embeddings and neighbor-embeddings on heterophilic graphs makes nodes less distinguishable. To overcome these challenges, we propose the principle of "**K**now **Y**our **N**eighbors" and introduce our model, KyN. By selecting nodes along with their neighbors, KyN yields accurate estimations of local homophily distribution. Our model is based on a novel ℓ_1 Lewis weights subgraph sampling method and offers a solid theoretical guarantee. Importantly, our analysis does not assume homophily in the underlying graph structure, and thus remains valid for both homophilic and heterophilic graphs. We evaluate KyN against various baselines on real-world datasets.

Our contributions are summarized as follows:

- We uncover the unexpected failure of active learning methods on heterophilic graphs. While they demonstrate strong performance on homophilic graphs, they cannot consistently outperform naive random sampling on heterophilic graphs. To the best of our knowledge, this is the first paper to reveal this phenomenon.
- We propose a novel method called KyN for active learning on heterophilic graphs. Our method is well-motivated by the identified issues of previous GALs. KyN selects training nodes along with their neighbors to accurately estimate the local homophily distribution, thereby reflecting the true homophilic or heterophilic nature of graphs.
- We conduct comprehensive experiments that demonstrate the superior performance of KyN on the heterophilic GAL task. Evaluations across 9 datasets, including a large-scale heterophilic graph with over 2 million nodes, highlight the potential of our framework. Notably, *KyN is the only GAL method that consistently outperforms random sampling on both homophilic and heterophilic graphs.*

Related Work

Active learning is a classic research direction that aims to mitigate annotation expenses (Ren et al. 2021; Matsushita, Matsushita, and Hasebe 2018). It is studied in many fields and under different settings, including computer vision (Bengar et al. 2021; Kim et al. 2021), nature language processing (Zhang, Strubell, and Hovy 2022; Margatina et al. 2023) and general deep learning (Huang et al. 2024b; Yan and Huang 2018; Tang and Huang 2022). In the graph realm, AGE (Cai, Zheng, and Chang 2017) is one of the earliest works that measure the informativeness of nodes by combining centrality, density, and uncertainty. ANRMAB (Gao et al. 2018) improves AGE by learning weights using reinforcement learning. ALG (Zhang et al. 2021a) considers both the importance and correlation via the effective reception field maximization. FeatProp (Wu et al. 2019b) first propagates features and then employs a clustering algorithm on the propagated node features. GraphPart (Ma et al. 2023) further enhances FeatProp by applying it to each graph partition. DOCTOR (Song, Zhang, and King 2023) is a GAL method based on the expected model change maximization. GreedyET (Huang et al. 2024a) treat GAL as the aggregation involvement maximization. Some other papers focus on different settings that fit certain applications, e.g., noise/soft label (Zhang et al. 2022, 2021b, 2024), fairness (Han et al. 2024) and transfer learning (Hu et al. 2020).

Graph neural networks under heterophily is an emerging topic in the graph realm. In heterophilic graphs, the nodes with the same labels are not more, sometimes even less, likely to be connected. The fusion phase of ordinary GNNs in these diverse neighborhoods makes nodes indistinguishable, leading to unsatisfactory performance. Various model architectures are proposed to address this challenge. H_2GCN (Zhu et al. 2020) is an early work on heterophily identifying designs crucial to the heterophily setting. CPGNN (Zhu et al. 2021) models different levels of homophily using a learnable class compatibility matrix in the aggregation step. GPR-GNN (Chien et al. 2021) is the generalized PageRank-inspired architecture designed to adapt to different label patterns. FAGCN (Bo et al. 2021) adaptively integrates different signals in the process of message passing with a self-gating mechanism. GloGNN (Li et al. 2022) generates node embedding by aggregating information from global nodes in the graph. GGCN (Yan et al. 2022) learns degree corrections and signed messages based on a unified theoretical perspective for heterophily and oversmoothing.

Coreset is a research field that is very close to active learning. The main difference between the two problems is that we have access to labels before training set selection, but many coreset methods do not use labels so that they can be used for active learning. There are sampling works that focus on ℓ_2-regression (Drineas, Mahoney, and Muthukrishnan 2006; Li, Miller, and Peng 2013; Cohen et al. 2015) and ℓ_1-regression (Clarkson 2005; Sohler and Woodruff 2011; Clarkson et al. 2016). Recent works show that coresets with relative error can be constructed on bounded complexity data for the logistic loss and hinge loss (Munteanu et al.

2019; Mai, Musco, and Rao 2021). Sampling-based coreset methods are also used for fields of active learning, e.g., multiple deep models active learning (Huang et al. 2024b). To the best of our knowledge, this paper is the first to explore Lewis weight sampling for graph active learning.

Preliminaries

Notations. Let $G = (V, E, \boldsymbol{X}, \boldsymbol{Y})$ be a simple graph with node set V and edge set E. $\boldsymbol{X} \in \mathbb{R}^{|V| \times f}$ is the node feature matrix, where f is the number of dimensions of each feature. $\boldsymbol{Y} \in \mathbb{R}^{|V| \times C}$ is the one-hot label matrix with C classes. We use $\mathbf{x}_i$ to represent the feature vector of the i-th node and y_i as its label. We can also use the adjacency matrix $\boldsymbol{A} \in \{0,1\}^{|V| \times |V|}$, where the (i,j)-th entry is 1 if and only if the i-th node and the j-th node are connected. A k-hop neighborhood of node $i \in V$, $N_k(i)$ denotes the subgraph induced by the nodes that are reachable within k-steps of i.

Homophily of graphs. Homophily is a graph property describing the tendency of edges to connect similar nodes (Platonov et al. 2023a). Throughout our paper, a graph is *homophilic* if the nodes with the same labels are more likely to be connected. And a graph is *heterophilic* if the nodes with the same labels are less likely to be connected. Many statistics can measure the degree of homophily of a graph. We will mainly use the following two definitions of homophily/heterophily from previous works (Loveland et al. 2023).

Definition 1 (Global Homophily)**.** *The global homophily of a graph is defined as:*

$$h = \frac{|\{(u,v) : (u,v) \in E \wedge y_u = y_v\}|}{|E|}, \tag{1}$$

where y_u is the label of node u.

Definition 2 (Local Homophily)**.** *The local homophily of a node t is defined as:*

$$h_t = \frac{|\{(u,t) : u \in N_1(t) \wedge y_u = y_t\}|}{|N_1(t)|}. \tag{2}$$

Intuitively, global homophily describes the overall property of a graph, while local homophily focuses on the specific neighborhood of each node. Previous works (Mao et al. 2023; Loveland et al. 2023) show that crucial properties (e.g., the prediction accuracy of GNNs) vary across local homophily levels, highlighting the importance of zooming in and analyzing the diversity of node neighborhoods.

Graph active learning. Active learning algorithms aim to select a training set that maximizes the performance of the models trained on it. Specifically, let $\mathcal{A}_{\text{GAL}}$ be a certain GAL algorithm that takes a graph G and a labeling budget B as inputs, the GAL-selected training set is $V_{\text{train}} = \mathcal{A}_{\text{GAL}}(G, B)$ with $|V_{\text{train}}| = B$. We acquire the labels of V_{train} from an oracle, then train a GNN with them. The performance of the trained GNN can be used to measure the quality of V_{train}, which in turn reflects the effectiveness of $\mathcal{A}_{\text{GAL}}$.

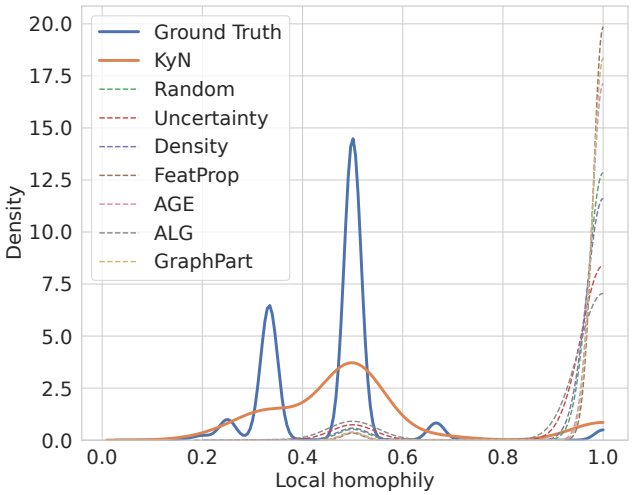


Figure 2: **Current GAL-selected training sets imply homophily even on a heterophilic graph**. The local node homophily distribution plot of different GAL-selected training sets and that of the ground truth on the Roman-empire dataset. For a clear comparison, we also include our algorithm KyN. It is clear that KyN is the most similar to the ground truth distribution, and the only one reflecting heterophilic nature.

Methodology

Motivation: Do current GAL-selected training sets reveal graph heterophily?

Before designing a GAL algorithm specifically for heterophilic graphs, we first investigate why existing GAL methods tend to underperform in these settings. Fundamentally, a GNN cannot inherently distinguish between homophilic and heterophilic graphs without explicit information from the training set. Therefore, it is critical to assess whether GAL-selected training sets, denoted as V_{train}, preserve the homophily-related signals accurately.

To this end, we plot the distribution of local homophily for GAL-selected training sets on the heterophilic graph Roman-Empire (Platonov et al. 2023b), as shown in Figure 2. The ground truth local homophily distribution (solid blue line) is right-skewed, reflecting the dataset's heterophilic nature. However, we observe that previous GAL-selected training sets (dotted lines) exhibit homophilic properties on this heterophilic dataset, with local homophily distributions being left-skewed.

This misalignment explains why GNNs trained on these labeled sets fail to achieve optimal performance: the models are provided with misleading or even contradictory homophily-related information. We formally demonstrate that this misalignment directly impacts accuracy. Specifically, let $\mathcal{P}_{h_t}(G)$ denote the ground truth local homophily distribution and $\mathcal{P}_{\hat{h}_t}(G_{\text{train}})$ represent the local homophily distribution estimated from the training set. For a statistical distance $\mathcal{D}$, a GAL-selected training set that retains accurate homophily-related information should have small

$\mathcal{D}(\mathcal{P}_{h_t}(G), \mathcal{P}_{\hat{h}_t}(G_{\text{train}}))$, as this indicates the training set reflects the true distribution of local homophily. We then formalize the relationship between homophily estimation and accuracy as follows:

Proposition 1 (Inaccurate homophily estimation harms accuracy). *For predictions $\hat{\mathbf{y}} = \{\hat{y}_1, \cdots, \hat{y}_n\}$, let $\text{Acc} = \sum_{i=1}^{n} \mathbb{1}(y_i = \hat{y}_i)/n$, where $\mathbb{1}(\cdot)$ is the indicator function, let the accuracy of the ego-graph of node i be $\text{Acc}_i = \sum_{j \in N(i)} \mathbb{1}(y_j = \hat{y}_j)/|N(i)|$, and measure the correctness of local homophily with $\mathcal{D}(\boldsymbol{h}, \hat{\boldsymbol{h}}) = \frac{1}{n}\sum_{i=1}^{n} |h_i - \hat{h}_i|$, where $\hat{h}_i = \frac{1}{|N(i)|}\sum_{j \in N(i)} \mathbb{1}(\hat{y}_i = \hat{y}_j)$ is the estimated local homophily and $\boldsymbol{h}$ is the vector of homophily. We have that a correct label homophily (i.e., small $\mathcal{D}(\boldsymbol{h}, \hat{\boldsymbol{h}})$) is necessary for high accuracy. Formally,*

$$\mathcal{D}(\boldsymbol{h}, \hat{\boldsymbol{h}}) \leq \frac{1}{n}\sum_{i=1}^{n}(1 - \text{Acc}_i) + (1 - \text{Acc}). \tag{3}$$

Proposition 1 reveals that inaccurate local homophily estimation (i.e., a large LHS in Eq. (3)) inherently limits model accuracy. Thus, it is crucial to select a training set that faithfully reflects the degree of homophily. But why do prior GAL methods select homophilic training sets for heterophilic graphs? We argue that these methods often query nodes without their neighbors, resulting in many isolated nodes in the induced subgraph. When fed to GNNs, these isolated nodes are viewed as strongly homophilic nodes (i.e., $h_t = 1$) since they are the only labeled nodes in their own neighborhood, resulting in an inaccurate estimation of homophily distribution. Therefore, the solution is straightforward: For heterophilic graphs, we should label the neighbors of selected nodes as well, embodying the principle of "know your neighbors." This approach ensures a more accurate estimation of local homophily, as demonstrated below.

Proposition 2 ("Know your neighbors" improves homophily estimation). *For any labeled node i in a graph G, the more its neighbors are known, the more accurate the estimation of local homophily will be. Formally, suppose we query n_i node, then $\forall \epsilon \in (0, h_i)$,*

$$\mathcal{P}(|\hat{h}_i - h_i| \geq \epsilon) \leq 2\exp(-2\epsilon^2 n_i), \tag{4}$$

where $\hat{h}_i$ is the estimated local homophily of node i and h_i is the ground truth.

Proposition 2 demonstrates that adhering to the principle of KyN yields more accurate local homophily estimates, which is essential for achieving high performance as discussed earlier.

Implementation: Subgraph importance sampling with ℓ_1 Lewis weights

After motivating the principle of "know your neighbors", we now introduce how to implement this through a subgraph importance sampling framework. The detailed pseudocode is deferred to the appendix. Specifically, there are two ways to "know your neighbors":

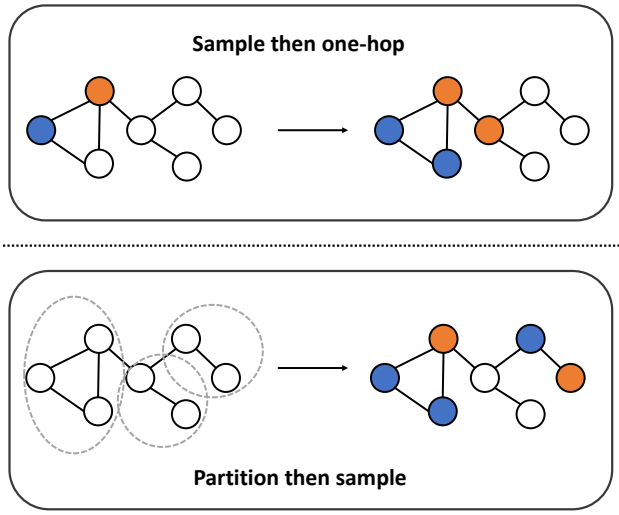


Figure 3: A toy example to illustrate two approaches to "know your neighbors". The colored nodes are labeled and will be used for training GNNs. We set $k = 1$ in the "sample then select k-hop" scheme.

- Sample then select k-hop: This approach first samples nodes with some GAL methods, then selects the k-hop neighbors of each node.
- Partition then sample: This approach first partitions the graph into disjoint subgraphs, and selects subgraphs with some GAL methods.

Figure 3 shows these two methods on a toy example. Due to neighbor explosion, the "sample then select k-hop" scheme tends to select a huge connected component, while "partition then sample" usually produces reasonably diverse subgraphs. Consider two nodes u and v are selected in the first stage of "sample then select one-hop", where v is in the k-hop neighborhood of u, i.e., $v \in N_k(u)$. In the second stage, the union neighborhood of these two nodes will be gigantic, draining the labeling budget. Therefore, we design our algorithm within the "partition then sample" scheme. We introduce the details of each stage separately.

Partitioning. We adopt the classic graph clustering algorithms METIS (Karypis and Kumar 1998) to partition the graphs. For the graph G, we partition its nodes into c groups: $V = \{V_1, V_2, \cdots V_c\}$, where V_i is the i-th part. We then have c subgraphs as

$$G_i = (V_i, E_i), \forall i \in [c], \tag{5}$$

where $E_i = \{(u, v) : (u, v) \in E \wedge u \in V_i \wedge v \in V_i\}$. Before moving on to the sampling phase, we need to generate a representation $\mathbf{R}_{G_i}$ for each subgraph G_i. It is possible to use naive readouts, e.g., take the average of node features:

$$\mathbf{R}_{G_i} = \frac{1}{|V_i|}\sum_{u \in V_i} \mathbf{x}_u. \tag{6}$$

However, since we are dealing with heterophilous graphs, Eq. (6) will lead to an inter-class fusion that makes sub-

graphs indistinguishable. Therefore, we propose a more sophisticated way to produce the representations. We first find a central node for each subgraph. The graph center is defined as follows:

Definition 3 (Jordan center (Wasserman and Faust 1994))**.** *The center of a graph is the set of all vertices of minimum eccentricity, i.e.,*

$$\arg\min_{u} \max_{v} d(u, v), \tag{7}$$

where $d(\cdot, \cdot)$ is the geodesic distance.

By finding a central node n_c, we are able to view the subgraph G_i as an ego-graph centered at n_c. We can then separate the ego-embedding and neighbor-embeddings to yield a reasonable subgraph representation $\mathbf{R}_{G_i}$. This separation is known to be effective on heterophilic graphs (Zhu et al. 2020). Specifically, we compute the representation as follows:

$$\mathbf{R}_{G_i} = \mathrm{CONCAT}(\mathbf{x}_{n_c}, \frac{1}{|N_1(n_c)| - 1} \sum_{i \in N_1(n_c) \setminus \{n_c\}} \mathbf{x}_i), \tag{8}$$

where $\mathrm{CONCAT}(\cdot, \cdot)$ is the concatenation function. Note that this readout can also serve as a proxy of GraphSAGE (Hamilton, Ying, and Leskovec 2017) without learnable parameters, which is one of the few basic GNN encoders that work with heterophily (Platonov et al. 2023b).

Sampling. The goal of the sampling phase is to approximate the training loss of all subgraphs with only a small fraction of them. We sample these subgraphs with probabilities proportional to their ℓ_1 Lewis weights. The formal definition of ℓ_1 Lewis weights is:

Definition 4 (ℓ_1 Lewis weights (Cohen and Peng 2015))**.** *For any matrix $\boldsymbol{M} \in \mathbb{R}^{n \times f}$ the ℓ_1 Lewis weights are the unique values $\tau_1(\boldsymbol{M}), \cdots, \tau_n(\boldsymbol{M})$ such that,*

$$\tau_i(\boldsymbol{M})^2 = \mathbf{m}_i^T (\boldsymbol{M}^T \boldsymbol{W} \boldsymbol{M})^\dagger \mathbf{m}_i, \tag{9}$$

where $\boldsymbol{W}$ is the diagonal matrix with $1/\tau_1(\boldsymbol{M}), \cdots, 1/\tau_n(\boldsymbol{M})$ as its diagonal, and the dagger symbol represents the pseudoinverse.

The ℓ_1 Lewis weights sampling has solid theoretical guarantees. In practice, we let $\boldsymbol{M} = \boldsymbol{R}$, where $\boldsymbol{R} = (\boldsymbol{R}_{G_1}, \cdots, \boldsymbol{R}_{G_c})^T$ is the subgraph representation matrix. We compute and normalize the ℓ_1 Lewis weights $\tau_1(\boldsymbol{M}), \cdots, \tau_c(\boldsymbol{M})$ and select subgraphs with these probabilities. Once a subgraph is selected, we query all nodes within the subgraph, achieving "know your neighbors". After the number of labeled nodes reaches the budget, we feed the training set to GNNs for parameter optimization.

Theoretical guarantees

The ℓ_1 Lewis weights sampling has solid theoretical guarantees. For linear classification, the ℓ_1 Lewis weights sampling yields a relative error coreset (Mai, Musco, and Rao 2021). A type of binary classification loss called nice hinge function is considered in (Mai, Musco, and Rao 2021):

Definition 5 (Nice Hinge Function (Mai, Musco, and Rao 2021))**.** *A function $f : \mathbb{R} \to \mathbb{R}^+$ is an (L, a_1, a_2)-nice hinge function if for fixed constants L, a_1 and a_2,*

(1) f is L-Lipschitz; (2) $|f(z) - \mathrm{ReLU}(z)| \leq a_1, \forall z$; (3) $f(z) \geq a_2, \forall z \geq 0$.

We extend the theory to our multi-class classification on graphs. We show that ℓ_1 Lewis sampling on $\boldsymbol{R}$ gives a relative error coreset for the cross-entropy (CE) loss. Specifically, minimizing the objective function on this selected coreset (i.e., the training set V_{train} in experiments) will yield a near minimizer over all subgraphs. Considering that the GCN encoder is not suitable for heterophilic graphs, while SAGE-Mean performs stably on heterophilic graphs (Platonov et al. 2023b), we follow previous work and use SAGE-Mean as the encoder. For simplicity, we use a one-layer SAGE-Mean encoder, the results are similar on any multi-layer linear GNNs. We reformulate the CE loss to show it is also a $(1, \ln 2, \ln 2)$-nice hinge function. The detailed proof is deferred to the appendix.

Theorem 1. *For a one-layer GNN encoder, the CE loss is given by $L(\beta) = -\sum_{i=1}^{c} \ln(p(y_i)|\boldsymbol{R}_{G_i}, \beta)$, where β is the learnable parameter. For a set of sampling values p_i with $\sum_{i=1}^{c} p_i = m$ and $p_i \geq \frac{\mathcal{C} \max(\tau_i(\boldsymbol{R}), 1/c) \cdot \mu(\boldsymbol{R})^2}{\epsilon^2}$ for all i, where $\mathcal{C} = a \cdot \max(1, L, a_1, 1/a_2)^{10} \cdot \ln(\frac{\ln(c \max(1, L, a_1, 1/a_2) \cdot \mu(\boldsymbol{R})/\epsilon) m}{\delta})$ and a is a fixed constant, $\mu(\boldsymbol{R}) = \sup_{\beta \neq 0} \frac{||(\boldsymbol{R}\beta)^+||_1}{||(\boldsymbol{R}\beta)^-||_1}$. If the sampling matrix $\boldsymbol{S} \in \mathbb{R}^{m \times c}$ has each row chosen independently as the i^{th} standard basis vector scaled by $1/p_i$ with probability p_i/m, then with probability at least $1 - \delta$, we have the following relative error coreset:*

$$\left| \sum_{i=1}^{m} [\boldsymbol{S} f(z)]_i - L(\beta) \right| \leq \epsilon \cdot L(\beta), \tag{10}$$

where $\boldsymbol{S}$ has $m = \tilde{O}(\frac{f \mu(\boldsymbol{R})^2}{\epsilon^2})$ rows.

Thus, assuming $\beta^* = \arg\min_\beta L(\beta)$, and $\tilde{\beta}$ is the minimizer of the weighted loss $\sum_{i=1}^{m} [\boldsymbol{S} f(Z)]_i$, we have $L(\tilde{\beta}) \leq \frac{1+\epsilon}{1-\epsilon} \cdot L(\beta^*)$. This shows that minimizing the objective function on the sampled subset of size m can produce an approximation close to the minimizer over all subgraphs, achieving the goal of subgraph sampling. On the other hand, selecting all nodes within each subgraph achieves "know your neighbors", revealing the degree of homophily of a graph. To ensure a fair comparison, we sample subgraphs until the number of labeling nodes exceeds the budget, and keep the first B nodes in experiments.

Experiments

Experimental setup

We first compare KyN with other GAL methods on real-world datasets: Roman-empire, Amazon-ratings, Tolokers, and Minesweeper (Platonov et al. 2023b), Wisconsin and Texas (Pei et al. 2020). Additionally, we evaluate the scalability of KyN using the large-scale Snap-Patents graph (Leskovec and Krevl 2014). We set the labeling budget to

Dataset	Roman-empire			Amazon-ratings		
Budget	$5C$	$10C$	$20C$	$5C$	$10C$	$20C$
Random	43.1 ± 2.9	50.7 ± 1.0	56.5 ± 0.8	30.2 ± 2.6	30.7 ± 1.5	31.3 ± 0.6
Uncertainty	32.7 ± 4.4	44.7 ± 3.0	52.3 ± 2.6	30.6 ± 2.9	30.8 ± 2.7	31.4 ± 1.1
Density	38.2 ± 3.3	44.5 ± 2.4	50.2 ± 2.2	30.5 ± 2.1	30.9 ± 2.2	31.1 ± 0.8
AGE	36.3 ± 2.8	48.2 ± 2.4	54.6 ± 1.6	29.3 ± 1.8	30.2 ± 2.5	30.8 ± 1.6
ALG	41.8 ± 2.3	48.4 ± 1.8	53.8 ± 1.5	30.8 ± 1.5	31.0 ± 1.5	31.6 ± 1.0
FeatProp	42.4 ± 1.0	50.6 ± 2.1	52.4 ± 1.7	30.2 ± 1.3	30.3 ± 1.5	30.9 ± 0.6
GraphPart	42.7 ± 1.6	44.8 ± 2.5	52.3 ± 1.9	30.4 ± 2.2	31.0 ± 1.4	32.1 ± 0.7
KyN(Ours)†	**44.8** ± 2.4	**51.4** ± 1.3	**57.5** ± 1.4	**31.2** ± 1.7	**31.3** ± 1.1	**32.3** ± 0.4
Dataset	Tolokers			Wisconsin		
Budget	$5C$	$10C$	$20C$	$5C$	$10C$	$20C$
Random	65.4 ± 3.9	68.8 ± 4.7	69.0 ± 3.2	71.7 ± 4.0	78.6 ± 3.3	86.1 ± 2.3
Uncertainty	68.9 ± 8.6	71.4 ± 8.0	71.7 ± 4.6	71.6 ± 5.9	78.7 ± 3.9	88.1 ± 2.1
Density	62.7 ± 9.2	68.5 ± 6.4	68.6 ± 4.2	68.7 ± 1.3	72.5 ± 1.1	83.7 ± 2.0
AGE	66.6 ± 7.8	69.4 ± 5.6	70.9 ± 4.7	69.2 ± 2.2	78.2 ± 0.7	87.4 ± 3.0
ALG	67.3 ± 6.4	69.6 ± 6.1	70.8 ± 4.3	70.8 ± 3.7	78.5 ± 3.2	86.9 ± 2.6
FeatProp	62.3 ± 7.1	70.6 ± 5.3	66.8 ± 3.9	71.9 ± 2.8	78.8 ± 1.7	87.9 ± 2.5
GraphPart	69.8 ± 6.8	71.2 ± 4.3	71.5 ± 4.1	69.7 ± 3.1	78.9 ± 1.5	87.2 ± 2.9
KyN(Ours)†	**71.0** ± 4.5	**71.8** ± 3.5	**72.9** ± 4.2	**72.5** ± 3.5	**79.1** ± 1.1	**88.5** ± 2.2
Dataset	Minesweeper			Texas		
Budget	$5C$	$10C$	$20C$	$5C$	$10C$	$20C$
Random	72.9 ± 5.2	75.0 ± 3.6	77.1 ± 3.1	73.3 ± 2.9	82.6 ± 3.0	92.8 ± 2.2
Uncertainty	68.7 ± 7.9	75.4 ± 6.2	76.7 ± 4.1	73.4 ± 2.9	84.1 ± 2.5	**94.7** ± 1.4
Density	67.3 ± 9.8	73.0 ± 7.9	75.1 ± 3.2	73.5 ± 2.5	78.6 ± 2.7	91.8 ± 1.6
AGE	71.0 ± 3.8	75.7 ± 3.8	76.4 ± 2.5	74.3 ± 2.3	80.4 ± 2.1	89.3 ± 0.7
ALG	71.6 ± 4.7	75.5 ± 5.4	76.8 ± 2.7	74.6 ± 2.7	83.5 ± 2.6	91.3 ± 1.1
FeatProp	73.1 ± 4.4	75.6 ± 2.9	76.2 ± 2.3	76.2 ± 2.8	82.2 ± 2.4	92.9 ± 1.5
GraphPart	72.8 ± 5.6	75.9 ± 3.1	76.8 ± 2.1	77.1 ± 2.4	83.9 ± 2.2	92.7 ± 1.9
KyN(Ours)†	**73.3** ± 5.3	**76.5** ± 3.6	**77.8** ± 2.8	**77.4** ± 2.9	**84.3** ± 4.1	93.2 ± 1.6

Table 1: The experimental results of KyN and other graph active learning methods. We report the mean classification accuracy and standard deviation trained on the training set selected by each GAL. The best results are **bolded**. The superscript † denotes that GAL consistently performs better than random sampling across all cases in this row.

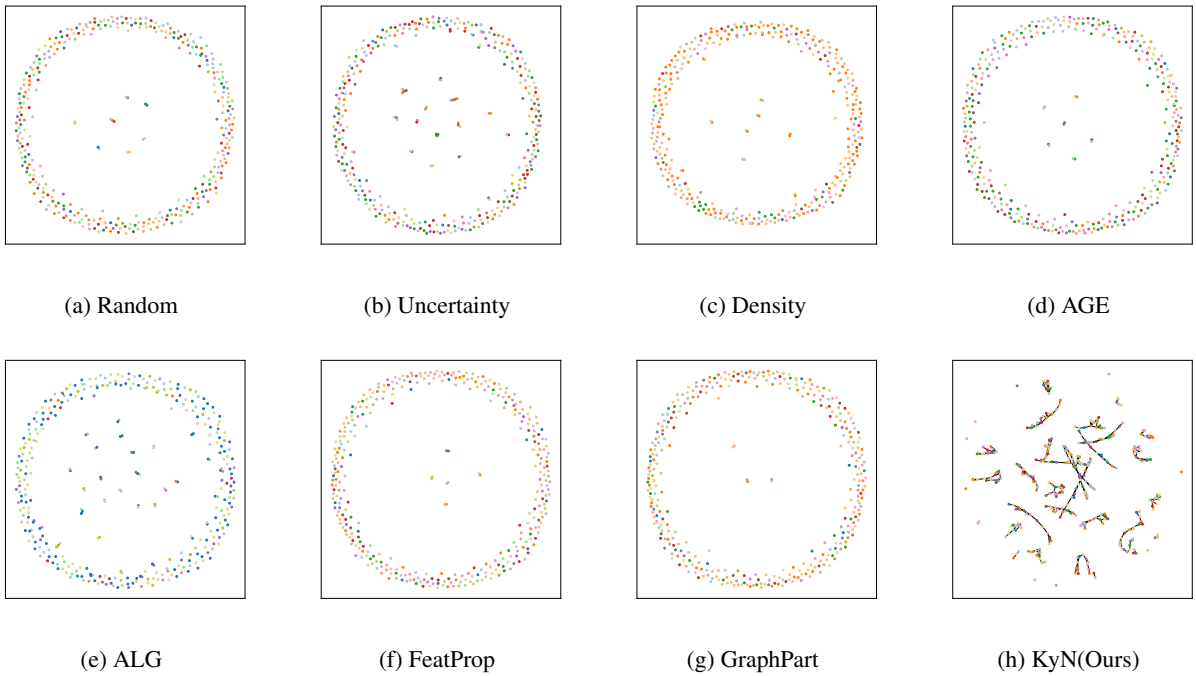
$5C$, $10C$, and $20C$, where C is the number of classes in each dataset. This budget configuration aligns with common practices in previous GAL research (e.g., (Han et al. 2024)). Further implementation details and additional experiments are provided in the appendix.

Experimental results

Performance on heterophilic graphs. Table 1 shows the performance of GALs on heterophilic graphs. The results show that KyN achieves the best performance on all heterophilic graphs with different labeling budgets. As mentioned earlier, we observe that on many heterophilic datasets (e.g., Roman-empire and Minesweeper), previous GALs fail to consistently outperform the naive random sampling. The gap between previous GAL methods and random sampling can even reach as high as 10.4% and 5.6%. Compared to previous GAL methods, the performance improvement of KyN on six datasets can reach up to 12.1%, 1.9%, 8.7%, 6.6%, 6.0% and 5.7%, respectively. More importantly, KyN is the only method that consistently outperforms random sampling across all datasets. The success of KyN is due to the unveiling of the heterophilic nature by the selection training sets. Previous GAL-selected training sets imply homophilic property even on heterophilic graphs. This is because these GALs are only designed for informativeness and coverage of graphs, not homophily. In contrast, we address this issue by the principle of "know your neighbors".

Method	FAGCN	M2M-GNN
Random	52.0 ± 0.5	58.3 ± 1.3
Uncertainty	47.7 ± 1.8	54.9 ± 1.0
Density	45.3 ± 1.1	51.5 ± 1.2
AGE	50.5 ± 1.9	55.7 ± 1.4
ALG	51.2 ± 1.3	56.3 ± 1.2
FeatProp	51.7 ± 0.9	57.0 ± 0.8
GraphPart	51.6 ± 1.2	56.1 ± 0.8
KyN(Ours)	**53.5** ± 1.4	**59.2** ± 1.0

Table 2: The experimental results with heterophilic GNNs as backbones on the Roman-empire dataset.



(a) Random (b) Uncertainty (c) Density (d) AGE

(e) ALG (f) FeatProp (g) GraphPart (h) KyN(Ours)

Figure 4: Case study of selected nodes by different GAL methods on the Roman-Empire dataset. KyN tends to select nodes with their neighbors, which has been shown to facilitate a more accurate estimation of the homophily distribution.

Method	Accuracy (↑)	Runtime (↓)
Random	32.9	0.06
Uncertainty	25.5	0.45
Density	25.1	752
AGE	23.7	3504
ALG	OOT	-
FeatProp	21.6	7245
GraphPart	OOT	-
KyN(Ours)	**33.7**	651

Table 3: The accuracy and runtime (in seconds) on a large heterophilic graph, snap-patents. OOT (out-of-time) indicates the algorithm failed to finish within 24 hours.

Generalization to other backbones. We use two heterophilic GNNs, FAGCN (Bo et al. 2021) and M2M-GNN (Liang et al. 2024), as backbones to compare different GALs on the Roman-empire dataset. The results are presented in Table 2. We observe that KyN still achieves the best performance with these two backbones. In other experiments in this article, we stick to GraphSAGE as the backbone so that readers who are not familiar with heterophilic GNN can understand it more easily.

Performance on a large heterophilic graph. To verify the scalability of KyN, we compare different GAL methods on a large heterophilic graph, snap-patents. This dataset contains more than 2 million nodes and 13 million edges. The results are presented in Table 3. We observe that KyN achieves the best performance and the runtime is also reasonable. This experiment demonstrates the scalability of KyN.

Case study. Figure 4 presents a case study on selected nodes using different GAL methods on the Roman-Empire dataset. We observe that KyN consistently selects more connected nodes compared to previous GAL baselines, aligning with our proposed principle of "know your neighbors". This selection strategy enhances the estimation of local homophily, as supported by Proposition 2, which is a crucial prerequisite for achieving high accuracy (as proved in Proposition 1).

Conclusion

In this paper, we investigate a new research problem: heterophilic graph active learning. We observe that while previous GAL methods perform well on homophilic graphs, they fail to consistently outperform naive random sampling on heterophilic graphs. Through an insightful investigation of the local homophily distribution, we find that previous GAL-selected training sets imply homophilic properties, even on heterophilic graphs. We argue that the previous design principle of informativeness and coverage on graphs will inevitably produce isolated training nodes that are harmful to heterophilic GALs. To address this issue, we propose a novel principle called "know your neighbors" and build the KyN framework. By labeling nodes along with their neighbors, KyN effectively captures the homophilic or heterophilic nature of graphs. We implement KyN using ℓ_1 Lewis weights sampling, which provides strong theoretical guarantees.

Acknowledgments

This work is supported by National Natural Science Foundation of China No. U2241212, No. 62276066, and the Postdoctoral Fellowship Program of CPSF under Grant No. GZC20252742. The computations in this research were performed using the CFFF platform of Fudan University.

References

Bengar, J. Z.; van de Weijer, J.; Twardowski, B.; and Raducanu, B. 2021. Reducing Label Effort: Self-Supervised meets Active Learning. In *ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*, 1631–1639. IEEE.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *AAAI 2021*, 3950–3957. AAAI Press.

Cai, H.; Zheng, V. W.; and Chang, K. C. 2017. Active Learning for Graph Embedding. *CoRR*, abs/1705.05085.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and Deep Graph Convolutional Networks. In *ICML 2020, 13-18 July 2020, Virtual Event*, volume 119.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *ICLR 2021, Virtual Event*. OpenReview.net.

Clarkson, K. L. 2005. Subgradient and sampling algorithms for l1 regression. In *SODA '05*. USA: SIAM.

Clarkson, K. L.; Drineas, P.; Magdon-Ismail, M.; Mahoney, M. W.; Meng, X.; and Woodruff, D. P. 2016. The Fast Cauchy Transform and Faster Robust Linear Regression. *SIAM J. Comput.*, 45(3): 763–810.

Cohen, M. B.; Lee, Y. T.; Musco, C.; Musco, C.; Peng, R.; and Sidford, A. 2015. Uniform Sampling for Matrix Approximation. In *ITCS 2015*. ACM.

Cohen, M. B.; and Peng, R. 2015. L_p Row Sampling by Lewis Weights. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 183–192. ACM.

Drineas, P.; Mahoney, M. W.; and Muthukrishnan, S. 2006. Sampling algorithms for l_2 regression and applications. In *SODA 2006, Miami, Florida, USA, January 22-26, 2006*, 1127–1136. ACM Press.

Gao, L.; Yang, H.; Zhou, C.; Wu, J.; Pan, S.; and Hu, Y. 2018. Active Discriminative Network Representation Learning. In *IJCAI 2018, Stockholm, Sweden*.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023. Addressing Heterophily in Graph Anomaly Detection: A Perspective of Graph Spectrum. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, 1528–1538. ACM.

Gasteiger, J.; Becker, F.; and Günnemann, S. 2021. GemNet: Universal Directional Graph Neural Networks for Molecules. In *NeurIPS 2021, December 6-14, 2021, virtual*.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 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.

Han, H.; Liu, X.; Ma, L.; Torkamani, M.; Liu, H.; Tang, J.; and Yamada, M. 2024. Structural Fairness-aware Active Learning for Graph Neural Networks. In *ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Hu, S.; Xiong, Z.; Qu, M.; Yuan, X.; Côté, M.; Liu, Z.; and Tang, J. 2020. Graph Policy Network for Transferable Active Learning on Graphs. In *NeurIPS 2020, virtual*.

Huang, S.; Lee, G.; Bao, Z.; and Pan, S. 2024a. Cost-effective Data Labelling for Graph Neural Networks. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, 353–364. ACM.

Huang, S.; Li, Y.; Sun, Y.; and Tang, Y. 2024b. One-shot Active Learning Based on Lewis Weight Sampling for Multiple Deep Models. In *ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Karypis, G.; and Kumar, V. 1998. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM J. Sci. Comput.*, 20(1): 359–392.

Kim, Y.; Song, K.; Jang, J.; and Moon, I. 2021. LADA: Look-Ahead Data Acquisition via Augmentation for Deep Active Learning. In *NeurIPS, virtual*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017*.

Lam, H. Y. I.; Pincket, R.; Han, H.; Ong, X. E.; Wang, Z.; Hinks, J.; Wei, Y.; Li, W.; Zheng, L.; and Mu, Y. 2023. Application of variational graph encoders as an effective generalist algorithm in computer-aided drug design. *Nat. Mac. Intell.*, 5(7): 754–764.

Leskovec, J.; and Krevl, A. 2014. SNAP Datasets: Stanford Large Network Dataset Collection.

Li, M.; Miller, G. L.; and Peng, R. 2013. Iterative Row Sampling. In *FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, 127–136. IEEE Computer Society.

Li, X.; Zhu, R.; Cheng, Y.; Shan, C.; Luo, S.; Li, D.; and Qian, W. 2022. Finding Global Homophily in Graph Neural Networks When Meeting Heterophily. In *ICML Baltimore, Maryland, USA*.

Liang, L.; Kim, S.; Shin, K.; Xu, Z.; Pan, S.; and Qi, Y. 2024. Sign is Not a Remedy: Multiset-to-Multiset Message Passing for Learning on Heterophilic Graphs. In *ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Loveland, D.; Zhu, J.; Heimann, M.; Fish, B.; Schaub, M. T.; and Koutra, D. 2023. On Performance Discrepancies Across Local Homophily Levels in Graph Neural Networks. In *Learning on Graphs Conference, 27-30 November 2023, Virtual Event*.

Luan, S.; Hua, C.; Lu, Q.; Ma, L.; Wu, L.; Wang, X.; Xu, M.; Chang, X.; Precup, D.; Ying, R.; Li, S. Z.; Tang, J.; Wolf, G.; and Jegelka, S. 2024. The Heterophilic Graph Learning Handbook: Benchmarks, Models, Theoretical Analysis, Applications and Challenges. *CoRR*, abs/2407.09618.

Ma, C.; Ren, Y.; Castells, P.; and Sanderson, M. 2024. Temporal Conformity-aware Hawkes Graph Network for Recommendations. In *WWW*.

Ma, J.; Ma, Z.; Chai, J.; and Mei, Q. 2023. Partition-Based Active Learning for Graph Neural Networks. *Trans. Mach. Learn. Res.*, 2023.

Mai, T.; Musco, C.; and Rao, A. 2021. Coresets for Classification - Simplified and Strengthened. In *NeurIPS 2021, December 6-14, 2021, virtual*, 11643–11654.

Mao, H.; Chen, Z.; Jin, W.; Han, H.; Ma, Y.; Zhao, T.; Shah, N.; and Tang, J. 2023. Demystifying Structural Disparity in Graph Neural Networks: Can One Size Fit All? In *NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Margatina, K.; Schick, T.; Aletras, N.; and Dwivedi-Yu, J. 2023. Active Learning Principles for In-Context Learning with Large Language Models. In *EMNLP 2023*.

Matsushita, K.; Matsushita, K.; and Hasebe. 2018. *Deep active learning*. Springer.

Munteanu, A.; Schwiegelshohn, C.; Sohler, C.; and Woodruff, D. P. 2019. On Coresets for Logistic Regression. In *INFORMATIK 2019, Kassel, Germany, September 23-26, 2019*, volume P-294 of *LNI*, 267–268. GI.

Ni, X.; Xiong, F.; Zheng, Y.; and Wang, L. 2024. Graph Contrastive Learning with Kernel Dependence Maximization for Social Recommendation. In *WWW 2024, Singapore*.

Pan, E.; and Kang, Z. 2023. Beyond Homophily: Reconstructing Structure for Graph-agnostic Clustering. In *ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*.

Pei, H.; Wei, B.; Chang, K. C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR 2020*. OpenReview.net.

Platonov, O.; Kuznedelev, D.; Babenko, A.; and Prokhorenkova, L. 2023a. Characterizing Graph Datasets for Node Classification: Homophily-Heterophily Dichotomy and Beyond. In *NeurIPS*.

Platonov, O.; Kuznedelev, D.; Diskin, M.; Babenko, A.; and Prokhorenkova, L. 2023b. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-Y.; Li, Z.; Gupta, B. B.; Chen, X.; and Wang, X. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9): 1–40.

Sohler, C.; and Woodruff, D. P. 2011. Subspace embeddings for the L_1-norm with applications. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 755–764. ACM.

Song, Z.; Zhang, Y.; and King, I. 2023. No Change, No Gain: Empowering Graph Neural Networks with Expected Model Change Maximization for Active Learning. In *NeurIPS 2023,*.

Tang, Y.; and Huang, S. 2022. Active Learning for Multiple Target Models. In *NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Tao, X.; Wang, L.; Liu, Q.; Wu, S.; and Wang, L. 2024. Semantic Evolvement Enhanced Graph Autoencoder for Rumor Detection. In *WWW 2024*. ACM.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR 2018*. OpenReview.net.

Wasserman, S.; and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019a. Simplifying Graph Convolutional Networks. In *ICML 2019, 9-15 June 2019, Long Beach, California, USA*.

Wu, J.; and Hooi, B. 2023. DECOR: Degree-Corrected Social Graph Refinement for Fake News Detection. In *KDD*.

Wu, Y.; Xu, Y.; Singh, A.; Yang, Y.; and Dubrawski, A. 2019b. Active Learning for Graph Neural Networks via Node Feature Propagation. *CoRR*, abs/1910.07567.

Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; and Koutra, D. 2022. Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks. In *ICDM 2022*.

Yan, Y.; and Huang, S. 2018. Cost-Effective Active Learning for Hierarchical Multi-Label Classification. In *IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2962–2968. ijcai.org.

Zhang, W.; Shen, Y.; Li, Y.; Chen, L.; Yang, Z.; and Cui, B. 2021a. ALG: Fast and Accurate Active Learning Framework for Graph Convolutional Networks. In *SIGMOD '21, Virtual Event, China, June 20-25, 2021*, 2366–2374. ACM.

Zhang, W.; Wang, Y.; You, Z.; Cao, M.; Huang, P.; Shan, J.; Yang, Z.; and Cui, B. 2021b. RIM: Reliable Influence-based Active Learning on Graphs. In *NeurIPS 2021, December 6-14, 2021, virtual*, 27978–27990.

Zhang, W.; Wang, Y.; You, Z.; Cao, M.; Huang, P.; Shan, J.; Yang, Z.; and Cui, B. 2022. Information Gain Propagation: a New Way to Graph Active Learning with Soft Labels. In *ICLR 2022*.

Zhang, W.; Wang, Y.; You, Z.; Li, Y.; Cao, G.; Yang, Z.; and Cui, B. 2024. NC-ALG: Graph-Based Active Learning Under Noisy Crowd. In *ICDE 2024*.

Zhang, Z.; Strubell, E.; and Hovy, E. H. 2022. A Survey of Active Learning for Natural Language Processing. In *EMNLP 2022, Abu Dhabi, United Arab Emirates*.

Zhu, J.; Rossi, R. A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021. Graph Neural Networks with Heterophily. In *AAAI 2021*, 11168–11176. AAAI Press.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *NeurIPS 2020, December 6-12, 2020, virtual*.