Learnable Coreset Selection for Graph Active Learning

Anonymous authors
Paper under double-blind review

Abstract

Graph Neural Networks (GNNs) have demonstrated their effectiveness in a variety of graph-based tasks. However, their performance heavily depends on the availability of a sufficient amount of labeled data, which is often costly to acquire in real-world applications. To tackle this, GNN-based Active Learning (AL) methods aim to enhance labeling efficiency by selecting the most informative nodes for labeling. However, existing methods often rely on heuristic or implicit approaches that fail to fully capture the influence of labeled data on unlabeled nodes, thereby limiting their adaptability across diverse graph types. In this paper, we propose LearnAL, a Learnable coreset labeling framework for graph Active Learning to address these limitations. Unlike traditional heuristic-based methods, LearnAL explicitly models the correlations between labeled and unlabeled nodes using an attention architecture, linking these correlations directly to prediction performance. Leveraging global influence (attention) scores, LearnAL selects and labels samples that maximize representational diversity, enhancing sample coverage. We provide theoretical analysis demonstrating that this attention-based selection reduces the covering radius bound, improving prediction performance on unlabeled data. Our experimental results show that the labeled coreset significantly enhances the generalizability of various graph models across different graph datasets, as well as CNN models in image classification tasks.

1 Introduction

Graph neural networks (GNNs) (Duvenaud et al., 2015; Kipf & Welling, 2017) have emerged as powerful approaches for learning representations of graph-structured data. It has been noted (Zhang et al., 2022b) that success of GNNs in various graph-based learning tasks (Xu et al., 2018; Klicpera et al., 2019; Zhang & Chen, 2018) requires plenty of labeled data. However, sufficiently informative training data is often not available, as human annotation is expensive and time-consuming, particularly for biological graphs that contain specialized structures requiring expert labeling.

Active learning (AL) provides solutions by selecting and annotating a few highly informative and representative points that can depict a large portion of the data space, especially uncertain regions. While various active learning methods have been proposed and applied for CNN models (Sener & Savarese, 2018; Caramalau et al., 2021; Wang et al., 2016; Yoo & Kweon, 2019) on independent and identically distributed (i.i.d) data, these methods fail to capture both the graph structure and node features, leading to suboptimal performance when applied to GNNs (Gao et al., 2018a; Madhawa & Murata, 2020). Additionally, the interconnected and interdependent nature of nodes in a graph means that the choice of labeled data partitions has a significant impact on GNN performance (Shchur et al., 2018; Fu et al., 2024). Therefore, directly applying active learning methods from CNNs to GNNs is not effective.

To select more representative data on graphs, GNN-based active learning methods (Gao et al., 2018b; Cai et al., 2017; Wu et al., 2019) incorporate graph structural information into query heuristics (uncertainty, diversity, or density). Recent works (Zhang et al., 2022b; 2021e;a;c) considered the characteristic of influence propagation in the graph and proposed a series of graph active learning methods aiming to identify nodes with maximum influence for the rest. For example, the Grain method (Zhang et al., 2021e) formulates labeled node selection as a social influence maximization problem, selecting nodes that influence as many unlabeled nodes as possible. These methods are based on the intuition that nodes close in graph space tend to share the same label, i.e., they primarily exploit the local structure of the graph.

While efficient, these strategies 1) lack a direct correlation with the expected prediction performance on unlabeled nodes in the final task, and 2) mainly focus on the local graph structure, failing to comprehensively explore the influence between labeled and unlabeled data across the entire graph space. However, in real-world applications, graphs can be complex; for instance, in heterophilic graphs, connected nodes may have different labels. These challenges raise a critical question for graph annotation: *Given a fixed labeling budget, how can we develop a general framework that efficiently and effectively identifies core data in the graph by considering both the graph structure (local and global) and features, ultimately improving model performance?*

In this paper, we propose LearnAL, a general graph active learning framework with an attention-based, learnable measure. We frame graph active learning as an unlabeled coreset selection problem for GNNs, with the goal of selecting data that maximizes the coverage of the remaining data in the graph representation space. The key challenge lies in developing an effective metric that 1) is directly linked to prediction accuracy, and 2) comprehensively evaluates the correlations between labeled and unlabeled data by considering the complex graph information (local and global) and feature dependencies. To address this, LearnAL explicitly connects the labeled and unlabeled pools beyond the original graph connections, construct the influence between them using an attention-aggregation strategy in the embedding space, iteratively select core data based on learned attention coefficients from a global perspective. We theoretically demonstrate that selecting unlabeled data with the maximum representation difference from the labeled pool, based on an attention-based metric, not only reduces the coreset selection bound radius δ but also directly decreases the total loss of the graph data. Empirically, we demonstrate the effectiveness of the proposed method across various GNN architectures and different types of graph data (homophily and heterophily) as well as different data scales (small-scale and large-scale). Additionally, we illustrate how LearnAL can serve as a general active learning framework, extending its applicability to image classification. In summary, our main contributions are:

- We propose a general active learning framework with learned measure for graph models, which iteratively selects and annotate data in a graph by addressing the coreset selection problem for non-i.i.d. graph data.
- We theoretically prove the superiority of the attention-based selection strategy: selecting unlabeled data with the maximum representation difference from the current labeled pool can help reduce the bound in graph coreset selection and directly enhance the performance of the graph model.
- Our proposed LearnAL is a general active learning framework that can be applied to both graph data and image tasks. We conduct extensive experiments on both types of data to demonstrate the effectiveness of the proposed method for various classification tasks.

2 Related Works

2.1 Graph Neural Networks

In recent years, GNNs have attracted increasing attention due to their superiority in the processing of graph-structured data (Henaff et al., 2015; Gilmer et al., 2017; Bronstein et al., 2017; Velickovic et al., 2018). To improve the expressive power of GNNs, different message-passing schemes have been developed to propagate and aggregate neighborhood information (Kipf & Welling, 2017; Velickovic et al., 2018; Feng et al., 2020). Recently, some studies tried to understand the generalization ability of GNNs from the perspective of training data. Zhu et al. (2021) explored the influence of training data and presented Shift-Robust GNN (SR-GNN), designed to account for distributional differences between biased training data and a graph's true inference distribution. Ma et al. (2021) extended PAC-Bayesian analysis for graph data to analyze the generalization performance of GNNs, and demonstrated that the distance between a test subgroup and the training set can be a key factor affecting the GNN performance. Su et al. proved that the distance of the training set to the rest of the vertexes in the graph is negatively correlated to the learning outcome of GNNs.

2.2 Active Learning on Graphs

In practice, obtaining sufficient informative training data is challenging, as human annotation is expensive and time-consuming. Active learning and semi-supervised representation learning with few labels are both designed to address the scarcity of labeled data, but from different perspectives. While few-labeled semi-supervised learning focuses on comprehensively leveraging the small amount of labeled data and the large amount of unlabeled data to achieve better performance, active learning focuses on selecting and labeling the most informative nodes to maximize model performance with minimal cost.

Generally, active learning is an iterative labeling process in which a model is learned at every iteration, and a set of data points is chosen to be labeled from a pool of unlabelled points to maximize model performance. Based on the query strategy, the majority of work can be divided into three categories (Settles, 2009): theoretically-motivated methods (MacKay, 1992), ensemble approaches (McCallum et al., 1998; Freund et al., 1997) and uncertainty based (Tong & Koller, 2001; Li & Guo, 2013; Settles & Craven, 2008). Demir et al. (2010) used a heuristic to first filter the pool based on uncertainty and then choose the points to label using diversity. Sener & Savarese (2018) proposed an effective batch active learning method for deep CNNs. In this method, the active learning problem is defined as coreset selection; however, it is only for nonstructural data. Several attempts have been made for applying AL on graph-structured data (Bilgic et al., 2010; Gu et al., 2013; Kuwadekar & Neville, 2011) based on a graph signal processing framework. Subsequently, a series of GNN-based AL methods (Cai et al., 2017; Gao et al., 2018a) have been studied using different metrics, including uncertainty, information density, and graph centrality to evaluate training data. However, simply combining these metrics may not select informative data. Recently, several works (Cui et al., 2022; Ma et al., 2022; Zhang et al., 2022a; Fuchsgruber et al., 2024; Wu et al., 2019; Li et al., 2020; Zhang et al., 2021c; Song et al., 2024; Han et al., 2023; Hardiman-Mostow et al., 2025) were proposed to further consider the graph information. To maximize the coverage of the labeled data, a new node selection metric is proposed in ALG (Zhang et al., 2021a) to maximize the effective reception field. Grain (Zhang et al., 2021e) further generalizes the reception field to the number of activated nodes in social influence maximization. Reinforcement learning (Hu et al., 2020a; Yu et al., 2024) and LLM (Chen et al., 2023; Li et al., 2024) are also used to improve active learning on graphs. Despite the progress in node-level tasks, several studies (Samoaa et al., 2024; 2023) have extended active learning to graph-level regression, proposing unified frameworks and batch-mode strategies that leverage graph embeddings to improve sample efficiency and scalability under limited labeling budgets. While most existing approaches are based on some query heuristics to implicitly encode the relationships between labeled and unlabeled data, they often struggle to identify truly informative data points in the face of complicated graph structures.

2.3 Coreset Selection

Coresets are small, informative subsets of data that allow models trained on them to perform well on the full dataset. Several works, such as those by Wei et al. (2015); Mirzasoleiman et al. (2020); Killamsetty et al. (2021a), have studied the efficient training of deep learning models using selected coresets. Mirzasoleiman et al. (2020) focused on selecting representative coresets of the training data that closely estimate the full training gradient. Killamsetty et al. (2021b) treated coreset selection as an optimization problem for the validation set loss, aiming for efficient learning with a focus on generalization. Killamsetty et al. (2021a) proposed GRAD-MATCH, which selects subsets approximating the full training loss or validation loss gradient using orthogonal matching pursuit. Meanwhile, coreset selection methods (Sener & Savarese, 2018; Ash et al., 2019) were also used for active learning scenarios, where a subset of data instances from the unlabeled set is selected to be labeled.

3 Problem Statement and Notations

In this section, we formally define the problem of active learning for GNNs under the semi-supervised node classification setting.

We are given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V}$ and edge set $\mathcal{E}$. Suppose there are N nodes in $\mathcal{V}$ and each node $v_i \in \mathcal{V}$ has an associated feature vector $x_i \in \mathbf{X} \in \mathbb{R}^{N \times d}$ and a label vector $y_i \in \mathbf{Y} \in \{0, 1\}^{N \times C}$.

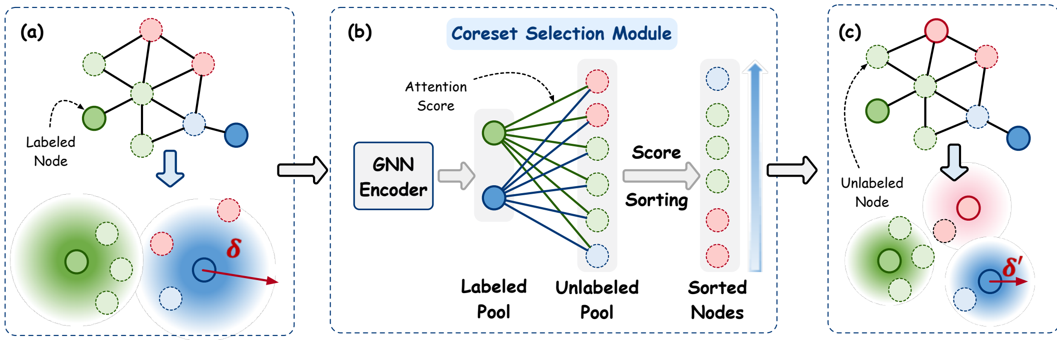

Figure 1: (a) The input graph and a visualization of the influence of labeled data with bound δ in embedding space. (b) LearnAL explicitly construct the influence of data in unlabeled pool for labeled data by an attention-based networks. The unlabeled data which have minimum representation influence are selected into labeled pool. (c) The output graph with selected informative data and a visualization of the decreased bound δ'.

The connection among nodes can be described by the adjacency matrix $\mathbf{A}$, with $\mathbf{A}_{ij} = 1$ if there exists an edge (v_i, v_j), otherwise $\mathbf{A}_{ij} = 0$. Here, we focus on the C class node classification task on graph $\mathcal{G}$, with a label space $\mathcal{Y} = \{1, \ldots, C\}$.

In active learning on graphs, we consider that the full node set $\mathcal{V}$ is partitioned into training set $\mathcal{V}_{train}$, validation set $\mathcal{V}_{val}$, and test set $\mathcal{V}_{test}$. The training set $\mathcal{V}_{train}$ contains labeled and unlabeled data. An active learning algorithm $A_{\mathbf{s}}$ iteratively selects data from the unlabeled pool $\mathcal{V}_u$ and gives labels to them into a labeled pool $\mathcal{V}_l$. With a labeling budget $\mathcal{B}$ and an initial labeled pool $\mathbf{s}^0 = \left\{s_j^0 \in \mathcal{V}_{train}\right\}_{j<m}$, an active learning algorithm expects to minimize the future expected loss with a GNN model $\mathcal{M}$ by:

$$\min_{\mathbf{s}^{k+1}:|\mathbf{s}^{k+1}|\leq\mathcal{B}} E_{x,y}\left[l_{\mathcal{M}}\left(\mathcal{G}, x, y; A_{\mathbf{s}^0\cup\ldots\mathbf{s}^{k+1}}\right)\right]. \tag{1}$$

4 Learnable Corset Labeling

In this section, we present LearnAL, a general graph active learning (as shown in Fig. 1) that leverages a learnable approach to select a coreset from graph data, enhancing the generalization ability of various GNN models. We define graph active learning as a coreset selection problem without labels on GNNs in Section 4.1. To address this problem, we theoretically show that labeling the unlabeled data that have maximum representation difference compared to the existing labeled pool obtains a smaller bound radius δ and reduce the total prediction loss in Section 4.2. Hence we construct an attention-based model to explicitly evaluate the influence of each unlabeled sample u for each labeled sample v. While capturing the correlations between labeled and unlabeled data from a global perspective, LearnAL selects and labels data according to attention scores at each batch of data labeling, enhancing the information complexity in the labeled pool. The above process is repeated until the labeling budget $\mathcal{B}$ runs out. We introduce each component of LearnAL in Section 4.3.

4.1 Coreset Selection on Graphs

Generally, the active learning algorithm acquires one label at a time by querying the oracle in each iteration. However, for graphs containing a large number of nodes and edges, this is infeasible. Therefore, we focus on the batch active learning technique (Contardo et al., 2017) in which the active learning algorithm chooses a set of data to be labeled by an oracle at each iteration. In a classification problem, given a training set, the loss of the training set can be calculated as:

$$R_{emp} = \frac{1}{|\mathbf{s}|}\sum_{i\in\mathbf{s}} l_{\mathcal{M}}\left(\mathcal{G}, x_i, y_i; A_{\mathbf{s}}\right), \tag{2}$$

where $|\mathbf{s}|$ is the number of labeled points and the empirical risk R_{emp} is the average loss of all training samples. After training a GNN model $\mathcal{M}$, the aim is to predict the outputs for new or unseen data. Among the generated hypotheses, the best hypothesis is the one that minimizes the expected value of the loss over the whole input space, which is defined as:

$$R = \frac{1}{N} \sum_{j \in [N]} l_{\mathcal{M}} \left(\mathcal{G}, x_j, y_j; A_{\mathbf{s}} \right). \tag{3}$$

When designing an active learning algorithm method for GNNs, the goal is to minimize the generalization gap between R_{emp} and R:

$$\min_{\mathbf{s}^1 : |\mathbf{s}^1| \leq \mathcal{B}} \left| \frac{1}{N} \sum_{i \in [N]} l_{\mathcal{M}} \left(\mathcal{G}, x_i, y_i; A_{\mathbf{s}^0 \cup \mathbf{s}^1} \right) - \frac{1}{|\mathbf{s}^0 + \mathbf{s}^1|} \sum_{j \in \mathbf{s}^0 \cup \mathbf{s}^1} l_{\mathcal{M}} \left(\mathcal{G}, x_j, y_j; A_{\mathbf{s}^0 \cup \mathbf{s}^1} \right) \right|, \tag{4}$$

where $[N] = \{1, \ldots, N\}$. In other words, given the initial labeled set $\mathbf{s}^0$ and the budget $\mathcal{B}$, we aim to find a set of points to query labels $\mathbf{s}^1$ such that when we train a GNN model, the performance of the model on the labeled subset is as close as possible to its performance on the entire dataset.

4.2 Theoretical Analysis

The optimization objective equation 4 is not directly computable since we do not have access to all the labels. In Sener & Savarese (2018), an upper bound is given to the objective function of coreset on CNNs. As shown in Theorem 1 (Sener & Savarese, 2018), we can bound this loss with covering radius δ., i.e., $\left| \frac{1}{N} \sum_{i \in [N]} l \left(\mathbf{x}_i, y_i, A_{\mathbf{s}} \right) - \frac{1}{|\mathbf{s}|} \sum_{j \in \mathbf{s}} l \left(\mathbf{x}_j, y_j; A_{\mathbf{s}} \right) \right| \leq \mathcal{O} \left(\delta_{\mathbf{s}} \right) + \mathcal{O} \left(\sqrt{\frac{1}{N}} \right)$ where $\delta_{\mathbf{s}}$ with radius δ centered at each labeled sample in s can cover the entire representation space. The visualization about covering radius δ can be found in Figure 1 (a). Similarly, Theorem 1 in FeatProp (Wu et al., 2019) suggests that the classification loss of GNNs is informally bounded by the covering radius of the labeled set. Obviously, if we want to reduce the loss, we need to decrease the covering radius.

Although this bound provides the original analysis of coreset selection, it is also important for directly analyzing the influence of training/labeled data on the prediction performance of testing/unlabeled data, offering theoretical guarantees for AL performance, particularly in the context of graph AL.

Proposition 4.1 *Given a graph $\mathcal{G}$, for any labeled data v with the hidden representation $\boldsymbol{h}_v$, there exist a $\delta_v > 0$, such that for two unlabeled nodes $\{u, u'\}$ with representations $\boldsymbol{h}_u$ and $\boldsymbol{h}_{u'}$, if the distance $\Delta(u, v) < \Delta(u', v) < \delta_v$, then $l(f(\boldsymbol{h}_u)) < l(f(\boldsymbol{h}_{u'}))$, where $f(\cdot)$ is the prediction function, and $l(\cdot)$ is the loss function.*

Assume that each training sample v has a hidden representation $\boldsymbol{h}_v$ with a neighborhood covered within a radius $\delta_v$in the embedding space. This proposition states that the prediction loss of samples within a radius δ_v around point v increases monotonically with their distance to v. Proposition 4.1 is formulated in terms of abstract points in the embedding space, making it applicable to both i.i.d. data and non-i.i.d. graph data. In the case of graph data, the embedding space inherently captures the complex structural relationships encoded in the graph.

Extending this to the entire graph data, we can conclude that a GNN trained on a training set with closer distances to the remaining data—indicating an approximate coverage of the whole representation space with a smaller covering radius—exhibits better performance in Lemma 4.1.

Lemma 4.1 *Assume there are two training set $\mathcal{V}_{train}$ and $\mathcal{V}'_{train}$, and test set $\mathcal{V}_{test}$. Based on two training set, we get two trained GNN models $\mathcal{M}(\mathcal{V}_{train})$ and $\mathcal{M}(\mathcal{V}'_{train})$. If $\sum_{v \in \mathcal{V}_{train}} \Delta(v, \mathcal{V}_{test}) < \sum_{u \in \mathcal{V}'_{train}} \Delta(u, \mathcal{V}_{test})$, thus the covering radius $\delta_{v \in \mathcal{V}_{train}} < \delta_{u \in \mathcal{V}'_{train}}$, we have $\sum_{u \in \mathcal{V}_{test}} l(f_{\mathcal{M}(\mathcal{V}_{train})}(u)) < \sum_{u \in \mathcal{V}_{test}} l(f_{\mathcal{M}(\mathcal{V}'_{train})}(u))$.*

According to Lemma 4.1, addressing coreset selection on graphs involves identifying nodes that exhibit the smallest distances to the remaining nodes in the embedding space. The key challenge then is to define a measure (distance) that is directly related to the final prediction performance, rather than relying on heuristic or implicit methods.

Lemma 4.2 *Assume a graph has a labeled data, v, with the hidden representation $\boldsymbol{h}_v$, and two unlabeled nodes, $\{u_1, u_2\}$, with representations $\{\boldsymbol{h}_{u_1}, \boldsymbol{h}_{u_2}\}$. If $\boldsymbol{h}_v = \alpha_1 \boldsymbol{h}_{u_1} + \alpha_2 \boldsymbol{h}_{u_2}$, where the score $\alpha_1 > \alpha_2 \approx 0$, then selecting u_2 into the labeled pool (as $u_2 \to v_2$) results in a smaller total loss across the entire graph space than selecting u_1 into the labeled pool (as $u_1 \to v_1$).*

Proof. Let v be a labeled input, and u_1 and u_2 be two unlabeled inputs. Let the hidden representation of v be $\boldsymbol{h}_v = \alpha_1 \boldsymbol{h}_{u_1} + \alpha_2 \boldsymbol{h}_{u_2}$, where $\alpha_1 > \alpha_2 \approx 0$, i.e., $\Delta(v, u_1) < \Delta(v, u_2) \leq \delta_v$ where δ_v is the covering radius of point v. The loss of v is then given by $l(\boldsymbol{h}_v)$, where $l(\cdot)$ is the loss function. For simplicity, we assume zero training loss (see Assumption 2 in Appendix A), leading to: $l(\boldsymbol{h}_v) = 0$.

Selecting the data u_2 that has the maximum difference with v according to scores $\{\alpha_1, \alpha_2\}$ and adding it to the labeled pool as $u_2 \to v_2$, we now have two training nodes, v and v_2, and one testing node, u_1. The total loss L_1 on the entire input space is: $L_1 = l(\boldsymbol{h}_v) + l(\boldsymbol{h}_{v_2}) + l(\boldsymbol{h}_{u_1}) \approx l(\boldsymbol{h}_{u_1})$ since the training loss on the training set is zero. From the perspective of the covering radius, the loss $l(\boldsymbol{h}_{u_1}) \leq \mathcal{O}(\delta_1) \leq \mathcal{O}(\max(\Delta(v, u_1), \Delta(v_2, u_1)))$.

Consider the scenario in which u_1 is selected to the labeled pool, we get the loss $L_2 = l(\boldsymbol{h}_v) + l(\boldsymbol{h}_{v_1}) + l(\boldsymbol{h}_{u_2}) \approx l(\boldsymbol{h}_{u_2})$, $l(\boldsymbol{h}_{u_2}) \leq \mathcal{O}(\delta_2) \leq \mathcal{O}(\max(\Delta(v, u_2), \Delta(v_1, u_2)))$.

As $\Delta(v, u_1) < \Delta(v, u_2)$ and $\Delta(v_2, u_1) = \Delta(v_1, u_2)$, we have $\max(\Delta(v, u_1), \Delta(v_2, u_1)) < \max(\Delta(v, u_2), \Delta(v_1, u_2))$, thus, $\delta_1 < \delta_2$. According to Proposition 4.1, we have $l(\boldsymbol{h}_{u_1}) < l(\boldsymbol{h}_{u_2})$, $L_1 < L_2$. ☐

Lemma 4.2 establishes a connection between coreset selection and prediction loss through an attention-based scoring mechanism. This lemma clearly demonstrates the benefit of selecting the coreset based on an explicit metric (representation scores) between the labeled and unlabeled pools in improving expected prediction accuracy, supporting the learnable coreset selection for active learning. Intuitively, in graph learning, the attention scores reflect the representational redundancy between labeled and unlabeled nodes in the learned embedding space. Nodes with higher scores are already well captured by the current labeled set, whereas nodes with lower scores tend to reside in under-explored regions of the representation manifold.

With Lemma 4.1 and 4.2, we establish a connection between the core data selection and prediction loss on GNNs, leading to the following theorem.

Theorem 4.1 *Given the whole sample $\mathcal{V}$ drawn from $\mathcal{G}$, let $\mathcal{V}_l$ represents the labeled pool consisting of points with labels, and let $\mathcal{V}_u$ denotes the set of unlabeled data. $\exists s \in \mathcal{V}_u : \forall v \in \mathcal{V}_l, A_{s,v} < A_{k,v}$ with $k \in \mathcal{V}_u \backslash \{s\}$, where $A_{i,j}$ measures the representation similarity between nodes i and j, the larger $A_{i,j}$, the closer nodes i and j are. Thus, we have $\delta_{v \in \mathcal{V}_l \cup \{s\}} < \delta_{v \in \mathcal{V}_l \cup \{k\}} < \delta_{u \in \mathcal{V}_l}$, such that $\sum_{i \in \mathcal{V}} l(f_{\mathcal{M}(\mathcal{V}_l \cup \{s\})}(i)) < \sum_{i \in \mathcal{V}} l(f_{\mathcal{M}(\mathcal{V}_l \cup \{k\})}(i)) < \sum_{i \in \mathcal{V}} l(f_{\mathcal{M}(\mathcal{V}_l}(i))$.*

Theorem 4.1 indicates that node s is the most informative data point in $\mathcal{V}_u$ with respect to the the existing labeled pool $\mathcal{V}_l$ to improve prediction performance.

4.3 Attention-based Message-Passing and Data Selection

According to Theorem 4.1, we design an attention-based graph coreset labeling method to effectively identify the unlabeled data with minimum representation influence to improve the generalization ability of the model (as illustrated in Figure 1 (b)).

To obtain the correlations between the labeled and the unlabeled pool, we first learn the hidden representation of nodes by GNN layers to encode the structural information in a graph. Let $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times d}$ be the node features, the l^{th} layer GNN is given by:

$$\begin{aligned} \boldsymbol{a}_v^{(l)} &= \text{Aggregation}^{(l)} \left(\left\{ \boldsymbol{h}_u^{(l-1)} : u \in \mathcal{N}(v) \right\} \right), \\ \boldsymbol{h}_v^{(l)} &= \text{Update}^{(l)} \left(\boldsymbol{h}_v^{(l-1)}, \boldsymbol{a}_v^{(l)} \right), \end{aligned} \tag{5}$$

where $\boldsymbol{h}_v^{(l)}$ is the hidden feature vector of node v at the l^{th} layer. We initialize $\boldsymbol{h}_v^{(0)} = x_v$, and $\mathcal{N}(v)$ is a set of nodes connected to v. We call Aggregation$(\cdot)$ an aggregation function and Update$(\cdot)$ an update function. For example, the layer-wise message-passing in GCN Kipf & Welling (2017) is defined as $\boldsymbol{h}_v^{(l)} = \mathrm{ReLU}\left(\mathbf{W} \cdot \mathrm{MEAN}\left\{\boldsymbol{h}_u^{(l-1)}, \forall u \in \mathcal{N}(v) \cup \{v\}\right\}\right)$, where $\mathbf{W}$ is a layer-specific trainable weight matrix. We obtain the l^{th}-layer hidden representations as $\boldsymbol{h}^{(l)} = [\boldsymbol{h}_1^{(l)}, \ldots, \boldsymbol{h}_N^{(l)}] \in \mathbb{R}^{N \times d'}$.

Then, we aim to assess the influence of unlabeled data for the existing labeled dataset from a global perspective. Specifically, we employ an attention architecture to explicitly model the relationships between the labeled and unlabeled data pools in the representation space. Assume the labeled pool $\mathcal{V}_l = \{v_1, \ldots, v_m\}$ with features $\boldsymbol{h}^{v\,(l)} = [\boldsymbol{h}_1^{v\,(l)}, \ldots, \boldsymbol{h}_m^{v\,(l)}]$ and unlabeled pool $\mathcal{V}_u = \{u_1, \ldots, u_n\}$ with features $\boldsymbol{h}^{u\,(l)} = [\boldsymbol{h}_1^{u\,(l)}, \ldots, \boldsymbol{h}_n^{u\,(l)}]$ in l^{th} layer. Consider the global information in the graph, for each labeled data v_i, we expect to represent it by aggregating the information from the unlabeled pool. Connecting v_i with all unlabeled data in $\mathcal{V}_u$, the hidden representation $\boldsymbol{h}_i^{v\,(l)}$ can be obtained by the labeled node v_i acting as the query $\boldsymbol{q}_i^v$ with $\boldsymbol{q}_i^{v\,(l)} = \boldsymbol{h}_i^{v\,(l-1)}\mathbf{W}^Q$:

$$\begin{aligned} A_i^{s\,(l)} &= \alpha \boldsymbol{q}_i^{v\,(l)} \mathbf{K}_{\mathcal{V}_u}^\top, \\ \boldsymbol{h}_i^{v\,(l)} &= \mathrm{softmax}\left(A_i^{s\,(l)}\right) \mathbf{V}_{\mathcal{V}_u}, \end{aligned} \tag{6}$$

where α is a constant scalar ($\alpha = \frac{1}{\sqrt{d'}}$), $\mathbf{K}_{\mathcal{V}_u} = \boldsymbol{h}^u \mathbf{W}^K$ and $\mathbf{V}_{\mathcal{V}_u} = \boldsymbol{h}^u \mathbf{W}^V$ are the key and value matrices of unlabeled pool, respectively. In this way, each labeled node aggregates the information from all unlabeled data in $\mathcal{V}_u$, and the attention score $A_i^{s\,(l)}$ measures the importance of samples in the unlabeled pool for labeled data v_i in representation space.

Similarly, viewing each unlabeled node u_i as query $\boldsymbol{q}_i^u$, its hidden representation can be achieved by aggregating the information from all labeled data:

$$\boldsymbol{q}_i^{u\,(l)} = \boldsymbol{h}_i^{u\,(l-1)} \mathbf{W}^Q, \boldsymbol{h}_i^{u\,(l)} = \mathrm{softmax}\left(\alpha \boldsymbol{q}_i^{u\,(l)} \mathbf{K}_{\mathcal{V}_v}^\top\right) \mathbf{V}_{\mathcal{V}_v}, \tag{7}$$

where $\mathbf{K}_{\mathcal{V}_v} = \boldsymbol{h}^v \mathbf{W}^K$ and $\mathbf{V}_{\mathcal{V}_v} = \boldsymbol{h}^v \mathbf{W}^V$ are the key and value matrices of labeled pool, respectively.

The equation 6 and equation 7 indicate the computation on single-head attention. In practice, LearnAL adopts multi-head attention (MHA) followed by feed-forward blocks (FFN) and layer normalization (LN($\cdot$)) as:

$$\begin{aligned} \boldsymbol{h}'^{(l)} &= \mathrm{LN}\left(\mathrm{MHA}\left(\boldsymbol{h}^{(l-1)}\right)\right) + \boldsymbol{h}^{(l-1)}; \\ \boldsymbol{h}^{(l)} &= \mathrm{LN}\left(\mathrm{FNN}\left(\boldsymbol{h}'^{(l)}\right)\right) + \boldsymbol{h}'^{(l)}, \end{aligned} \tag{8}$$

where $\boldsymbol{h}^{(l)}$ is the representation of labeled and unlabeled data in l^{th} layer. In addition, we incorporate positional encoding, including random walk positional encoding Dwivedi & Bresson (2021) and Laplacian positional encoding Dwivedi et al. (2021), which are crucial components in transformers, into our proposed LearnAL.

The model is trained using a supervised loss over the labeled set:

$$\mathcal{L} = \frac{1}{|\mathcal{V}_l|} \sum_{v_i \in \mathcal{V}_l} \ell\big(f(\boldsymbol{h}_i^{(L)}), y_i\big), \tag{9}$$

where $f(\cdot)$ is the prediction head, $\boldsymbol{h}_i^{(L)}$ is the final-layer representation, and $\ell(\cdot)$ denotes the cross-entropy loss.

Data selection. According to Theorem 4.1, to reduce the total loss of input data, we need to select nodes in the unlabeled pool that have the maximum representation difference to the nearest labeled data. Intuitively, data with the smallest similarity to the existing labeled data in the representation space will help maximize sample diversity and complexity. Based on the LearnAL, the attention matrix A^s has explicitly show the importance of nodes in the unlabeled pool for labeled data, thus, we sample node by:

$$u = \arg\min_{u \in \mathcal{V}_u} \max_{v \in \mathcal{V}_v} A_{v,u}^s. \tag{10}$$

Then, we get the labeled pool $\mathcal{V}_l = \mathcal{V}_l \cup \{u\}$.

4.4 Complexity Analysis and Algorithm of LearnAL

In LearnAL, the complexity of the attention-based networks is $\mathcal{O}(LU)$, where L is the number of labeled data, U is the number of unlabeled nodes. For data selection with a greedy searching method, the complexity is $\mathcal{O}(LUm + Ulog(U))$, where m is the dimension of the low-dimensional embedding. The whole computation process is shown in Algorithm 1.

Algorithm 1 Learnable Coreset Labeling with Attention Architecture

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, batch budget b, labeling budget $\mathcal{B}$.
Output: Labeled pool s.

Initialize labeled pool $s = s^0$
Initialize model parameters Θ
while $|s| < \mathcal{B}$ **do**
    // **Model training**
    Compute hidden node embeddings via a GNN model
    Compute the node representations with attention-based matrix $\mathbf{A}$
    Compute supervised loss:
    $\mathcal{L} = \frac{1}{|s|} \sum_{v_i \in s} \ell\big(f(\boldsymbol{h}_i^{(L)}), y_i\big)$
    Update Θ via backpropagation
    // **Node selection**
  for $i = 1$ to b **do**
        Select node u_i with maximum representation difference to its nearest labeled node based on attention $\mathbf{A}$ (Eq. 10)
        $s \leftarrow s \cup \{u_i\}$
  end for
end while

5 Experiments

We conduct experiments to verify that the labeled data selected by our proposed LearnAL can enhance the generalization of different graph models. We focus on four popular GNN models: GCN Kipf & Welling (2017), GAT Velickovic et al. (2018), H2GCN Zhu et al. (2020) and GPRGNN Chien et al. (2021). The framework is adaptable to any general GNN model; additional results using APPNP Klicpera et al. (2019) and GraphSAGE Hamilton et al. (2017) are provided in Appendix E. Our results show that the core graph data identified by our method can achieve improved performance regardless of the GNN architecture. Additionally, we apply LearnAL to image classification tasks to demonstrate the generalizability of the proposed method. We conduct an ablation study to analyze how positional encoding in the self-attention framework influences the performance of LearnAL, followed by a robustness evaluation under noisy node features and graph structures.

Datasets. Focusing on semi-supervised node classification, we experiment on a range of graph benchmarks: (1) homophilic graph datasets (Cora, Citeseer, Pubmed, and ogbn-arxiv) (Pei et al., 2020; Hu et al., 2020b) and (2) heterophilic graph datasets (Actor, Squirrel, roman-empire, Penn94) (Zhu et al., 2020; Platonov et al., 2023; Lim et al., 2021) involving diverse domains and sizes (roman-empire, Penn94 and ogbn-arxiv are large-scale datasets). We also perform experiments on CIFAR-10 (Krizhevsky et al., 2009) and FashionMNIST (Xiao et al., 2017; Griffin et al., 2007) datasets for image classification. The details of these datasets are provided in Appendix C.

5.1 Experimental Setting

We compare our proposed method with other active learning approaches for graphs: Random, Standard, FeatProp (Wu et al., 2019), AGE (Cai et al., 2017), GRAIN (Zhang et al., 2021e), RIM (Zhang et al., 2021d), ALG (Zhang et al., 2021b), GraphPart (Ma et al., 2022), UQ_AL (Fuchsgruber et al., 2024), and NC-ALG (Zhang et al., 2024). In general active learning, the initial pool is usually uniformly randomly selected from

Table 1: Classification accuracy (%) on three citation datasets with different training sets (mean accuracy (%) and standard deviation over 5 different runs).

Methods	Training Data	Cora	Citeseer	Pubmed
GCN	Random	79.81 ± 1.73	70.24 ± 2.04	76.54 ± 2.60
	Standard	82.31 ± 0.47	71.45 ± 0.69	79.59 ± 0.41
	AGE	80.95 ± 1.14	70.34 ± 7.01	79.50 ± 2.69
	FeatProp	77.3 ± 1.36	64.0 ± 3.21	73.2 ± 1.94
	GRAIN	80.96 ± 0.40	70.96 ± 0.42	79.94 ± 0.33
	RIM	81.78 ± 0.52	72.45 ± 0.70	76.04 ± 0.83
	ALG	83.01 ± 0.28	71.8 ± 0.09	78.52 ± 0.04
	GraphPart	82.50 ± 0.43	71.67 ± 0.64	79.64 ± 0.35
	UQ_AL	78.86 ± 0.54	65.80 ± 2.45	74.22 ± 5.62
	NC-ALG	83.02 ± 0.53	72.11 ± 0.37	**80.23 ± 0.81**
	LearnAL	**83.92 ± 0.54**	**73.10 ± 0.58**	79.83 ± 0.34
	LearnAL (cosine)	81.02 ± 0.27	70.90 ± 0.42	76.75 ± 0.91
GAT	Random	80.60 ± 1.42	70.94 ± 1.77	76.84 ± 3.72
	Standard	82.06 ± 0.56	71.38 ± 0.76	77.74 ± 0.84
	AGE	81.42 ± 0.66	70.32 ± 0.74	79.50 ± 1.81
	FeatProp	76.9 ± 1.69	59.0 ± 2.81	68.3 ± 3.19
	GRAIN	80.44 ± 0.81	70.76 ± 0.37	79.67 ± 0.60
	RIM	82.30 ± 0.70	73.08 ± 0.67	76.44 ± 1.07
	ALG	82.92 ± 0.47	71.28 ± 0.35	78.86 ± 0.53
	GraphPart	82.59 ± 0.82	70.78 ± 0.65	77.76 ± 0.61
	UQ_AL	79.42 ± 1.67	66.74 ± 3.40	77.00 ± 3.95
	NC-ALG	82.63 ± 0.93	71.27 ± 0.47	79.22 ± 1.32
	LearnAL	**83.68 ± 0.39**	**72.92 ± 0.57**	**79.82 ± 0.50**
	LearnAL (cosine)	81.03 ± 1.01	69.95 ± 1.30	78.70 ± 0.24

the whole data. For i.i.d. data, this initial data selection method is reasonable. However, for non-i.i.d. graphs in which nodes are connected by edges, it is of great importance to utilize initial knowledge of the graph. Thus, except for the random selection, we propose a structure and feature-based initial pool selection method.

Considering both the features and graph structure, we propagate features among nodes with the layer-wise propagation rule:

$$\boldsymbol{H}^{(l+1)} = \hat{\mathbf{A}}\boldsymbol{H}^{(l)}, \tag{11}$$

where $\hat{\mathbf{A}} = \hat{\mathbf{D}}^{-1/2}(\mathbf{A}+\mathbf{I})\hat{\mathbf{D}}^{-1/2}$ is a symmetric normalized adjacency matrix, $\mathbf{I}$ is the identity matrix, $\hat{\mathbf{D}}$ is the corresponding degree matrix of $\mathbf{A}+\mathbf{I}$, and $\boldsymbol{H}^{(l)}$ is the hidden node representation in l^{th} layer with $\boldsymbol{H}^{(0)} = \mathbf{X}$. After k iterations of aggregation, the representation of a node $\boldsymbol{h}_i^k$ captures the structural information within its k-hop neighborhood. Then, we select k nodes into the initial pool using the k-medoids method.

Our method introduces several hyperparameters, including the number of initial labels $|s^0|$, batch budget b, and final labeling budget $\mathcal{B}$, where $|s^0| < \mathcal{B}$. For fair comparison, all methods select the same number of core samples as the standard training set on Cora, Citeseer, and Pubmed. For the ogbn-arxiv dataset, the labeling budget is set to 800, and for the heterophilic datasets, the labeling budget is 600. For $|s^0|$ and b, we perform a hyperparameter search for each dataset. The initial labels in $|s^0|$ are chosen to provide sufficient supervision for early training stages, ensuring the acquisition of reliable attention coefficients. In our experiments, we observed that setting $|s^0| = 0.3 \times \mathcal{B}$ is effective for commonly used datasets, enabling the selection of a representative coreset from the unlabeled data. For other hyperparameters used in our experiments, including the learning rate, early stopping patience, hidden layer size, dropout rates of the input layer and hidden layer, we usually adopt a similar setting as in Kipf & Welling (2017); Velickovic et al. (2018); Klicpera et al. (2019). Furthermore, all the experiments are conducted on a Linux server equipped with NVIDIA A100. The detailed parameters used in the experiments are listed in Appendix B.

Table 2: Classification accuracy (%) on three heterophilic datasets with different training sets (mean accuracy (%) and standard deviation over 5 different runs).

Methods	Training Data	Actor	Squirrel	roman-empire
GCN	Random	28.47 ± 0.93	25.94 ± 1.67	16.63 ± 2.12
	AGE	25.38 ± 0.38	23.50 ± 1.32	10.15 ± 3.83
	GRAIN	26.47 ± 0.51	25.13 ± 0.31	4.17 ± 0.00
	LearnAL	**29.14 ± 2.52**	**27.53 ± 1.19**	**20.31 ± 0.96**
H2GCN	Random	31.51 ± 0.68	34.68 ± 0.95	21.36 ± 0.30
	AGE	28.71 ± 1.68	27.26 ± 2.25	18.25 ± 1.92
	GRAIN	31.63 ± 0.95	33.24 ± 0.47	12.23 ± 0.24
	LearnAL	**32.12 ± 0.39**	**35.81 ± 0.82**	**21.71 ± 0.22**
GPRGNN	Random	28.37 ± 1.41	25.55 ± 1.35	13.93 ± 0.06
	AGE	25.67 ± 0.83	21.88 ± 1.15	7.83 ± 3.51
	GRAIN	26.61 ± 0.51	26.26± 0.57	7.20 ± 3.94
	LearnAL	**28.75 ± 0.66**	**28.45 ± 0.63**	**14.01 ± 0.03**

Table 4: Classification accuracy (%) on the Penn94 dataset with different training sets selected by graph active learning methods.

Method	H2GCN	GPRGNN	LINKX
Random	66.63 ± 0.67	67.29 ± 0.82	65.26 ± 1.06
AGE	66.83 ± 0.22	66.75 ± 0.35	65.49 ± 0.47
GRAIN	66.49 ± 0.09	67.45 ± 0.40	65.71 ± 0.24
LearnAL	66.91 ± 0.20	67.89 ± 0.36	67.93 ± 0.81

5.2 Results on Graph Tasks

We conducted experiments on active learning for semi-supervised node classification on homophilic datasets. From Table 1, we can observe that our proposed LearnAL outperforms other methods across different graph datasets. Specifically, GCN with the training set selected by LearnAL demonstrates improvements of approximately 2.9% and 2.1% over the model trained on the training set selected by GRAIN on Cora and Citeseer, respectively.

Table 3: Classification accuracy (%) on ogbn-arxiv dataset. OOM denotes out-of-memory.

Training Data	GCN
Random	63.35 ± 1.01
AGE	63.64 ± 0.78
GRAIN	OOM
LearnAL	64.48 ± 0.11

The effectiveness of LearnAL extends to other GNN models, including GAT, APPNP and GraphSAGE (in Appendix E). We further evaluate the influence of the labeling budget, and report the test accuracy of the GCN model versus the number of labeled nodes for training in Fig. 2. Compared with the other baselines, learnAL quickly boosts its accuracy at the beginning of the training and consistently outperforms the baselines as the number of labeled nodes increases. Specifically, to achieve an accuracy of approximately 70% on Citeseer, LearnAL requires labeling only 40 samples, whereas other methods need over 60 nodes. This result highlights the efficiency of LearnAL. Notably, the differences in the starting points observed in Fig. 2 arise from the use of method-specific initialization strategies. Our approach adopts a structure and feature-based initial pool selection method, which jointly considers both node attributes and graph topology, providing more informative and representative labeled samples at the early stage of training.

While the homophilic datasets are graphs with high **Homo.** (indicating the proportion of edges connecting nodes with the same label (Zhu et al., 2020)), we also consider heterophilic datasets with low **Homo.**. The prediction accuracies for node classification on three different heterophilic datasets are reported in Table 2. LearnAL demonstrates state-of-the-art or competitive performance across all heterophilic datasets and various GNN models, including H2GCN and GPRGNN, which are specifically designed for heterophilic settings. The baseline methods fail to achieve better performance compared to random sampling because they cannot explore and exploit more complex structural information, such as the long-range dependent information in heterophilic datasets. In contrast, LearnAL captures global-level graph structural information by directly learning the correlations between labeled and unlabeled data from a global perspective, which provides a significant advantage. We conduct statistical significance tests to assess the reliability of LearnAL's improvements over representative baselines in Appendix D. Results show that LearnAL achieves statistically significant gains on most datasets, with p-values below 0.05 in the majority of comparisons. On Pubmed, improvements are still positive but less statistically pronounced, likely due to its small number of classes and simpler label structure. Overall, the results demonstrate that LearnAL's improvements are consistent and statistically reliable across diverse datasets.

We further evaluate the contribution of the learned attention score within the same framework by conducting an ablation study. Specifically, we replace the learned cross-attention score in our selection module with a fixed cosine similarity computed on the same GNN node embeddings, while keeping all other components identical (including the GNN encoder, the coreset selection objective, and the training pipeline). We refer to this variant as LearnAL (cosine). As shown in Table 1, the learned attention mechanism consistently outperforms the fixed similarity measure on the same GNN representations. The improvements are consistent across datasets, demonstrating the effectiveness of the proposed attention module.

The results in Tables 3 and 4 demonstrate that our proposed method effectively extends to large-scale datasets, including ogbn-arxiv and Penn94.

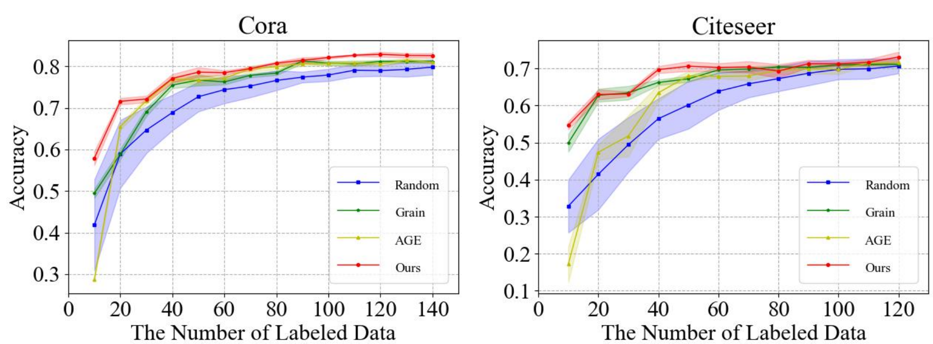

Figure 2: The test accuracy across different labeling budgets for GCN model training.

Training Efficiency. Table 5 reports the training time of different graph AL methods on Cora, Citeseer, and Pubmed. We observe that LearnAL is orders of magnitude faster than some graph AL methods. Specifically, LearnAL yields 3x training speedup over AGE on cora, and 2x training speedup over ACL on citeseer. In terms of memory usage, LearnAL shows memory consumption of 1034.45 MB and 1496.38 MB on the Cora and Citeseer datasets, respectively. Heuristic-based methods (e.g., diversity or density) use less memory but tend to incur higher time costs and less effective. LearnAL offers a better balance between memory use and speed, making it more scalable across datasets.

5.3 Results on Image Classification

To demonstrate the generalization ability of our proposed method, we report the performance comparison of LearnAL with six existing methods on CIFAR-10 and FashionMNIST datasets in Figure 3. Our proposed attention-based graph coreset labeling method can achieve the comparative or even better performance compared to some CNN baselines including Random sampling, CoreSet (Sener & Savarese, 2018), VAAL (Sinha et al., 2019), and CoreGCN (Caramalau et al., 2021). Especially, after selecting 4000 labeled examples, the LearnAL achieves highest performances with 82.23% and 89.97% on CIFAR-10 and FashionMNIST, respectively.

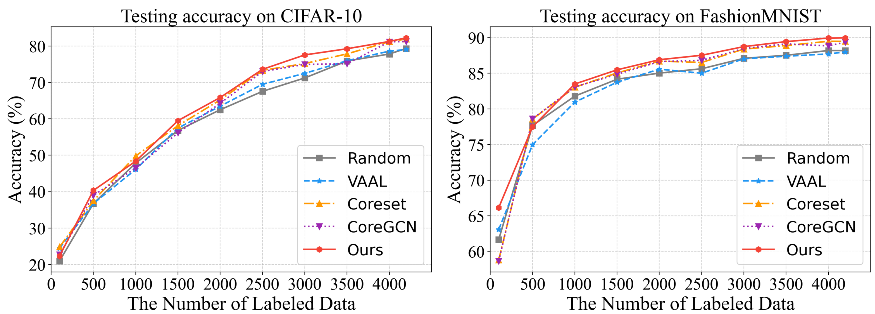

Figure 3: The comparison of several active learning methods on CIFAR-10 and FashionMNIST. The accuracy is averaged over 5 runs.

5.4 Ablation Study

Positional encoding plays a crucial role in the self-attention framework. In graph coreset selection, since the goal is to select a diverse and representative subset of nodes for labeling. Positional encoding injects structural information (e.g., based on topology or spectral signals), allowing the attention mechanism to jointly consider feature similarity and structural context when estimating node importance.

Table 5: Efficiency comparison of LearnAL and other graph AL competitors w.r.t. training time (s) on NVIDIA A100.

Method	Cora	Citeseer	Pubmed
AGE	57.45	71.68	978.14
ACL	38.23	62.36	176.08
GRAIN	21.55	37.81	172.73
LearnAL	20.15	27.92	145.83

Based on GCNs, we test its impact on the proposed LearnAL framework by comparing two common positional encoding methods: Laplacian-based (lpe) and random walk positional encoding (rwpe) against LearnAL without any positional encoding (w/o pe). As shown in Table 6, we observe that the GCN achieves superior performance when trained on data selected by LearnAL with positional encoding, compared to LearnAL without positional encoding. The difference in performance is minimal with either Laplacian-based or random walk positional encoding methods across all three datasets.

Table 6: Ablation study on positional encoding in LearnAL.

Methods	Training Data	Cora	Citeseer	pubmed
GCN	LearnAL w/o pe	82.59 ± 0.37	71.93 ± 0.64	79.53 ± 0.55
	LearnAL (lpe)	83.92 ± 0.54	73.16 ± 0.46	79.10 ± 0.90
	LearnAL (rwpe)	82.89 ± 0.38	73.10 ± 0.58	79.83 ± 0.34

5.5 Robustness Analysis of LearnAL with noisy data

We further demonstrate the robustness of the proposed method on noisy data. Specifically, we simulate noisy data by randomly removing a certain percentage (10%) of the graph node features, as reported in Table 7, and by additionally perturbing the graph structure through randomly adding or removing 10% of the edges, as reported in Table 8. As shown in Table 7 and 8, we observe that LearnAL continues to achieve strong performance, on different graph datasets, in the presence of noise. This demonstrates the method's robustness and its ability to perform well in settings that might better resemble real-world, noisy data scenarios.

Table 7: Classification accuracy (%) on three citation datasets with different training sets (noise and clear features).

Methods	**Training data**	**Cora**	**Citeseer**	**Pubmed**
GCN	noise	82.58 ± 0.40	72.04 ± 0.19	76.08 ± 0.46
	clear	83.92 ± 0.54	73.10 ± 0.58	79.83 ± 0.34

Table 8: The performance of LearnAL with perturbed structures on Cora, Citeseer, and Pubmed. The edge ratio of 90% means removing 10% edges while 110% means adding 10% edges randomly.

Method	**Edge Ratio**	**Cora**	**Citeseer**	**Pubmed**
	90%	78.40 ± 0.77	72.62 ± 0.48	78.04 ± 0.58
GCN	100%	83.92 ± 0.54	73.10 ± 0.58	79.83 ± 0.34
	110%	80.22 ± 0.96	70.12 ± 0.82	76.18 ± 1.04

6 Conclusion and Limitations

In conclusion, we proposed a learnable core-set labeling framework to address limitations in existing graph AL methods. By leveraging an attention mechanism to connect labeled and unlabeled data, LearnAL identifies the most informative samples, gradually expanding the labeled dataset to cover the graph representation space. This improves GNN performance across various graph data. Our theory shows that choosing samples with large representation differences reduces the bound radius δ, and experiments confirm LearnAL's effectiveness on both graph and image datasets. While scalable to large datasets via subgraph sampling, future work will explore more efficient methods. LearnAL also assumes few outliers—when present, it may select atypical samples. Future research will address anomaly handling and class-balanced selection.

References

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Christopher Berlind and Ruth Urner. Active nearest neighbors in changing environments. In *International conference on machine learning*, pp. 1870–1879. PMLR, 2015.

Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active learning for networked data. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 79–86, 2010.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Process. Mag.*, 34(4):18–42, 2017.

Hongyun Cai, Vincent Wenchen Zheng, and Kevin Chen-Chuan Chang. Active learning for graph embedding. *CoRR*, abs/1705.05085, 2017.

Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Sequential graph convolutional network for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9583–9592, 2021.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). *arXiv preprint arXiv:2310.04668*, 2023.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021.

Gabriella Contardo, Ludovic Denoyer, and Thierry Artières. A meta-learning approach to one-step active learning. *arXiv preprint arXiv:1706.08334*, 2017.

Limeng Cui, Xianfeng Tang, Sumeet Katariya, Nikhil Rao, Pallav Agrawal, Karthik Subbian, and Dongwon Lee. Allie: Active learning on large-scale imbalanced graphs. In *Proceedings of the ACM Web Conference 2022*, pp. 690–698, 2022.

Begüm Demir, Claudio Persello, and Lorenzo Bruzzone. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(3):1014–1031, 2010.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pp. 2224–2232, 2015.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.

Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. In *Advances in Neural Information Processing Systems*, 2020.

Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2-3):133, 1997.

Sichao Fu, Xueqi Ma, Yibing Zhan, Fanyu You, Qinmu Peng, Tongliang Liu, James Bailey, and Danilo Mandic. Finding core labels for maximizing generalization of graph neural networks. *Neural Networks*, pp. 106635, 2024.

Dominik Fuchsgruber, Tom Wollschläger, Bertrand Charpentier, Antonio Oroz, and Stephan Günnemann. Uncertainty for active learning on graphs. *arXiv preprint arXiv:2405.01462*, 2024.

Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. 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*, pp. 2142–2148, 2018a.

Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. Active discriminative network representation learning. In *IJCAI International Joint Conference on Artificial Intelligence*, 2018b.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272, 2017.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

Quanquan Gu, Charu Aggarwal, Jialu Liu, and Jiawei Han. Selective sampling on graphs for classification. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 131–139, 2013.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 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*, pp. 1024–1034, 2017.

Haoyu Han, Xiaorui Liu, Li Ma, MohamadAli Torkamani, Hui Liu, Jiliang Tang, and Makoto Yamada. Structural fairness-aware active learning for graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2023.

Harris Hardiman-Mostow, Jack Mauro, Adrien Weihs, and Andrea L Bertozzi. Topology-aware active learning on graphs. *arXiv preprint arXiv:2510.25892*, 2025.

Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pp. 409–426, 1994.

Shengding Hu, Zheng Xiong, Meng Qu, Xingdi Yuan, Marc-Alexandre Côté, Zhiyuan Liu, and Jian Tang. Graph policy network for transferable active learning on graphs. *Advances in Neural Information Processing Systems*, 33:10174–10185, 2020a.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv:2005.00687*, 2020b.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021b.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ankit Kuwadekar and Jennifer Neville. Relational active learning for joint collective classification models. In *Proceedings of the 28th international conference on machine learning (icml-11)*, pp. 385–392, 2011.

Quan Li, Tianxiang Zhao, Lingwei Chen, Junjie Xu, and Suhang Wang. Enhancing graph neural networks with limited labeled data by actively distilling knowledge from large language models. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 741–746. IEEE, 2024.

Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 859–866, 2013.

Yayong Li, Jie Yin, and Ling Chen. Seal: Semisupervised adversarial active learning on attributed graphs. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3136–3147, 2020.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021.

Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems*, 34:1048–1061, 2021.

Jiaqi Ma, Ziqiao Ma, Joyce Chai, and Qiaozhu Mei. Partition-based active learning for graph neural networks. *Transactions on Machine Learning Research*, 2022.

David JC MacKay. Information-based objective functions for active data selection. *Neural computation*, 4(4): 590–604, 1992.

Kaushalya Madhawa and Tsuyoshi Murata. Active learning for node classification: An evaluation. *Entropy*, 22(10):1164, 2020.

Andrew McCallum, Kamal Nigam, et al. Employing em and pool-based active learning for text classification. In *ICML*, volume 98, pp. 350–358. Citeseer, 1998.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *ICLR*, 2023.

Peter Samoaa, Linus Aronsson, Philipp Leitner, and Morteza Haghir Chehreghani. Batch mode deep active learning for regression on graph data. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 5904–5913. IEEE, 2023.

Peter Samoaa, Linus Aronsson, Antonio Longa, Philipp Leitner, and Morteza Haghir Chehreghani. A unified active learning framework for annotating graph data for regression tasks. *Engineering Applications of Artificial Intelligence*, 138:109383, 2024.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations*, 2018.

Burr Settles. Active learning literature survey. 2009.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 1070–1079, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *CoRR*, abs/1811.05868, 2018.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5972–5981, 2019.

Zixing Song, Yifei Zhang, and Irwin King. No change, no gain: empowering graph neural networks with expected model change maximization for active learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Junwei Su, Jiaqi Han, and Chuan Wu. On locality in graph learning via graph neural network.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations*, 2018.

Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12):2591–2600, 2016.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pp. 1954–1963. PMLR, 2015.

Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Active learning for graph neural networks via node feature propagation. *CoRR*, abs/1910.07567, 2019. URL `http://arxiv.org/abs/1910.07567`.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 93–102, 2019.

Chengcheng Yu, Jiapeng Zhu, and Xiang Li. Class-balanced and reinforced active learning on graphs. *arXiv preprint arXiv:2402.10074*, 2024.

Jifan Zhang, Julian Katz-Samuels, and Robert Nowak. Galaxy: Graph-based active learning at the extreme. In *International Conference on Machine Learning*, pp. 26223–26238. PMLR, 2022a.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

Wentao Zhang, Yu Shen, Yang Li, Lei Chen, Zhi Yang, and Bin Cui. 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*, pp. 2366–2374, 2021a.

Wentao Zhang, Yu Shen, Yang Li, Lei Chen, Zhi Yang, and Bin Cui. Alg: Fast and accurate active learning framework for graph convolutional networks. In *Proceedings of the 2021 international conference on management of data*, pp. 2366–2374, 2021b.

Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. RIM: reliable influence-based active learning on graphs. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 27978–27990, 2021c.

Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. Rim: Reliable influence-based active learning on graphs. *Advances in Neural Information Processing Systems*, 34:27978–27990, 2021d.

Wentao Zhang, Zhi Yang, Yexin Wang, Yu Shen, Yang Li, Liang Wang, and Bin Cui. Grain: Improving data efficiency of graph neural networks via diversified influence maximization. *Proc. VLDB Endow.*, 14(11): 2473–2482, 2021e.

Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. Information gain propagation: a new way to graph active learning with soft labels. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022b.

Wentao Zhang, Yexin Wang, Zhenbang You, Yang Li, Gang Cao, Zhi Yang, and Bin Cui. Nc-alg: Graph-based active learning under noisy crowd. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 2681–2694. IEEE, 2024.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust gnns: Overcoming the limitations of localized graph training data. *Advances in Neural Information Processing Systems*, 34:27965–27977, 2021.

A Proof

We introduce the following assumptions:

Assumption 1 (local curvature)**.** *For a representation $\boldsymbol{h}$, both $\frac{d}{dh}l(f(h))$ and $\frac{d^2}{d^2h}l(f(h))$ exist and and are continuous and bounded.*

Assumption 2 (well trained model)**.** For a given training set $\mathcal{V}_l$, and a well-trained GNN model $\mathcal{M}$, for any $\epsilon > 0$ and $v \in \mathcal{V}_l$, we have $l\left(f_{\mathcal{M}}\left(\boldsymbol{h}_v\right)\right) < \epsilon$.

Assumption 1 regarding local curvature is a standard technical assumption to make the analysis feasible. Assumption 2 has been formally proved Keriven & Peyré (2019), demonstrating that GNNs can achieve universal approximation power. Here, $f(\cdot)$ is the prediction function and $l(\cdot)$ is the loss function. Under mild conditions and with enough parameters in the model, a model with universal approximation power can achieve zero loss on the training set upon convergence, i.e., the property in Assumption 2 Su et al., with high probability.

Proof of Proposition 4.1

Proof. Let θ_D be the set of parameters learnt by the GNN model $\mathcal{M}$ that satisfy properties given in Assumption 2. For a labeled sample v with representation $\boldsymbol{h}_v$, we have $l\left(f_{\theta_D}\left(\boldsymbol{h}_v\right)\right) < \epsilon$ with $\epsilon > 0$, i.e., it achieves the global minimum of the loss function in the embedding space, where $\boldsymbol{h}_v = \mathcal{M}(v)$, $f(\cdot)$ is the prediction function, and $l(\cdot)$ is the loss function. Here, we assume that both $f(\cdot)$ and $l(\cdot)$ are smooth.

According to Assumption 1, we have $\frac{d}{dh}l\left(f\left(\boldsymbol{h}_v\right)\right) = 0$ since it achieves a local minimum. Furthermore, the global minimum also states that $\frac{d^2}{d^2h}l\left(f\left(\boldsymbol{h}_v\right)\right) \geq 0$.

Assume there are two embeddings $\boldsymbol{h}$ and $\boldsymbol{h}'$, where $\Delta\left(\boldsymbol{h}, \boldsymbol{h}_v\right) \leq \Delta\left(\boldsymbol{h}', \boldsymbol{h}_v\right) \leq r_v$, with Δ measuring the distance in the embedding space and r_v defining the width of the neighborhood around h_v in the embedding space. This indicates that $\boldsymbol{h}$ is closer to $\boldsymbol{h}_v$ than $\boldsymbol{h}'$, we have $\boldsymbol{h}' = \boldsymbol{h} + \delta$, $\delta > 0$. Thus we can get:

$$l\left(f\left(\boldsymbol{h}'\right)\right) = l(f(\boldsymbol{h} + \delta)) \approx l(f(\boldsymbol{h})) + \frac{d}{dh}l(f(\boldsymbol{h}))\delta + \frac{1}{2}\frac{d^2}{d^2h}l\left(f\left(\boldsymbol{h}\right)\right)\|\delta\|^2 + \mathcal{O}\left(\|\delta\|^3\right) \tag{12}$$

As $\frac{d}{dh}l\left(f\left(\boldsymbol{h}\right)\right) \geq 0$ and $\delta > 0$, we have $l(f(\boldsymbol{h})) < l(f(\boldsymbol{h}'))$. □

Based on Proposition 4.1, we provide the proof for Lemma 4.1.

Proof of Lemma 4.1 Assume there are two training set $\mathcal{V}_{train}$ and $\mathcal{V}'_{train}$, and test set $\mathcal{V}_{test}$. Based on two training set, we get two trained GNN models $\mathcal{M}(\mathcal{V}_{train})$ and $\mathcal{M}(\mathcal{V}'_{train})$. If $\sum_{v \in \mathcal{V}_{train}} \Delta(v, \mathcal{V}_{test}) < \sum_{u \in \mathcal{V}'_{train}} \Delta(u, \mathcal{V}_{test})$, thus the covering radius $\delta_{v \in \mathcal{V}_{train}} < \delta_{u \in \mathcal{V}'_{train}}$, we have $\sum_{u \in \mathcal{V}_{test}} l(f_{\mathcal{M}(\mathcal{V}_{train})}(u)) < \sum_{u \in \mathcal{V}_{test}} l(f_{\mathcal{M}(\mathcal{V}'_{train})}(u))$.

Proof. Assume there are two training set $\mathcal{V}_{train}$ and $\mathcal{V}'_{train}$, and test set $\mathcal{V}_{test}$. Let $G = (V, E)$ be the input graph with node feature vector X_v for all $v \in V$. Let $\mathcal{M}$ be a given GNN model and f be the prediction function that maps the output of $\mathcal{M}$ to the class representation. The loss function l is λ^l Lipschitz continuous for all y bounded by L. According to Proposition 4.1, we can obtain near-zero error for labeled data and have

$$
\begin{aligned}
&\sum_{v \in \mathcal{V}_{train}} l\left(f_{\mathcal{M}(\mathcal{V}_{train})}(v)\right) \approx 0, \\
&\sum_{v \in \mathcal{V}'_{train}} l\left(f_{\mathcal{M}(\mathcal{V}'_{train})}(v)\right) \approx 0.
\end{aligned}
$$

Then, we consider the loss on the test set $\mathcal{V}_{test}$ with the model trained on two training sets, which can be written as:

$$
\begin{aligned}
&\sum_{u \in \mathcal{V}_{test}} \mathcal{L}\left(f_{\mathcal{M}(\mathcal{V}_{train})}(u)\right), \\
&\sum_{u \in \mathcal{V}_{test}} \mathcal{L}\left(f_{\mathcal{M}(\mathcal{V}'_{train})}(u)\right).
\end{aligned}
$$

From Proposition 4.1, we known that the loss function is monotonically increasing with respect to the embedding distance in $\delta_{\boldsymbol{h}_v}$, where $\boldsymbol{h}_v$ is the hidden representation based on trained model $\mathcal{M}$.

According to Theorem 1 in Sener & Savarese (2018), we know that the loss function is bounded by convering radius δ. Now we extend the similar conclusion to GNN: We have a condition which states that there exists h_j in δ ball around h_i such that h_j has near-zero loss.

$$
\begin{aligned}
E_{y_i \sim \eta(h_i)}\left[l_{\mathcal{M}}\left(\mathcal{G}, y_i; A_{\mathbf{s}}\right)\right] &= \sum_{k \in [C]} p_{y_i \sim \eta_k(h_i)}(y_i = k) l_{\mathcal{M}}\left(\mathcal{G}, k; A_{\mathbf{s}}\right) \\
&\overset{(d)}{\leq} \sum_{k \in [C]} p_{y_i \sim \eta_k(h_j)}(y_i = k) l_{\mathcal{M}}\left(\mathcal{G}, k; A_{\mathbf{s}}\right) \\
&+ sum_{k \in [C]} \left|\eta_k\left(h_i\right) - \eta_k\left(h_j\right)\right| l_{\mathcal{M}}\left(\mathcal{G}, k; A_{\mathbf{s}}\right) \\
&\overset{(e)}{\leq} \sum_{k \in [C]} p_{y_i \sim \eta_k(h_j)}(y_i = k) l_{\mathcal{M}}\left(\mathcal{G}, k; A_{\mathbf{s}}\right) + \delta \lambda^{\eta} L C
\end{aligned}
\tag{13}
$$

We use the Claim in Berlind & Urner (2015), i.e., fix $p, p' \in [0,1]$ and $y' \in [0,1]$, then, $p_{y \sim p}\left(y \neq y'\right) \leq p_{y \sim p'}\left(y \neq y'\right) + |p - p'|$ to achieve (d), and use Lipschitz property of regression function and bound of loss in (e). Then, we further bound

$$
\begin{aligned}
\sum_{k \in [C]} p_{y_i \sim \eta_k(h_j)}\left(y_i = k\right) l_{\mathcal{M}}\left(\mathcal{G}, k; A_{\mathbf{s}}\right) = &\sum_{k \in [C]} p_{y_i \sim \eta_k(h_j)}\left(y_i = k\right)\left[l\left(h_i, k; A_{\mathbf{s}}\right) - l\left(h_j, k; A_{\mathbf{s}}\right)\right] \\
&+ \sum_{k \in [C]} p_{y_i \sim \eta_k(h_j)}\left(y_i = k\right) l\left(h_j, k; A_{\mathbf{s}}\right) \\
\leq &\delta \lambda^l
\end{aligned}
\tag{14}
$$

where last step is coming from the fact that the trained classifier assumed to have 0 loss over training data. Here, $l_{\mathcal{M}}\left(\mathcal{G}, y_i; A_{\mathbf{s}}\right) = l\left(h_i, y_i; A_{\mathbf{s}}\right)$, as h_i is the low-dimensional embedding of x_i by GNN $\mathcal{M}$. Then, we can get

$$
E_{y_i \sim \eta(h_i)}\left[l_{\mathcal{M}}\left(\mathcal{G}, k; A_{\mathbf{s}}\right)\right] \leq \delta\left(\lambda^l + \lambda^{\eta} L C\right). \tag{15}
$$

We further use Hoeffding's inequality Hoeffding (1994) and finally obtain

$$
\left|\frac{1}{N} \sum_{i \in [N]} l_{\mathcal{M}}\left(\mathcal{G}, y_i; A_{\mathbf{s}}\right) - \frac{1}{|\mathbf{s}|} \sum_{j \in \mathbf{s}} l_{\mathcal{M}}\left(\mathcal{G}, y_j; A_{\mathbf{s}}\right)\right| \leq \delta\left(\lambda^l + \lambda^{\eta} L C\right) + L \sqrt{\frac{\log(1/\gamma)}{2N}} \tag{16}
$$

with probability at least $1 - \gamma$.

Thus, while $\sum_{v \in \mathcal{V}_{train}} \Delta(v, \mathcal{V}_{test}) < \sum_{u \in \mathcal{V}'_{train}} \Delta(u, \mathcal{V}_{test})$, we have he covering radius $\delta_{v \in \mathcal{V}_{train}} < \delta_{u \in \mathcal{V}'_{train}}$. The smaller covering radius means the smaller loss for the whole samples, thus, we have $\sum_{u \in \mathcal{V}_{test}} l(f_{\mathcal{M}(\mathcal{V}_{train})}(u)) < \sum_{u \in \mathcal{V}_{test}} l(f_{\mathcal{M}(\mathcal{V}'_{train})}(u))$.

□

Proof of Theorem 4.1

Proof. Given the whole sample $\mathcal{V}$ drawn from $\mathcal{G}$, let $\mathcal{V}_l$ represent the labeled pool consisting of points with labels, and let $\mathcal{V}_u$ denote the set of unlabeled data. $\exists s \in \mathcal{V}_u : \forall v \in \mathcal{V}_l, \Delta(s, v) < \Delta(k, v)$ with $k \in \mathcal{V}_u \setminus \{s\}$, thus we have $\sum_{v \in \mathcal{V}_l \cup \{s\}} \Delta(v, \mathcal{V}) < \sum_{u \in \mathcal{V}_l \cup \{k\}} \Delta(u, \mathcal{V})$ where $\mathcal{V}$ denote the whole graph data. In other word, we have $\sum_{v \in \mathcal{V}_l \cup \{s\}} \Delta(v, \mathcal{V}_{test}) < \sum_{u \in \mathcal{V}_l \cup \{k\}} \Delta(u, \mathcal{V}_{test})$ for test set $\mathcal{V}_{test}$. Acoording to Lemma 4.1, we can get that $\delta_{v \in \mathcal{V}_l \cup \{s\}} < \delta_{v \in \mathcal{V}_l \cup \{k\}} < \delta_{u \in \mathcal{V}_l}$. Thus, we have $\sum_{i \in \mathcal{V}} l(f_{\mathcal{M}(\mathcal{V}_l \cup \{s\})}(i)) < \sum_{i \in \mathcal{V}} l(f_{\mathcal{M}(\mathcal{V}_l \cup \{k\}}(i)) < \sum_{i \in \mathcal{V}} l(f_{\mathcal{M}(\mathcal{V}_l}(i))$. □

B Experimental part

Table 9: Implementation Details

Model	**Dataset**	**Epochs**	**Learning Rate**	**Weight Decay**	**Hidden Units**
GCN, GAT, APPNP, GraphSAGE	Cora, Citeseer, Pubmed	200	1e-2	5e-4	64
	ogbn-arxiv	300	1e-2	0	64

Implementation details for image classification. ResNet-18 [15] is the favourite choice as learner due to its relatively higher accuracy and better training stability. During training the learner, we set a batch size of 64. We use Stochastic Gradient Descent (SGD) with a weight decay $5e - 4$ and a momentum of 0.9. At every selection stage, we train the model for 200 epochs. We set the initial learning rate of 0.1 and decrease it by the factor of 10 after 160 epochs. We use the same set of hyper-parameters in all the experiments. For quantitative evaluation, we report the mean average accuracy of 5 trials on the test sets.

C Dataset Statistic.

Table 10: Statistics of graph benchmark datasets.

	Cora	Citeseer	Pubmed	ogbn-arxiv	Actor	roman-empire	Squirrel	Penn94
# Nodes	2,708	3,327	19,717	169,343	7,600	22,662	24,492	41,554
# Edges	5,429	4,732	44,338	1,166,343	26,752	32,927	93,050	1,362,229
Class	7	6	3	40	5	18	5	2

D Statistical analysis.

We conduct statistical significance tests to evaluate whether the improvements of LearnAL over representative baselines are statistically reliable. Tables 11 and 12 report the p-values of pairwise comparisons between LearnAL and other baselines across both homogeneous and heterogeneous graph datasets.

Across different datasets, most p-values are significantly smaller than 0.05, and many are even close to zero, indicating that the performance gains of LearnAL are statistically significant. In particular, LearnAL consistently achieves highly significant improvements on Cora, Citeseer, and Roman-empire, as well as on Actor and Squirrel datasets.

These results confirm that the improvements of LearnAL are not due to random variation, but are robust and statistically reliable across different graph settings.

Table 13: The performance of APPNP on three citation datasets with different training sets.

Methods	Training Data	Cora	Citeseer	pubmed
APPNP	Random	82.15 ± 0.85	72.03 ± 1.07	77.84 ± 4.18
	Standard	82.86 ± 0.28	71.07 ± 0.76	80.12 ± 0.32
	AGE	83.68 ± 0.26	71.43 ± 0.48	80.42 ± 1.18
	FeatProp	78.1 ± 1.56	66.3 ± 1.91	75.2 ± 1.32
	GRAIN	82.27 ± 0.74	71.35 ± 0.20	80.55 ± 0.36
	RIM	83.18 ± 0.34	74.22 ± 0.37	76.29 ± 0.42
	ALG	84.59 ± 0.19	72.17 ± 0.10	80.05 ± 0.09
	GraphPart	82.86 ± 0.28	71.21 ± 0.89	80.12 ± 0.32
	NC-ALG	84.66 ± 0.40	71.73 ± 0.59	80.25 ± 0.30
	AGCL	**84.93 ± 0.42**	**73.53 ± 0.42**	**80.91 ± 0.34**

Table 14: The performance of GraphSAGE on three citation datasets with different training sets.

Methods	Training data	Cora	Citeseer	Pubmed
GraphSAGE	GRAIN	81.55 ± 0.50	71.06 ± 0.44	**79.23 ± 0.30**
	GraphPart	81.27 ± 0.43	70.42 ± 0.58	77.29 ± 0.35
	LearnAL	**82.56 ± 0.39**	**72.64 ± 1.11**	79.14 ± 0.33

Table 11: Statistical significance comparison (p-values) between LearnAL and representative baselines on Cora, Citeseer, and Pubmed based on GCN.

Method	Cora	Citeseer	Pubmed
LearnAL/Grain	0.000001	0.000000	0.442680
LearnAL/Random	0.000026	0.003183	0.030452
LearnAL/ALG	0.011246	0.000815	0.301010
LearnAL/GraphPart	0.000124	0.000039	0.222167

Table 12: Statistical significance comparison (p-values) between LearnAL and representative baselines on heterogeneous graphs based on GCN.

Method	Actor	Squirrel	Roman-empire
LearnAL/Grain	0.016123	0.008812	0.000000
LearnAL/Random	0.016295	0.076061	0.001587
LearnAL/AGE	0.020687	0.000105	0.001470

E The results on APPNP and GraphSAGE.

We have conducted additional experiments to evaluate LearnAL with APPNP Klicpera et al. (2019) and GraphSAGE Hamilton et al. (2017). The results, shown in the Table 13 and 14, indicate that LearnAL performs consistently well across different datasets. Specifically, LearnAL outperforms other methods when using other model including APPNP and GraphSAGE, further demonstrating its versatility and effectiveness in core data selection, irrespective of the underlying GNN model.