Few-Shot Graph Out-of-Distribution Detection with LLMs

Haoyan Xu[1*], Zhengtao Yao[1*], Yushun Dong[2], Ziyi Wang[3], Ryan Rossi[4], Mengyuan Li[1], and Yue Zhao[1](✉)

[1] University of Southern California, Los Angeles CA 90007, USA
{haoyanxu,zyao9248,mengyuanli,yzhao010}@usc.edu
[2] Florida State University, 600 W College Ave, Tallahassee, FL 32306, USA
yushun.dong@fsu.edu
[3] University of Maryland, College Park, 1000 Hilltop Cir, College Park, MD 20742, USA zoewang@umd.edu
[4] Adobe Research, 345 Park Ave, San Jose, CA 95110, USA ryrossi@adobe.com

Abstract. Graph out-of-distribution (OOD) detection usually relies on training a graph neural network (GNN) with a large set of labeled in-distribution (ID) nodes. However, acquiring high-quality labeled nodes in text-attributed graphs (TAGs) is challenging and costly due to their complex textual and structural characteristics. Large language models (LLMs) offer strong zero-shot language capabilities but overlook graph connectivity, limiting their utility for graph OOD detection.

In this work, we propose LLM-GOOD, a general framework that effectively combines the strengths of LLMs and GNNs to enhance data efficiency in graph OOD detection. Specifically, we first leverage LLMs' strong zero-shot capabilities to filter out likely OOD nodes, significantly reducing the human annotation burden. To minimize the usage and cost of the LLM, we employ it only to annotate a small subset of unlabeled nodes. We then train a lightweight GNN filter using these noisy labels, enabling efficient predictions of ID status for all other unlabeled nodes by leveraging both textual and structural information. After obtaining node embeddings from the GNN filter, we can apply informativeness-based methods to select the most valuable nodes for precise human annotation. Finally, we train the target ID classifier using these accurately annotated ID nodes. Extensive experiments on four real-world TAG datasets demonstrate that LLM-GOOD significantly reduces human annotation costs and outperforms state-of-the-art baselines in terms of both ID classification accuracy and OOD detection performance.

Keywords: Graph OOD Detection · Large Language Models · Data-Efficient Learning · Text-Attributed Graphs · Graph Neural Networks · Few-Shot Learning · Zero-Shot Annotation

[*] Equal contribution.

1 Introduction

Out-of-distribution (OOD) detection [16,10,7,14] has emerged as a critical task in machine learning, particularly for safety-critical applications where models must reliably identify inputs that differ significantly from the training data [18,17]. Recently, several OOD detection methods [27,31,39,38] and open-set learning approaches [37] have been proposed and applied to graph-structured data. Existing graph OOD detection methods typically operate within *a semi-supervised, transductive framework*, where the entire set of nodes is accessible during training, but only a portion of the class labels (in-distribution (ID) classes) are provided [27]. These methods generally rely on *a sufficient number* of labeled ID nodes to train a GNN-based ID classifier, from which they derive the ID classification logits for all nodes. Post-hoc OOD detectors [31,10,20] are then applied to these logits for OOD detection. In particular, nodes with higher energy scores [31] or higher entropy scores are identified as OOD nodes.

While these graph OOD detection methods are effective, they invariably rely on the assumption that ground truth ID labels are readily available. This assumption often overlooks a critical challenge: obtaining sufficient high-quality labels for graph-structured data. Specifically, (1) the diverse and complex nature of graph-structured data makes human labeling inherently difficult, and (2) the large scale of real-world graphs renders annotating a significant portion of nodes both time-consuming and resource-intensive [4,34].

Our Observations and Motivation. In this paper, we aim to address the challenge of few-shot OOD detection and ID classification on text-attributed graphs (TAGs) within the commonly used semi-supervised transductive setting, as described above. Consider a text-attributed social network where nodes represent individuals, node attributes correspond to their textual descriptions, and edges denote interactions or connections between them. Initially, the entire network is unlabeled, and the goal is to classify individuals into specific interest groups, such as technology enthusiasts, sports fans, or musicians, while operating within a limited human annotation budget. However, the network also contains individuals whose interests fall outside these predefined categories, such as those primarily engaged in political discussions or travel blogging. Identifying and labeling these OOD nodes would be inefficient, as they do not contribute to training an effective classifier for the targeted interest groups. Instead, the focus is on accurately classifying only the ID nodes while detecting and filtering out OOD nodes that do not belong to the intended classification space. Furthermore, zero-shot [30,5] and few-shot [23,1] OOD detection for images have been extensively studied using multi-modal foundation models. However, to date, no similarly powerful graph foundation model exists to support zero-shot or few-shot graph OOD detection. As a result, we turn to LLMs to tackle the data-efficiency challenge of OOD detection on TAGs.

In summary, current graph OOD detection methods typically rely heavily on large amounts of labeled ID nodes to perform well. Conversely, while LLMs demonstrate remarkable zero-shot capabilities on text-attributed graphs (TAGs), they inherently lack the ability to interpret and leverage the structural informa-

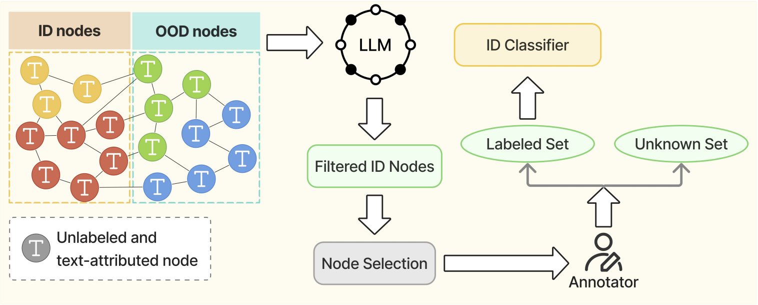

Fig. 1. An illustration of our method. To reduce annotation costs, we use an LLM to filter out OOD nodes before selecting nodes for human annotation. The annotated ID nodes are then used to train the target ID classifier.

tion essential to TAGs. In this study, we take the first step toward integrating the strengths of both GNNs and LLMs to tackle the data-efficiency challenges in graph OOD detection.

Present work. As shown in Fig. 1, to address these challenges, we propose to leverage LLMs to filter out OOD nodes before human annotation, thereby **reducing human costs**. Specifically, we provide the LLM with ID knowledge (i.e., the names of ID classes) and prompt it to determine whether an unlabeled query node belongs to one of the ID classes, using the text information associated with the query node. Note that while we enable LLMs to directly perform zero-shot OOD detection and ID classification, using LLMs for zero-shot annotation is very slow during inference. Therefore, we aim to leverage LLMs to reduce human annotation costs during training and rely solely on well-trained GNNs for faster inference during testing. However, prompting the LLM to annotate all unlabeled nodes in the training set is costly for large graphs, although the cost of using an LLM is significantly lower than that of human annotation. To further **reduce the LLM's cost**, we propose prompting the LLM to annotate only a small subset of nodes and then using these pseudo-labels to train a lightweight GNN filter. With this GNN filter, we can predict whether every unlabeled node in the training set belongs to one of the ID classes. If not, it's very likely that this node is an OOD node, and we then filter it out before human annotation.

In addition, we can obtain the embeddings of all unlabeled nodes in the graph after training the GNN filter with pseudo-labels. Based on these embeddings, the most informative nodes can be selected using existing informativeness-aware node selection methods. These selected nodes are then annotated by a human annotator, and the final annotated ID nodes are used to train the target ID classifier. Optionally, we can combine the accurate labels from human annotation with the noisy labels from the LLM to train a robust ID classifier under **severe data scarcity scenarios**. Compared to other active learning methods that require multiple rounds of selection [32,3], our approach requires only a single round of annotation. Moreover, relying solely on noisy labels from an LLM to train an ID classifier imposes a performance upper bound (see the results in Section 5.3). Furthermore, leveraging LLM knowledge to train smaller models, such as GNNs, facilitates faster inference, particularly in domains where time efficiency is crucial.

We summarize our key contributions as follows:

- To the best of our knowledge, we are the first to investigate LLM's zero-shot learning ability for the graph OOD detection problem. With the zero-shot learning ability of LLMs, our method achieves high performance with only one round of node selection, compared to traditional multi-round active learning selection methods.
- We design a general framework LLM-GOOD that can filter out many OOD nodes before annotation to reduce human costs and use LLM's zero-shot annotations to train a light GNN filter to further reduce LLM costs.
- We apply LLM-GOOD to node classification datasets consisting of different properties under label budget constraints. Experimental results show that our method effectively filters out OOD nodes and achieves much better ID classification and OOD detection performance compared to baselines within an annotation budget. Our code is available at: `https://github.com/zhengtaoyao/LLM_GOOD`.

2 Related Work

2.1 Graph OOD Detection

In recent years, OOD detection in graph data has presented new challenges, especially in the context of multi-class classification for in-distribution data, which further complicates the task of identifying outlier data [20]. For instance, OODGAT [27] leverages a graph neural network (GNN) that explicitly models interactions among different types of nodes, enabling effective separation of inliers and outliers during feature propagation. GNNSafe [31] highlights the inherent OOD detection capabilities of standard GNN classifiers and proposes a robust OOD discriminator using an energy-based function derived from GNNs trained with standard classification loss. GRASP [20] explores the potential of OOD score propagation and derives the conditions under which the score propagation is beneficial. They also propose an edge augmentation strategy with theoretical guarantees for post-hoc node-level OOD detection.

While effective, these methods rely heavily on the assumption of abundant ID labels in open-set scenarios. However, in real-world applications, labeled data are costly and challenging to obtain, limiting the practicality of such approaches.

2.2 Data-Efficient Graph Learning

Graphs have a wide range of applications across various domains [33,36,35,29], and researchers have conducted extensive and focused studies on graph machine learning in low-resource settings, aiming to reduce the cost and time required for annotation [12]. Current data-efficient graph learning methods can be broadly divided into three categories: self-supervised graph learning, semi-supervised graph learning, and few-shot graph learning.

Few-shot graph learning aims at enabling models to generalize effectively and make accurate predictions using only a small number of labeled examples. The primary objective is to train models to learn from a limited set of annotated instances and apply this knowledge to predict new and unseen data [12]. To achieve this, researchers typically adopt one of two approaches: metric learning, which encourages query nodes to align closely with their respective prototypes [28], or parameter optimization, which employs meta-learning to generate node representations [11]. Some graph active learning methods [32,2,3] have been developed to enhance the performance of semi-supervised node classification while adhering to a label budget constraint. For instance, FeatProp [32] identifies nodes by propagating their features throughout the graph structure and applying K-Medoids clustering, mitigating the impact of under-trained model representations. However, both current few-shot graph learning methods [6,41] and graph active learning techniques are restricted to the closed-set node classification scenario. Recently, [37] applied active learning methods to the graph open-set classification scenario. However, their approach involves using real OOD nodes and requires multiple rounds of node selection for human annotation.

2.3 LLMs as Prefix for Graphs

In this paper, we focus on utilizing information generated by LLMs to enhance the training of GNNs. These techniques can be divided into two main categories: (i) Embeddings from LLMs for GNNs, which involves incorporating embeddings produced by LLMs into GNNs, and (ii) Labels from LLMs for GNNs, which focuses on leveraging labels generated by LLMs to guide GNN training [25]. We mainly focus on the second category that leverages generated labels from LLMs as supervision to improve the training of GNNs.

LLM-GNN [4] utilizes LLMs as annotators to produce node category predictions accompanied by confidence scores, which are treated as labels. A post-filtering process is applied to remove low-quality annotations while ensuring label diversity. These refined labels are then used to train GNNs. Similarly, GraphEdit [9] uses LLMs to create an edge predictor, which evaluates and refines candidate edges by comparing them to the edges of the original graph.

3 Setting

3.1 Text-Attributed Graphs

Our study focuses on TAGs, represented as $G_T = (\mathcal{V}, \mathbf{A}, \mathbf{T}, \mathbf{X})$. The set of nodes is $\mathcal{V} = \{v_1, \ldots, v_n\}$, where each node is associated with raw text attributes $\mathbf{T} = \{t_1, t_2, \ldots, t_n\}$. These text attributes can be converted into sentence embeddings $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ using SentenceBERT [24]. The adjacency matrix

$\mathbf{A} \in \{0,1\}^{n \times n}$ encodes graph connectivity, where $\mathbf{A}[i,j] = 1$ indicates an edge between nodes i and j.

3.2 Graph OOD Detection

The node set can be partitioned as $\mathcal{V} = \mathcal{V}_{\text{in}} \cup \mathcal{V}_{\text{out}}$, where $\mathcal{V}_{\text{in}}$ denotes the set of ID nodes, and $\mathcal{V}_{\text{out}}$ represents the set of OOD nodes. We assume that ID nodes are drawn from the distribution $P_{\mathcal{V}}^{\text{in}}$, while OOD nodes are sampled from the distribution $P_{\mathcal{V}}^{\text{out}}$. The OOD node detection task is formally defined as follows: Given a collection of nodes sampled from $P_{\mathcal{V}}^{\text{in}}$ and $P_{\mathcal{V}}^{\text{out}}$, the objective is to accurately determine the source distribution—either $P_{\mathcal{V}}^{\text{in}}$ or $P_{\mathcal{V}}^{\text{out}}$—for each node.

We study OOD node detection in graphs under the transductive learning paradigm, where ID and OOD nodes coexist in the same graph, the most common framework for node-level OOD detection. During training, only the node attributes $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, and the ID labels of a subset of nodes, $\mathcal{V}' \subseteq \mathcal{V}_{\text{in}}$, are provided. In general, the task consists of two main objectives: (1) **OOD Detection**: For each node $v \in \mathcal{V}$, determine whether it belongs to one of the ID known classes or to an OOD unknown class. (2) **ID Classification**: For nodes identified as ID, assign them to one of the predefined K classes.

3.3 Few-Shot Graph OOD Detection

Assume that we have a validation set $\mathcal{V}_{val}$ and a test set $\mathcal{V}_{test}$. The remaining nodes form the candidate set $\mathcal{V}_{can} = \mathcal{V} \setminus (\mathcal{V}_{val} \cup \mathcal{V}_{test})$. All nodes in $\mathcal{V}_{can}$ are initially **unlabeled**. Given a human label budget $\mathcal{B}$, our goal is to select a subset of nodes from $\mathcal{V}_{can}$ such that the trained model f achieves the lowest expected loss in the test set $\mathcal{V}_{test}$:

$$\arg \min_{\mathcal{V}_{can}^{s} \subset \mathcal{V}_{can}, |\mathcal{V}_{can}^{s}| = \mathcal{B}} \mathbb{E}_{v_i \in \mathcal{V}_{test}} \left[\ell(y_i, \tilde{y}_i) \right] \tag{1}$$

where f is our target ID classifier, y_i is the ground truth label of node v_i, and $\tilde{y}_i$ denotes the label prediction of node v_i by f. Compared with other label-efficient graph learning methods, such as active learning approaches, **we do not require any initial set of labeled nodes and, more importantly, we only select nodes for one round of annotation**.

4 Method

In real-world scenarios, graphs typically include a large number of unlabeled nodes, many of which may be OOD nodes and irrelevant to the target task. Our goal is to train an ID classifier using a limited set of ID labels, striving for high accuracy in ID classification while effectively identifying OOD data, where the classifier should exhibit low confidence.

To reduce human efforts, we seek to exclude as many OOD nodes as possible from the training set prior to labeling. To achieve this, the first step is to use

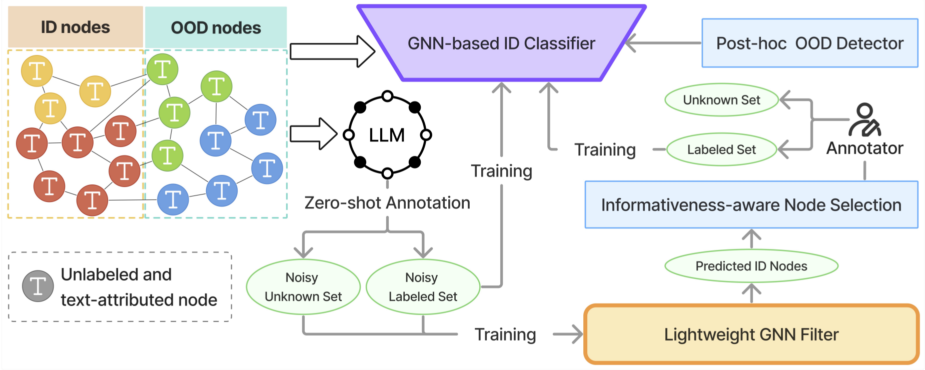

Fig. 2. An overview of our framework LLM-GOOD. To reduce human cost, we use LLM to filter out OOD nodes before human annotation (§4.1). To further reduce LLM cost, we use LLM to annotate a small subset of nodes, and then train a lightweight GNN filter on these noisy annotations to predict labels for the remaining nodes in the graph (§4.2). After obtaining node embeddings from the GNN filter, informativeness-aware selection methods identify the most informative unlabeled potential ID nodes (§4.3). After these selected nodes are annotated, the labeled accurate ID nodes are used to train the target ID classifier for ID classification and OOD detection (§4.4).

an LLM as an annotator to identify potential OOD nodes (see §4.1). However, annotating all unlabeled nodes in the training set using the LLM still incurs a high cost. Therefore, we propose to annotate a small subset of nodes with the LLM and use these pseudo-labels from LLM to train a lightweight GNN filter (see §4.2). This approach further reduces the cost of using the LLM. After training, the GNN filter can predict which unlabeled nodes are ID nodes, allowing us to identify potential ID nodes with minimal use of the LLM.

Furthermore, based on the node embeddings from the GNN filter, informativeness-aware node selection methods, such as K-Medoids-based node selection, can be applied to choose the most informative nodes from the unlabeled potential ID nodes (see §4.3). Once these informative nodes are annotated, the labeled ID nodes can be used to train the target ID classifier. Optionally, accurate labels from humans and noisy labels from the LLM can be combined to train a robust ID classifier, especially in scenarios of extreme data scarcity. Finally, post-hoc OOD detection methods can be applied to the classifier to enhance its ability to recognize unseen classes (see §4.4). Fig. 2 illustrates the pipeline of the proposed framework LLM-GOOD.

4.1 LLM as Zero-shot Open-world Annotator

We randomly select a small set of nodes $\mathcal{V}_{LLM}$ from $\mathcal{V}_{can}$ and then let LLM annotate them. We provide the LLM with ID knowledge (ID classes' names) and prompt it to determine whether an unlabeled query node belongs to one of the ID classes, incorporating the text information of the query node. An example prompt for zero-shot OOD detection is shown in the following box. We instruct LLM to output "none" if it predicts that the node does not belong to any of the ID classes. Therefore, the noisy labels of $\mathcal{V}_{LLM}$ from LLM are $\mathcal{Y}_{LLM} = \{y_1^n, y_2^n, \ldots, y_m^n\}$, where m is the number of annotated nodes. Given K known ID classes, the LLM's label set extends to $K+1$ classes, with the $(K+1)$-th class representing the unknown class.

Zero-Shot OOD Detection and ID Classification Prompt

As a research scientist, your task is to analyze and classify **{object}** based on their main topics, meanings, background, and methods. Please first read the content of the **{object}** carefully. Then, identify the **{object}**'s key focus. Finally, match the content to one of the given categories.
There are the following categories: `[Category 1, Category 2, Category 3, ...]`
Given the current possible categories, determine if it belongs to one of them. If so, specify that category; otherwise, say `"none"`.
`[Insert {Object} Content Here]`

4.2 Train Lightweight GNN with Pseudo-Labels

To further reduce LLM's cost, we first use the LLM to annotate a small subset of nodes and then train a GNN on these annotations to predict labels for the remaining nodes in the graph. With the labeled node set $\mathcal{V}_{LLM}$ and its noisy labels $\mathcal{Y}_{LLM}$, we can train a $K+1$ class classifier. As an aside, any GNN can be used as the lightweight OOD filter. In this paper, we use a two-layer standard graph convolutional network (GCN) as the OOD filter, and set the output dimension of the last layer as $K+1$. The output of the first layer is as follows:

$$\mathbf{H}^{(1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}^{(0)}\right) \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$, $\mathbf{I}$ is the identity matrix, and $\mathbf{W}^{(0)}$ is the weight matrix. The OOD filter's final output for all nodes is $\mathbf{H}^{(2)} \in \mathbb{R}^{N\times(K+1)}$:

$$\mathbf{H}^{(2)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(1)}\mathbf{W}^{(1)}\right) \tag{3}$$

Embeddings $\mathbf{H}^{(\mathbf{1})}$ capture graph structure information and can be leveraged in the subsequent module for selecting nodes based on informativeness. In addition, with $\mathbf{H}^{(2)}$, we can determine whether each unlabeled node belongs to the unknown $(K+1)$-th class, with the goal of filtering out as many OOD nodes as possible prior to human annotation. As a result, we retain nodes predicted to belong to one of the first K ID classes for further processing, while excluding those identified as unknown. Specifically, our goal is to filter out OOD nodes from $\mathcal{V}_{can}$ based on $\mathbf{H}^{(2)}$ to get the filtered ID node set $\mathcal{V}_{can}^{ID}$ and then select the most informative nodes from $\mathcal{V}_{can}^{ID}$ based on $\mathbf{H}^{(1)}$.

$$\mathcal{V}_{can}^{ID} = \left\{ v_i, \arg\max_k \mathbf{H}^{(2)}[i, K] \leq K \right\} \tag{4}$$

The cross-entropy loss function of the OOD filter is defined as:

$$\mathcal{L} = -\frac{1}{|\mathcal{V}_{LLM}|} \sum_{i \in \mathcal{V}_{LLM}} \sum_{k=1}^{K+1} y_{ik}^{n} \log \hat{y}_{ik}^{n} \tag{5}$$

4.3 Informativeness-aware Node Selection

Most node selection methods typically prioritize nodes with high prediction uncertainty or diverse representations for labeling. However, in the presence of open-set noise, these metrics become unreliable, as OOD nodes also exhibit high uncertainty and diversity while lacking class-specific features or shared inductive biases with ID examples. By utilizing our OOD filter to remove a significant number of OOD nodes, we can more effectively identify and select the most informative nodes from the remaining potential ID nodes. Any graph active selection method, such as FeatProp [32] or MITIGATE [3], can be applied.

4.4 ID Classification and OOD Detection

With the help of the OOD filter, we can train the target ID classifier with more labeled ID nodes while adhering to the label budget constraint.

Assume that we have selected $\mathcal{V}_{can}^{s}$ from $\mathcal{V}_{can}^{ID}$ and annotated it with accurate labels $\mathcal{Y}_{can}^{s}$. We now have a set of nodes, $\mathcal{V}_{can}^{s}$, with accurate labels $\mathcal{Y}_{can}^{s}$, and a set of nodes, $\mathcal{V}_{LLM}$, with noisy labels $\mathcal{Y}_{LLM}$. From $\mathcal{V}_{can}^{s}$ and $\mathcal{V}_{LLM}$, we can derive the ID node set $\mathcal{V}_{can}^{s-ID}$ with accurate labels and $\mathcal{V}_{LLM}^{ID}$ with noisy labels. We can then use $\mathcal{V}_{can}^{s-ID}$ and $\mathcal{V}_{LLM}^{ID}$ to train the target ID classifier. Similarly, any graph neural network can serve as the ID classifier. The design of noise-resistant GNNs to better leverage the noisy labels from LLM is left for future study.

Specifically, the output of the ID classifier is $\mathbf{Z} \in \mathbb{R}^{N \times K}$:

$$\mathbf{Z} = GNN(\mathbf{A}, \mathbf{X}) \tag{6}$$

Note that if a node is in both $\mathcal{V}_{can}^{s-ID}$ and $\mathcal{V}_{LLM}^{ID}$, its label is taken from $\mathcal{Y}_{can}^{s-ID}$. Using noisy labels from $\mathcal{Y}_{LLM}^{ID}$ is extremely helpful when there are very few accurate labels available, particularly in situations of extreme data scarcity.

After training the ID classifier, any post-hoc OOD detector [15,13,10,40,20] can be applied to the output logits of the ID classifier. As an example, consider the well-known post-hoc OOD detector, MSP [10]. Correctly classified examples generally exhibit higher maximum softmax probabilities compared to misclassified and out-of-distribution examples. Consequently, given $\mathbf{Z}$, we can compute the softmax probability of the predicted class, i.e., the maximum softmax probability, which serves as the OOD score.

5 Experiments

Our experiments answer the following research questions (RQ): **RQ1** (§5.2): How effective is the proposed LLM-GOOD in ID classification and OOD detection compared to other leading baselines? **RQ2** (§5.2): Whether LLMs can filter out OOD nodes effectively? **RQ3** (§5.4): Will LLM-GOOD be robust to different settings, such as varying levels of label scarcity? **RQ4** (§5.5): What are the differences in cost and effectiveness between various LLMs?

5.1 Experimental Setup

Datasets We utilize the following TAG datasets, which are commonly used for node classification: Cora [21], Citeseer [8], Pubmed [26] and Wiki-CS [22]. For each dataset, we split all classes into ID and OOD sets, and the ID classes for the four datasets are shown in Appendix B. Additionally, the number of ID classes is set to a minimum of two to perform the ID classification task.

For each dataset with K ID classes, we randomly select $10 \times K$ ID nodes and an equal number of OOD nodes for validation. The test set consists of 500 ID and 500 OOD nodes, while the remaining nodes form $\mathcal{V}_{can}$.

Baselines We evaluate LLM-GOOD against two categories of baselines: (1) OOD detection methods, including MSP [10], Entropy, GNNSafe [31], and GRASP [20]; (2) node selection methods for node classification, including uncertainty-based selection [19], FeatProp [32], and MITIGATE [3], where different selection strategies are integrated into GCNs with MSP as the OOD score.

For all methods, including baselines and LLM-GOOD, we use two GCN layers as the ID classifier.

Settings For all datasets, we use GPT-4o-mini to annotate 200 randomly selected nodes and train the lightweight GNN filter using these annotated noisy nodes with two standard GCN layers. The results for other LLM are given in Section 5.5. For LLM-GOOD, we use the energy score [31] as the OOD score. Additionally, we evaluate an alternative approach (LLM-GOOD-f), where LLMs filter all unlabeled nodes in the initial graph, and a subset of ID-labeled nodes is randomly selected for manual labeling.

Evaluation Metrics For the ID classification task, we use classification accuracy (ID ACC) as the evaluation metric. For the OOD detection task, we employ three commonly used metrics from the OOD detection literature [27]: the area under the ROC curve (AUROC), the precision-recall curve (AUPR), and the false positive rate when the true positive rate reaches 95% (FPR@95). In all experiments, the OOD nodes are considered positive cases. Details about these metrics are provided in Appendix A.

Table 1. Performance comparison (best highlighted in bold) of different models on ID classification and OOD detection tasks for the Cora and Citeseer datasets under label budget $10 \times K$. All values are percentages (%).

Model	Cora				Citeseer			
	ID ACC ↑	**AUROC ↑**	**AUPR ↑**	**FPR@95 ↓**	**ID ACC ↑**	**AUROC ↑**	**AUPR ↑**	**FPR@95 ↓**
GCN-Uncertainty	79.04±7.98	77.02±4.46	79.51±3.61	75.80±7.17	75.36±4.81	69.73±5.08	69.57±5.58	87.72±4.93
GCN-FeatProp	81.04±2.45	78.24±3.25	79.92±4.07	75.92±5.40	79.48±2.83	71.45±4.47	71.42±4.60	86.56±6.41
GCN-MITIGATE	81.64±2.31	79.04±2.31	80.52±2.56	73.40±4.71	80.44±3.12	72.19±4.33	71.92±3.99	**84.52±6.08**
MSP	77.68±7.60	75.40±6.85	78.19±5.53	81.32±9.72	70.92±7.46	62.12±7.09	64.63±5.02	90.64±3.78
GNNSafe	74.76±8.99	84.05±7.44	84.62±6.42	61.20±19.24	71.16±7.44	65.84±5.73	65.97±5.13	89.12±3.29
Entropy	76.80±8.65	76.10±8.08	78.12±6.68	76.24±12.87	73.20±4.28	63.26±6.65	65.24±4.81	88.56±4.69
GRASP	77.88±8.36	83.00±6.43	82.30±6.33	61.48±21.59	71.72±5.37	60.64±6.62	63.04±4.67	91.20±2.58
LLM-GOOD-f	84.00±4.40	86.59±2.32	87.36±3.10	60.56±3.94	72.52±10.43	70.71±4.49	72.99±4.77	88.92±6.85
LLM-GOOD	**85.20±2.68**	**88.06±3.77**	**87.85±3.68**	**48.04±1.19**	**80.60±3.38**	**73.29±4.12**	**75.26±3.34**	86.48±5.45

Implementation Details We evaluate all methods under the total label budgets $10 \times K$ and $5 \times K$, respectively. Since baseline methods require an initial set of labeled nodes and multiple rounds of node selection, in each selection round, K nodes are chosen from the unlabeled pool and annotated for all baselines. In addition, we allocate an initial label budget of $5 \times K$ for the total budget of $10 \times K$ and K for the total budget of $5 \times K$. In contrast, our method does not require an initial set of labeled nodes and involves only a single round of random node selection for annotation.

All GCNs have 2 layers with hidden dimensions of 32. All models use a learning rate of 0.01, a dropout probability of 0.5 and a weight decay of 5e-4. For all K-Medoids-based selection methods, the number of clusters is fixed at 48. For LLM-GOOD, 200 nodes are randomly selected and annotated by the LLM. The weight assigned to the unknown class in the GNN filter's loss function is selected from $\{0.05, 0.1, 0.2, 0.3, 0.5\}$ based on the performance of the validation set. For all experiments, we average all results across 5 different random seeds.

Table 2. Performance comparison (best highlighted in bold) of different models on ID classification and OOD detection tasks for the Pubmed and Wiki-CS datasets under label budget $10 \times K$. All values are percentages (%).

Model	Pubmed				Wiki-CS			
	ID ACC ↑	**AUROC ↑**	**AUPR ↑**	**FPR@95 ↓**	**ID ACC ↑**	**AUROC ↑**	**AUPR ↑**	**FPR@95 ↓**
GCN-Uncertainty	84.48±9.41	57.15±6.61	55.96±5.69	91.60±2.39	81.88±4.70	77.31±6.18	79.59±5.82	78.48±10.79
GCN-FeatProp	83.00±8.57	53.07±11.45	53.58±9.42	92.96±5.64	76.48±5.49	73.58±6.99	75.26±7.90	83.52±6.43
GCN-MITIGATE	83.24±8.48	57.91±10.41	57.93±9.57	92.60±3.76	77.76±7.96	71.69±4.91	73.26±4.77	86.16±7.90
MSP	81.04±7.51	52.65±8.86	53.74±7.15	93.44±3.60	77.52±5.60	75.10±4.56	77.54±5.48	84.80±4.72
GNNSafe	82.16±8.02	49.65±17.49	55.55±14.82	93.04±6.90	78.80±4.72	83.71±4.38	84.95±06.08	82.16±11.53
Entropy	81.36±7.57	52.22±09.70	52.61±8.47	92.68±3.27	77.76±4.80	72.85±3.71	75.47±4.68	88.00±3.46
GRASP	82.68±8.18	49.97±19.59	54.86±14.99	93.76±5.46	78.28±5.11	78.34±8.54	79.17±10.88	86.28±6.14
LLM-GOOD-f	87.00±2.19	61.09±19.05	66.81±15.58	91.24±4.89	83.76±2.46	86.84±1.84	89.42±1.57	79.24±10.66
LLM-GOOD	**87.08±2.58**	**64.87±16.70**	**70.60±14.26**	**90.72±5.10**	**83.92±3.53**	**87.71±2.41**	**89.84±2.64**	**71.04±16.20**

5.2 Main Results

As shown in Tables 1, 2, 3 and 4, LLM-GOOD consistently outperforms state-of-the-art graph OOD detection methods by a significant margin across all TAG datasets. Specifically, for ID classification on four datasets, the most substantial improvement is observed on the Cora dataset when the label budget is set to $5 \times K$. In this case, the ID accuracy increases from 63.80% (achieved by the best baseline, GNNSafe) to 81.52%, reflecting a notable improvement of 17.72%. It is important to note that all baselines have an initial set of labeled ID nodes and use multiple selection rounds to improve performance. In contrast, LLM-GOOD selects nodes randomly in a single round yet still outperforms the baselines.

Furthermore, LLM-GOOD exhibits remarkable advancements in OOD detection metrics, achieving higher AUROC and AUPR scores while maintaining a lower FPR@95 across all datasets. The most significant improvement is observed in the Pubmed dataset when the label budget is $10 \times K$, where the AUROC increases from 57.91% (achieved by the best baseline, GCN with MITIGATE node selection) to 64.87%, marking an improvement of 6.96%.

Moreover, we calculate the final proportion of ID nodes, defined as the ratio of true ID nodes among all selected and annotated nodes, across various selection methods. The results are in Appendix C. Our method achieves the highest proportion across all datasets compared to the baselines. This shows that our method effectively filters out OOD nodes before human annotation, thereby reducing annotation costs. While LLM-GOOD and LLM-GOOD-f achieve similar ID node proportions, LLM-GOOD significantly reduces LLM costs by annotating only a small number of nodes and leveraging a GNN filter to label the rest.

Table 3. Performance comparison (best highlighted in bold) of different models on ID classification and OOD detection for the **Cora** and **Citeseer** TAG datasets under label budget $5 \times K$. All values are percentages (%).

Model	**Cora**				**Citeseer**			
	ID ACC↑	**AUROC↑**	**AUPR↑**	**FPR@95↓**	**ID ACC↑**	**AUROC↑**	**AUPR↑**	**FPR@95↓**
GCN-Uncertainty	51.20	65.51	65.90	87.88	59.96	67.53	69.00	89.76
GCN-FeatProp	55.76	71.00	72.28	85.88	69.32	67.00	65.87	**86.20**
GCN-MITIGATE	58.68	67.36	69.36	86.64	67.64	64.42	65.27	90.72
MSP	63.32	72.18	73.41	82.92	71.20	63.59	65.51	89.72
GNNSafe	62.52	79.29	81.12	68.76	70.88	67.47	67.16	87.52
Entropy	63.80	73.03	73.06	77.08	69.00	65.47	67.14	87.20
GRASP	63.52	75.30	75.07	68.96	72.56	60.64	62.40	91.12
$\text{LLM}_{\text{GOOD-f}}$	**81.52**	**82.26**	**83.71**	**64.72**	69.20	**70.99**	**73.39**	88.60
LLM-GOOD	78.60	80.21	80.82	68.88	**72.12**	67.81	69.65	91.92

Table 4. Performance comparison (best highlighted in bold) of different models on ID classification and OOD detection for the **Pubmed** and **Wiki-CS** TAG datasets under label budget $5 \times K$. All values are percentages (%).

Model	Pubmed				Wiki-CS			
	ID ACC↑	AUROC↑	AUPR↑	FPR@95↓	ID ACC↑	AUROC↑	AUPR↑	FPR@95↓
GCN-Uncertainty	73.36	57.53	57.55	90.56	61.40	61.94	64.86	92.12
GCN-FeatProp	71.60	50.89	51.19	95.04	71.64	69.30	71.73	90.32
GCN-MITIGATE	71.76	57.27	57.73	92.84	70.36	60.36	62.29	92.84
MSP	82.04	57.10	57.52	92.32	67.60	74.64	77.92	84.16
GNNSafe	81.48	53.71	59.51	96.44	68.20	79.38	81.26	83.28
Entropy	82.04	56.80	55.75	92.16	66.40	72.45	74.96	84.04
GRASP	81.40	52.35	57.30	94.96	67.84	77.54	79.95	85.60
$\text{LLM}_{\text{GOOD-f}}$	**83.52**	**63.29**	**68.68**	93.32	75.24	**84.62**	**86.30**	**77.88**
LLM-GOOD	78.08	60.34	65.13	**89.76**	**78.56**	83.35	86.26	84.84

5.3 OOD Detection Performance Upper Bound

We use different number of LLM's noisy labels and human's annotated accurate labels to train the ID classifier respectively. Given the different label budgets, we randomly select a corresponding number of nodes and use the ID nodes from the selected nodes to train the ID classifier. The results in Figure 3 shows that:

- When the number of noisy ID labels or accurate ID labels increases, both ID classification and OOD detection performance improve. However, the improvement rate is significantly higher when using accurate ID labels.
- When training the ID classifier with LLM-generated noisy labels, both ID classification and OOD detection performance reach an upper bound substantially lower than that of training with accurate labels. This highlights the importance of our method, which utilizes LLM to reduce human annotation costs without relying entirely on the LLM for OOD detection.
- When the label budget for accurate labels reaches $10 \times K$, ID classification and OOD detection performance nearly reach the upper bound achieved with a large number of noisy labels. At $20 \times K$, both exceed the upper bound of using any number of LLM-generated noisy labels.

5.4 Combine Accurate Labels and Noisy Labels

We test different methods' performance under severe data-scarcity situation on Cora. The human label budget is set to $1 \times K$, $2 \times K$ and $3 \times K$. For LLM-GOOD-combined, we use 100 noisy labels along with a small number of corresponding clean labels to train the ID classifier. The results are shown in Table 5.

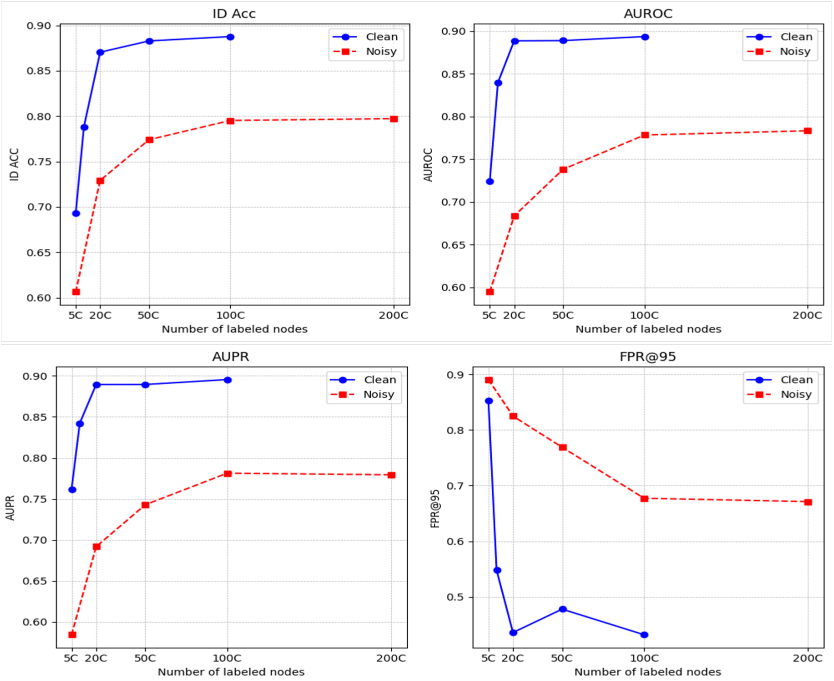

Fig. 3. ID classification and OOD detection performance upper bound.

Table 5. Different methods' performance under severe data-scarcity situation on Cora. LLM-GOOD-combined achieves the best performance.

	$1 \times K$	$2 \times K$	$3 \times K$
GCN-MSP	0.3228	0.4260	0.4856
LLM-GOOD	0.4580	0.6156	0.7492
LLM-GOOD-combined	**0.7832**	**0.8072**	**0.8164**

From the results, we observe that for all methods, increasing the label budget leads to improved performance, and our method consistently outperforms the baseline. When the accurate label budget is extremely small, incorporating additional noisy labels is particularly beneficial. For instance, when the accurate label budget is $1 \times K$, the performance gap between LLM-GOOD and LLM-GOOD-combined is 32.52%. However, as the accurate label budget increases to $3 \times K$, the performance gap decreases to 6.72%.

Currently, most graph machine learning research assumes either that the entire training set is clean or that all training labels are uniformly affected by a specific type of noise. However, in real-world scenarios, it is more likely that a graph contains a small set of clean labels alongside another set of noisy labels.

Table 6. Comparison of zero-shot OOD detection performance using different prompts across various LLMs on the Cora and Pubmed datasets.

	Cora			**Pubmed**		
	AUROC	**AUPR**	**OOD Proportion**	**AUROC**	**AUPR**	**OOD Proportion**
GPT-3.5-turbo-short prompt	0.5077	0.6609	0.0200	0.5000	0.7100	0.0000
GPT-3.5-turbo-long prompt	0.5909	0.7468	0.1700	0.5255	0.6440	0.0300
GPT-4o-mini-short prompt	0.7159	0.8200	0.5600	0.5060	0.7135	0.0050
GPT-4o-mini-long prompt	**0.7366**	**0.8323**	0.5150	0.8524	0.8796	0.3650
ds-v3-short prompt	0.6185	0.7589	0.2350	0.5000	0.7100	0.0000
ds-v3-long prompt	0.6887	0.8170	0.7000	**0.9364**	**0.9293**	0.4700

We leave the design of a more effective pipeline to leverage both label sets to denoise and train a robust, noise-resistant GNN as a direction for future study.

5.5 LLMs as Open-world Zero-shot Annotators

We record the annotation cost and zero-shot OOD detection performance of the following LLMs: GPT-3.5-turbo, GPT-4, GPT-4o, GPT-4o-mini, DeepSeek-V3, DeepSeek-R1. Severe rate limitation prevented DeepSeek-R1 from annotating 200 nodes in a reasonable time, so its results are not included.

Zero-shot annotation accuracy We use the baseline short prompt (as shown in Section 4.1) and the long prompt (as shown in Appendix D) for zero-shot OOD detection on the Cora dataset. We randomly select 200 nodes and have different LLMs perform zero-shot open-world annotation using these two prompts. The true OOD proportion of the selected nodes is 56%. The OOD detection performance and the LLMs' predicted OOD proportions are presented in Table 6. From the results, we can observe that, sometimes, GPT-3.5-turbo does not dare to say 'none', but our long prompt mitigates this issue. Additionally, both the OOD detection performance and the predicted OOD proportion improve significantly with the long prompt, suggesting that GPT-3.5-turbo becomes more willing to say 'none.' Furthermore, when using the same prompt for open-world annotation, GPT-4o-mini generally outperforms GPT-3.5-turbo in OOD detection.

We further evaluate open-world annotation on the PubMed dataset by randomly selecting 200 nodes and having the LLMs annotate them using two prompts. The true OOD proportion of the selected nodes is 42%. We can observe that our proposed prompt outperforms the baseline short prompt in zero-shot OOD detection, even though the latter explicitly instructs the LLM to respond with "none" for OOD nodes.

Table 7. The cost (dollars) of different LLMs for annotating 200 nodes on Cora dataset.

	GPT-3.5-turbo	**GPT-4o-mini**	**GPT-4o**	**GPT-4**	**ds-v3**	**ds-r1**
Cost	0.07	0.02	0.50	3.70	0.03	0.55

Cost We randomly select 200 nodes from the Cora dataset and have different LLMs annotate them. The costs associated with each LLM are shown in Table 7. As observed, GPT-4o-mini incurs the lowest cost while achieving significantly better zero-shot open-world annotation performance than GPT-3.5-turbo. Therefore, in this paper, we use GPT-4o-mini for node annotation to reduce human costs in open-set scenarios.

6 Conclusion and Future Directions

In this paper, we introduce a novel approach leveraging the powerful zero-shot learning capabilities of LLMs for label-efficient graph OOD detection. We propose a general framework, LLM-GOOD, which filters out a large number of OOD nodes before annotation, significantly reducing human labeling costs. Additionally, LLM-GOOD utilizes zero-shot annotations from LLMs to train a lightweight GNN filter, further minimizing the reliance on LLMs. Unlike traditional multi-round active learning methods, our approach achieves high performance with a single round of node selection. A potential future research direction is to investigate more effective ways to leverage both clean and noisy labels to train a more noise-resistant ID classifier for graph OOD detection. Additionally, it would be interesting to explore whether in-context learning can improve node-level OOD detection performance compared to zero-shot OOD detection with LLMs.

Acknowledgments

This work was partially supported by the National Science Foundation under Award No. 2428039 and No. 2346158. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Bai, Y., Han, Z., Cao, B., Jiang, X., Hu, Q., Zhang, C.: Id-like prompt learning for few-shot out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17480–17489 (2024)
2. Cai, H., Zheng, V.W., Chang, K.C.C.: Active learning for graph embedding. arXiv preprint arXiv:1705.05085 (2017)
3. Chang, W., Liu, K., Ding, K., Yu, P.S., Yu, J.: Multitask active learning for graph anomaly detection. arXiv preprint arXiv:2401.13210 (2024)
4. Chen, Z., Mao, H., Wen, H., Han, H., Jin, W., Zhang, H., Liu, H., Tang, J.: Label-free node classification on graphs with large language models (llms). arXiv preprint arXiv:2310.04668 (2023)
5. Ding, C., Pang, G.: Zero-shot out-of-distribution detection with outlier label exposure. arXiv preprint arXiv:2406.01170 (2024)

6. Ding, K., Wang, J., Li, J., Shu, K., Liu, C., Liu, H.: Graph prototypical networks for few-shot learning on attributed networks. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 295–304 (2020)
7. Dong, H., Zhao, Y., Chatzi, E., Fink, O.: Multiood: Scaling out-of-distribution detection for multiple modalities. arXiv preprint arXiv:2405.17419 (2024)
8. Giles, C.L., Bollacker, K.D., Lawrence, S.: Citeseer: An automatic citation indexing system. In: Proceedings of the third ACM conference on Digital libraries. pp. 89–98 (1998)
9. Guo, Z., Xia, L., Yu, Y., Wang, Y., Yang, Z., Wei, W., Pang, L., Chua, T.S., Huang, C.: Graphedit: Large language models for graph structure learning. arXiv preprint arXiv:2402.15183 (2024)
10. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
11. Ju, W., Liu, Z., Qin, Y., Feng, B., Wang, C., Guo, Z., Luo, X., Zhang, M.: Few-shot molecular property prediction via hierarchically structured learning on relation graphs. Neural Networks **163**, 122–131 (2023)
12. Ju, W., Yi, S., Wang, Y., Long, Q., Luo, J., Xiao, Z., Zhang, M.: A survey of data-efficient graph learning. arXiv preprint arXiv:2402.00447 (2024)
13. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems **31** (2018)
14. Li, S., Gong, H., Dong, H., Yang, T., Tu, Z., Zhao, Y.: DPU: Dynamic prototype updating for multimodal out-of-distribution detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
15. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017)
16. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020)
17. Liu, Q., Paparrizos, J.: The elephant in the room: Towards a reliable time-series anomaly detection benchmark. Advances in Neural Information Processing Systems, 37, 108231–108261 (2024)
18. Liu, Q., Boniol, P., Palpanas, T., Paparrizos, J.: Time-series anomaly detection: Overview and new trends. Proceedings of the VLDB Endowment (PVLDB), 17(12), 4229–4232 (2024)
19. Luo, W., Schwing, A., Urtasun, R.: Latent structured active learning. Advances in neural information processing systems **26** (2013)
20. Ma, L., Sun, Y., Ding, K., Liu, Z., Wu, F.: Revisiting score propagation in graph out-of-distribution detection. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems
21. McCallum, A.K., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval **3**, 127–163 (2000)
22. Mernyei, P., Cangea, C.: Wiki-cs: A wikipedia-based benchmark for graph neural networks. arXiv preprint arXiv:2007.02901 (2020)
23. Miyai, A., Yu, Q., Irie, G., Aizawa, K.: Locoop: Few-shot out-of-distribution detection via prompt learning. Advances in Neural Information Processing Systems **36** (2024)
24. Reimers, N.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)

25. Ren, X., Tang, J., Yin, D., Chawla, N., Huang, C.: A survey of large language models for graphs. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 6616–6626 (2024)
26. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008)
27. Song, Y., Wang, D.: Learning on graphs with out-of-distribution nodes. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1635–1645 (2022)
28. Tan, Z., Ding, K., Guo, R., Liu, H.: Graph few-shot class-incremental learning. In: Proceedings of the fifteenth ACM international conference on web search and data mining. pp. 987–996 (2022)
29. Wang, Y., Duan, Z., Huang, Y., Xu, H., Feng, J., Ren, A.: MTHetGNN: A heterogeneous graph embedding framework for multivariate time series forecasting. Pattern Recognition Letters 153, 151–158 (2022)
30. Wang, H., Li, Y., Yao, H., Li, X.: Clipn for zero-shot ood detection: Teaching clip to say no. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1802–1812 (2023)
31. Wu, Q., Chen, Y., Yang, C., Yan, J.: Energy-based out-of-distribution detection for graph neural networks. arXiv preprint arXiv:2302.02914 (2023)
32. Wu, Y., Xu, Y., Singh, A., Yang, Y., Dubrawski, A.: Active learning for graph neural networks via node feature propagation. arXiv preprint arXiv:1910.07567 (2019)
33. Xiao, Z., Song, W., Xu, H., Ren, Z., Sun, Y.: TIMME: Twitter ideology-detection via multi-task multi-relational embedding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2258–2268 (2020)
34. Xia, Y., Mukherjee, S., Xie, Z., Wu, J., Li, X., Aponte, R., Lyu, H., Barrow, J., Chen, H., Dernoncourt, F., et al.: From selection to generation: A survey of llm-based active learning. arXiv preprint arXiv:2502.11767 (2025)
35. Xu, H., Duan, Z., Wang, Y., Feng, J., Chen, R., Zhang, Q., Xu, Z.: Graph partitioning and graph neural network based hierarchical graph matching for graph similarity computation. Neurocomputing 439, 348–362 (2021)
36. Xu, H., Chen, R., Wang, Y., Duan, Z., Feng, J.: CoSimGNN: Towards large-scale graph similarity computation. arXiv preprint arXiv:2005.07115 (2020)
37. Xu, H., Liu, K., Yao, Z., Yu, P.S., Ding, K., Zhao, Y.: Lego-learn: Label-efficient graph open-set learning. arXiv preprint arXiv:2410.16386 (2024)
38. Xu, H., Yao, Z., Wang, Z., Cheng, Z., Hu, X., Li, M., Zhao, Y.: Graph synthetic out-of-distribution exposure with large language models. arXiv preprint arXiv:2504.21198 (2025)
39. Xu, H., Yao, Z., Zhang, X., Wang, Z., He, L., Dong, Y., Yu, P.S., Li, M., Zhao, Y.: Glip-ood: Zero-shot graph ood detection with foundation model. arXiv preprint arXiv:2504.21186 (2025)
40. Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., et al.: Openood: Benchmarking generalized out-of-distribution detection. Advances in Neural Information Processing Systems **35**, 32598–32611 (2022)
41. Yu, T., He, S., Song, Y.Z., Xiang, T.: Hybrid graph neural networks for few-shot learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 3179–3187 (2022)