

# ALLIE: Active Learning on Large-scale Imbalanced Graphs

Limeng Cui\*, Xianfeng Tang, Sumeet Katariya, Nikhil Rao, Pallav Agrawal,  
Karthik Subbian, Dongwon Lee\*

\*The Pennsylvania State University, USA {lzc334, dongwon}@psu.edu  
Amazon.com, USA {xianft, katsumee, nikhilr, pplg, ksubbian}@amazon.com

## ABSTRACT

Human labeling is time-consuming and costly. This problem is further exacerbated in extremely imbalanced class label scenarios, such as detecting fraudsters in online websites. Active learning selects the most relevant example for human labelers to improve the model performance at a lower cost. However, existing methods for active learning for graph data often assumes that both data and label distributions are balanced. These assumptions fail in extreme rare-class classification scenarios, such as classifying abusive reviews in an e-commerce website.

We propose a novel framework ALLIE to address this challenge of active learning in large-scale imbalanced graph data. In our approach, we efficiently sample from both majority and minority classes using a reinforcement learning agent with imbalance-aware reward function. We employ focal loss in the node classification model in order to focus more on rare class and improve the accuracy of the downstream model. Finally, we use a graph coarsening strategy to reduce the search space of the reinforcement learning agent. We conduct extensive experiments on benchmark graph datasets and real-world e-commerce datasets. ALLIE out-performs state-of-the-art graph-based active learning methods significantly, with up to 10% improvement of F1 score for the positive class. We also validate ALLIE on a proprietary e-commerce graph data by tasking it to detect abuse. Our coarsening strategy reduces the computational time by up to 38% in both proprietary and public datasets.

## KEYWORDS

Graph neural networks, fraud detection, active learning, reinforcement learning

### ACM Reference Format:

Limeng Cui\*, Xianfeng Tang, Sumeet Katariya, Nikhil Rao, Pallav Agrawal,, Karthik Subbian, Dongwon Lee\*. 2022. ALLIE: Active Learning on Large-scale Imbalanced Graphs. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*, April 25–29, 2022, Virtual Event, Lyon, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3485447.3512229>

## 1 INTRODUCTION

Graph structured data are ubiquitous and are widely used in social network analysis [29], financial fraud detection [46], molecular design [17], search engines [44] and recommender systems [36, 38].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
WWW '22, April 25–29, 2022, Virtual Event, Lyon, France.

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9096-5/22/04...\$15.00

<https://doi.org/10.1145/3485447.3512229>

Recently, Graph Neural Networks (GNNs) have emerged as state-of-the-art models on these types of datasets, due to their ability to learn and aggregate complex interactions between (K-hop) neighborhoods, as opposed to traditional pointwise or pairwise models [24]. Despite their appealing advantages, GNNs, like other deep learning models, require a large amount of labeled data for training in supervised settings. It is often time-consuming, labor-intensive, and expensive to acquire sufficient labeled data for training in many domains, hindering the application of GNNs.

**Active Learning (AL)** is a promising solution to obtain labels faster, cheaper, and train models efficiently. AL dynamically queries candidate samples<sup>1</sup> for labeling to maximize the performance of the machine learned model with limited budget. The recent developments in AL on graphs [7, 10, 16, 17, 21, 31, 35, 49, 54] have proven to be effective on several benchmark datasets, such as citation graphs and gene networks. However, AL methods for large-scale *imbalanced* scenarios (e.g., finding a small fraction of fraudulent reviews on an e-commerce website) is less explored. This motivates us to study how to query the most “informative” samples so as to ameliorate the effect of imbalance and to reduce the training cost of GNNs.

Training GNNs with AL algorithm on imbalanced graphs is non-trivial. The **low prevalence rate** of positive samples<sup>2</sup> prevents traditional AL methods from learning the whole data distribution, because under-represented positive samples are less likely to be selected by traditional AL methods. For example, finding abusive reviews on a shopping website can be formulated as a binary classification problem, where positive samples (i.e., abusive reviews) are a very small portion of the labeled data. Training an AL model to sample reviews for labeling will mostly yield non-abusive reviews, resulting in limited model performance improvement. Most of the AL sampling methods proposed in natural language processing and computer vision [3, 52, 53] to balance class distribution assume independent and identically distributed (i.i.d.) data. These approaches are not directly applicable to graph structured data due to the heterogeneous relational structure and dense connections. Moreover, existing AL methods tend to reinforce or even worsen the prediction bias on minority classes when querying unlabeled data [3].

It is challenging to build an AL approach for **large-scale graph data**. For example, popular social network platforms (e.g., Facebook, Snapchat) have hundreds of millions of monthly active users; online e-commerce websites (e.g., Amazon, Walmart) host millions of products and conduct billions of transactions. Searching over all the unlabeled samples in the graph at this scale is impractical, as the computational complexity of AL methods grows exponentially with the size of the unlabeled set. Therefore, it is critical to reduce the search space for AL algorithms on large-scale graphs.

<sup>1</sup>Samples in the case of node classification in a GNN will be nodes.

<sup>2</sup>We assume positive samples are the rare class in the imbalanced setting

To tackle the aforementioned two challenges, we propose an Active Learning based method for Large-scale Imbalanced graphs (ALLIE), which combines the idea of AL on graphs with reinforcement learning for accurate and efficient node classification. ALLIE can effectively select informative unlabeled samples for labeling, using multiple uncertainty measures as its criteria. Moreover, our approach gives labeling priority to less confident and “under-represented” samples. To scale our approach to large graphs, we further introduce a graph coarsening strategy for ALLIE that categorizes similar nodes into clusters. With a better representation of nodes in each cluster, the search space for the AL algorithm is reduced. To the best of our knowledge, this work is the first to jointly model the imbalance issue on large-scale graphs and active learning. Our contributions are as follows:

- **Imbalance-aware reinforcement learning based graph policy network.** We apply a reinforcement learning strategy by maximizing the performance of the classifier to find a representative subset of the unlabeled dataset. The queried nodes will be more representative for the minority class (Section. 3.2.1).
- **Graph coarsening strategy to handle large-scale graph data.** Existing methods seldom pay attention to scalability, making them less efficient when applied to real-world applications. To reduce running time, we apply a graph coarsening strategy to reduce the action space in the policy network (Section. 3.2.2).
- **Robust learning for more accurate node classification.** Unlike conventional methods that do not distinguish the majority and minority classes when optimizing the objective function, we construct a node classifier with focal loss that down-weights the well-classified examples (Section. 3.2.3).

We evaluated ALLIE on both balanced and imbalanced datasets. Our balanced datasets use public citation graphs (Section. 4.2) and the imbalanced dataset is from a proprietary e-commerce website (Section. 4.3). We report the performance on node classification on both datasets. The reported results show that on balanced graph datasets, ALLIE improved an average of 2.39% in Macro F1 and 2.71% in Micro F1 over the best baseline. On the e-commerce website dataset, ALLIE achieved an average increase of 4.75% in Precision, 1.96% in Recall and 3.45% in F1 (with 10.54%, 3.7% and 7.71% relative improvement respectively) on the positive classes (i.e., the abusive users and reviews) over the best baseline. We also conduct a comprehensive ablation study to demonstrate the necessity of each component of ALLIE. Additional experiments show ALLIE performs well over baselines with various initial training set sizes and query budgets.

## 2 RELATED WORK

### 2.1 Active Learning on Graphs

Active learning [2, 4] has been widely studied in different domains such as computer vision [25, 30] and natural language processing [14, 50]. More recently, some pioneering works explore AL for graph structured data [17, 31, 54]. For example, AGE [7] selects the clustering center of node features. It combines several measurements together, including information entropy [41], density and centrality to find the best candidate(s) from all unlabeled nodes. FeatProp [49] extends AGE and also uses cluster centers as selected candidates. The authors proved an upper bound on the classification loss, and

discussed why they chose K-Medoids instead of K-Center as the clustering method. ANR<sub>MAB</sub> [16] uses Multi-Armed Bandit to select one metric from the measurements in AGE. Chen et al. [10] propose ActiveHNE to further extend ANR<sub>MAB</sub> to cover heterogeneous graphs. GPA [21] uses a policy network to perform AL on graphs. The goal is to select a sequence of nodes by using reinforcement learning which maximizes the performance of the GNN. Different from the heuristics-based AL methods, MetAL [35] uses meta-gradients to evaluate the importance of labeling each unlabeled instance. Despite these achievements, existing work mainly focus on balanced datasets, and perform poorly when the datasets are imbalanced. In addition, the measurements that are commonly used to estimate the representativeness of the samples are centrality and density. Though these criteria can help characterize data distribution, they do not favor the most “underrepresented” samples at the borders between classes. We explicitly focus on functions that can be adapted to imbalanced datasets.

### 2.2 Graph Coarsening

Learning on graphs is too time-consuming for large-scale graph data that model the dense connections among millions of nodes. The computational cost grows exponentially with the number of nodes [22]. In addition to using state-of-the-art models, we can compress graphs to reduce the running time of AL on them. Sparsification and reduction are two common ways of simplifying graphs. Sparsification reduces the number of edges, such as spanners, edge cut, and spectral sparsifiers [26, 43]. Such methods have been previously used for recommendation systems [12]. Reduction is conducted on the number of vertices as well as the number of edges. Related methods include graph coarsening [28, 51] and Kron reduction [8].

Graph coarsening is the merging of vertices in a graph to obtain a coarser version of the original graph with similar spectral properties [27]. We can use the same algorithm to process the coarser graph as with the original. Graph coarsening can be repeated several times until we get a sufficiently coarse graph [20, 47]. DiffPool [51] uses an assignment matrix to transform the original graph to a coarser one. It pools nodes given an assignment matrix at each layer. SAGPool [28] generalizes convolution operations to graphs. Researchers have incorporated graph coarsening into GNNs as a way to implement efficient pooling [5, 13]. For example, GraphSAGE [45] with DiffPool is 12 times faster than the original model.

### 2.3 Imbalanced Learning

Many real-world applications in computer vision [37], medical diagnosis [11] and fraud detection [34] suffer from class imbalance. Learning from an imbalanced dataset may result in a prediction model that favors the majority class over the minority class [19]. A comprehensive review of class imbalance problems in deep learning can be found in [6].

Methods for dealing with imbalance can be roughly divided into two categories: data-level and algorithmic-level methods. Oversampling [9, 18] and undersampling [39] are two data-level methods that are commonly used in deep learning. Oversampling replicates selected samples from minority classes, while undersampling removes samples from majority classes. Algorithm-level methods keep the

data unchanged while adjusting the training or inference process. Focal loss is a scaled cross-entropy loss, where the scaling factor goes to zero for well-classified samples [33]. Cost-sensitive learning assigns different costs to the misclassified samples from different classes [15].

### 3 THE PROPOSED METHOD: ALLIE

#### 3.1 Task Description

Let  $G = (V, E)$  denote a graph, where  $V$  is a set of nodes and  $E$  is a set of edges. We consider a classification setting where each node  $v \in V$  has a label  $y \in \mathcal{Y} = \{1, \dots, C\}$  ( $C$  is the number of classes). The node set is divided into three subsets including  $V_{\text{train}}$ ,  $V_{\text{valid}}$  and  $V_{\text{test}}$ , with corresponding label sets  $Y_{\text{train}}$ ,  $Y_{\text{valid}}$  and  $Y_{\text{test}}$ . In traditional supervised learning, the goal is to learn a classifier  $f(G, V_{\text{train}}; \theta_d)$  parameterized by  $\theta_d$  with the graph  $G$  and labels  $V_{\text{train}}$  to predict the labels of the nodes in  $V_{\text{test}}$ .

In AL setting, a query budget  $B$  is given, which allows us to query the labels of  $B$  samples from  $V_{\text{train}}$  ( $B \ll |V_{\text{train}}|$ ) in total. Suppose the initial label set is denoted as  $V_{\text{query}}^0$ . At each step  $t$ , we select an unlabeled node  $v^t$  using an AL policy  $\pi$  from the remaining candidate nodes  $V_{\text{train}} \setminus V_{\text{query}}^{t-1}$  that have not been queried, and query the label of the node  $v^t$ . Then we update  $V_{\text{query}}^t$  by  $V_{\text{query}}^{t-1} \cup \{v^t\}$  and train the classifier  $f(G, V_{\text{query}}^t; \theta_d)$  for one epoch. After the query budget is used up, we continue training  $f(G, V_{\text{query}}^B; \theta_d)$  with  $V_{\text{query}}$  until convergence.

The learning process of policy  $\pi$  can be naturally formulated as a Markov Decision Process (MDP), in which the AL network is sequentially querying unlabeled nodes into a sequence over time. Formally, the MDP is defined as follows.

- **State space  $\mathcal{S}$ :** A state matrix  $S^t \in \mathcal{S}$  is defined as the state of graph  $G$  at time  $t$  where each row  $s^t$  is the state representation of a node. More specifically, a state  $s^t$  consists of a node's degree, entropy, average KL divergence and reverse KL divergence between its predicted label distribution and its neighbor's.
- **Action space  $\mathcal{A}$ :** At time  $t$ , the action  $a^t \in \mathcal{A}$  is to determine which node should be queried next. The AL network will append the node to the node sequence. The number of actions taken should satisfy the given budget constraint.
- **Reward  $\mathcal{R}$ :** After the network has selected a sequence of nodes, we evaluate the performance of the classifier on the validation set  $V_{\text{valid}}$  as the final reward. Since the size of the initial training set  $|V_{\text{train}}|$  is limited in AL, calculating the immediate reward after each action will change the policy estimation a lot. Hence, in order to measure the policy's quality more accurately, we choose to calculate only the final reward, which provides a more stable estimation.
- **Transition probability  $\mathcal{P}$ :** Transition probability  $p(S^{t+1}|S^t, a^t)$  defines the state transition from  $S^t$  to  $S^{t+1}$  after taking action  $a^t$  at time  $t$ .

We parameterize the policy network using a deep neural network, which is defined as follows:

**DEFINITION 1 (POLICY NETWORK).** A policy network  $\pi(\cdot; \theta_p)$  parameterized by  $\theta_p$  is used to select a node sequence from the candidate training nodes to query, which yields a probability score for each node in the unlabeled set. We learn the optimal parameter  $\theta_p^*$  by maximizing the performance of the classifier  $f$  on the validation set:  $\mathcal{M}(f(G, V_{\text{valid}}), Y_{\text{valid}})$ , where  $\mathcal{M}$  is an evaluation metric,  $f$  is trained on  $V_{\text{query}}^B$  chosen by  $\theta_p^*$  and  $Y_{\text{valid}}$  is the labels of the validation set.

Hence we can formally define the MDP based AL problem on graphs as follows:

**PROBLEM 1 (MDP BASED ACTIVE LEARNING PROBLEM).** Given a graph  $G = (V, E)$ , with a query budget  $B$ , our goal is to learn a policy  $\pi$  to select the best node sequence to query, in order to optimize the prediction performance throughout the query process.

#### 3.2 Framework

Figure 1 illustrates the proposed framework. First, we alternately use a policy network to query the label of a candidate node and train the GNN classifier to update the current state of the graph, until the query budget is reached. In what follows, we evaluate the GNN classifier on the validation set to update the policy.

**3.2.1 Reinforcement Learning Architecture.** The AL algorithm takes an action by selecting the next node to query. In addition to the heuristic metrics, we can choose the nodes that can maximize the performance of the GNN classifier on the validation set. As this problem can be naturally formalized as a reinforcement learning architecture, we followed the GPA framework [49] in our paper.

We denote the state of graph  $G$  at step  $t$  as a matrix  $S^t$ , where each row  $s_v^t$  is the state representation of node  $v$ . In order to represent the state representation, we adopt degree as representativeness measure and entropy and KL divergence as the uncertainty measures in the policy network:

- **Degree:** We use the degree of a node to represent its representativeness. The higher the degree of the nodes, the more important the nodes are. Thus their labels are more likely to be informative. The degree is denoted by

$$s_{v,1}^t = \min(\text{degree}(v)/\delta, 1), \quad (1)$$

where  $\delta$  is a scaling hyperparameter.

- **Entropy:** The entropy of the label distribution is to predict the uncertainty of each node. In other words, if the classifier has low confidence about a node's predicted label, then the node's label is more likely to be useful. We divide the entropy by  $\log(C)$  to normalize it into range  $[0, 1]$ :

$$s_{v,2}^t = -\frac{1}{\log(C)} \sum_{i=1}^C \hat{y}_i(v^t) \log(\hat{y}_i(v^t)), \quad (2)$$

where  $\hat{y}_i(v^t)$  is the class probability of node  $v$  belonging to the  $i$ -th class predicted by the classifier at step  $t$ .

- **Divergence:** The divergence is calculated based on a node's label prediction distribution and its neighbor's. It measures how different the node and its neighbors are, which can better identify the decision boundaries in the graph:

$$s_{v,3}^t = \frac{1}{|N_v|} \sum_{u \in N_v} \text{KL}(\hat{y}(v^t) \parallel \hat{y}(u^t)), \quad s_{v,4}^t = \frac{1}{|N_v|} \sum_{u \in N_v} \text{KL}(\hat{y}(u^t) \parallel \hat{y}(v^t)). \quad (3)$$

We use an indicator to represent whether the node has been labeled or not, and concatenate it with the above metrics to form the feature vector  $s_v^t$  for each node  $v$ . The graph state matrix  $S^t$  will be passed into the policy network to generate the action probabilities.

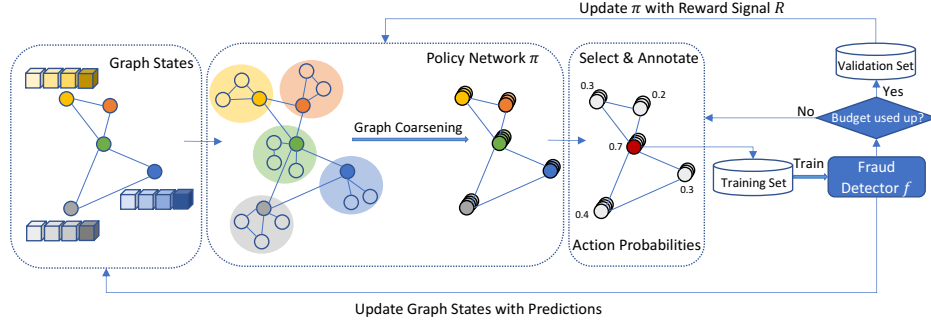


Figure 1: The overview of ALLIE with two main parts: policy network and fraud detector.

**Imbalance-aware Reward Function Design:** In order to fix the imbalanced data distribution issue and boost the model's performance on the minority classes, we introduce a balancing strategy on the reward in AL to make the method query more nodes that can represent the minority class better. Specifically, for the reward signal, we use a performance metric that treats each class equally instead of each sample and thus assigns more weights to the minority samples.

Below we detail how we calculate the reward signal and what metrics we choose. The policy network is rewarded by the performance gain of the GNN classifier  $f$  trained with the updated set of labeled nodes. The reward of the selected node sequence is calculated based on the performance of  $f$  on the validation set:

$$R(V_{\text{query}}^B) = \mathcal{M}(f(G, V_{\text{valid}}), Y_{\text{valid}}), \quad (4)$$

where  $\mathcal{M}$  is the evaluation metric,  $f$  trained on graph  $G$  and labels of  $V_{\text{query}}^B$ ,  $V_{\text{valid}}$  and  $Y_{\text{valid}}$  are the nodes and labels of the validation set.

We implement  $\mathcal{M}$  using the following metrics:

- **Weighted reward:** When the sample belongs to the minority class, the reward is +1 if the prediction  $\hat{y}$  is correct; -1 if not. When the sample belongs to the majority class, the reward is multiplied by the imbalanced ratio  $\rho$ , which is the number of samples in the minority class divided by the number of samples in the majority class [32].
- **Micro-F1** calculates metrics by counting the total true positives, false negatives and false positives globally, which favors the majority classes, e.g., the benign buyer.
- **Macro-F1** averages the F1 score per class, which can get a sense of effectiveness on the small classes (e.g., the abusive buyer).

We empirically compare the three reward functions in Section 4.4.1 and find ALLIE with Macro-F1 achieves the best results.

**Reinforcement Learning Algorithm:** The training framework is shown in Figure 1. At every step, we first update the graph state matrix  $S_G^t$ . The policy network selects a node  $v^t$  from  $V_{\text{train}} \setminus V_{\text{query}}^{t-1}$  based on the probability of each action  $\pi(\cdot|S^t)$ , gets its label, and puts it into the label set  $V_{\text{query}}^t$ . Then the GNN classifier  $f$  is trained for one epoch on graph state matrix  $S_G^t$  and the label set  $V_{\text{query}}^t$ . After that, we can get the new label prediction of each node and update the heuristic metrics such as  $s_{v,2}^t$ ,  $s_{v,3}^t$  and  $s_{v,4}^t$ . The heuristic metrics are used to generate the graph state matrix  $S^{t+1}$  for the next step. When the query budget  $B$  is used up, we train the GNN classifier  $f$  until convergence without querying more nodes.

**3.2.2 Policy Network Design.** The policy network takes graph state as an input and produces the probability distribution of each action (where an action is querying a node's label). GNNs can better characterize the graph's topology and help find the most informative nodes in the graph. Hence we set up the policy network architecture as a GNN. We use GCN [24] to implement the policy network. In GCN, the nodes are assigned to an initial feature matrix  $H^{(0)} \in \mathbb{R}^{N \times F}$ , where  $N$  is the number of nodes and  $F$  is the feature dimension size. Here we use the initial state of graph  $H^{(0)} = S^t$  as the initial input feature. The layer-wise propagation rule updates the node representations using the representations of its neighbors in the graph in the  $(l+1)$ -th layer, yielding the feature matrix:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W), \quad (5)$$

where  $\tilde{A} \in \mathbb{R}^{N \times N}$  is the adjacency matrix with self-connections ( $A+I$ ),  $\tilde{D}$  is the degree matrix of  $\tilde{A}$ ,  $W \in \mathbb{R}^{N \times F}$  is the weight matrix and  $\sigma(\cdot)$  denotes an activation function (we use ReLU in this paper). We apply a linear layer to map the final output to a probability score indicating whether this node should be queried:

$$\pi(\cdot|S^t) = \text{Softmax}(WH^{(l)} + b). \quad (6)$$

**Graph Coarsening:** The computational cost will grow exponentially as the number of GCN layers increases. In addition, as the search space covers all the candidate nodes for annotation, the large number of discrete actions makes reinforcement learning methods difficult to apply for large graphs. Thus, we introduce the graph coarsening strategy SAGPool [28] into the GCN policy network in Eq. (5) to distinguish between the nodes that should be dropped and the nodes that should be retained, which will reduce the running time and shrink the action space at the same time.

The self-attention score matrix  $Z \in \mathbb{R}^{N \times 1}$  is calculated as follows:

$$Z^{(l)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} \Theta), \quad (7)$$

where  $\Theta \in \mathbb{R}^{N \times 1}$  is the parameter matrix to be learned. With the attention score matrix  $Z$ , we can select the top  $k$  percent nodes to keep in each layer, yielding a list of top  $[kN]$  nodes' indices:

$$\text{idx} = \text{top}(Z, [kN]). \quad (8)$$

The output feature matrix and the corresponding adjacency matrix of each layer are calculated as:

$$H^{(l+1)} = H_{\text{idx},:}^{(l)} \odot Z_{\text{idx},:}^{(l)}, A = A_{\text{idx}, \text{idx}}, \quad (9)$$

where  $id_{x,:}$  represents the row-wise (i.e. node-wise) index notation,  $\odot$  is the broadcasted elementwise product, and  $id_{x,id_x}$  represents the row-wise and col-wise index notation.

**3.2.3 Robust Classification.** The GCN fraud detector can classify both labeled and unlabeled nodes. On the top of the policy network, we apply a linear layer, taking the final output embedding  $H^{(L)}$  as input. The output of the fraud detector is the probability of a node being positive.

The goal of the fraud detector is to determine whether a node is positive (abusive buyer) or not (benign buyer). On the shopping website, the positive class makes up only a very small portion (<5%). This data imbalanced issue causes two learning problems: (1) the easy negative samples do not contain much information to facilitate the training; (2) the easy negative samples may degenerate the model. To efficiently train on all samples, we employ focal loss [33]. Denote positive nodes as  $v^+ \sim p_{\mathbb{R}^+}(v)$ , and negative nodes  $v^- \sim p_{\mathbb{R}^-}(v)$ , where  $\mathbb{R}^+$  and  $\mathbb{R}^-$  represent the positive samples' and negative samples' spaces respectively. The loss function is denoted as follows:

$$\mathcal{J}_c(\theta_c) = -\mathbb{E}_{v^+ \sim p_{\mathbb{R}^+}} [\alpha(1 - f(v; \theta_c))^\gamma \log f(v; \theta_c)] \\ - \mathbb{E}_{v^- \sim p_{\mathbb{R}^-}} [(1 - \alpha)f(v; \theta_c)^\gamma \log(1 - f(v; \theta_c))]$$

where  $\gamma$  is a focusing parameter, which focuses more on hard and easily misclassified examples, and  $\alpha$  is the weight assigned to the rare class.  $\gamma = 2$  and  $\alpha = 0.25$  work best based on the rule of thumb [33].

As for the multi-class scenario in general, we exclude  $\alpha$  as it is not applicable for multiple classes. We still set  $\gamma$  as 2 based on rule of thumb. The multi-class focal loss is calculated as follows:

$$\mathcal{J}_c(\theta_c) = -(1 - f(v; \theta_c))^\gamma \log f(v; \theta_c) \quad (10)$$

**3.2.4 Training and Inference.** For training the classifier, we minimize the focal loss  $\mathcal{J}_c$  in Eq. (10). The objective function of the policy network is:

$$\mathcal{J}_p(\theta_p) = \mathbb{E}_{\pi(V_{\text{query}}^B; \theta_p)} [R(V_{\text{query}}^B)], \quad (11)$$

where  $B$  is the query budget, and  $R$  is the reward on graph  $G$ . We use a classic policy gradient method REINFORCE [48] to train the policy network  $\pi$ .

In order to train the policy network  $\pi(\cdot; \theta_p)$  parametered by  $\theta_p$ , we alternately update  $\theta_c$  by optimizing the focal loss  $\mathcal{J}_c$  on the training data queried by policy  $\pi(\cdot; \hat{\theta}_p)$ , and update  $\theta_p$  by maximizing the sum of expected rewards obtained from the classifier  $f(\cdot; \hat{\theta}_c)$  on the validation set:

$$\theta_p^* = \operatorname{argmax}_{\theta_p} \mathcal{J}_p(\theta_p), \quad \theta_c^* = \operatorname{argmin}_{\theta_c} \mathcal{J}_c(\theta_c). \quad (12)$$

The training process is divided into two stages. In the first stage, we train the classifier  $f(\cdot; \theta_c)$  to minimize the loss function  $\mathcal{J}_c(\theta_c)$ , while actively querying the unlabeled nodes. When the query budget is used up, we train the classifier  $f$  until convergence. In the second stage, we evaluate the trained classifier  $f$  on the validation set to get the reward signal and use that to update  $\theta_p$  together with the policy gradient. The detailed training steps are summarized in Algorithm 1.

## 4 EXPERIMENTS

In this section, we compare the performance of ALLIE with state-of-the-art AL methods on graphs. We aim to answer the following evaluation questions (EQ):

### Algorithm 1: ALLIE for AL on Graphs.

---

**Input:** Graph  $G$ , validation set  $V_{\text{valid}}$  and corresponding label set  $Y_{\text{valid}}$ , initial query set  $V_{\text{query}}^0$ , query budget  $B$  and training epochs  $N$

**Output:** Well-trained node classifier  $f$  and AL policy  $\pi$

```

1 for  $e = 1, \dots, N$  do
2   for  $t = 1, \dots, B$  do
3     Update the graph state  $S_G^t$ ;
4     Use policy  $\pi$  to sample a node based on
        $S_G^t$  for annotation, and add it to the query set  $V_{\text{query}}^t$ ;
5     Minimize the detection loss  $\mathcal{J}_c(\theta_c)$ 
       in Eq. (10) with the updated  $V_{\text{query}}^t$  for one epoch;
6   end
7   while not converged do
8     Minimize
       the detection loss  $\mathcal{J}_c(\theta_c)$  in Eq. (10) with  $V_{\text{query}}^B$ ;
9   end
10  Evaluate classifier  $f$  on the validation set  $V_{\text{valid}}$ 
    and  $Y_{\text{valid}}$  to get the reward signal  $R(V_{\text{query}}^B)$  in Eq. (4);
11  Use the sum of expected
    rewards to learn the optimal policy  $\pi^*$  in Eq. (11);
12 end

```

---

**Table 1: Statistics of citation graph datasets.**

	Cora	Citeseer	Pubmed
# nodes	2,485	2,110	19,717
# edges	5,068	3,668	44,338
# classes	7	6	3

- **EQ1:** Is ALLIE able to improve the node classification performance on both benchmark dataset and real-world e-commerce dataset?
- **EQ2:** How effective are graph coarsening, focal loss, and reward function adaptation method in ALLIE?
- **EQ3:** How robust is ALLIE with respect to its hyperparameter values?

To this end, we introduce the datasets used and baselines, followed by experiments to answer these questions.

### 4.1 Experimental Setting

**4.1.1 Datasets.** We use several benchmark citation graph datasets (Cora, Citeseer, Pubmed [40]). The statistics of the citation graph datasets are presented in Table 1. We also use datasets created from sampled, anonymized logs from an e-commerce website. We construct a graph consisting of sellers, buyers, reviews and products. Table 2 shows the approximate numbers of the nodes and edges that we sampled. This dataset is heavily sampled, and is not reflective of production traffic. We merely use it here to highlight the utility of ALLIE. We randomly initialized the attributes of each node when training the graph neural network. Sampling is done by randomly picking 10K buyers, performing Breadth First Search (BFS), and add the additional nodes to our dataset. A similar sampling method is used previously in [1].

**Table 2: Summary of the e-commerce dataset. The dataset is heavily subsampled, and is used to show the efficacy of ALLIE on a real world use case.**

Data property	Value
Node types	{buyer, seller, review, product}
# nodes (post sampling)	~50K
% abusive buyers	5.2
# edges (post sampling)	~61K
% abusive reviews	1.7

**4.1.2 Implementation details.** We implement ALLIE with PyTorch. We vary the learning rate in  $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$  and found learning rate  $10^{-2}$  worked best. For baselines, we follow the exact network architecture detailed in the corresponding works. We outline the baselines used in Section 4.1.3. For all models, we use Adam [23] with 100 epochs. On citation graph datasets, we use 5 samples from each class to construct the initial training set, and set the query budget as 20 for each class. On the e-commerce dataset, we use 200 samples from each class to construct the initial training set, and set query budget as 250 for each class. We repeat all experiments five times and report the averages and the standard deviations of the metrics.

**4.1.3 Baselines.** We compare ALLIE with the following representative and state-of-the-art AL on graph algorithms:

- Random: Random selects several candidate nodes uniformly at random to annotate in each epoch, and uses GNN to re-train the classifier using these nodes.
- AGE<sup>3</sup> [7]: AGE uses the weighted sum of entropy, density and centrality to find the best candidate(s) from all unlabeled nodes.
- FeatProp<sup>4</sup> [49]: FeatProp uses cluster centers as selected candidates through k-medoids clustering.
- GPA<sup>5</sup> [21]: GPA formalizes AL on graphs as an MDP (Markov Decision Process) and learns the optimal query strategy with reinforcement learning.
- MetAL<sup>6</sup> [35]: MetAL uses an AL algorithm that selects a set of unlabeled instances based on an informative metric, gets their labels, and updates the labeled dataset.

We choose the above methods that based on the following aspects: (1) only heuristic metrics, such as AGE and FeatProp; (2) heuristic metrics and reinforcement learning, such as GPA; and (3) heuristic metrics and meta-learning, such as MetAL. This allows us to compare ALLIE to multiple kinds of methods.

**4.1.4 Evaluation measures.** We use Micro F1 and Macro F1 to evaluate the performance of all methods on the citation datasets. We report per-class precision, recall and F1 score on the e-commerce dataset. The latter dataset lends itself to a highly imbalanced classification problem.

## 4.2 EQ1: Performance on Public Datasets

To answer EQ1, we first compare ALLIE with the state-of-the-art AL algorithms introduced in Section 4.1.3 on benchmark graph datasets. We conduct experiments in both balanced and imbalanced settings.

<sup>3</sup><https://github.com/vwz/AGE>

<sup>4</sup>[https://github.com/CrickWu/active\\_graph](https://github.com/CrickWu/active_graph)

<sup>5</sup><https://github.com/ShengdingHu/GraphPolicyNetworkActiveLearning>

<sup>6</sup><https://github.com/Kaushalya/metall>

**4.2.1 Balanced Setting.** We use the original datasets as-is to conduct the experiments in this setting. The problem is that of multi-class node classification. Table 3 summarizes the node classification performance of all competing methods (reporting the average of 5 runs). From the table, we can make the following observations:

- For the metric-based methods AGE and FeatProp, the performance is unsatisfactory. Though they use several heuristic metrics to capture the representativeness of nodes, they do not leverage node interactions to better measure node informativeness.
- The meta-learning based method MetAL performs better than metric-based methods, demonstrating the effectiveness of using the classifier’s performance as feedback. MetAL is inferior to ALLIE. We hypothesize that MetAL needs a moderate- to large-sized initial training set to learn accurate model weights.
- ALLIE outperforms other methods in terms of Macro F1 and Micro F1 on three datasets. This shows that ALLIE effectively leverages both graph information as well as feedback.

**4.2.2 Imbalanced Setting.** In this setting, we manually adapt the datasets into binary classes to make the data distribution imbalanced. Following [42], we treat the smallest class in Cora, Citeseer and PubMed as the positive class and the rest as the negative class. The positive class ratios are 7%, 8% and 21% respectively. Results are shown in Table 4 (reporting the average of 5 runs). From the table, we can see that:

- All the model performances on Macro-F1 and Micro-F1 degrade. This reinforces our hypothesis that when the data distribution becomes imbalanced, the classifier tends to predict most samples as belonging to the majority class.
- ALLIE outperforms the other models. It demonstrates the effectiveness of the balancing strategies, including the imbalance-aware reinforcement learning framework and focal loss.

## 4.3 EQ1: Performance on e-commerce dataset

In order to test EQ1 in a real-world setting with large-scale imbalanced graphs, we define two node classification tasks on the e-commerce dataset. The tasks are detecting abusive users behavior and abusive reviews. Both these tasks are important in e-commerce to ensure high customer trust. Because the dataset is proprietary, we report relative changes in each metric with respect to a baseline.

**4.3.1 Classification on Buyers.** Here we investigate the performance of ALLIE when distinguishing abusive buyers from benign buyers, and make the following observations.

- ALLIE outperforms the other models. It again shows the importance of applying reinforcement learning to query nodes from the unlabeled data, which directly optimizes the performance of the GNN classifier.
- It is worthwhile to point out that ALLIE has a higher performance improvement with the abusive buyer class compared with the benign buyer class. This indicates the effectiveness of adapting the reward function to better capture the minority class (abusive buyer) and using focal loss to down-weight the well-classified samples (benign buyers far from the classification boundary).

**4.3.2 Classification on Reviews.** We summarize the observations of ALLIE in classifying abusive reviews and benign reviews.

**Table 3: Node classification performance (Macro F1 and Micro F1  $\pm$ Std) on the balanced setting of citation graph datasets. The best and 2nd best are noted in bold font and underlined, respectively.**

	Cora		Citeseer		PubMed	
	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1
Random	0.6819 $\pm$ 0.041	0.7031 $\pm$ 0.071	0.5438 $\pm$ 0.093	0.5762 $\pm$ 0.079	0.7033 $\pm$ 0.051	0.7296 $\pm$ 0.118
AGE	0.7322 $\pm$ 0.046	0.7725 $\pm$ 0.031	0.6152 $\pm$ 0.013	0.6722 $\pm$ 0.098	0.7735 $\pm$ 0.019	0.7737 $\pm$ 0.019
FeatProp	<u>0.7826<math>\pm</math>0.018</u>	0.7645 $\pm$ 0.009	0.6417 $\pm$ 0.041	0.6097 $\pm$ 0.087	0.7392 $\pm$ 0.068	0.7321 $\pm$ 0.124
GPA	0.7677 $\pm$ 0.063	<u>0.8105<math>\pm</math>0.012</u>	<u>0.6635<math>\pm</math>0.039</u>	<u>0.7130<math>\pm</math>0.102</u>	<u>0.7912<math>\pm</math>0.091</u>	0.7996 $\pm$ 0.043
MetAL	0.7455 $\pm$ 0.012	0.7985 $\pm$ 0.023	0.6184 $\pm$ 0.016	0.7018 $\pm$ 0.065	0.7711 $\pm$ 0.086	0.7764 $\pm$ 0.075
ALLIE	<b>0.8025<math>\pm</math>0.071</b>	<b>0.8242<math>\pm</math>0.027</b>	<b>0.6838<math>\pm</math>0.020</b>	<b>0.7425<math>\pm</math>0.082</b>	<b>0.8228<math>\pm</math>0.038</b>	<b>0.8376<math>\pm</math>0.049</b>

**Table 4: Node classification performance (Macro F1 and Micro F1  $\pm$ Std) on the imbalanced setting of citation graph datasets.**

	Cora		Citeseer		PubMed	
	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1
Random	0.1781 $\pm$ 0.048	0.5645 $\pm$ 0.042	0.2755 $\pm$ 0.063	0.5695 $\pm$ 0.072	0.1735 $\pm$ 0.036	0.5691 $\pm$ 0.193
AGE	0.1631 $\pm$ 0.084	0.6424 $\pm$ 0.038	0.3716 $\pm$ 0.093	0.6293 $\pm$ 0.049	0.1871 $\pm$ 0.048	0.6266 $\pm$ 0.051
FeatProp	0.3582 $\pm$ 0.071	0.6427 $\pm$ 0.193	0.4295 $\pm$ 0.104	0.6817 $\pm$ 0.094	0.2594 $\pm$ 0.088	0.6692 $\pm$ 0.038
GPA	<u>0.4892<math>\pm</math>0.042</u>	<u>0.7384<math>\pm</math>0.098</u>	0.4239 $\pm$ 0.078	<u>0.7084<math>\pm</math>0.053</u>	<u>0.3813<math>\pm</math>0.085</u>	0.7442 $\pm$ 0.158
MetAL	0.4583 $\pm$ 0.035	0.6979 $\pm$ 0.025	<u>0.4328<math>\pm</math>0.062</u>	0.7035 $\pm$ 0.067	0.3602 $\pm$ 0.105	<b>0.7791<math>\pm</math>0.084</b>
ALLIE	<b>0.5391<math>\pm</math>0.027</b>	<b>0.7692<math>\pm</math>0.015</b>	<b>0.4894<math>\pm</math>0.041</b>	<b>0.7684<math>\pm</math>0.074</b>	<b>0.4391<math>\pm</math>0.037</b>	<u>0.7694<math>\pm</math>0.056</u>

**Table 5: Buyer classification performance relative change (Precision, Recall and F1  $\pm$ Std) on the e-commerce dataset.**

	Benign Buyer			Abusive Buyer		
	Precision	Recall	F1	Precision	Recall	F1
Random	-	-	-	-	-	-
AGE	+0.1650 $\pm$ 0.013	+0.0640 $\pm$ 0.053	+0.1408 $\pm$ 0.028	+0.1119 $\pm$ 0.025	+0.0337 $\pm$ 0.024	+0.0685 $\pm$ 0.017
FeatProp	+0.1547 $\pm$ 0.042	+0.0522 $\pm$ 0.041	+0.0876 $\pm$ 0.031	+0.0987 $\pm$ 0.016	+0.0028 $\pm$ 0.036	+0.0388 $\pm$ 0.028
GPA	<u>+0.1872<math>\pm</math>0.042</u>	<b>+0.0950<math>\pm</math>0.032</b>	<u>+0.1647<math>\pm</math>0.035</u>	<u>+0.1519<math>\pm</math>0.013</u>	+0.0435 $\pm$ 0.042	<u>+0.0891<math>\pm</math>0.042</u>
MetAL	+0.0546 $\pm$ 0.031	+0.0401 $\pm$ 0.018	+0.0743 $\pm$ 0.039	+0.0941 $\pm$ 0.031	+0.0482 $\pm$ 0.013	+0.0771 $\pm$ 0.031
ALLIE	<b>+0.1936<math>\pm</math>0.025</b>	<u>+0.0834<math>\pm</math>0.018</u>	<b>+0.1671<math>\pm</math>0.024</b>	<b>+0.2024<math>\pm</math>0.013</b>	<b>+0.0521<math>\pm</math>0.041</b>	<b>+0.1184<math>\pm</math>0.027</b>

**Table 6: Review classification performance relative change (Precision, Recall and F1  $\pm$ Std) on the e-commerce dataset.**

	Benign Review			Abusive Review		
	Precision	Recall	F1	Precision	Recall	F1
Random	-	-	-	-	-	-
AGE	+0.1864 $\pm$ 0.013	+0.1968 $\pm$ 0.019	+0.2037 $\pm$ 0.173	+0.1544 $\pm$ 0.012	+0.1299 $\pm$ 0.695	+0.1187 $\pm$ 0.031
FeatProp	+0.1432 $\pm$ 0.022	+0.0612 $\pm$ 0.017	+0.1286 $\pm$ 0.022	+0.1858 $\pm$ 0.023	+0.1259 $\pm$ 0.031	+0.1409 $\pm$ 0.032
GPA	<b>+0.3934<math>\pm</math>0.032</b>	<u>+0.3167<math>\pm</math>0.009</u>	<u>+0.3885<math>\pm</math>0.025</u>	<u>+0.4515<math>\pm</math>0.013</u>	+0.3815 $\pm$ 0.058	<u>+0.3979<math>\pm</math>0.023</u>
MetAL	+0.1168 $\pm$ 0.017	+0.1968 $\pm$ 0.041	+0.2054 $\pm$ 0.026	+0.4139 $\pm$ 0.052	<u>+0.2758<math>\pm</math>0.032</u>	+0.3800 $\pm$ 0.077
ALLIE	<u>+0.3864<math>\pm</math>0.013</u>	<b>+0.3968<math>\pm</math>0.019</b>	<b>+0.4207<math>\pm</math>0.173</b>	<b>+0.4960<math>\pm</math>0.056</b>	<b>+0.4409<math>\pm</math>0.013</b>	<b>+0.4375<math>\pm</math>0.042</b>

- ALLIE still outperforms baselines, especially in the abusive review class, which indicates that ALLIE is suitable for imbalanced graphs.
- The scores of metric-based methods, AGE and FeatProp, on review classification task is generally much lower than their results on the buyer classification task. This indicates that their performance worsens when the data is more imbalanced.

#### 4.4 EQ2: Ablation Study

In order to answer EQ2, we explore each component of ALLIE separately. We first study the influence of different reward function designs. Then we examine the influence of graph coarsening and balancing strategies.

**4.4.1 Effect of reward functions.** To explore the impact of various reward function designs in Section 3.2.1, we consider several variants

of ALLIE that use different reward functions: weighted reward, Micro-F1 and Macro-F1. We term these methods ALLIE<sub>weighted reward</sub>, ALLIE<sub>Micro-F1</sub> and ALLIE<sub>Macro-F1</sub> respectively.

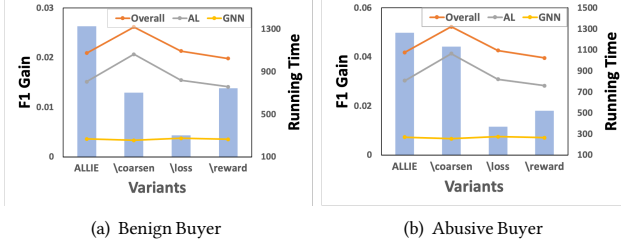
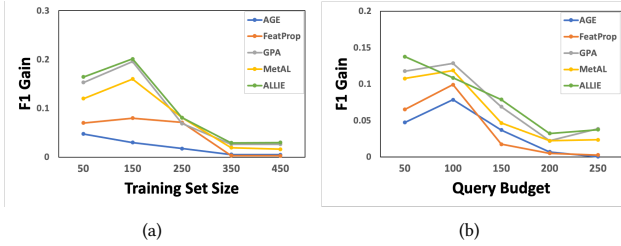
Table 7 summarizes the results on the imbalanced setting of the public datasets in Section 4.2.2. We find that ALLIE<sub>Macro-F1</sub> is superior than ALLIE<sub>weighted reward</sub> and ALLIE<sub>Micro-F1</sub>. This verifies that incorporating sample balancing into the reward function design can address the class imbalance issue.

**4.4.2 Effect of graph coarsening and balancing strategies.** We define several variants of ALLIE to study the effects of graph coarsening, focal loss and reward function adaptation:

- **\coarsen:** This is a variant of ALLIE which does not integrate the graph coarsening module.

**Table 7: Effect of reward functions (Macro F1 and Micro F1  $\pm$ Std) on imbalanced citation graph datasets.**

Reward Function	Cora		Citeseer		PubMed	
	Macro F1	Micro F1	Macro F1	Micro F1	Macro F1	Micro F1
Weighted Reward	0.5274 $\pm$ 0.034	0.7524 $\pm$ 0.080	0.4789 $\pm$ 0.016	0.7639 $\pm$ 0.011	0.4323 $\pm$ 0.141	0.7543 $\pm$ 0.187
Micro-F1	0.5245 $\pm$ 0.138	0.7534 $\pm$ 0.046	0.4716 $\pm$ 0.022	0.7563 $\pm$ 0.048	0.4287 $\pm$ 0.069	0.7685 $\pm$ 0.098
Macro-F1	<b>0.5391<math>\pm</math>0.027</b>	<b>0.7692<math>\pm</math>0.015</b>	<b>0.4894<math>\pm</math>0.041</b>	<b>0.7684<math>\pm</math>0.074</b>	<b>0.4391<math>\pm</math>0.037</b>	<b>0.7694<math>\pm</math>0.056</b>

**Figure 2: Performance gain comparison over variants on the buyer classification task. The background histograms indicate the F1 gain over AGE of each variant. The lines indicate the running time (in seconds).****Figure 3: Performance gain over Random with different training set sizes and query budgets.**

- **\loss**: This is a variant which does not specifically down-weight the well-classified samples. The focal loss function is replaced by the standard cross entropy loss function.
- **\reward**: This is a variant which uses Micro-F1 as its reward metric.

We also record the running time (in seconds) of each variant and summarize the experimental results in Figure 2. We have the following findings:

- Removing the graph coarsening module slightly degrades the model’s performance, as SAGPool has an attention mechanism that can improve the performance of GNN. Furthermore, this variant takes the longest time compared to all methods in the AL part.
- Changing the reward function to Macro-F1 is more effective for improving the F1 of abusive buyers, as Micro-F1 favors large classes (benign buyer) while Macro-F1 averages F1 per class.
- When we do not use focal loss, the false positive rate increases, which results in lower Precision and F1. This variant performs the worst, which indicates that the focal loss function contributes the most in ALLIE.

## 4.5 EQ3: Hyperparameter Sensitivity Analysis

We vary the initial training set sizes and query budget to test how ALLIE varies along these dimensions. The buyer classification task on the e-commerce dataset is used as the example task here.

**4.5.1 Performance under different initial training set sizes.** We start ALLIE with  $\{50, 150, 250, 350, 450\}$  initial training samples. We run each method five times and report the averaged F1 score in Figure 3(a). From the results, we see that ALLIE outperforms all the baselines regardless of the initial training set size. Importantly, when the training set sizes are small, ALLIE significantly outperforms the baseline methods.

**4.5.2 Performance under different query budgets.** We train ALLIE with  $\{50, 100, 150, 200, 250\}$  budgets, and then evaluate the learned model. All the methods are tested using the same initial training set. We run each method five times and report the averaged F1 score in Figure 3(b). Again, we see that ALLIE outperforms baselines significantly when the budgets are small.

The above experiments show that ALLIE is indeed well suited to the problem of active learning on graphs when the labeled data is highly imbalanced.

## 5 CONCLUSION

In this paper, we propose ALLIE, a novel active learning framework designed for large-scale imbalanced graphs. ALLIE leverages a graph policy network to query the candidate nodes to label by optimizing the long-term performance of the GNN classifier. With two balancing strategies, ALLIE can better deal with an imbalanced data distribution compared with several state-of-the-art methods. Moreover, ALLIE has a graph coarsening module which makes it scalable on large-scale applications. Experiments on three benchmark datasets and a real-world shopping website dataset demonstrate the strong performance of ALLIE.

## ACKNOWLEDGMENTS

This work was in part supported by NSF awards #1940076, #1909702, and #1934782.

## REFERENCES

- [1] Bijaya Adhikari, Liangyue Li, Nikhil Rao, and Karthik Subbian. 2021. Finding Needles in Heterogeneous Haystacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15232–15239.
- [2] Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and S Yu Philip. 2014. Active learning: A survey. In *Data Classification: Algorithms and Applications*. CRC Press, 571–605.
- [3] Umang Aggarwal, Adrian Popescu, and Céline Hudelot. 2020. Active learning for imbalanced datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1428–1437.
- [4] Philip Bachman, Alessandro Sordani, and Adam Trischler. 2017. Learning algorithms for active learning. In *International conference on machine learning*. PMLR, 301–310.

- [5] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34, 4 (2017), 18–42.
- [6] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks* 106 (2018), 249–259.
- [7] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2017. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085* (2017).
- [8] Sina Yamac Caliskan and Paulo Tabuada. 2014. Towards Kron reduction of generalized electrical networks. *Automatica* 50, 10 (2014), 2586–2590.
- [9] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.
- [10] Xia Chen, Guoxian Yu, Jun Wang, Carlotta Domeniconi, Zhao Li, and Xiangliang Zhang. 2019. Activehne: Active heterogeneous network embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2123–2129.
- [11] Limeng Cui, Siddharth Biswal, Lucas M Glass, Greg Lever, Jimeng Sun, and Cao Xiao. 2020. Conan: Complementary pattern augmentation for rare disease detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 614–621.
- [12] G Dasarathy, N Rao, and R Baraniuk. 2017. On computational and statistical tradeoffs in matrix completion with graph information. In *Signal Processing with Adaptive Sparse Structured Representations Workshop SPARS*.
- [13] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016), 3844–3852.
- [14] Liat Ein Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active learning for BERT: An empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7949–7962.
- [15] Charles Elkan. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, Vol. 17. Lawrence Erlbaum Associates Ltd, 973–978.
- [16] Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. 2018. Active discriminative network representation learning. In *IJCAI International Joint Conference on Artificial Intelligence*.
- [17] Zhongkai Hao, Chengqiang Lu, Zhenya Huang, Hao Wang, Zheyuan Hu, Qi Liu, Enhong Chen, and Cheekong Lee. 2020. ASGN: An active semi-supervised graph neural network for molecular property prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 731–752.
- [18] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 1322–1328.
- [19] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21, 9 (2009), 1263–1284.
- [20] B Hendrickson and R Leland. 1995. A Multi-Level Algorithm For Partitioning Graphs. In *Supercomputing'95: Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*. IEEE, 28–28.
- [21] Shengding Hu, Zheng Xiong, Meng Qu, Xingdi Yuan, Marc-Alexandre Côté, Zhiyuan Liu, and Jian Tang. 2020. Graph Policy Network for Transferable Active Learning on Graphs. In *NeurIPS*.
- [22] Vassilis Kalofolias and Nathanaël Perraudin. 2018. Large Scale Graph Learning From Smooth Signals. In *International Conference on Learning Representations*.
- [23] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- [24] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=SJU4ayYgl>
- [25] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. 2015. Introducing geometry in active learning for image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2974–2982.
- [26] Ioannis Koutis, Gary L Miller, and Richard Peng. 2011. A nearly-m log n time solver for sdd linear systems. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. IEEE, 590–598.
- [27] Stephane Lafon and Ann B Lee. 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE transactions on pattern analysis and machine intelligence* 28, 9 (2006), 1393–1403.
- [28] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*. PMLR, 3734–3743.
- [29] Ruirui Li, Xian Wu, Xian Wu, and Wei Wang. 2020. Few-shot learning for new user recommendation in location-based social networks. In *Proceedings of The Web Conference 2020*. 2472–2478.
- [30] Xin Li and Yuhong Guo. 2013. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 859–866.
- [31] Yayong Li, Jie Yin, and Ling Chen. 2020. SEAL: Semisupervised Adversarial Active Learning on Attributed Graphs. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
- [32] Enlu Lin, Qiong Chen, and Xiaoming Qi. 2020. Deep reinforcement learning for imbalanced classification. *Applied Intelligence* 50, 8 (2020), 2488–2502.
- [33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.
- [34] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference 2021*. 3168–3177.
- [35] Kaushalya Madhawa and Tsuyoshi Murata. 2020. MetaL: Active Semi-Supervised Learning on Graphs via Meta-Learning. In *Asian Conference on Machine Learning*. PMLR, 561–576.
- [36] Federico Monti, Michael M Bronstein, and Xavier Bresson. 2017. Geometric matrix completion with recurrent multi-graph neural networks. *arXiv preprint arXiv:1704.06803* (2017).
- [37] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. 2020. Imbalance problems in object detection: A review. *IEEE transactions on pattern analysis and machine intelligence* (2020).
- [38] Nikhil Rao, Hsiang-Fu Yu, Pradeep Ravikumar, and Inderjit S Dhillon. 2015. Collaborative Filtering with Graph Information: Consistency and Scalable Methods. In *NIPS*, Vol. 2. Citeseer, 7.
- [39] Chris Seifert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2009. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 40, 1 (2009), 185–197.
- [40] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.
- [41] Claude Elwood Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review* 5, 1 (2001), 3–55.
- [42] Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. 2020. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*.
- [43] Daniel A Spielman and Shang-Hua Teng. 2011. Spectral sparsification of graphs. *SIAM J. Comput.* 40, 4 (2011), 981–1025.
- [44] Sriram Srinivasan, Nikhil S Rao, Karthik Subbian, and Lise Getoor. 2019. Identifying facet mismatches in search via micrographs. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1663–1672.
- [45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
- [46] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 598–607.
- [47] Lu Wang, Yanghua Xiao, Bin Shao, and Haixun Wang. 2014. How to partition a billion-node graph. In *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 568–579.
- [48] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.
- [49] Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. 2019. Active learning for graph neural networks via node feature propagation. *arXiv preprint arXiv:1910.07567* (2019).
- [50] Yang Xu, Yu Hong, Huibin Ruan, Jianmin Yao, Min Zhang, and Guodong Zhou. 2018. Using active learning to expand training data for implicit discourse relation recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 725–731.
- [51] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 4805–4815.
- [52] Ruilong Zhang, Lei Li, Yuhong Zhang, and Chenyang Bu. 2018. Imbalanced networked multi-label classification with active learning. In *2018 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 290–297.
- [53] Yifan Zhang, Peilin Zhao, Jiezhong Cao, Wenye Ma, Junzhou Huang, Qingyao Wu, and Minghui Tan. 2018. Online adaptive asymmetric active learning for budgeted imbalanced data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2768–2777.
- [54] Yanqiao Zhu, Weizhi Xu, Qiang Liu, and Shu Wu. 2020. When Contrastive Learning Meets Active Learning: A Novel Graph Active Learning Paradigm with Self-Supervision. *arXiv preprint arXiv:2010.16091* (2020).