ABSTRACT

Different from deep neural networks for non-graph data classification, graph neural networks (GNNs) leverage the information exchange between nodes (or samples) when representing nodes. The category distribution shows an imbalance or even a highly-skewed trend on nearly all existing benchmark GNN data sets. The imbalanced distribution will cause misclassification of nodes in the minority classes, and even cause the classification performance on the entire data set to decrease. This study explores the effects of the imbalance problem on the performances of GNNs and proposes new methodologies to solve it. First, a node-level index, namely, the label difference index \( \text{LDI} \), is defined to quantitatively analyze the relationship between imbalance and misclassification. The less samples in a class, the higher the value of its average \( \text{LDI} \); the higher the \( \text{LDI} \) of a sample, the more likely the sample will be misclassified. We define a new loss and propose four new methods based on \( \text{LDI} \). Experimental results indicate that the classification accuracies of the three among our proposed four new methods are better in both transductive and inductive settings. The \( \text{LDI} \) can be applied to other GNNs.

1 INTRODUCTION

As an increasing number of applications involve graph data, researchers have designed graph neural networks (GNNs) to process graph data. GNNs generally use the connections between nodes to exchange information in neighborhoods to obtain a better representation for each node. Therefore, a unique problem faced by GNNs is misclassification caused by neighbor nodes. Nodes in minority classes are more likely to be neighbored by those in majority classes, resulting in low accuracy.

Real-world data usually conforms to an imbalanced even a highly-skewed distribution, in which majority classes occupy most of the data proportion. Deep neural networks also suffer when the training data are highly imbalanced [2, 3, 12].

This study investigates the imbalance problem for GNNs, and reveals more details for the relationship between imbalance and misclassification on graph data. Furthermore, based on the state-of-the-art algorithms for dealing with the imbalance of non-graph data, new methodologies are proposed. First, we define the label difference index \( \text{LDI} \) to measure the likelihood of a node being misclassified based on the category distribution of its neighborhood. The larger the index, the more likely the node can be misclassified. The \( \text{LDI} \) of a node is determined by the category distribution of its neighbor nodes. Nodes neighbored by heterophilic nodes\(^1\) are more likely to be with larger \( \text{LDIs} \), and easy to be negatively affected, resulting in misclassification. In addition, \( \text{LDI} \) is also affected by the global category distribution of the entire data set, and statistics show that the average \( \text{LDIs} \) of the nodes in minority classes are usually large. Thus, the relationship between the imbalance and misclassification is established. Second, we define a new loss and propose four new methods, namely, improved focal loss (iFL), Graph Re-sampling (GRS), Graph Re-weighting (GRW), Graph Metric Learning (GML), and Graph Bilateral-branch Network (GBBN). Experiments on several graph benchmark data sets show that except for GML, the new methods are better than the original GNNs. Imbalanced sampling with \( \text{LDI} \) can help to further improve performance.

Our contributions are summarized as follows:

(1) A node-level index (i.e., \( \text{LDI} \)) is defined to characterize the neighborhood of a node (or a sample) in a graph. On the basis of \( \text{LDI} \), the relationship between imbalance and misclassification is analyzed and useful observations are obtained.

(2) A new loss and four new methods are proposed based on the index \( \text{LDI} \). Extensive experiments on benchmark data sets indicate that the proposed methodology achieves better results than existing imbalance learning methods on graphs. Further, several classical GNNs are improved by utilizing \( \text{LDI} \).

2 RELATED WORK

Classical GNNs include GCN [7], SGC [17], GAT [15], and so on. Kipf et al. [7] expanded the traditional convolution neural network (CNN) on high-dimensional graph data to obtain GCN. GCN iteratively updated each node’s representation through the message exchange with their neighbor nodes. Wu et al. [17] reduced the complexity of GCN by repeatedly eliminating the nonlinearity between GCN layers and folding the resulting function into a linear transformation to obtain SGC. Veličković et al. [15] proposed GAT based on the attention mechanism to classify graph data.

---

\(^1\)Heterophilic nodes are the samples (or nodes) that belong to different categories [21].
In the research of graph, researchers usually define indices (e.g., degree and centrality) to characterize the properties of a graph. Likewise, a number of indices are also defined to measure the properties of the involved graphs in the research of GNNS. For example, graph homophily was proposed by Pei et al. [11] to characterize the degree that similar nodes connect together. Graph heterophily was proposed by Zhu et al. [22] to measure the graph homophily level. Two graph smoothness metrics, namely, feature smoothness and label smoothness, were proposed by Hou et al. [5] to help understand the use of graph information in GNNS. Almost all existing graph indices in GNNS are graph-level or category-level. These indices are only used to analyze graph characteristics and the learning performances. They are not involved in the training process. To our knowledge, there is no node-level index. This study will define a node-level index which can be directly used in both characteristics analyses and model training.

Machine learning has been increasingly applied in recent years. The imbalance has become a hot research topic. Kang et al. [6] divided the learning of a classification model into two steps. Zhou et al. [20] proposed a new learning model with two branches, both of which involve re-sampling. Zong et al. [23] proposed GNN-XML to overcome the imbalance problem in extreme multi-label text classification. Using re-weighting, Cui et al. [3] used a better weighting design to achieve a better imbalance classifier. Liu et al. [9] created some virtual samples around the minority classes samples to increase the number of minority classes samples. Liu et al. [10] transferred the visual knowledge of the majority to the minority classes by learning a set of dynamic element vectors. Lin et al. [8] proposed a new loss function, focal loss, to deal with the imbalance problem in object detection.

As to the imbalance of graph data, Min et al. [14] conducted a pilot study for this problem. They proposed DR-GCN, adopting a conditional adversarial training together with distribution alignment to learn node representations. They only focus on the transductive setting. Therefore, methods are compared on the transductive setting in Section 5.2.

3 EFFECT OF IMBALANCE FOR GNNS

3.1 Qualitative analysis.

We first analyze the interaction between the majority and the minority classes in an imbalanced data set as a non-graph classification task from the perspective of training loss. The loss can be expressed as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{majority}} + \mathcal{L}_{\text{minority}},$$  \hspace{1cm} (1)

where $\mathcal{L}_{\text{majority}}$ is the loss of the majority classes and $\mathcal{L}_{\text{minority}}$ is the loss of the minority classes. Because the number of samples in the majority classes (majority samples for brevity) is much greater than that of samples in the minority classes (minority samples for brevity), the loss is composed mainly of majority samples, and the gradients of model optimization are determined more by majority samples. As a result, both the feature representation and the decision layers of the training model are optimized towards a high classification accuracy of the majority classes in each training epoch, which may lead to the overfitting of the majority classes and ignores the minority classes while training. Ultimately, the performance of the overall model is damaged.

In graph classification tasks, because the loss function can still be expressed in the above form, the training for GNNS will also be affected by the imbalance, that is, the majority samples determine the gradients in training. However, because the majority and the minority nodes in the graph usually have direct edge links, in addition to the negative effects of the loss of coupling on the minority classes, the nodes in the majority classes may affect the feature representation of the nodes in the minority classes directly through edges. When node heterophily holds, this kind of effect between neighbor nodes is very likely to be negative.

Information exchange between nodes of the same category is beneficial, while information exchange between nodes of different categories is likely to be harmful in GNNS. Because there are fewer minority nodes, the proportion of nodes in minority classes with heterophily nodes in their neighborhoods may be relatively larger, which may have a greater negative effect. In other words, minority nodes are more likely to be misclassified.

To sum up, imbalance in non-graph data affects both the decision and the feature representation layers through the loss minimization; imbalance in graph data affects both the decision and the feature representation layers according to both the loss minimization and the direct information exchange between nodes. Therefore, imbalance in graph data can have a more serious impact on GNNS training.

3.2 Quantitative analysis.

Suppose a graph $G = (V, E)$ and let $x_i \in V$. The category distribution of the K-neighbor nodes of $x_i$ is $P_{N_i} = \{p_{i,1}, \cdots, p_{i,C}\}$, where $N_i$ denotes the K-neighborhood and $p_{i,c}$ represents the proportion of the $c$th category in $N_i$. We use 1-neighborhood to reduce the computation complexity in this study. Meanwhile, if $x_i$ belongs to the $c$th category, the distribution of itself is marked as a one-hot vector $P_L = \{0, \cdots, 0, 1, 0, \cdots, 0\}$, where the $c$th element is 1.

We define a node-level index, label difference index (LDI), on graph to better characterize the relationship between imbalance and misclassification.

**Definition 3.1.** Label Difference Index (LDI): Given a node $x_i$, the LDI of $x_i$ is

$$\text{LDI}_i = \frac{1}{\sqrt{2}} \|P_{N_i} - P_L\|_2.$$  \hspace{1cm} (2)

The range of $\text{LDI}_i$ is $[0, 1]$. The higher the LDI value, the larger the difference between the two distributions. If $x_i$ is an isolated node, then we define its LDI as the average LDI of nodes in the same category. We have the following proposition.

**Proposition 3.2.** Given a node $x_i$ belonging to the $c$th category. When the proportion of homophily nodes in its neighborhood is fixed (i.e., $p_{i,c}$ is fixed), the less categories of heterophily nodes, the larger the LDI.

The proof is shown in the appendix. Figure 1 shows an illustrative example. The colors denote the categories. Assume that $P_{R} = \{1, 0, 0, 0\}$. In Figure 1(a), $P_{N_5} = \{1, 0, 0, 0\}$, then $\text{LDI}_5 = 0$. In Figure 1(b), $P_{N_6} = \{\frac{3}{4}, \frac{1}{4}, 0, 0\}$, then $\text{LDI}_6 = \frac{1}{4}$. In Figure 1(c), $P_{N_7} = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$, then $\text{LDI}_7 = \frac{\sqrt{3}}{4}$. In Figure 1(d), $P_{N_8} = \{0, \frac{1}{2}, \frac{1}{3}, \frac{1}{3}\}$, $\text{LDI}_8 = \frac{1}{2}$.
we analyze the average accuracies of nodes in different 
intervals. That is, samples with large 
values in Figure 1 indicate that our 
set Citeseer [14]. As shown in Figure 3, samples in the majority 
categories, the relationship between 
and the classification accuracy decreases as the 
minority classes.

Figure 1: Examples of nodes with different neighborhoods. The 
values of node 5 in (a)-(f) are 0, $\frac{1}{2}$, $\frac{\sqrt{3}}{2}$, $\frac{\sqrt{6}}{2}$, 1, respectively.

then $\text{LDI}_5 = \frac{\sqrt{3}}{2}$. In Figure 1(f), $P_{N_5} = \{0, 1, 0, 0\}$, then $\text{LDI}_5 = 1$. In other words, 
node 5 in Figure 1(f) is the most easiest to be misclassified. The six 
LDI values in Figure 1 indicate that our LDI definition is reasonable and Proposition 3.2 holds.

LDI is a node-level index that can be easily applied to concrete 
GNN algorithms. Different from the homophily index mentioned in [11], LDI utilizes the distribution of neighbors, and is more 
finely-grained. Proposition 3.2 verifies this. The indices of node 5 in 
Figures 1 (d), (e), and (f) are the same (equal to 0) if the homophily 
index in [11] is used. However, LDI can distinguish these three 
cases better.

Based on LDI, we conduct analyses in terms of the differences 
among categories, the differences among samples within specific 
categories, the relationship between LDI and accuracies, and the relationship between LDI and misclassification under different numbers of layers (#layers).

Average LDIs of different categories. We calculate the aver-
age LDIs of different categories on five benchmark graph data sets, 
as shown in Figure 2. Although the trend of the category distribution 
(Figure 2 (up)) is not completely the opposite of the trend of 
the LDI distribution (Figure 2 (middle)) on each data set, the overall 
trend is that the LDIs of minority classes are large, indicating that 
the proportions of heterophily nodes around the nodes in the 
minority classes are relatively high. In Figure 2 (down), the categories 
with large average LDIs usually have low accuracies.

LDI distributions within specific categories. Within a cate-
gory, the LDIs of different samples are also distinct. We arbitrarily 
select a majority class and a minority class from the graph data 
set Citeseer [14]. As shown in Figure 3, samples in the majority 
class are more concentrated in the small LDI intervals. Contrarily, 
samples in the minority class not only concentrate in the large 
LDI intervals. That is, samples with large LDIs can be found in the 
majority classes and samples with small LDIs can be found in the 
minority classes.

Average accuracies of different LDI intervals. Furthermore, 
we analyze the average accuracies of nodes in different LDI inter-
vals. As shown in Figure 4, nodes concentrate in small LDI intervals, 
and the classification accuracy decreases as the LDI increases on different data sets. Nodes with large LDIs are more likely to 
exchange information with heterophily nodes in their neighborhoods, 
resulting in low accuracies. As shown in Figure 3, within a specific category, the average accuracies of different LDI intervals show 
that samples with large LDI values in the majority classes can also 
be more likely to be misclassified. Thus establishing a reasonable 
ode-level index to supplement the existing category-level index is 
necessary.

Relationship between LDI and misclassification under differ-
ent #layers. The performance of GNNs decreases with the in-
crease of layers (e.g., #layers > 4), which is mainly caused by 
oversmoothing. We explore the relationship between LDI and the 
misclassification caused by oversmoothing. Let the correctly pre-
dicted sample set be $R_n$ when the number of layers equals to $n$. We calculate the ratio of the average LDI of the newly wrongly 
predicted samples (i.e., $R_n \cap R_{n-1}$) when the layers increase by one 
to the average LDI of the correctly predicted samples before the 
layers increases, that is,

$$r_n = \frac{\text{LDI}_{\text{avg}}(R_n \cap R_{n-1})}{\text{LDI}_{\text{avg}}(R_n)},$$

where $\text{LDI}_{\text{avg}}(\Omega)$ represents the average LDI of set $\Omega$.

In Figure 5, as the #layers increases, the value of $r_n$ is greater 
than 1 in nearly all cases. It shows that as the number of layers in-
creases, samples with large LDIs are more likely to be misclassified. 
More results are shown in the appendix.

To sum up, the category distribution of neighbors that LDI 
relies on is related closely to the global category distribution as 
shown in Figure 2. LDI reflects the possibility of each node being 
misclassified as shown in Figure 4. Therefore, on the basis of the 
global distribution and LDI, we proposed several new methods to 
tackle the imbalance for GNNs in the succeeding section.

4 METHODOLOGY

A new loss and four new methods, namely, iFL, GRS, GRW, GML, 
and GBBN, are proposed in this section.

4.1 Improved Focal Loss (iFL).

Focal loss is designed to solve the imbalance problem in object 
classification. For binary classification, the focal loss can be defined as

$$L_{FL} = - \sum_i \alpha_{yi} (1 - p_{i,yi})^\gamma \log(p_{i,yi}),$$

where $p_{i,yi}$ is the probability that node $i$ is correctly classified; $\alpha_{yi}$ and $\gamma$ are hyperparameters. For binary classification, the value of $\alpha_{yi}$ is easy to set. Nevertheless, the number of categories for GNN benchmark data sets is usually large, resulting in that it is difficult 
to set $\alpha_{yi}$ properly. Therefore, to avoid the manually setting for $\alpha_{yi}$, an improved focal loss is defined as follows:

$$L_{iFL} = - \sum_i \alpha_{yi} (1 - \bar{p}_{yi})^\gamma \log(p_{i,yi}),$$

where $\bar{p}_{yi}$ is the average probability that nodes in the category 
$yi$ are correctly classified. For categories with low classification 
accuracy, the value of $(1 - \bar{p}_{yi})^\gamma$ will be large.
4.2 Graph Re-sampling (GRS) and Graph Re-weighting (GRW).

Re-sampling/re-weighting essentially resamples/rewights the samples during network training to strengthen the learning of the minority classes.

Our proposed GRS and GRW are based on re-sampling and re-weighting, respectively. In this study, re-sampling and re-weighting are performed according to the numbers of samples in different categories, and give more weights to the minority classes with smaller sample sizes. LDI is also utilized. Suppose the category of $x_i$ is $c$. $N_c$ is the number of samples of category $c$. $N$ is the total number of samples of the data set. For $x_i$, there are three weighting strategies:

W1: Label-based weighting:
$$w^L_i = \frac{N}{N_c}.$$  
(6)

W2: LDI-based weighting:
$$w^D_i = e^{LDI_i}.$$  
(7)

W3: Combination of label and LDI:
$$w^{LD}_i = w^L_i \cdot w^D_i.$$  
(8)

4.3 Graph Metric Learning (GML).

Metric learning aims to learn a feature space in which the distances between homophily samples are close and the distances between heterophily samples are far. Dong et al. [4] applied a triplet loss-based metric learning to deal with the imbalance problem in image classification.

Triplet loss is used in this study. The key to triplet loss-based metric learning is to select the appropriate triplets as training samples. When choosing a triplet, the anchor point is first selected; the positive and negative sample pairs corresponding to it are then selected from other samples. According to the probability of being easy to be classified correctly, anchor points are divided into easy ones and hard ones. Because the hard triplets are not conducive to the learning of the model, Wang et al. [16] proposed that only choosing the easy anchors would obtain better performance.

In this study, LDI is used to measure the hardness of a sample. The top 10% samples with the highest LDI values are removed (i.e., the top 10% hardest samples are removed). For the remaining

Because $LDIs$ of some samples are zero, to prevent the weights from being zero, $e^{LDI}$ is used as the weight.
Figure 4: For all data sets, nodes concentrate in small \( \text{LDI} \) intervals. Nodes with large \( \text{LDIs} \) have low accuracies.

Figure 5: Variations in \( r \) (solid lines) and accuracy (dot lines) with increasing \#layers using GCN and GAT under inductive (left) and transductive (right) settings on the Citeseer data set.

Figure 6: Node 1 is the selected anchor. Let \( K = 2 \), nodes 2, 3, 4, and 5 represent negative candidate nodes, and the outermost two red nodes 7 and 8 are positive candidate nodes.

For all data sets, nodes concentrate in small \( \text{LDI} \) intervals. Nodes with large \( \text{LDIs} \) have low accuracies.

Figure 5: Variations in \( r \) (solid lines) and accuracy (dot lines) with increasing \#layers using GCN and GAT under inductive (left) and transductive (right) settings on the Citeseer data set.

samples, the larger the \( \text{LDI} \) of a sample, the more likely it is to be selected. For each anchor, in its \( K \)-neighborhood, the negative samples are selected according to the closeness of distances. Outside its \( K \)-neighborhood, the positive samples are selected according to the further distances. The candidates are shown in Figure 6. Let the \( \text{th} \) triplet be \( (x_t, x_t^+, x_t^-) \), its score can be calculated by

\[
\text{score}_t = \max(0, m + d(x_t, x_t^+) - d(x_t, x_t^-)),
\]

where \( m \) refers to margin, and \( d(\cdot) \) represents the distance between two feature vectors. Assume that \( T \) is the set of selected triplets with relatively high scores. The triplet loss can be represented as

\[
\mathcal{L}_{\text{ML}} = \frac{1}{|T|} \sum_t \text{score}_t.
\]

The triplet loss and the focal loss form the final loss function. Following the setting of Zhou et al. [20], we also adopt a cumulative learning strategy to combine the above two losses:

\[
\mathcal{L}_{\text{GML}} = f(\epsilon) \mathcal{L}_{\text{ML}} + \mathcal{L}_{\text{IFL}},
\]

where \( f(\epsilon) \) refers to the weight in the current epoch.

4.4 Graph Bilateral-Branch Network (GBBN).

Our GBBN is based on the bilateral-branch network proposed by Zhou et al. [20] to solve the imbalance in visual recognition tasks. The network consists of two branches: the conventional learning branch and the re-balancing branch. The conventional branch samples uniformly from the original data, maintaining the original data distribution for feature learning. The re-balancing branch is a reverse sampling, which aims to increase the probability of minority classes and reduce data imbalance [1].

Figure 7 shows the structure of GBBN. The conventional (upper) branch uses conventional models such as GCN and GAT. The re-balancing (lower) branch no longer adopts the same structure as the upper branch, but uses DropEdge GNN [13] to reduce information exchange between nodes. In the experiments of this study, we drop all edges to reduce the training complexity. In addition, the proposed iFL is used rather than the standard cross entropy loss. Comparing with GNN-XML proposed in [23] which aims to overcome the imbalance for text classification, the settings of two branches and the loss functions are different.

The upper branch learns the representations of the nodes according to uniform sampling. The sampled nodes also present an imbalanced distribution. The reverse sampling of the lower branch takes the label and \( \text{LDI} \) of a node into account. For samples with higher \( \text{LDIs} \), the sampling probabilities are greater, which not only balances the data from the category-level, but also focuses on the samples that are easily misclassified from the node-level. For this branch, this study leverages the three weighting strategies defined in Section 4.2, namely, W1, W2, and W3.

The graph is entered into the two branches. Two logit outputs of each sample under GNN and DropEdge GNN are obtained and denoted as \( x^G \) and \( x^{DE} \), respectively. Then, a weighted logit vector is obtained by controlling the adaptive weight parameter \( \alpha \). The
We design extensive experiments to evaluate the effectiveness of the where
\[ z = a\alpha^G + (1 - a)\alpha^{DE}, \]  
where \( z \in \mathbb{R}^C \) refers to the final logic vector. The prediction is \( \hat{y} = \text{Softmax}(z) \).

The classification loss of GBBN can be expressed as follows:
\[
L_{\text{GBBN}} = \alpha L_{\text{IFL}}(\hat{p}, y^G) + (1 - \alpha) L_{\text{IFL}}(\hat{p}, y^{DE}).
\]  
(13)

The network training also adopts the cumulative learning strategy. By changing the parameter \( \alpha \), the output logit and the loss also vary, resulting in that different training stages have different learning emphases. In the early stage, the emphasis is on the upper branch. Then, the emphasis gradually shifts to the minority categories and the lower branch. This is reasonable. With the increase of training epoch, minority nodes (especially nodes with large LDIs) should rely on their own representations more than neighborhoods because their neighborhoods contain more heterophily nodes than others. In our experiments, we use linear decay, i.e. \( \alpha = 1 - \epsilon / \text{Epoch} \), where \( \epsilon \) is the current epoch and \( \text{Epoch} \) is the expected max epoch.

## 5 EXPERIMENT

We design extensive experiments to evaluate the effectiveness of the four proposed methods. Both transductive and inductive settings are tested.

### 5.1 Settings.

**Data sets.** Five\(^4\) benchmark data sets in Figure 2 are used: Cora [14], Citeseer, Pubmed [19], CoauthorCS (CS) [19], and Flickr [18]. For Cora and Citeseer, the train, validation, and test data sets. The CS data set is not involved because this data set is not used in the DR-GCN study. The results in Table 1 show that our methods are better than DR-GCN on Cora, Citeseer, and Pubmed data sets. The accuracies of DR-GCN are directly from [14].

### 5.2 Results.

**Comparison with DR-GCN.** DR-GCN [14] only focuses on the transductive setting. Therefore, only the weighting strategy W1 is used in GRS, GRW, and the lower branch of GBBN. The reason why W2 and W3 are not used is because the input data under transductive setting is the entire graph, when calculating the LDIs of the training set, the labels of the validation set and the test set will be referred to, resulting in label leak in the validation and test sets. The CS data set is not involved because this data set is not used in the DR-GCN study. The results in Table 1 show that our methods are better than DR-GCN on Cora, Citeseer, and Pubmed data sets. The accuracies of DR-GCN are directly from [14].

**Results on the transductive setting with shuffle.** For the same reason for why W2 and W3 are not used in the transductive setting, GML is also not utilized in this experiment. The accuracy, G-mean, and macro F1-score are used as the evaluation metrics. The results are shown in Tables 2 and 3. More results are shown in the appendix.

**Results on the inductive setting with shuffle.** In this part, we compare the effect of the model considering whether the LDI is used in the inductive setting. Accordingly, all the weighting strategies (W1, W2, and W3) are leveraged. The macro F1-score and accuracy/G-mean based on SGC with shuffle are shown in Tables 4 and 5, respectively. When without shuffle, the results are shown in the appendix. The results on the inductive setting based on GCN and GAT are shown in the appendix.

---

\(^4\)The rest three data sets Reddit [18], Amazon-Computer [18], and Amazon-Photo [18] shown in the appendix are relatively large. A subgraph sampling-based strategy (e.g., GraphSAINT [18]) should be employed. We leave it as our future work.
Table 2: Accuracy and G-mean on the transductive setting with shuffle.

| Data set       | Acc  | G-mean | Acc  | G-mean | Acc  | G-mean | Acc  | G-mean |
|----------------|------|--------|------|--------|------|--------|------|--------|
| GCN            | 0.339| 0.449  | 0.354| 0.523  | 0.380| 0.658  | 0.370| 0.456  |
| GRS(GCN,W1)    | 0.343| 0.420  | 0.357| 0.527  | 0.384| 0.661  | 0.366| 0.450  |
| GRW(GCN,W1)    | 0.347| 0.483  | 0.393| 0.567  | 0.581| 0.667  | 0.199| 0.382  |
| GBBN(GCN,W1)   | 0.586| 0.710  | 0.581| 0.714  | 0.783| 0.833  | 0.798| 0.811  |
| SGC            | 0.327| 0.405  | 0.350| 0.519  | 0.571| 0.638  | 0.238| 0.265  |
| GRS(SGC,W1)    | 0.331| 0.415  | 0.351| 0.520  | 0.571| 0.638  | 0.239| 0.266  |
| GRW(SGC,W1)    | 0.347| 0.483  | 0.393| 0.567  | 0.581| 0.667  | 0.207| 0.307  |
| GBBN(SGC,W1)   | 0.586| 0.710  | 0.581| 0.714  | 0.783| 0.833  | 0.798| 0.811  |
| GAT            | 0.337| 0.422  | 0.360| 0.530  | 0.560| 0.618  | 0.254| 0.335  |
| GRS(GAT,W1)    | 0.338| 0.414  | 0.352| 0.522  | 0.568| 0.623  | 0.250| 0.320  |
| GRW(GAT,W1)    | 0.307| 0.472  | 0.375| 0.558  | 0.580| 0.666  | 0.206| 0.370  |
| GBBN(GAT,W1)   | 0.553| 0.663  | 0.575| 0.693  | 0.804| 0.845  | 0.782| 0.804  |

Table 3: Macro F1-score on the transductive setting with shuffle.

| Data set       | Cora  | Citeseer | Pubmed | CS  | Acc  | G-mean | GRW(CS) Acc  | G-mean |
|----------------|-------|----------|--------|-----|------|--------|-------------|--------|
| GCN            | 0.223 | 0.307    | 0.561  | 0.227| 0.175| 0.566  | 0.536  | 0.633  |
| GRS(GCN,W1)    | 0.262 | 0.356    | 0.572  | 0.068| 0.165| 0.356  | 0.536  | 0.041  |
| GRW(GCN,W1)    | 0.554 | 0.549    | 0.783  | 0.689| 0.161| 0.303  | 0.536  | 0.041  |
| SGC            | 0.176 | 0.304    | 0.536  | 0.041| 0.065| 0.545  | 0.547  | 0.006  |
| GBBN(GCN,W1)   | 0.623 | 0.626    | 0.838  | 0.826| 0.180| 0.311  | 0.499  | 0.118  |
| GAT            | 0.168 | 0.307    | 0.502  | 0.106| 0.249| 0.350  | 0.570  | 0.112  |
| GRS(GAT,W1)    | 0.492 | 0.519    | 0.803  | 0.680| 0.450| 0.529  | 0.633  | 0.404  |
| GRW(GAT,W1)    | 0.449 | 0.530    | 0.503  | 0.633| 0.482| 0.643  | 0.670  | 0.385  |
| GBBN(GAT,W1)   | 0.454 | 0.528    | 0.527  | 0.658| 0.454| 0.643  | 0.694  | 0.385  |
| GBBN(GAT,W1)   | 0.502 | 0.635    | 0.527  | 0.670| 0.493| 0.648  | 0.645  | 0.385  |
| GBBN(GAT,W1)   | 0.493 | 0.635    | 0.648  | 0.645| 0.518| 0.642  | 0.692  | 0.385  |
| GBBN(GAT,W1)   | 0.367 | 0.448    | 0.332  | 0.488| 0.462| 0.716  | 0.764  | 0.417  |
| GBBN(GAT,W1)   | 0.493 | 0.635    | 0.468  | 0.645| 0.462| 0.716  | 0.764  | 0.417  |
| GBBN(GAT,W1)   | 0.666 | 0.762    | 0.675  | 0.770| 0.675| 0.762  | 0.835  | 0.878  |
| GBBN(GAT,W1)   | 0.675 | 0.757    | 0.669  | 0.761| 0.675| 0.762  | 0.835  | 0.878  |
| GBBN(GAT,W1)   | 0.687 | 0.784    | 0.683  | 0.774| 0.683| 0.784  | 0.837  | 0.879  |

Table 4: Macro F1-score on the inductive setting based on SGC with shuffle.

| Data set       | Cora  | Citeseer | Pubmed | CS  | Acc  | G-mean | GRW(CS) Acc  | G-mean |
|----------------|-------|----------|--------|-----|------|--------|-------------|--------|
| GCN            | 0.450 | 0.529    | 0.604  | 0.363| 0.449| 0.530  | 0.503  | 0.633  |
| GRS(GCN,W1)    | 0.482 | 0.643    | 0.529  | 0.670| 0.454| 0.527  | 0.658  | 0.694  |
| GRW(GCN,W1)    | 0.502 | 0.635    | 0.527  | 0.670| 0.493| 0.468  | 0.645  | 0.645  |
| GBBN(GCN,W1)   | 0.518 | 0.641    | 0.542  | 0.679| 0.367| 0.448  | 0.332  | 0.488  |
| GBBN(GCN,W1)   | 0.493 | 0.635    | 0.468  | 0.645| 0.502| 0.635  | 0.645  | 0.645  |
| GBBN(GCN,W1)   | 0.666 | 0.762    | 0.675  | 0.770| 0.675| 0.762  | 0.835  | 0.878  |
| GBBN(GCN,W1)   | 0.675 | 0.757    | 0.669  | 0.761| 0.675| 0.762  | 0.835  | 0.878  |
| GBBN(GCN,W1)   | 0.687 | 0.784    | 0.683  | 0.774| 0.683| 0.784  | 0.837  | 0.879  |

The following observations can be obtained: (1) The overall performances of our three methods (i.e., GRS, GRW, and GBBN) are better than the conventional model SGC. (2) When with shuffle, GBBN achieved the best performance. GML did not improve the performance of its base models. The reason is discussed in the next subsection. (3) Among the three weighting strategies, the combination of label and LDI weighting strategy (i.e., W3) performed best.
As far as we know, there is currently no research on metric learning. The effectiveness of W3 indicates that the index \( \mathcal{L}_D \) is defined to establish the connection between the imbalance and misclassification. Initial qualitative and quantitative analyses between \( \mathcal{L}_D \) and GNNs performances are conducted to reveal that samples with high \( \mathcal{L}_D \) values are more likely to be misclassified when layers increase. A new loss and four new methods (i.e. iFL, GRS, GRW, GML, and GBBN) are proposed based on \( \mathcal{L}_D \). Comparative experimental results show that the proposed methods are better than DR-GCN. Overall, the proposed GBBN with iFL and \( \mathcal{L}_D \) performs better than the other methods when with shuffle. The effectiveness of W3 indicates that the index \( \mathcal{L}_D \) does contain useful cues for training.

5.3 Discussion.

The better performances of our three methods (GML is poor) over existing methods indicate the importance of considering the distribution characteristics of the imbalance. The combination of label and \( \mathcal{L}_D \) weighting strategy W3 is better in most cases under the inductive setting, indicating that \( \mathcal{L}_D \) contains meaningful training cues. Our future research will study the significance of this index in terms of larger size of the neighborhood (only 1-neighborhood is considered in this study) and more layers.

Further, we discuss why GML failed in the experiment. The principle of metric learning is to decrease the distance between samples of the same category and increase the distance between samples of different categories. However, in the graph data, edge connections exist among samples. The network in training exchanges information among the feature representations of connected samples during training, which will cause metric learning to reduce the distance between an anchor and a positive sample. However, it is also likely to reduce the distance between the anchor and the heterophily samples in the positive sample’s neighbors. The feature coupling caused by data connection causes the metric learning strategy proposed in this study to be unsuitable for graph data sets. As far as we know, there is currently no research on metric learning specifically for GNNs, and exploring more sophisticated methods in the future would be a worthwhile step.

6 CONCLUSION

This study focuses on the imbalanced distribution in learning with GNNs for graph data. Most benchmark data sets used in GNN studies exhibit an imbalanced distribution over categories. A node-level index called \( \mathcal{L}_D \) is defined to establish the connection between the imbalance and misclassification. Initial qualitative and quantitative analyses between \( \mathcal{L}_D \) and GNNs performances are conducted to reveal that samples with high \( \mathcal{L}_D \) values are more likely to be misclassified when layers increase. A new loss and four new methods (i.e. iFL, GRS, GRW, GML, and GBBN) are proposed based on \( \mathcal{L}_D \). Comparative experimental results show that the proposed methods are better than DR-GCN. Overall, the proposed GBBN with iFL and \( \mathcal{L}_D \) performs better than the other methods when with shuffle. The effectiveness of W3 indicates that the index \( \mathcal{L}_D \) does contain useful cues for training.

REFERENCES

[1] Grant Van Horn and Pietro Perona. 2017. The Devil is in the Tails: Fine-grained Classification in the Wild. CoRR (2017).
[2] 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.
[3] Yin Cui, Mengjin Jia, Tao Peng, and Liang Wang. 2019. Class-Balanced Loss Based on Effective Number of Samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 9268–9277.
[4] Qi Dong, Shaoqiang Gong, and Xianfeng Zhu. 2019. Imbalanced Deep Learning by Minority Class Incremental Rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence 41, 6 (2019), 1367–1381.
[5] Yifan Hou, Jian Zhang, James Cheng, Kaiyi Ma, Richard T. B. Ma, Hongzhi Chen, and Ming-Chang Yang. 2020. Measuring and Improving the Use of Graph Information in Graph Neural Networks. In International Conference on Learning Representations (ICLR).
[6] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling Representation and Classifier for Long-Tailed Recognition. In International Conference on Learning Representations (ICLR).
[7] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In International Conference on Learning Representations (ICLR).
[8] Yu-Rong Zheng, Priyabrat Giri, Ross Girshick, and Jitendra Malik. 2017. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision. 2990–2998.
[9] Jialun Liu, Yifan Sun, Chuhsen Han, Zhaozheng Dou, and Wenhui Li. 2020. Deep Representation Learning on Long-Tailed Data: A Learnable Embedding Augmentation Perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2970–2979.
[10] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. 2019. Large-Scale Long-Tailed Recognition in an Open World. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2537–2546.
[11] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. GeoM-gen: Geometric graph convolutional networks. arXiv preprint arXiv:2002.05287 (2020).
[12] Mengyue Ren, Wenyan Yuan, Ben Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. In ICLR.
[13] Yu Rong, Wenhua Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In International Conference on Learning Representations (ICLR).
[14] Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. 2020. Multi-Class Balanced Graph Convolutional Network Learning. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20).
[15] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. International Conference on Learning Representations (ICLR) (2018).
[16] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. 2019. Dynamic Curriculum Learning for Imbalanced Data Classification. In ICCV.
[17] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. PMLR 89:6861–6871.
[18] Hangqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2020. GraphSAN: Graph sampling based inductive learning method. In International Conference on Learning Representations (ICLR).
[19] Lingxiao Zhao and Leman Akoglu. 2019. Pairnorm: Tackling oversmoothing in gnns. arXiv preprint arXiv:1909.12223 (2019).
[20] Boyan Zhou, Quan Cui, Xin-Shen Wei, and Zhao-Min Chen. 2020. BBN: Bilateral-Branch Network With Cumulative Learning for Long-Tailed Visual Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 9719–9728.
[21] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. 2020. Graph Neural Networks with Heterophily. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.
[22] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Graph Neural Networks for Classification in the Wild. In International Conference on Learning Representations (ICLR).
[23] Daoming Zong and Shiliang Sun. 2020. GNN-XML: Graph Neural Networks for XML Document Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2970–2979.

APPENDIX

A CATEGORY DISTRIBUTIONS ON THREE DIFFERENT DATA SETS.

As shown in Figure S1, these three GNN benchmark data sets show imbalanced or highly-skewed distributions.

Figure S1: Category distributions on three different datasets.
B PROOF OF PROPOSITION.

Proposition B.1. Given a node \( x_i \) belonging to the \( c \)th category. When the proportion of homophily nodes in its neighborhood is fixed (i.e., \( p_{i,c} \) is fixed), the less categories of heterophily nodes, the larger the \( LDI \).

Proof. Let \( \sum_{c \neq c'} p_{i,c'} = 1 - p_{i,c} = \Delta, \) then

\[
LDI_i = \frac{1}{\sqrt{2}} \left( \frac{(1 - p_{i,c})^2 + \sum_{c \neq c'} p_{i,c}^2}{C-1} \right).
\]

(14)

By using Cauchy Inequality, we have

\[
\frac{(\sum_{c \neq c'} p_{i,c'})^2}{C-1} \leq \sum_{c \neq c'} p_{i,c'}^2 \leq (\sum_{c \neq c'} p_{i,c'})^2.
\]

(15)

If and only if \( \forall c' \), \( p_{i,c'} = \frac{\Delta}{C-1} \), then

\[
\frac{(\sum_{c \neq c'} p_{i,c'})^2}{C-1} = \sum_{c \neq c'} p_{i,c'}^2 = \frac{\Delta^2}{(C-1)^2}.
\]

(16)

If and only if \( p_{i,c'} = 1 - p_{i,c} \) and \( p_{i,c'} = 0(c' \neq c') \), then

\[
\sum_{c \neq c'} p_{i,c'}^2 = (\sum_{c \neq c'} p_{i,c'})^2 = \Delta^2.
\]

(17)

Accordingly,

\[
\Delta \sqrt{\frac{1 + (C-1)^2}{2(C-1)^2}} \leq LDI_i \leq \Delta.
\]

(18)

The upper bound is attained only when all the heterophily nodes belong to the same category. To sum up, the more concentrated the categories of heterophily nodes, the larger the \( LDI \). The proof ends.

C MORE VALIDATIONS IN \( r \) AND ACCURACY WITH INCREASING \#LAYERS.

In Figure S2, as the \#layers increase, the value of \( r \) is greater than 1 in almost all cases. It shows that as the number of layers increases, samples with higher \( LDI \)s are more likely to be misclassified.

D DATASET STATISTICS.

The data statistics are shown in Table S1.

| Name          | #Node | #Edge | #Features | #Class |
|---------------|-------|-------|-----------|--------|
| Cora          | 2,708 | 5,429 | 1,433     | 7      |
| Citeseer      | 3,327 | 4,732 | 3,703     | 6      |
| Pubmed        | 19,717| 44,338| 500       | 3      |
| CoauthorCS    | 18,333| 81,894| 6,805     | 15     |
| Flickr        | 89,250| 899,756| 500       | 7      |

E RESULTS ON THE TRANSDUCTIVE SETTING WITH SHUFFLE.

In this part, the results on transductive setting without shuffle are shown in Tables S2 and S3, respectively. The following observations can be obtained: (1) Overall, GRS, GRW, and GBBN performed better as compared to the original models. (2) Our label-based sampling strategy (W1) is effective. (3) GRW performed better than GRS and GBBN overall.

F RESULTS ON THE INDUCTIVE SETTING.

F.1 Results on the inductive setting based on SGC without shuffle.

In this part, the results on inductive setting based on SGC without shuffle are shown in Tables S4 and S5, respectively. The following observations can be obtained: (1) The overall performances of our three methods (i.e., GRS, GRW, and GBBN) are better than the conventional model SGC. (2) When without shuffle, GRW achieved the best performance. (3) The effectiveness of W3 indicates that the index \( LDI \) does contain useful cues for training.

F.2 Results on the inductive setting based on GCN and GAT.

We compare the effect of the model considering whether the \( LDI \) is used on the inductive setting. Accordingly, all the three weighting strategies (W1, W2, and W3) are leveraged. The results based on GCN are shown in Tables S6, S7, S8 and S9, and the results based on GAT are shown in Tables S10, S11, S12 and S13.

The following observations can be obtained: (1) The overall performances of our three methods (i.e., GRS, GRW, and GBBN) are better than the conventional models GCN and GAT. (2) When with shuffle, GBBN achieved the best performance based on GCN and GAT. (3) When without shuffle, GBBN achieved the best performance based on GCN, and GRW achieved the best performance based on GAT. (4) The effectiveness of W2 and W3 indicates that the index \( LDI \) does contain useful cues for training.

Overall, the proposed GBBN and GRW algorithms performed better than the other methods. The effectiveness of W2 and W3 indicates that the index \( LDI \) does contain useful cues for training.
### Table S2: Accuracy and G-mean on transductive setting without shuffle.

| Data set   | Cora  | Citeseer | Pubmed  | CS     |
|------------|-------|----------|---------|--------|
|            | Acc   | G-mean   | Acc     | G-mean | Acc   | G-mean   | Acc   | G-mean   | Acc   | G-mean   |
| GCN        | 0.849 | 0.903    | 0.748   | 0.814  | 0.841 | 0.878    | 0.918 | 0.938    |
| GRS(GCN,W1)| 0.854 | 0.905    | 0.744   | 0.812  | 0.840 | 0.878    | 0.917 | 0.936    |
| GRW(GCN,W1)| 0.845 | 0.901    | 0.748   | 0.819  | 0.828 | 0.871    | 0.882 | 0.928    |
| GBBN(GCN,W1)| 0.839 | 0.898    | 0.737   | 0.812  | 0.831 | 0.891    | 0.929 | 0.939    |
| SGC        | 0.847 | 0.894    | 0.736   | 0.802  | 0.816 | 0.858    | 0.881 | 0.880    |
| GRS(SGC,W1)| 0.848 | 0.894    | 0.739   | 0.804  | 0.816 | 0.858    | 0.904 | 0.924    |
| GRW(SGC,W1)| 0.713 | 0.864    | 0.706   | 0.795  | 0.775 | 0.834    | 0.111 | 0.522    |
| GBBN(SGC,W1)| 0.796 | 0.860    | 0.736   | 0.802  | 0.842 | 0.884    | 0.906 | 0.914    |
| GAT        | 0.855 | 0.904    | 0.748   | 0.810  | 0.846 | 0.881    | 0.914 | 0.934    |
| GRS(GAT,W1)| 0.849 | 0.905    | 0.743   | 0.810  | 0.845 | 0.880    | 0.914 | 0.936    |
| GRW(GAT,W1)| 0.857 | 0.907    | 0.754   | 0.824  | 0.858 | 0.892    | 0.894 | 0.940    |
| GBBN(GAT,W1)| 0.835 | 0.890    | 0.735   | 0.804  | 0.855 | 0.875    | 0.930 | 0.934    |

### Table S3: Macro F1-score on transductive setting without shuffle.

| Dataset   | Cora  | Citeseer | Pubmed  | CS     |
|-----------|-------|----------|---------|--------|
| GCN       | 0.836 | 0.696    | 0.835   | 0.895  |
| GRS(GCN,W1)| 0.841 | 0.692    | 0.835   | 0.893  |
| GRW(GCN,W1)| 0.833 | 0.706    | 0.824   | 0.858  |
| GBBN(GCN,W1)| 0.827 | 0.694    | 0.827   | 0.901  |
| SGC       | 0.834 | 0.672    | 0.811   | 0.813  |
| GRS(SGC,W1)| 0.835 | 0.676    | 0.811   | 0.874  |
| GRW(SGC,W1)| 0.724 | 0.673    | 0.774   | 0.158  |
| GBBN(SGC,W1)| 0.771 | 0.673    | 0.841   | 0.867  |
| GAT       | 0.843 | 0.683    | 0.840   | 0.889  |
| GRS(GAT,W1)| 0.837 | 0.685    | 0.839   | 0.892  |
| GRW(GAT,W1)| 0.845 | 0.714    | 0.854   | 0.871  |
| GBBN(GAT,W1)| 0.820 | 0.678    | 0.852   | 0.902  |

### Table S4: Macro F1-score on inductive setting based on SGC without shuffle.

| Dataset   | Cora  | Citeseer | Pubmed  | CS     |
|-----------|-------|----------|---------|--------|
| GCN       | 0.659 | 0.720    | 0.701   | 0.777  |
| GRS(GCN,W1)| 0.653 | 0.715    | 0.700   | 0.776  |
| GRS(W1)   | 0.750 | 0.830    | 0.711   | 0.791  |
| GRS(W2)   | 0.655 | 0.715    | 0.705   | 0.780  |
| GRS(W3)   | 0.732 | 0.822    | 0.712   | 0.799  |
| GRW(W1)   | 0.764 | 0.844    | 0.691   | 0.801  |
| GRW(W2)   | 0.366 | 0.415    | 0.582   | 0.687  |
| GRW(W3)   | 0.767 | 0.846    | 0.719   | 0.803  |
| GBBN(W1)  | 0.721 | 0.801    | 0.691   | 0.776  |
| GBBN(W2)  | 0.703 | 0.781    | 0.691   | 0.773  |
| GBBN(W3)  | 0.721 | 0.804    | 0.673   | 0.763  |

### Table S5: Accuracy and G-mean on inductive setting based on SGC without shuffle.

| Dataset   | Cora  | Citeseer | Pubmed  | CS     |
|-----------|-------|----------|---------|--------|
| GCN       | 0.659 | 0.720    | 0.701   | 0.777  |
| GRS(GCN,W1)| 0.653 | 0.715    | 0.700   | 0.776  |
| GRS(W1)   | 0.750 | 0.830    | 0.711   | 0.791  |
| GRS(W2)   | 0.655 | 0.715    | 0.705   | 0.780  |
| GRS(W3)   | 0.732 | 0.822    | 0.712   | 0.799  |
| GRW(W1)   | 0.764 | 0.844    | 0.691   | 0.801  |
| GRW(W2)   | 0.366 | 0.415    | 0.582   | 0.687  |
| GRW(W3)   | 0.767 | 0.846    | 0.719   | 0.803  |
| GBBN(W1)  | 0.721 | 0.801    | 0.691   | 0.776  |
| GBBN(W2)  | 0.703 | 0.781    | 0.691   | 0.773  |
| GBBN(W3)  | 0.721 | 0.804    | 0.673   | 0.763  |
Table S6: Accuracy and G-mean on inductive setting based on GCN with shuffle.

| Dataset | Cora  | Citeseer | Pubmed | CS   | Flickr |
|---------|-------|----------|--------|------|--------|
| GCN     | 0.524 | 0.627    | 0.554  | 0.678| 0.666  |
| GML     | 0.499 | 0.626    | 0.534  | 0.668| 0.665  |
| GRS(W1) | 0.522 | 0.635    | 0.541  | 0.680| 0.670  |
| GRS(W2) | 0.509 | 0.614    | 0.538  | 0.670| 0.670  |
| GRS(W3) | 0.485 | 0.645    | 0.540  | 0.676| 0.670  |
| GRW(W1) | 0.526 | 0.648    | 0.555  | 0.692| 0.664  |
| GRW(W2) | 0.514 | 0.602    | 0.552  | 0.670| 0.662  |
| GRW(W3) | 0.525 | 0.648    | 0.545  | 0.691| 0.670  |
| GBBN(W1)| 0.640 | 0.769    | 0.667  | 0.773| 0.807  |
| GBBN(W2)| 0.674 | 0.778    | 0.668  | 0.775| 0.808  |
| GBBN(W3)| 0.643 | 0.744    | 0.659  | 0.774| 0.807  |

Table S7: Macro F1-score on inductive setting based on GCN with shuffle.

| Dataset | Cora  | Citeseer | Pubmed | CS   | Flickr |
|---------|-------|----------|--------|------|--------|
| GCN     | 0.449 | 0.507    | 0.325  | 0.129|        |
| GML     | 0.440 | 0.496    | 0.293  | 0.131|        |
| GRS(W1) | 0.450 | 0.513    | 0.367  | 0.101|        |
| GRS(W2) | 0.432 | 0.499    | 0.334  | 0.131|        |
| GRS(W3) | 0.446 | 0.508    | 0.373  | 0.181|        |
| GRW(W1) | 0.463 | 0.526    | 0.346  | 0.190|        |
| GRW(W2) | 0.408 | 0.486    | 0.229  | 0.116|        |
| GRW(W3) | 0.465 | 0.521    | 0.362  | 0.163|        |
| GBBN(W1)| 0.625 | 0.640    | 0.786  | 0.123|        |
| GBBN(W2)| 0.644 | 0.642    | 0.801  | 0.101|        |
| GBBN(W3)| 0.643 | 0.744    | 0.707  | 0.115|        |

Table S8: Macro F1-score on inductive setting based on GCN without shuffle.

| Dataset | Cora  | Citeseer | Pubmed | CS   | Flickr |
|---------|-------|----------|--------|------|--------|
| GCN     | 0.449 | 0.507    | 0.325  | 0.129|        |
| GML     | 0.440 | 0.496    | 0.293  | 0.131|        |
| GRS(W1) | 0.450 | 0.513    | 0.367  | 0.101|        |
| GRS(W2) | 0.432 | 0.499    | 0.334  | 0.131|        |
| GRS(W3) | 0.446 | 0.508    | 0.373  | 0.181|        |
| GRW(W1) | 0.463 | 0.526    | 0.346  | 0.190|        |
| GRW(W2) | 0.408 | 0.486    | 0.229  | 0.116|        |
| GRW(W3) | 0.465 | 0.521    | 0.362  | 0.163|        |
| GBBN(W1)| 0.625 | 0.640    | 0.786  | 0.123|        |
| GBBN(W2)| 0.644 | 0.642    | 0.801  | 0.101|        |
| GBBN(W3)| 0.643 | 0.744    | 0.707  | 0.115|        |

Table S9: Accuracy and G-mean on inductive setting based on GCN without shuffle.

| Dataset | Cora  | Citeseer | Pubmed | CS   | Flickr |
|---------|-------|----------|--------|------|--------|
| GCN     | 0.524 | 0.627    | 0.554  | 0.678| 0.666  |
| GML     | 0.499 | 0.626    | 0.534  | 0.668| 0.665  |
| GRS(W1) | 0.522 | 0.635    | 0.541  | 0.680| 0.670  |
| GRS(W2) | 0.509 | 0.614    | 0.538  | 0.670| 0.670  |
| GRS(W3) | 0.485 | 0.645    | 0.540  | 0.676| 0.670  |
| GRW(W1) | 0.526 | 0.648    | 0.555  | 0.692| 0.664  |
| GRW(W2) | 0.514 | 0.602    | 0.552  | 0.670| 0.662  |
| GRW(W3) | 0.525 | 0.648    | 0.545  | 0.691| 0.670  |
| GBBN(W1)| 0.640 | 0.769    | 0.667  | 0.773| 0.807  |
| GBBN(W2)| 0.674 | 0.778    | 0.668  | 0.775| 0.808  |
| GBBN(W3)| 0.643 | 0.744    | 0.659  | 0.774| 0.807  |
Table S10: Accuracy and G-mean on inductive setting based on GAT with shuffle.

| Dataset | Cora | Citeseer | Pubmed | CS | Flickr |
|---------|------|----------|--------|----|--------|
|         | Acc  | G-mean   | Acc    | G-mean | Acc  | G-mean |
| GAT     | 0.476 | 0.565 | 0.532 | 0.659 | 0.666 | 0.722 |
| GML     | 0.491 | 0.578 | 0.528 | 0.655 | 0.659 | 0.710 |
| GRS(W1) | 0.488 | 0.629 | 0.538 | 0.678 | 0.669 | 0.740 |
| GRS(W2) | 0.480 | 0.572 | 0.530 | 0.660 | 0.653 | 0.712 |
| GRS(W3) | 0.484 | 0.627 | 0.509 | 0.655 | 0.673 | 0.742 |
| GRW(W1) | 0.493 | 0.632 | 0.544 | 0.682 | 0.669 | 0.733 |
| GRW(W2) | 0.481 | 0.567 | 0.547 | 0.671 | 0.661 | 0.712 |
| GRW(W3) | 0.481 | 0.615 | 0.537 | 0.677 | 0.674 | 0.743 |
| GBBN(W1) | 0.628 | 0.727 | 0.657 | 0.750 | 0.818 | 0.860 |
| GBBN(W2) | 0.624 | 0.701 | 0.648 | 0.738 | 0.820 | 0.859 |
| GBBN(W3) | 0.629 | 0.728 | 0.658 | 0.757 | 0.822 | 0.870 |

Table S11: Macro F1-score on inductive setting based on GAT with shuffle.

| Dataset | Cora | Citeseer | Pubmed | CS | Flickr |
|---------|------|----------|--------|----|--------|
|         | Acc  | G-mean   | Acc    | G-mean | Acc  | G-mean |
| GCN     | 0.363 | 0.474 | 0.649 | 0.238 | 0.132 |
| GML     | 0.382 | 0.471 | 0.631 | 0.251 | 0.127 |
| GRS(W1) | 0.435 | 0.508 | 0.664 | 0.433 | 0.145 |
| GRS(W2) | 0.378 | 0.475 | 0.637 | 0.260 | 0.127 |
| GRS(W3) | 0.432 | 0.474 | 0.667 | 0.412 | 0.183 |
| GRW(W1) | 0.438 | 0.515 | 0.657 | 0.421 | 0.168 |
| GRW(W2) | 0.374 | 0.489 | 0.634 | 0.213 | 0.126 |
| GRW(W3) | 0.418 | 0.507 | 0.667 | 0.406 | 0.185 |
| GBBN(W1) | 0.580 | 0.601 | 0.819 | 0.812 | 0.112 |
| GBBN(W2) | 0.554 | 0.588 | 0.820 | 0.803 | 0.137 |
| GBBN(W3) | 0.586 | 0.617 | 0.821 | 0.817 | 0.124 |

Table S12: Accuracy and G-mean on inductive setting based on GAT without shuffle.

| Dataset | Cora | Citeseer | Pubmed | CS | Flickr |
|---------|------|----------|--------|----|--------|
|         | Acc  | G-mean   | Acc    | G-mean | Acc  | G-mean |
| GCN     | 0.742 | 0.811 | 0.716 | 0.796 | 0.838 | 0.868 |
| GML     | 0.732 | 0.801 | 0.713 | 0.788 | 0.836 | 0.866 |
| GRS(W1) | 0.764 | 0.850 | 0.721 | 0.803 | 0.839 | 0.875 |
| GRS(W2) | 0.744 | 0.810 | 0.702 | 0.786 | 0.839 | 0.871 |
| GRS(W3) | 0.759 | 0.836 | 0.715 | 0.801 | 0.839 | 0.875 |
| GRW(W1) | 0.780 | 0.849 | 0.718 | 0.803 | 0.836 | 0.873 |
| GRW(W2) | 0.749 | 0.814 | 0.717 | 0.793 | 0.836 | 0.875 |
| GRW(W3) | 0.788 | 0.856 | 0.723 | 0.806 | 0.840 | 0.876 |
| GBBN(W1) | 0.715 | 0.806 | 0.682 | 0.769 | 0.837 | 0.875 |
| GBBN(W2) | 0.711 | 0.791 | 0.672 | 0.761 | 0.839 | 0.874 |
| GBBN(W3) | 0.724 | 0.807 | 0.684 | 0.770 | 0.833 | 0.872 |

Table S13: Macro F1-score on inductive setting based on GAT without shuffle.

| Dataset | Cora | Citeseer | Pubmed | CS | Flickr |
|---------|------|----------|--------|----|--------|
|         | Acc  | G-mean   | Acc    | G-mean | Acc  | G-mean |
| GCN     | 0.723 | 0.673 | 0.832 | 0.829 | 0.148 |
| GML     | 0.709 | 0.647 | 0.830 | 0.821 | 0.149 |
| GRS(W1) | 0.747 | 0.684 | 0.835 | 0.852 | 0.151 |
| GRS(W2) | 0.717 | 0.658 | 0.835 | 0.828 | 0.149 |
| GRS(W3) | 0.740 | 0.681 | 0.835 | 0.857 | 0.191 |
| GRW(W1) | 0.755 | 0.683 | 0.834 | 0.857 | 0.151 |
| GRW(W2) | 0.728 | 0.665 | 0.836 | 0.789 | 0.139 |
| GRW(W3) | 0.765 | 0.689 | 0.838 | 0.865 | 0.144 |
| GBBN(W1) | 0.691 | 0.623 | 0.836 | 0.858 | 0.143 |
| GBBN(W2) | 0.680 | 0.614 | 0.832 | 0.841 | 0.148 |
| GBBN(W3) | 0.695 | 0.628 | 0.832 | 0.849 | 0.142 |