
TAM: Topology-Aware Margin Loss for Class-Imbalanced Node Classification

Jaeyun Song^{*1} Joonhyung Park^{*1} Eunho Yang¹²

Abstract

Learning unbiased node representations under class-imbalanced graph data is challenging due to interactions between adjacent nodes. Existing studies have in common that they compensate the minor class nodes ‘as a group’ according to their overall quantity (ignoring node connections in graph), which inevitably increase the false positive cases for major nodes. We hypothesize that the increase in these false positive cases is highly affected by the label distribution around each node and confirm it experimentally. In addition, in order to handle this issue, we propose Topology-Aware Margin (TAM) to reflect local topology on the learning objective. Our method compares the connectivity pattern of each node with the class-averaged counter-part and adaptively adjusts the margin accordingly based on that. Our method consistently exhibits superiority over the baselines on various node classification benchmark datasets with representative GNN architectures.

1. Introduction

The importance of learning qualitative node representation has been emerging to accurately classify the node property in real-world graphs such as social networks, commercial graphs, and chemical molecules (Mohammadrezaei et al., 2018; Ying et al., 2018; Hamilton et al., 2017). Recently, graph neural networks (GNNs) (Welling & Kipf, 2016; Veličković et al., 2018; Hamilton et al., 2017) are widely adopted to handle graph-structured data and have shown remarkable success in various fields. However, as natural graphs could be class-imbalanced inherently, GNNs are prone to be biased toward major classes. Learning from those graphs without handling class-imbalanced issue leads

to low accuracy for minor classes. Although the simple solution is to curate class-balanced graphs, collecting data in a balanced way is not always possible.

To address this problem, diverse imbalance handling strategies for node classification (Shi et al., 2020; Zhao et al., 2021; Qu et al., 2021; Park et al., 2021) have been recently proposed. These methods fortify minor classes in their own way such as extending SMOTE method (Chawla et al., 2002) to graph-structured data (Zhao et al., 2021), mixing nodes by considering neighbor structure (Park et al., 2021) or generating virtual minor nodes via the conditional GAN (Shi et al., 2020).

However, these approaches overlook the fact that when compensating the minor classes based on their quantity, certain nodes could significantly degrade the performance of other classes. Considering the innate characteristics of message passing algorithms of GNNs, we hypothesize that the entire representation learning procedure can be misled by weighted minor nodes in the aggregation of message passing and that the effect is more attributed to nodes with high connectivity rates with other (major) classes. Toward this direction, we observe that compensating such minor nodes with high connectivity rates to major class significantly increase false positives for major nodes. In line with this observation, we confirm that existing imbalance handling algorithms show sub-optimal performances as they do not reflect this local topology when weighting the minor classes. Although not directly related to our hypothesis, ReNode (Chen et al., 2021) is somehow related in terms of adjusting the weights of some nodes; this method decreases the weights of nodes close to topological class boundaries. However, this method can only work for homogeneously-connected graphs. Moreover, since the connectivity patterns and class-wise relative weighting for multi-class cases are not considered in the weighting process, the impacts of individual nodes affecting other classes are still not properly identified.

Armed with this hypothesis, in this paper we propose Topology-Aware Margin (TAM), a node-wise logit adjustment method, which takes into account their local topology in terms of class-pair connectivity and neighbor distribution statistics. Our key principle is as follows: if a (minor) node is highly likely to be confused with specific (major) classes considering its local topology, we should decrease

^{*}Equal contribution ¹Graduate School of AI, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea ²AITRICS, Seoul, South Korea. Correspondence to: Jaeyun Song <mercery@kaist.ac.kr>, Joonhyung Park <deepjoon@kaist.ac.kr>, Eunho Yang <eunhoy@kaist.ac.kr>.

the margins for those (major) classes so that GNNs can be trained in a well-calibrated manner (informally speaking, when some minor node has abnormally many major neighbors, we reduce the weight for it). Toward this, first we devise Anomalous Connectivity-aware Margin (ACM) that decreases the target class margin of a node if it has relatively high neighbor density for that target class. At the same time, we introduce Anomalous Distribution-aware Margin (ADM) that calculates the degree of confusion based on the average neighbor statistics of *target* class and additionally adjusts the margin of the target class.

Our method can be combined with most imbalance handling approaches seamlessly and consistently brings the performance enhancement over multiple node classification benchmark datasets such as citation networks (Sen et al., 2008), WebKB, and Wikipedia networks (Rozemberczki et al., 2021) with various architectures including GCN (Welling & Kipf, 2016), GAT (Veličković et al., 2018), and GraphSAGE (Hamilton et al., 2017).

Our contribution is threefold:

- We hypothesize and confirm that false positives due to compensating minor nodes do not appear evenly on the graph, and are highly affected by the neighbor label distribution around each node. Specifically, we demonstrate that a significantly high false positive ratio appears around minor nodes that have higher connectivity with major nodes.
- We propose a tailored solution to this hypothesis that can effectively decrease excessive false positives by individually adjusting the extent of compensation based on node topology compared to class statistics.
- Our method can be combined with existing imbalance handling methods regardless of their compensating strategies. When combined with our method, baselines consistently improve the imbalance handling performance on multiple benchmark datasets.

2. Preliminary

2.1. Notation and Definitions

We target a semi-supervised node classification task on an undirected graph $G(V, E)$ where V is a node set, E is the set of edges, and \mathcal{Y} is the set of possible class labels. Y is set of labels for V and V^L is the set of labeled nodes ($V^L \subseteq V$). $X \in \mathbb{R}^{|V| \times d}$ is the node feature matrix where the i -th node v_i has the node feature x_i (the i -th row of X). Let $\mathcal{N}(v)$ be the set of adjacent nodes to node v : $\{u \in V | u, v \in E\}$. d_v is the degree of node v : $|\mathcal{N}(v)|$.

We here introduce two key definitions leveraged for estimating the node- and class-level connectivity: neighbor label

distribution (\mathcal{D}) and class-wise connectivity matrix (\mathcal{C}).

Definition 2.1. (Neighbor Label Distribution \mathcal{D}). Let $\mathcal{D} \in \mathbb{R}^{|V| \times |\mathcal{Y}|}$. Then, neighbor label distribution (NLD) \mathcal{D} is defined as:

$$\mathcal{D}_{i,j} = \frac{|\{v \in \mathcal{N}(i) \cup \{i\} | y_v = j\}|}{d_i + 1}. \quad (1)$$

That is, the i -th row represents the distribution of neighbor labels of node i (including node y_i itself).

Definition 2.2. (Class-wise Connectivity Matrix \mathcal{C}). Let $\mathcal{C} \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$. Then, class-wise connectivity matrix \mathcal{C} is defined as:

$$\mathcal{C}_{i,j} = \frac{1}{|\{v \in V | y_v = i\}|} \sum_{u \in \{v \in V | y_u = j\}} \mathcal{D}_{u,j}. \quad (2)$$

We define these two terms similarly with homophily used in Pei et al. (2019) since our method requires to compute neighbor label distribution for a given node and its class statistics. Note that two concepts are computed under the assumption that neighbor label distributions for all nodes are given. However, this assumption is hardly satisfied in real world, hence we utilize model prediction for unlabeled nodes in main experiments (Section 4.3).

2.2. Node Classification with Graph Neural Networks

In this section, we briefly describe the GNNs in terms of node classification. The l -th layer of GNNs consists of three functions including message function m_l , feature aggregation function ψ_l , and node feature update functions γ_l . For node v , node feature $x_v^{(l+1)}$ is derived from $x_v^{(l)}$ as follows.

$$x_v^{(l+1)} = \gamma_l \left(x_v^{(l)}, \psi_l \left(\{m_l(x_v^{(l)}, x_u^{(l)}, w_{v,u}) | u \in \mathcal{N}(v)\} \right) \right) \quad (3)$$

where $w_{u,v}$ is the edge weight of edge $\{v, u\} \in E$. For example, node feature $x_v^{(l)}$ of Graph Convolutional Network (GCN) (Welling & Kipf, 2016) is computed as $x_v^{(l+1)} = \sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{\Theta_l^\top x_u^{(l)}}{\sqrt{\hat{d}_v \hat{d}_u}}$, where $\hat{d}_v = 1 + \sum_{u \in \mathcal{N}(v)} w_{v,u}$ and Θ_l is the matrix of filter parameters at the l -th layer.

2.3. Margin-based Class-Imbalance Handling

We revisit margin-based imbalance handling methods in the vision domain (Cao et al., 2019; Tan et al., 2020; Ren et al., 2020; Menon et al., 2020). Margin-based approaches alleviate the bias to major classes by increasing the margin of minor classes to major classes or decreasing the margin of major classes to minor classes in the training phase and show significantly superior performance than other loss modification algorithms (Ren et al., 2020). Specifically, let us define the quantity of the k -th class N_k , then cross entropy (CE)

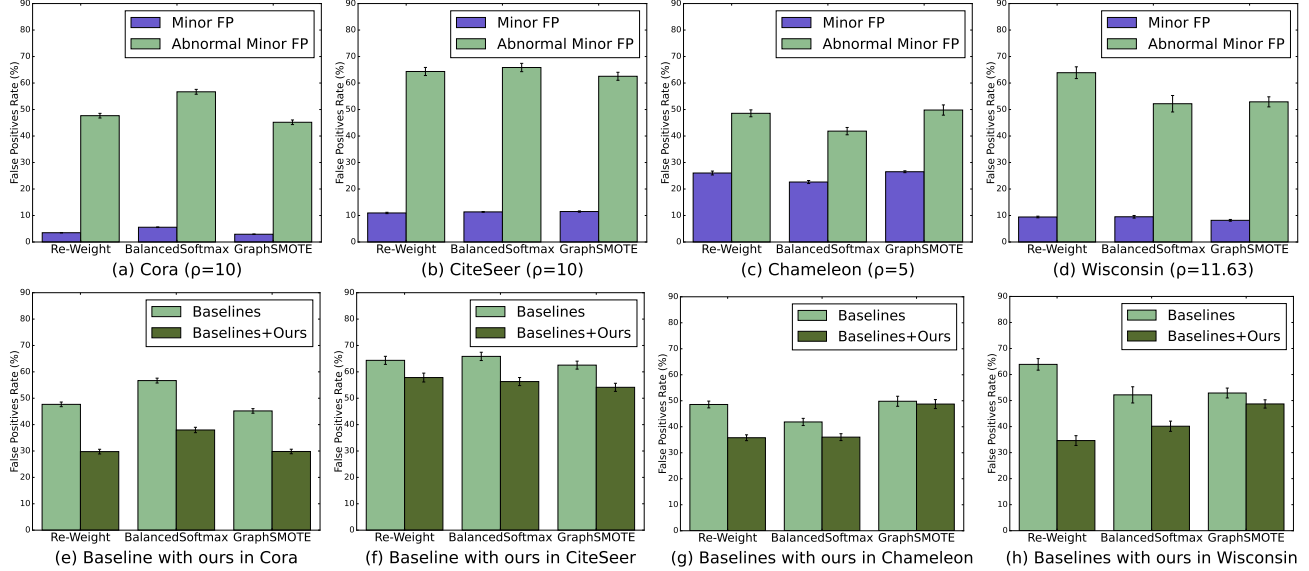


Figure 1. Comparison of false positive rates near normal minor nodes and anomalously-connected minor nodes. For (a) ~ (d), **Abnormal Minor FP** represents false positive rates when major nodes are connected with anomalous minor nodes. **Minor FP** presents the average probability of being false positives. In (e) ~ (h), the results of the change in false positive rates (caused by abnormal minor nodes) are presented when baselines are integrated with our method.

with Balanced Softmax (Ren et al., 2020) is computed for node v as:

$$\mathcal{L} = \mathcal{L}_{CE}(l_v + m, y_v) = -\log \left(\frac{e^{l_v, y_v + \log N_{y_v}}}{\sum_{k \in \mathcal{Y}} e^{l_v, k + \log N_k}} \right), \quad (4)$$

where l_v, y_v are the logit and the label of node v respectively, and $m = (\log N_1, \log N_2, \dots, \log N_{|\mathcal{Y}|})$. In multi-class Softmax regression, Balanced Softmax minimizes the generalization bound (Ren et al., 2020). In that margin-based approaches could adjust logits by considering the relative quantity ratio between two classes and are effective in the vision domain, we adopt margin-based approaches in our algorithm.

3. Analysis of Anomalous Connectivity

Our primary research hypothesis is minor nodes that deviate from the connectivity pattern induce excessive false positives during the quantity-based compensating process. To verify our assumption empirically, we investigate the *topological positions of false positives* on minor classes.

Experimental Design We design an experiment to compare the false positives ratios on neighbors of anomalously connected minor nodes with those of normal minor nodes assuming that neighbor label information is accessible. First, we define anomalously connected node set V^* as $V^* = \{v \in V^L \mid \max_{t \in |\mathcal{Y}| \setminus \{y_v\}} \frac{D_{v,t}}{c_{y_v,t}} > 1\}$, which is a set of nodes that has more edge connections with other classes

compared to class-averaged level. $V_{minor}^* \subset V^*$ is a set of minor class nodes belonging to V^* , $V_{major} \subset V \setminus V^L$ is a set of major nodes in validation set, and $FP(\cdot)$ is a function that counts the number of false positives for the minor classes.

We calculate the ratio $\frac{FP(\{\mathcal{N}(v) \cap V_{major} \mid v \in V_{minor}^*\})}{|\mathcal{N}(v) \cap V_{major} \mid v \in V_{minor}^*|}$, representing the probability of being false positives when major nodes are connected with anomalous minor nodes (**Abnormal Minor FP** in Figure 1). Then we compare the computed probability with the ratio $\frac{FP(V_{major})}{|V_{major}|}$, the average probability of being false positives (**Minor FP** in Figure 1). Experimental details are described in the following paragraph.

Settings We conduct experiments on two well-known node classification benchmark datasets - CiteSeer (homophilous graph) and Wisconsin (heterophilous graph) using GCN architecture. For CiteSeer dataset (Sen et al., 2008), we follow the split of Chen et al. (2018) and process the label distribution to follow step imbalance setting as existing works (Cao et al., 2019; Chen et al., 2021). In other words, all minor classes have n_{minor} labeled nodes and the major class nodes have $\rho * n_{minor}$ where ρ is an imbalance ratio. We set ρ to 10. All experiments are repeated 100 times.

Results To verify our assumption, we scrutinize three representative imbalance handling approaches: Re-Weight (Japkowicz & Stephen, 2002), Balanced Softmax (Ren et al., 2020), and GraphSMOTE (Zhao et al., 2021). In Figure 1 (a) and (b), we confirm that false positives on minor

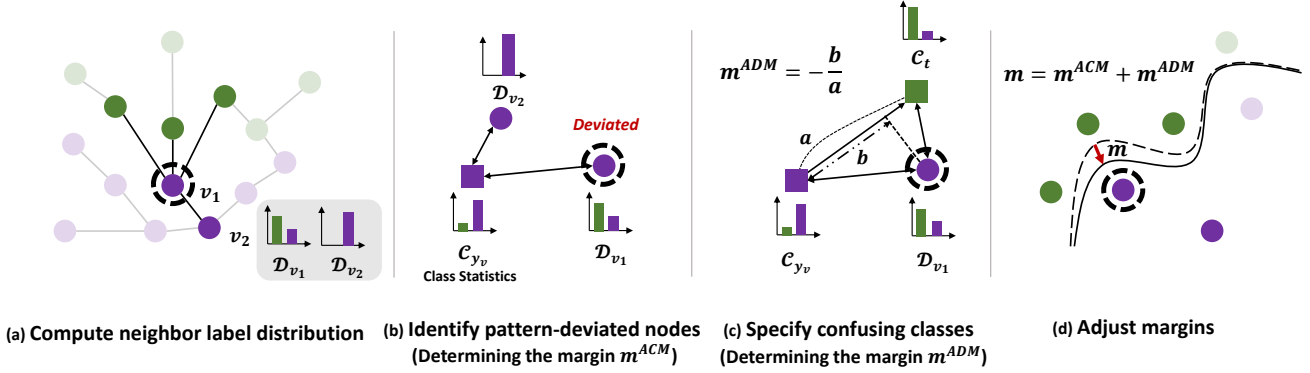


Figure 2. The overall pipeline of TAM. We first calculate neighbor label distribution \mathcal{D} by utilizing model prediction for unlabeled neighbors, then compute class-wise connectivity matrix \mathcal{C} in (a). According to \mathcal{D} and \mathcal{C} , determine ACM m^{ACM} in (b) and ADM m^{ADM} in (c). By applying two margins to logits, we adjust margins in (d).

classes are intensively concentrated around minor nodes that have higher connectivity with other classes (compared to class-averaged level) regardless of each baseline’s compensating strategy. Interestingly, the aptness of false positives is consistently exhibited in both homogeneously- and heterogeneously-connected graphs. It is worth noting that the increase in false negative rates is negligible to the decrease in false positive rates in Appendix A.1.

4. Proposed Method

We now introduce our effective margin adjustment strategy, TAM, which determines the intensity of imbalance compensation based on the local topology of individual node. In Section 3, we investigated that reinforcing minor nodes connected more with other classes than class-averaged level induces the false positives of minor classes. Inspired by this observation, we identify topologically improbable nodes and adaptively adjust the margins for those nodes. We devise two core components of TAM. First, Anomalous Connectivity Margin (ACM) decreases the class margin of a target node (one of the neighbor nodes) if the portion of the class of target node in neighbor label distribution (NLD) is larger than class-averaged connectivity (Section 4.1). Then, Anomalous Distribution-aware Margin (ADM) adjusts the margin according to the relative distance computed using both target class-averaged NLD and self class-averaged NLD (Section 4.2). The overall pipeline and full algorithm of TAM are provided in Figure 2 and in Algorithm 1, respectively.

Our learning objective function is formulated as:

$$\mathcal{L}_{TAM} = \frac{1}{|V^L|} \sum_{v \in V^L} \mathcal{L}(l_v + \alpha m_v^{ACM} + \beta m_v^{ADM}, y_v), \quad (5)$$

where \mathcal{L} is the loss function (such as cross entropy), $l_v \in \mathbb{R}^{|\mathcal{Y}|}$ is the logit of node v , and y_v is the label of node v . Our novel components add the margins here: $m_v^{ACM} \in \mathbb{R}^{|\mathcal{Y}|}$

represents the ACM term of node v and $m_v^{ADM} \in \mathbb{R}^{|\mathcal{Y}|}$ is the ADM term, with respective hyperparameters α and β .

It is worth noting that our method can be orthogonally combined with any imbalance handling methods and GNNs since our method simply adjusts output logits of the model before evaluating the loss function. We describe how to compute m_v^{ACM} and m_v^{ADM} in the following subsections.

4.1. Anomalous Connectivity-Aware Margin

To restrain the generation of false positives caused by abnormally connected nodes, we suggest ACM, which modifies the margin of each class by calibrating the deviation of $\mathcal{D}_{v,:}$ (NLD of node v) from $\mathcal{C}_{y_v,:}$ (connectivity pattern of class y_v). Given a node v , we first compare \mathcal{D}_{y_v,y_v} with averaged homophily ratio of class y_v , \mathcal{C}_{y_v,y_v} , to estimate how much node v follows the class-homophily. If $\frac{\mathcal{C}_{y_v,y_v}}{\mathcal{D}_{v,y_v}}$ is high, we decreases the margins on entire classes. The intuition behind here is: as nodes that do not follow the class-homophily tendency would be risky in the imbalance handling process, we make learning signals of these nodes weak in the training phase.

To further control the margin for each class t , we calculate the connecting ratio with class t over class-averaged level: $\frac{\mathcal{D}_{v,t}}{\mathcal{C}_{y_v,t}}$. The high $\frac{\mathcal{D}_{v,t}}{\mathcal{C}_{y_v,t}}$ implies that node v has a fair chance to be confused with class t considering the message-passing of GNNs. Hence, we decreases the margin of class t in this case so that GNNs can be trained in a well-calibrated manner.

Based on this motivation, the ACM of node v on class t is derived as:

$$m_{v,t}^{ACM} = -\max\left(\log\left(\frac{\mathcal{C}_{y_v,y_v}}{\mathcal{D}_{v,y_v}} \cdot \frac{\mathcal{D}_{v,t}}{\mathcal{C}_{y_v,t}}\right), 0\right) \quad (6)$$

Note that the margin of its own class m_{v,y_v}^{ACM} is set to 0 by the above equation.

Algorithm 1 Topology-Aware Margin

```

1: Input: Graph  $G(X, V, E, Y)$ , set of possible class labels  $\mathcal{Y}$ ,
   set of labeled nodes  $V^L$ , model  $f_\theta$ , loss function  $\mathcal{L}$ , hyperpar-
   ameters  $\alpha, \beta$ , learning rate  $\eta$ , label  $y_v$  onehot vector,  $e_{y_v}$ .
2: Initialize: Model parameter  $\theta \in \mathbb{R}^d$ .
3: Compute  $T_1, T_2, \dots, T_k$  following the Equation (9)
4: for  $t = 1, 2, \dots, T$  do
5:    $l \leftarrow f_\theta(X, V, E)$  {Model prediction}
6:   for  $k = 1, 2, \dots, |\mathcal{Y}|$  do
7:      $p_{:,k} \leftarrow \frac{l_{:,k}}{T_k}$ 
8:   end for{Class-wise temperature}
9:    $\mathcal{D} \leftarrow \mathbf{0}$ 
10:  for  $v \in V^L$  do
11:    for  $u \in \mathcal{N}(v) \cup \{v\}$  do
12:      if  $u \in V^L$  then
13:         $\mathcal{D}_{v,:} \leftarrow \mathcal{D}_{v,:} + \frac{1}{|\mathcal{N}(v)|+1} e_{y_u}$ 
14:      else
15:         $\mathcal{D}_{v,:} \leftarrow \mathcal{D}_{v,:} + \frac{1}{|\mathcal{N}(v)|+1} \text{Softmax}(p_{u,:})$ 
16:      end if
17:    end for
18:  end for{Neighbor label distribution}
19:  for  $k = 1, 2, \dots, |\mathcal{Y}|$  do
20:     $\mathcal{C}_{k,:} \leftarrow \frac{1}{|\{v \in V^L | y_v = k\}|} \sum_{u \in \{v \in V^L | y_v = k\}} \mathcal{D}_{u,:}$ 
21:  end for{Class-wise connectivity matrix}
22:  for  $v \in V^L$  do
23:    for  $k = 1, 2, \dots, |\mathcal{Y}|$  do
24:      Compute  $\cos A_{v,k}$  following the Equation (7)
25:       $m_{v,k}^{ACM} \leftarrow -\max \left( \log \left( \left( \frac{\mathcal{C}_{y_v, y_v}}{\mathcal{D}_{v, y_v}} \right) \cdot \left( \frac{\mathcal{D}_{v,t}}{\mathcal{C}_{y_v, t}} \right) \right), 0 \right)$ 
26:       $m_{v,k}^{ADM} \leftarrow -\frac{\text{JS}(\mathcal{D}_{v,:}, \mathcal{C}_{y_v,:}) \cos A_{v,k}}{\text{JS}(\mathcal{C}_{t,:}, \mathcal{C}_{y_v,:})}$ 
27:    end for
28:  end for{ACM & ADM}
29:   $\mathcal{L}_{TAM} \leftarrow \mathcal{L}(l^L + \alpha m^{ACM} + \beta m^{ADM}, Y^L)$ 
30:   $\theta \leftarrow \theta - \eta \nabla \mathcal{L}_{TAM}$ 
31: end for
32: Output:  $\theta$ 

```

4.2. Anomalous Distribution-Aware Margin

Although ACM can identify nodes that deviated from connectivity patterns, it is not sufficient to recognize whether a deviated node is confused with other classes or simply an outlier node. However, identifying which classes a node is likely to be indistinguishable is necessary to explicitly adjust the margin of confusing classes. Thus, we suggest Anomalous Distribution-aware Margin (ADM), which complementarily adjusts the target class margin according to the relative closeness to the target class compared to the self class (the class of a given node) in NLD space. Since discriminating two classes is more difficult as NLDs of two classes are closer, we design ADM to be sensitive to the distance between the target class and the self class.

The goal for ADM is to decrease the target class margin more intensively as a given node is closer to the target class compared to the self class in NLD space. Even though there are various methods to compute the relative distance, we

calculate the relative distance as in Figure 2 (c). Specifically, let us define the angle between the segment between self class-averaged NLD and node NLD, and the segment between self class-averaged NLD and target class-averaged NLD as $A_{v,t}$. Then, following the law of cosine, we can compute $\cos A_{v,t}$ for given node v as:

$$\cos A_{v,t} = \frac{\text{JS}(\mathcal{D}_{v,:}, \mathcal{C}_{y_v,:})^2 + \text{JS}(\mathcal{C}_{t,:}, \mathcal{C}_{y_v,:})^2 - \text{JS}(\mathcal{D}_{v,:}, \mathcal{C}_{t,:})^2}{2\text{JS}(\mathcal{D}_{v,:}, \mathcal{C}_{y_v,:})\text{JS}(\mathcal{C}_{t,:}, \mathcal{C}_{y_v,:})}, \quad (7)$$

where JS is Jensen-Shannon Divergence. Then, ADM is computed as:

$$m_{v,t}^{ADM} = -\frac{\text{JS}(\mathcal{D}_{v,:}, \mathcal{C}_{y_v,:}) \cos A_{v,t}}{\text{JS}(\mathcal{C}_{t,:}, \mathcal{C}_{y_v,:})}. \quad (8)$$

Note that ADM can also increase the margin if a node is clearly distinguishable with the given local topology.

4.3. Class-wise Temperature for Unlabeled Nodes

Until this point, we have assumed that the label information for neighbors of labeled nodes can be accessible during calculating the NLD \mathcal{D} and Class-wise Connectivity Matrix \mathcal{C} . However, in most node classification scenarios, label information is unknown except for a small set of labeled node set V^L . Therefore, to estimate the class information required when obtaining \mathcal{D} and \mathcal{C} , we exploit the model prediction of the model being trained.

To refine the model predictions, we introduce the class-wise temperature strategy. A similar concept is also adopted in the computer vision domain (Wang et al., 2020a), but we use the reverse direction of temperature compared to an existing method. As investigated in previous research (Park et al., 2021), GNNs are prone to overfit to minor class instances in class-imbalanced settings. This issue can bias the neighbors of minor nodes to be over-confident as minor classes. Inspired by this problem, we assign the temperature T_k to logits of each class k based on its quantity N_k . When N_k is small, the logits of class k are scaled by a large T_k , so the model predictions become more accurate for minor classes. We only use class-wise temperature T_k to obtain the label of neighbors and not for training. The temperature T_k of class k is derived as:

$$\pi_k = \delta \cdot \frac{N_k}{\frac{1}{|\mathcal{Y}|} \sum_{s \in |\mathcal{Y}|} N_s} + (1 - \delta)$$

$$T_k = \frac{1}{\phi(\pi_k + 1 - \max_j \pi_j)}, \quad (9)$$

where ϕ is a hyperparameter and δ is a parameter that determines the sensitivity to imbalance ratio. We fix δ to 0.4 for all experiments.

Table 1. Experimental results of our algorithm TAM and other baselines on three class-imbalanced node classification benchmark datasets (homophilous graphs). We report averaged balanced accuracy (bAcc.) and F1-score with the standard errors for 10 repetitions on three representative GNN architectures.

Dataset	Cora		CiteSeer		PubMed		
	bAcc.	F1	bAcc.	F1	bAcc.	F1	
Imbalance Ratio ($\rho = 10$)							
GCN	Cross Entropy	60.95 ±1.22	59.30 ±1.66	38.21 ±1.12	29.40 ±1.97	65.21 ±1.40	55.43 ±2.79
	Re-Weight	65.52 ±0.84	65.54 ±1.20	44.52 ±1.22	38.85 ±1.62	70.17 ±1.25	66.37 ±1.73
	PC Softmax	67.79 ±0.92	67.39 ±1.08	49.81 ±1.12	45.55 ±1.26	70.20 ±0.60	68.83 ±0.73
	DR-GCN	60.17 ±0.83	59.31 ±0.97	42.64 ±0.75	38.22 ±1.22	65.51 ±0.81	64.95 ±0.53
	GraphSMOTE	66.29 ±0.93	66.30 ±1.25	44.40 ±1.27	39.10 ±1.78	68.51 ±1.14	62.63 ±2.39
	BalancedSoftmax	68.46 ±0.67	68.41 ±0.80	53.70 ±1.40	50.73 ±1.64	72.97 ±0.80	70.80 ±1.11
	+ TAM	69.90 ±0.73	69.89 ±0.89	55.54 ±1.40	54.18 ±1.69	74.13 ±0.70	73.27 ±0.67
	ReNode	67.61 ±0.77	67.27 ±0.91	47.78 ±1.67	42.51 ±2.30	71.59 ±1.70	66.56 ±2.90
	+ TAM	67.18 ±1.32	67.39 ±1.62	48.36 ±1.63	42.48 ±2.10	71.00 ±1.86	67.18 ±3.42
	GraphENS	70.31 ±0.51	70.30 ±0.65	55.42 ±1.74	53.85 ±2.00	71.89 ±0.80	71.07 ±0.66
	+ TAM	71.52 ±0.30	71.71 ±0.45	57.47 ±1.56	56.23 ±1.87	74.01 ±0.73	72.41 ±0.94
	GAT	Cross Entropy	60.82 ±1.27	59.56 ±1.75	41.16 ±1.49	33.71 ±2.02	63.97 ±1.21
Re-Weight		66.72 ±0.80	66.52 ±1.06	45.59 ±1.73	39.43 ±2.03	69.13 ±1.25	64.81 ±1.70
PC Softmax		67.02 ±0.65	66.57 ±0.89	50.70 ±1.73	47.14 ±1.85	72.20 ±0.49	70.95 ±0.82
DR-GCN		59.30 ±0.76	57.79 ±1.03	44.04 ±1.26	39.44 ±1.76	69.56 ±1.01	68.49 ±0.71
GraphSMOTE		66.08 ±1.37	64.92 ±1.66	45.79 ±1.42	39.92 ±2.11	67.86 ±1.58	61.96 ±2.61
BalancedSoftmax		67.79 ±0.54	67.73 ±0.68	52.83 ±1.25	49.96 ±1.69	72.56 ±0.66	69.90 ±1.13
+ TAM		69.00 ±0.62	69.25 ±0.64	56.32 ±1.65	54.99 ±2.10	73.37 ±0.76	72.60 ±0.89
ReNode		68.34 ±1.25	68.59 ±1.51	48.99 ±1.69	43.90 ±2.15	67.55 ±2.20	64.46 ±2.56
+ TAM		68.39 ±1.15	68.69 ±1.36	48.81 ±1.26	44.40 ±1.91	69.00 ±2.39	67.46 ±2.66
GraphENS		70.45 ±0.49	69.84 ±0.53	52.35 ±1.57	49.35 ±2.31	71.99 ±0.72	70.59 ±0.85
+ TAM		70.00 ±0.60	69.93 ±0.76	55.86 ±1.48	53.85 ±1.98	73.42 ±0.77	71.95 ±1.01
SAGE		Cross Entropy	60.41 ±1.09	58.57 ±1.34	44.41 ±1.21	38.20 ±1.68	67.34 ±0.93
	Re-Weight	63.76 ±0.98	63.46 ±1.22	46.64 ±1.92	41.38 ±2.76	69.03 ±1.17	64.01 ±2.18
	PC Softmax	64.03 ±0.81	63.73 ±0.99	50.14 ±1.89	47.38 ±2.13	71.39 ±0.84	70.25 ±1.02
	DR-GCN	61.05 ±1.17	60.17 ±1.23	46.00 ±0.93	47.73 ±1.12	69.23 ±0.68	67.35 ±0.90
	GraphSMOTE	61.75 ±0.07	60.90 ±1.22	42.51 ±1.54	34.93 ±1.67	66.11 ±1.12	61.17 ±2.10
	BalancedSoftmax	66.10 ±0.54	66.26 ±0.63	54.18 ±1.79	52.67 ±1.96	70.32 ±0.92	68.81 ±0.99
	+ TAM	68.01 ±0.56	68.14 ±0.58	55.47 ±1.33	54.87 ±1.33	72.91 ±0.62	72.61 ±0.68
	ReNode	65.18 ±0.96	65.46 ±1.28	48.58 ±1.87	43.62 ±2.17	69.58 ±1.12	67.21 ±1.72
	+ TAM	65.54 ±1.02	65.96 ±1.30	49.53 ±1.94	45.96 ±2.52	69.96 ±1.40	66.34 ±2.53
	GraphENS	68.65 ±0.49	68.79 ±0.21	53.43 ±1.29	51.70 ±1.46	70.45 ±0.82	68.96 ±1.34
	+ TAM	69.12 ±0.81	69.16 ±0.87	55.43 ±1.32	53.82 ±1.49	72.31 ±1.05	71.21 ±1.31

5. Experiments

5.1. Experimental Settings

Datasets To show the effectiveness of our algorithm on both homophilous and heterophilous graphs, we evaluate our method on homophilous graphs: Cora, CiteSeer, and PubMed (Sen et al., 2008), and heterophilous graphs: Wisconsin¹, Chameleon, and Squirrel (Rozemberczki et al., 2021). We utilize the splits used in Yang et al. (2016) for Cora, CiteSeer, and PubMed, and in Pei et al. (2019) for Wisconsin, Chameleon, and Squirrel. To construct class-imbalanced datasets, we adopt the step imbalance method following Zhao et al. (2021); Park et al. (2021). Specifically, we select minor classes as half the number of classes ($|\mathcal{Y}|/2$) and alter labeled nodes of minor classes to unlabeled nodes randomly until the number of nodes in each minor class equals the ratio of the number of major nodes in the most

frequent class to imbalance ratio ($\frac{\max_{k \in \mathcal{Y}} N_k}{\rho}$). In this paper, we adopt imbalance ratios of five and ten. For Wisconsin, we do not modify the number of nodes in each class since the train splits in Pei et al. (2019) is already highly imbalanced (11.63). The detailed experimental settings such as evaluation protocol and implementation details of our algorithm are described in Appendix B.

5.2. Baselines

To validate our method, we first select vanilla (cross entropy) and re-weight (Japkowicz & Stephen, 2002) as baselines. We also adopt competitive baselines in the vision domain and node classification. For the vision domain, Balanced Softmax (Ren et al., 2020) and PC Softmax (Hong et al., 2021) are introduced as the representative algorithms of loss modification and post-hoc correction, respectively. In node classification, we compare our method with DR-GCN (Shi et al., 2020), GraphSMOTE (Zhao et al., 2021), and GraphENS (Park et al., 2021). For ReNode (Chen et al.,

¹<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/>

Table 2. Experimental results of our algorithm TAM and other baselines on three class-imbalanced node classification benchmark datasets (heterophilous graphs). We report averaged balanced accuracy (bAcc.) and F1-score with the standard errors for 10 repetitions on three representative GNN architectures.

	Dataset	Chameleon		Squirrel		Wisconsin	
		$(\rho = 5)$		$(\rho = 5)$		$(\rho = 11.63)$	
	Imbalance Ratio	bAcc.	F1	bAcc.	F1	bAcc.	F1
GCN	Cross Entropy	33.21 ±0.88	31.74 ±0.85	24.06 ±0.43	20.32 ±0.59	29.73 ±1.29	27.51 ±1.45
	Re-Weight	37.85 ±0.95	37.46 ±0.95	27.40 ±0.52	26.76 ±0.42	44.13 ±3.08	40.74 ±3.27
	PC Softmax	37.98 ±0.83	36.55 ±0.88	27.37 ±0.33	26.67 ±0.27	30.90 ±3.10	28.15 ±2.16
	DR-GCN	34.12 ±0.89	31.78 ±1.02	24.67 ±0.39	19.54 ±0.71	29.44 ±1.36	27.08 ±1.37
	GraphENS	41.13 ±0.59	39.61 ±0.77	26.79 ±0.43	26.35 ±0.41	44.09 ±2.96	40.86 ±3.29
	BalancedSoftmax	38.33 ±0.73	37.54 ±0.68	27.86 ±0.42	27.04 ±0.35	31.51 ±2.28	28.82 ±1.99
	+ TAM	41.48 ±0.93	40.43 ±1.02	28.67 ±0.54	27.84 ±0.45	35.97 ±3.68	31.41 ±2.16
	ReNode	37.43 ±0.90	36.75 ±0.89	28.38 ±0.46	27.81 ±0.44	36.85 ±2.14	33.30 ±1.79
	+ TAM	40.28 ±0.85	39.27 ±0.80	28.19 ±0.36	27.55 ±0.39	36.28 ±2.87	34.10 ±2.38
	GraphSMOTE	42.65 ±0.59	41.56 ±0.53	28.29 ±0.60	27.89 ±0.61	45.36 ±4.21	40.91 ±4.39
+ TAM	42.77 ±0.62	41.78 ±0.62	29.18 ±0.46	28.84 ±0.44	45.59 ±3.73	40.76 ±4.01	
GAT	Cross Entropy	34.33 ±0.74	31.54 ±0.95	24.89 ±0.37	21.33 ±0.52	32.15 ±2.72	30.92 ±2.76
	Re-Weight	39.63 ±0.49	39.08 ±0.50	26.49 ±0.41	25.92 ±0.41	42.15 ±2.33	37.66 ±2.27
	PC Softmax	41.47 ±0.78	40.51 ±0.89	27.31 ±0.51	26.74 ±0.50	41.89 ±3.95	38.03 ±3.35
	DR-GCN	36.85 ±0.77	34.61 ±0.62	25.40 ±0.43	22.83 ±0.59	33.93 ±2.34	31.75 ±2.50
	GraphENS	40.66 ±1.13	39.49 ±1.10	26.87 ±0.43	26.78 ±0.41	40.93 ±2.78	37.43 ±2.74
	BalancedSoftmax	41.47 ±0.71	40.52 ±0.78	26.66 ±0.39	25.97 ±0.35	41.20 ±3.08	37.93 ±2.99
	+ TAM	42.56 ±0.59	41.40 ±0.74	27.75 ±0.44	27.23 ±0.45	48.44 ±3.32	43.71 ±2.91
	ReNode	40.41 ±0.56	39.85 ±0.60	26.89 ±0.45	26.40 ±0.46	40.88 ±2.84	37.13 ±2.74
	+ TAM	41.53 ±0.35	40.76 ±0.50	26.53 ±0.40	26.00 ±0.42	46.64 ±3.35	41.60 ±3.02
	GraphSMOTE	42.27 ±0.51	41.43 ±0.54	28.17 ±0.56	27.38 ±0.66	40.77 ±2.24	38.96 ±2.48
+ TAM	42.83 ±0.82	42.26 ±0.83	28.44 ±0.33	28.02 ±0.37	41.82 ±2.94	38.23 ±3.13	
SAGE	Cross Entropy	35.76 ±0.57	33.55 ±0.68	27.59 ±0.31	25.87 ±0.14	68.76 ±3.57	64.16 ±3.26
	Re-Weight	40.85 ±0.69	40.40 ±0.71	29.88 ±0.48	27.59 ±0.42	68.13 ±3.19	63.45 ±2.27
	PC Softmax	42.90 ±0.85	42.34 ±0.87	30.54 ±0.62	29.41 ±0.65	70.57 ±3.34	67.13 ±2.91
	DR-GCN	39.58 ±0.58	38.37 ±0.72	28.78 ±0.50	25.01 ±0.70	69.30 ±1.99	64.60 ±2.00
	GraphENS	37.77 ±0.69	37.36 ±0.67	25.31 ±0.45	25.15 ±0.40	66.23 ±3.17	60.89 ±2.97
	BalancedSoftmax	43.03 ±0.98	42.40 ±0.96	30.29 ±0.48	29.37 ±0.44	67.50 ±2.47	63.95 ±2.43
	+ TAM	43.77 ±0.90	42.95 ±0.90	30.70 ±0.59	29.82 ±0.53	68.62 ±3.47	65.23 ±2.97
	ReNode	40.74 ±0.75	40.45 ±0.77	29.75 ±0.47	28.49 ±0.51	72.52 ±2.13	69.15 ±3.18
	+ TAM	41.45 ±0.86	41.00 ±0.86	29.79 ±0.39	28.75 ±0.39	70.97 ±2.05	66.95 ±2.62
	GraphSMOTE	34.43 ±0.86	31.16 ±1.21	26.26 ±0.31	23.73 ±0.30	65.14 ±3.84	62.53 ±3.40
+ TAM	36.94 ±0.88	35.00 ±0.93	26.70 ±0.48	24.71 ±0.41	64.07 ±3.15	62.65 ±2.96	

2021), we search the best algorithm in each setting among re-weight (Japkowicz & Stephen, 2002), focal loss (Lin et al., 2017), and class-balanced weight (Cui et al., 2019). To show that our algorithm could improve the performance of many imbalance handling methods, we combine our methods with competitive approaches in each domain: Balanced Softmax, ReNode, and GraphENS. Since GraphSMOTE shows superior performance only in heterophilous graphs than GraphENS, we combine our methods with GraphSMOTE rather than GraphENS in Wisconsin, Chameleon, and Squirrel. The implementation details of baselines are suggested in Appendix B.4.

5.3. Main Results

Homophilous graphs In Table 1, we report the averaged balanced accuracy (bAcc.) and F1 score with standard error for the baselines and ours on three homogeneously-

connected citation networks. Existing imbalance approaches integrated with TAM achieve the best performance for all 9 settings (3 datasets with 3 architectures). Since our method can detect the nodes of the topological boundaries and adjust their logits based on ACM module (Section 4.1), TAM successfully brings enhancement of imbalance-handling performance on homophilous graphs. We also confirm that TAM consistently improves the performance over various types of imbalance handling strategies such as logit adjustment (Balanced Softmax), node weighting (ReNode), and over-sampling (GraphENS). Note that we validate our method on another imbalance ratio ($\rho = 5$) and get consistent results. These results are deferred to Appendix A.2.

Heterophilous graphs As shown in Table 2, Baselines equipped with TAM also exhibit superior imbalance-handling performances in most cases for three heterogeneously-connected graphs. The rationale for these results is that TAM could model the class-wise

connectivity pattern of heterophilous graphs and identify the outlier nodes for each class using class-averaged statistic matrix \mathcal{C} . We also verify that TAM indeed reduces the false positives stemming from abnormally-connected minor nodes (see **Reducing False Positives** in Section 5.3). Note that we also provide the comparison of our method with other baselines under another imbalance ratio ($\rho = 10$). These results are deferred to Appendix A.3.

Reducing False Positives In Section 3, we have observed that false positives on minor classes are highly located near the minor nodes which have anomalous connectivity. To verify that TAM can alleviate this false positive issue, we compute the ratio of false positives on neighbor nodes of anomalous minor nodes using GNNs trained with TAM. As indicated in Figure 1 (c) and (d), TAM steadily reduces the ratio of false positives without dependence of imbalance handling approaches on both homophilous and heterophilous graphs.

5.4. Analysis of TAM

Ablation study To validate each component of our method, we conduct an ablation study on three node classification datasets. First, we compare ACM with ACM without using class-connectivity matrix \mathcal{C} to justify exploiting class connectivity statistics. Specifically ‘ACM w/o \mathcal{C} ’ solely uses $\frac{D_{v,t}}{D_{v,y_v}}$ to determine the ACM margins. In Table 3, ‘ACM w/o \mathcal{C} ’ consistently shows inferior performances compared to our ACM module implying that the class connectivity matrix plays an important role in modeling connectivity patterns.

We also verify our three key modules: ACM, ADM, and Class-wise Temperature (Cls-wise T_k). As shown in Table 3, each component of our method can bring performance improvement by *itself*. From these results, We carefully expect that our individual module contributes to alleviating the adversarial byproduct in the imbalance-handling process. Note that we use Balanced Softmax as a base imbalance-handling approach in the ablation study.

Table 3. Ablation Study of our method TAM.

Modules	ACM w/o \mathcal{C}	ACM	ADM	Cls-wise T_k	F1
CiteSeer + GCN	X	X	X	X	50.73 \pm 1.40
	✓	X	X	X	50.28 \pm 1.16
	X	✓	X	X	53.54 \pm 1.99
	X	X	✓	X	51.95 \pm 1.71
	X	✓	✓	X	54.08 \pm 1.81
	X	✓	✓	✓	55.54 \pm 1.40
PubMed + GCN	X	X	X	X	70.80 \pm 1.11
	✓	X	X	X	71.29 \pm 1.16
	X	✓	X	X	72.59 \pm 0.93
	X	X	✓	X	71.47 \pm 1.03
	X	✓	✓	X	72.84 \pm 1.16
	X	✓	✓	✓	73.27 \pm 2.39
chameleon + SAGE	X	X	X	X	42.40 \pm 0.96
	✓	X	X	X	42.34 \pm 0.88
	X	✓	X	X	42.53 \pm 1.07
	X	X	✓	X	42.75 \pm 1.01
	X	✓	✓	X	42.98 \pm 1.00
	X	✓	✓	✓	42.95 \pm 0.90

Adjusting both minor and major nodes TAM also adjusts the margins for nodes other than minor class nodes (*i.e.* major class nodes). The rationale for this design is that, even though the impact of anomalous major nodes is far weaker, they still occur false positives to their neighbors. To validate our principle, we explore the performances of the TAM that *only* regulates the margins of minor nodes on Cora and Chameleon datasets (Figure 3). These results support our choice in that adjusting only minor nodes shows sub-optimal performances while it brings considerable improvements compared to baselines.

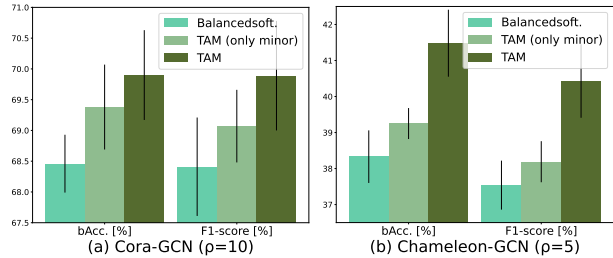


Figure 3. Comparison of balanced accuracy and F1-score. For (a) and (b), performance gains by adjusting all classes compared to adjusting only minor nodes are shown.

Sensitivity to hyperparameters α and β The two intensity terms - α and β - for ACM and ADM have been introduced in Section 4. We investigate the sensitivity of performance to ACM intensity α and ADM intensity β in Figure 4. We observe the performance drops when α or β have extreme values. We believe that small α or β might not sufficiently decrease false positives induced by anomalous nodes. In contrast, large α or β would increase false negatives of minor classes.

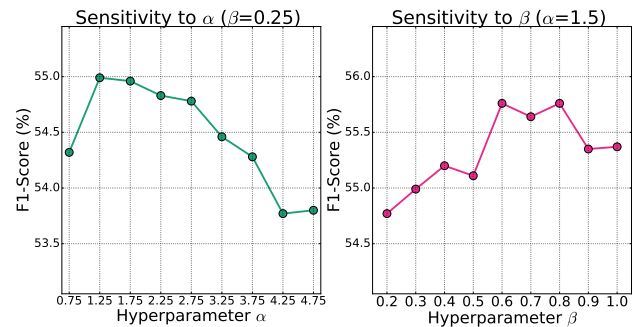


Figure 4. Sensitivity graphs on CiteSeer ($\rho=10$). Green and red graphs show the performance change as ACM intensity α and ADM intensity β increases, respectively.

6. Related Work

Imbalance handling in the vision domain The key objective of solving class-imbalanced problem is to mitigate the bias to major classes induced by the label distribution

in the training set. There are four major branches for imbalance handling: re-sampling methods, ensemble approaches, post-hoc correction, and loss modification. Re-sampling methods (Chawla et al., 2002; Kang et al., 2019; Zhang et al., 2021; Wang et al., 2021a) sample minor class data more frequently or augment minor class diversely. Ensemble approaches (Zhou et al., 2020; Xiang et al., 2020; Wang et al., 2020a; Cai et al., 2021) train multiple head classifiers and collaboratively inference test data with these classifiers. Post-hoc correction algorithms (Kang et al., 2019; Tian et al., 2020; Menon et al., 2020; Hong et al., 2021) reward minor classes only in the inference. Loss modification methods (Cui et al., 2019; Menon et al., 2020; Tang et al., 2020) compensate minor classes in the training phase by assigning more weights on the loss of minor class data (Japkowicz & Stephen, 2002; Lin et al., 2017; Cui et al., 2019; Xu et al., 2020) or expanding the margin of minor classes to major classes (Cao et al., 2019; Tan et al., 2020; Ren et al., 2020; Menon et al., 2020; Wang et al., 2021c;b). However, it is challenging to directly apply these methods to node classification due to connections between nodes in graphs.

Imbalance handling in node classification Recently, to utilize topology information in class-imbalanced node classification, several methods (Shi et al., 2020; Wang et al., 2020b; Zhao et al., 2021; Liu et al., 2021; Qu et al., 2021; Chen et al., 2021) are proposed. DR-GCN (Shi et al., 2020) introduces conditional GAN to generate virtual nodes which are similar to adjacent node features of source nodes. GraphSMOTE (Zhao et al., 2021) synthesizes the features of minor nodes by interpolating two minor nodes as SMOTE (Chawla et al., 2002) does and determines edges of synthesized nodes with edge predictor. ImGAGN (Qu et al., 2021) produces virtual minor nodes by mixing all minor nodes and these virtual nodes connect only to minor nodes according to the generated weight matrix. Since ImGAGN is designed to mainly target binary classification, the extension of ImGAGN to multi-class classification is non-trivial. GraphENS (Park et al., 2021) generates diverse minor nodes by mixing minor nodes with (other class) nodes in neighbor distribution level utilizing model prediction and node feature level using saliency map. However, these approaches do not consider the topologies of nodes when rewarding minor classes. ReNode (Chen et al., 2021) reduces loss weights of nearby nodes at topological class boundaries, but it is only effective when graphs are homophilous and does not consider class-pair connectivity. Compared to other imbalance handling methods, TAM reflects class-pair connectivity on logit adjustments and works on both homophilous and heterophilous graphs.

Algorithms for heterophilous graphs Although our method mainly targets the class-imbalanced problem in node classification, handling heterophilous graphs is related in

that we consider connectivity pattern of each class. Since many GNNs are designed under homophily assumption, many GNNs fail in heterophilous graphs. Recently, to overcome this limitation, various algorithms (Abu-El-Haija et al., 2019; Pei et al., 2019; Zhu et al., 2020; 2021; Jin et al., 2021; Lim et al., 2021; Yang et al., 2021) are suggested. CPGNN (Zhu et al., 2021) models Compatibility Matrix and conducts propagation with this matrix. DMP (Yang et al., 2021) determines edge attributes for each edge and propagates node features with these attributes.

7. Conclusion

In class-imbalanced node classification, we found that compensating minor nodes, which deviate from class-wise connectivity patterns, are prone to induce false positives of minor nodes. From this observation, we proposed TAM to adjust margins node-wisely according to the extent of deviation from connectivity patterns. We show that our algorithm consistently improves competitive imbalance handling methods by simply combining TAM on both homophilous and heterophilous graphs with various GNN architectures.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST), No.2019-0-01371, Development of brain-inspired AI with human-like intelligence No.2019-0-00075, Artificial Intelligence Innovation Hub, No.2022-0-00713, Meta-learning applicable to real-world problems) and the National Research Foundation of Korea (NRF) grants (No.2018R1A5A1059921) funded by the Korea government (MSIT). This work was also supported by Samsung Electronics Co., Ltd (No.IO201214-08133-01).

References

- Abu-El-Haija, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., Ver Steeg, G., and Galstyan, A. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pp. 21–29. PMLR, 2019.
- Cai, J., Wang, Y., and Hwang, J.-N. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 112–121, 2021.
- Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-

- aware margin loss. *Advances in neural information processing systems*, 32, 2019.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.
- Chen, D., Lin, Y., Zhao, G., Ren, X., Li, P., Zhou, J., and Sun, X. Topology-imbalance learning for semi-supervised node classification. *Advances in Neural Information Processing Systems*, 34:29885–29897, 2021.
- Chen, J., Ma, T., and Xiao, C. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*, 2018.
- Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.
- Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- Hong, Y., Han, S., Choi, K., Seo, S., Kim, B., and Chang, B. Disentangling label distribution for long-tailed visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 6626–6636. Computer Vision Foundation / IEEE, 2021.
- Japkowicz, N. and Stephen, S. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, 2002.
- Jin, D., Yu, Z., Huo, C., Wang, R., Wang, X., He, D., and Han, J. Universal graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:10654–10664, 2021.
- Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2019.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.
- Lim, D., Hohne, F., Li, X., Huang, S. L., Gupta, V., Bhalerao, O., and Lim, S. N. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., and He, Q. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, pp. 3168–3177, 2021.
- Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2020.
- Mohammadrezaei, M., Shiri, M. E., and Rahmani, A. M. Identifying fake accounts on social networks based on graph analysis and classification algorithms. *Secur. Commun. Networks*, 2018:5923156:1–5923156:8, 2018.
- Park, J., Song, J., and Yang, E. Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International Conference on Learning Representations*, 2021.
- Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., and Yang, B. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2019.
- Qu, L., Zhu, H., Zheng, R., Shi, Y., and Yin, H. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1390–1398, 2021.
- Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.
- Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale attributed node embedding. *J. Complex Networks*, 9(2), 2021.
- Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008.
- Shi, M., Tang, Y., Zhu, X., Wilson, D., and Liu, J. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020.
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

- Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., and Yan, J. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11662–11671, 2020.
- Tang, K., Huang, J., and Zhang, H. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020.
- Tian, J., Liu, Y.-C., Glaser, N., Hsu, Y.-C., and Kira, Z. Posterior re-calibration for imbalanced datasets. *Advances in Neural Information Processing Systems*, 33:8101–8113, 2020.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- Wang, J., Lukasiewicz, T., Hu, X., Cai, J., and Xu, Z. Rsg: A simple but effective module for learning imbalanced datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3784–3793, 2021a.
- Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C. C., and Lin, D. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9695–9704, 2021b.
- Wang, T., Zhu, Y., Zhao, C., Zeng, W., Wang, J., and Tang, M. Adaptive class suppression loss for long-tail object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3103–3112, 2021c.
- Wang, X., Lian, L., Miao, Z., Liu, Z., and Yu, S. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2020a.
- Wang, Z., Ye, X., Wang, C., Cui, J., and Yu, P. Network embedding with completely-imbalanced labels. *IEEE Transactions on Knowledge and Data Engineering*, 2020b.
- Welling, M. and Kipf, T. N. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.
- Xiang, L., Ding, G., and Han, J. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European Conference on Computer Vision*, pp. 247–263. Springer, 2020.
- Xu, Z., Dan, C., Khim, J., and Ravikumar, P. Class-weighted classification: Trade-offs and robust approaches. In *International Conference on Machine Learning*, pp. 10544–10554. PMLR, 2020.
- Yang, L., Li, M., Liu, L., Wang, C., Cao, X., Guo, Y., et al. Diverse message passing for attribute with heterophily. *Advances in Neural Information Processing Systems*, 34: 4751–4763, 2021.
- Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.
- Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.
- Zhang, Y., Wei, X.-S., Zhou, B., and Wu, J. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 3447–3455, 2021.
- Zhao, T., Zhang, X., and Wang, S. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841, 2021.
- Zhou, B., Cui, Q., Wei, X.-S., and Chen, Z.-M. 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*, pp. 9719–9728, 2020.
- Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.
- Zhu, J., Rossi, R. A., Rao, A., Mai, T., Lipka, N., Ahmed, N. K., and Koutra, D. Graph neural networks with heterophily. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11168–11176, 2021.

A. Additional Experimental Results

In this section, we provide additional experimental results which are omitted due to the space constraints.

A.1. The False Negative Rates in Section 3

To prove that TAM does not significantly sacrifice the false negative rates for reducing false positive cases, we present the false negative rates of the results in Section 3. As shown in Table 4, our approach effectively reduces the false positive cases for major nodes connected with non-typical minor nodes while increasing false negative rates slightly. These consistent results over multiple benchmark datasets strengthen our claim that TAM can successfully mitigate the false positive cases of major classes.

Table 4. Comparison of false positive rates (FPR) and false negative rates (FNR) near normal minor nodes and anomalously-connected minor nodes.

Method (GCN)	Cora, $\rho=10$		CiteSeer, $\rho=10$		Chameleon, $\rho=5$		Wisconsin, $\rho=12$	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
Re-Weight	47.68	9.97	64.63	4.74	48.57	29.06	63.90	36.40
+ TAM	29.77(-17.91)	15.83(+5.86)	57.84(-6.79)	4.78(+0.04)	35.78(-12.79)	33.73(+4.67)	34.63(-29.27)	39.67(+3.27)
BalancedSoftmax	56.69	5.91	65.86	4.95	41.85	31.37	52.19	45.77
+ TAM	37.97(-18.72)	11.78(+5.87)	56.32(-9.54)	5.85(+0.90)	36.01(-5.84)	28.27(-3.10)	40.14(-12.05)	40.46(-5.31)
GraphSMOTE	45.19	10.51	62.55	3.80	49.81	19.48	52.90	47.33
+ TAM	29.82(-15.37)	15.58(+5.07)	54.14(-8.41)	5.59(+1.79)	48.73(-1.08)	16.45(-3.03)	48.70(-4.20)	50.30(+2.97)

A.2. The Results of Three Benchmark Datasets (Homophilous Graphs)

In the main paper, we only report the comparison of our method with other baselines on Cora, CiteSeer, and PubMed (Sen et al., 2008) with the high imbalance ratio ($\rho = 10$) due to the space limitation. To show that our method is also effective under low imbalance ratio, we provide the results on Cora, CiteSeer, and PubMed with relatively low imbalance ratio ($\rho = 5$) over three GNN architectures: GCN, GAT, and SAGE in Table 5. We observe consistent results with the main paper in that our method improves various imbalance handling algorithms by combining ours with these baselines.

A.3. The Results of Two Benchmark Datasets (Heterophilous Graphs)

We also provide the additional experimental results on heterophilous graphs: Chameleon and Squirrel (Rozemberczki et al., 2021) with the different imbalance ratio ($\rho = 10$). In Table 6, we affirm consistent results with the main paper in that our method improves various imbalance handling algorithms by combining ours with these baselines in most cases. For the Squirrel dataset, our method shows comparable performance with baselines. We conjecture that low accuracy of baselines in Squirrel induces erroneous estimation of neighbor label distribution and class-wise connectivity matrix, resulting in insignificant improvements.

Table 5. Experimental results of our algorithm TAM and other baselines on three class-imbalanced node classification benchmark datasets (homophilous graphs). We report averaged balanced accuracy (bAcc.) and F1-score with the standard errors for 10 repetitions on three representative GNN architectures.

	Dataset	Cora		CiteSeer		PubMed		
		Imbalance Ratio ($\rho = 5$)	bAcc.	F1	bAcc.	F1	bAcc.	F1
GCN	Cross Entropy		69.15 ± 0.52	69.36 ± 0.75	48.56 ± 1.70	44.56 ± 2.30	71.89 ± 1.04	67.59 ± 1.54
	Re-Weight		71.86 ± 0.76	72.23 ± 0.75	54.32 ± 1.33	52.37 ± 1.65	73.91 ± 0.95	71.70 ± 0.77
	PC Softmax		72.69 ± 0.58	72.90 ± 0.56	58.86 ± 1.27	57.33 ± 1.46	74.13 ± 0.73	72.84 ± 0.72
	DR-GCN		67.56 ± 0.56	67.29 ± 0.73	50.47 ± 1.17	47.73 ± 1.59	70.36 ± 0.66	68.22 ± 0.93
	GraphSMOTE		73.46 ± 0.84	73.07 ± 0.67	54.79 ± 1.21	53.52 ± 1.46	72.49 ± 0.79	69.80 ± 1.17
	BalancedSoftmax		73.63 ± 0.68	73.40 ± 0.67	60.13 ± 1.48	59.30 ± 1.61	75.26 ± 0.58	73.83 ± 0.66
	+ TAM		73.75 ± 0.66	73.74 ± 0.66	60.97 ± 1.02	60.46 ± 0.98	76.03 ± 0.96	75.16 ± 1.10
	ReNode		74.91 ± 0.57	75.37 ± 0.62	58.01 ± 1.52	56.63 ± 1.85	73.99 ± 0.88	72.04 ± 1.01
	+ TAM		74.77 ± 0.42	75.57 ± 0.43	58.57 ± 1.59	57.56 ± 1.80	75.22 ± 1.05	74.22 ± 1.41
	GraphENS		75.68 ± 0.58	75.47 ± 0.58	62.24 ± 1.10	61.70 ± 1.11	74.30 ± 0.59	73.53 ± 0.52
+ TAM		75.72 ± 0.64	75.93 ± 0.62	63.01 ± 0.87	62.56 ± 0.80	75.62 ± 0.55	75.28 ± 0.51	
GAT	Cross Entropy		68.12 ± 0.51	68.81 ± 0.62	51.43 ± 1.67	48.85 ± 2.13	70.65 ± 1.11	66.73 ± 1.69
	Re-Weight		73.24 ± 0.81	72.40 ± 0.96	55.40 ± 1.59	53.97 ± 1.62	72.94 ± 0.77	70.59 ± 1.10
	PC Softmax		71.24 ± 0.52	71.53 ± 0.62	58.83 ± 1.28	57.45 ± 1.37	74.72 ± 0.69	72.66 ± 0.82
	DR-GCN		66.43 ± 0.72	66.31 ± 0.84	51.48 ± 1.63	49.48 ± 2.31	72.41 ± 0.57	71.74 ± 0.63
	GraphSMOTE		72.96 ± 0.67	72.39 ± 0.83	55.38 ± 1.52	53.72 ± 1.88	72.94 ± 0.85	70.65 ± 1.23
	BalancedSoftmax		72.50 ± 0.60	71.96 ± 0.67	59.72 ± 1.15	58.79 ± 1.18	73.38 ± 0.74	72.47 ± 0.83
	+ TAM		72.72 ± 0.66	72.78 ± 0.81	62.19 ± 0.87	61.55 ± 0.86	74.71 ± 0.74	74.14 ± 0.80
	ReNode		74.34 ± 0.69	74.77 ± 0.52	58.69 ± 1.64	57.05 ± 1.94	73.85 ± 0.96	71.79 ± 1.16
	+ TAM		75.07 ± 0.62	75.05 ± 0.69	59.11 ± 1.41	57.77 ± 1.55	73.79 ± 0.91	72.30 ± 0.88
	GraphENS		74.92 ± 0.57	74.58 ± 0.61	59.40 ± 1.08	58.98 ± 1.11	73.93 ± 0.66	72.99 ± 0.90
+ TAM		74.82 ± 0.40	75.13 ± 0.43	62.23 ± 0.79	61.89 ± 0.79	75.05 ± 0.65	74.47 ± 0.66	
SAGE	Cross Entropy		66.58 ± 0.78	66.43 ± 0.86	51.50 ± 1.55	49.01 ± 2.09	71.55 ± 0.74	70.38 ± 0.73
	Re-Weight		71.59 ± 0.77	71.91 ± 0.87	56.65 ± 1.50	55.38 ± 1.72	72.22 ± 0.95	70.33 ± 0.99
	PC Softmax		71.55 ± 0.72	71.22 ± 0.80	56.85 ± 1.52	55.27 ± 1.73	73.21 ± 0.46	72.33 ± 0.63
	DR-GCN		66.20 ± 0.68	66.03 ± 0.73	54.31 ± 1.42	53.36 ± 1.44	71.43 ± 0.83	70.22 ± 1.03
	GraphSMOTE		69.66 ± 0.78	69.98 ± 0.89	52.90 ± 1.19	50.70 ± 1.72	70.71 ± 1.39	69.12 ± 1.68
	BalancedSoftmax		71.50 ± 0.43	71.68 ± 0.47	58.49 ± 1.32	57.91 ± 1.44	72.82 ± 0.53	71.64 ± 0.62
	+ TAM		72.86 ± 0.39	72.81 ± 0.44	61.09 ± 1.28	60.41 ± 1.22	74.37 ± 0.56	73.98 ± 0.61
	ReNode		72.92 ± 0.48	73.58 ± 0.50	58.36 ± 1.69	57.09 ± 2.08	73.51 ± 1.04	71.98 ± 1.07
	+ TAM		73.09 ± 0.34	73.76 ± 0.35	58.69 ± 1.28	57.50 ± 1.51	73.84 ± 0.64	72.86 ± 0.92
	GraphENS		73.43 ± 0.62	73.47 ± 0.74	60.17 ± 1.33	59.71 ± 1.32	73.32 ± 0.75	72.74 ± 0.68
+ TAM		75.27 ± 0.37	75.36 ± 0.59	62.40 ± 1.10	61.96 ± 1.05	73.93 ± 0.60	73.74 ± 0.47	

Table 6. Experimental results of our algorithm TAM and other baselines on two class-imbalanced node classification benchmark datasets (heterophilous graphs). We report averaged balanced accuracy (bAcc.) and F1-score with the standard errors for 10 repetitions on three representative GNN architectures.

	Dataset	Chameleon		Squirrel		
		Imbalance Ratio ($\rho = 10$)	bAcc.	F1	bAcc.	F1
GCN	Cross Entropy		31.52 \pm 0.72	29.30 \pm 0.73	24.76 \pm 0.37	18.57 \pm 0.27
	Re-Weight		36.07 \pm 0.87	35.61 \pm 0.81	26.92 \pm 0.53	25.04 \pm 0.59
	PC Softmax		36.86 \pm 1.04	36.24 \pm 1.01	26.49 \pm 0.59	25.73 \pm 0.49
	DR-GCN		33.34 \pm 0.81	29.60 \pm 0.79	23.34 \pm 0.43	18.20 \pm 0.49
	GraphENS		41.54 \pm 0.63	40.19 \pm 0.68	26.75 \pm 0.35	26.31 \pm 0.27
	BalancedSoftmax		36.47 \pm 0.89	35.94 \pm 0.85	27.32 \pm 0.52	26.42 \pm 0.41
	+ TAM		38.85 \pm 1.00	37.44 \pm 0.96	27.81 \pm 0.45	27.25 \pm 0.45
	ReNode		34.26 \pm 1.13	33.66 \pm 1.09	25.42 \pm 0.34	24.55 \pm 0.41
	+ TAM		38.01 \pm 0.97	36.92 \pm 0.94	26.41 \pm 0.36	25.87 \pm 0.43
	GraphSMOTE		41.50 \pm 0.82	40.80 \pm 0.79	27.14 \pm 0.49	26.67 \pm 0.53
	+ TAM		42.80 \pm 0.89	41.91 \pm 0.88	28.30 \pm 0.46	27.81 \pm 0.48
	GAT	Cross Entropy		32.41 \pm 0.70	27.33 \pm 0.94	24.69 \pm 0.39
Re-Weight			35.72 \pm 0.65	34.19 \pm 0.74	25.79 \pm 0.52	24.32 \pm 0.62
PC Softmax			38.32 \pm 0.88	37.46 \pm 0.84	26.52 \pm 0.31	25.71 \pm 0.44
DR-GCN			34.84 \pm 0.72	31.53 \pm 0.86	24.69 \pm 0.46	21.81 \pm 0.42
GraphENS			39.71 \pm 0.55	38.75 \pm 0.60	26.55 \pm 0.49	26.00 \pm 0.52
BalancedSoftmax			39.27 \pm 0.83	38.53 \pm 0.87	26.09 \pm 0.43	25.28 \pm 0.38
+ TAM			41.40 \pm 0.57	40.25 \pm 0.72	26.91 \pm 0.36	26.19 \pm 0.38
ReNode			37.95 \pm 0.78	37.09 \pm 0.87	26.14 \pm 0.52	25.47 \pm 0.52
+ TAM			37.57 \pm 0.97	36.11 \pm 0.96	26.08 \pm 0.41	25.39 \pm 0.37
GraphSMOTE			40.18 \pm 0.67	39.43 \pm 0.76	27.10 \pm 0.49	26.63 \pm 0.63
+ TAM			41.19 \pm 0.55	40.51 \pm 0.68	26.56 \pm 0.46	25.74 \pm 0.47
SAGE		Cross Entropy		32.07 \pm 0.48	25.33 \pm 0.73	25.55 \pm 0.41
	Re-Weight		36.49 \pm 1.21	34.84 \pm 1.30	29.83 \pm 0.59	25.88 \pm 0.42
	PC Softmax		40.71 \pm 0.82	39.95 \pm 0.98	29.23 \pm 0.50	28.19 \pm 0.54
	DR-GCN		37.24 \pm 0.79	34.37 \pm 0.97	28.77 \pm 0.70	22.32 \pm 0.96
	GraphENS		34.91 \pm 0.68	33.47 \pm 0.81	24.09 \pm 0.36	23.03 \pm 0.32
	BalancedSoftmax		40.76 \pm 0.99	40.27 \pm 1.05	30.07 \pm 0.44	29.12 \pm 0.42
	+ TAM		41.19 \pm 1.08	40.41 \pm 1.13	29.91 \pm 0.50	28.56 \pm 0.58
	ReNode		37.07 \pm 1.02	36.02 \pm 0.99	29.48 \pm 0.64	26.09 \pm 0.58
	+ TAM		38.24 \pm 0.93	37.23 \pm 1.09	29.77 \pm 0.58	27.72 \pm 0.68
	GraphSMOTE		33.31 \pm 0.63	30.83 \pm 0.67	25.51 \pm 0.43	19.79 \pm 0.49
	+ TAM		33.23 \pm 0.54	30.66 \pm 0.74	25.34 \pm 0.56	22.29 \pm 0.48

B. Detailed Experimental Results

In this section, we describe detailed experimental settings: dataset statistics, evaluation protocol, and implementation details.

B.1. Label Distribution in Training Datasets

We provide the label distribution in class-imbalanced datasets in Table 7.

Table 7. Label distribution in training datasets [%]

Dataset	L ₀	L ₁	L ₂	L ₃	L ₄	L ₅	L ₆
Cora ($\rho = 5$)	21.74	21.74	21.74	21.74	4.35	4.35	4.35
Cora ($\rho = 10$)	23.26	23.26	23.26	23.26	2.33	2.33	2.33
CiteSeer ($\rho = 5$)	27.78	27.78	27.78	5.56	5.56	5.56	-
CiteSeer ($\rho = 10$)	30.30	30.30	30.30	3.03	3.03	3.03	-
PubMed ($\rho = 5$)	45.45	45.45	9.09	-	-	-	-
PubMed ($\rho = 10$)	47.62	47.62	4.76	-	-	-	-
Chameleon ($\rho = 5$)	29.51	29.31	29.03	6.07	6.07	-	-
Chameleon ($\rho = 10$)	31.44	31.22	30.93	3.20	3.20	-	-
Squirrel ($\rho = 5$)	29.54	29.53	29.07	5.98	5.98	-	-
Squirrel ($\rho = 10$)	31.43	31.31	30.93	3.17	3.17	-	-
Wisconsin ($\rho = 11.63$)	46.50	27.92	13.42	8.17	4.00	-	-

B.2. Evaluation Protocol

We validate our algorithm and baselines on various GNN architectures: GCN (Welling & Kipf, 2016), GAT (Veličković et al., 2018) and GraphSAGE (Hamilton et al., 2017). We follow the detailed architecture used in Chen et al. (2021). All GNNs consist of their own convolutional layers with ReLU activation and dropout (Srivastava et al., 2014) is applied with dropping rate of 0.5 before the last layer. For 1-layer GNNs, we do not adopt dropout and we use multi-head attention with 4 heads for GAT. We search the best architecture based on the average of validation accuracy and F1 score among the number of layers $l \in \{1, 2, 3\}$ and the hidden dimension $d \in \{64, 128, 256\}$. For optimization, we train models for 2000 epochs with Adam optimizer (Kingma & Ba, 2015). The initial learning rate is set to 0.01 and the learning rate is halved if the validation loss has not improved for 100 iterations. Weight decay is applied to all learnable parameters as 0.0005 except for the last convolutional layer.

B.3. Implementation Details

For our algorithm, we search the best hyperparameters based on the average of validation accuracy and F1 among the coefficient of ACM term $\alpha \in \{0.25, 0.5, 1.5, 2.5\}$, the coefficient of ADM term $\beta \in \{0.125, 0.25, 0.5\}$, and the minimum temperature of class-wise temperature $\phi \in \{0.8, 1.2\}$. The sensitivity to imbalance ratio of class-wise temperature δ is fixed as 0.4 for all main experiments. We adopt warmup for 5 iterations since we utilize model prediction for unlabeled nodes.

B.4. Baselines

For DR-GCN (Shi et al., 2020), we only utilize the module to keep representations having structure information with conditional GAN and do not adopt the component, which exploits unlabeled nodes, mainly targeting semi-supervised learning for a fair comparison. For GraphSMOTE (Zhao et al., 2021), we select the model whose predicted edges have discrete values among multiple versions in that this setting shows superior performance in many datasets. Since the interpolation happens in the representation space, we search the best architecture for GraphSMOTE among the number of layers $l \in \{2, 3\}$. For ReNode (Chen et al., 2021), we search hyperparameters among lower bound of cosine annealing $w_{min} \in \{0.25, 0.5, 0.75\}$ and upper bound of the cosine annealing $w_{max} \in \{1.25, 1.5, 1.75\}$ as following Chen et al. (2021). PageRank teleport probability is fixed as $\alpha = 0.15$, which is the default setting in the released codes.