A Unified Framework against Topology and Class Imbalance

Junyu Chen
SKLOIS, IIE, CAS
SCS, UCAS
chenjunyu@iie.ac.cn

Qianqian Xu*
IIP, ICT, CAS
xuqianqian@ict.ac.cn

Zhiyong Yang
SCST, UCAS
yangzhiyong21@ucas.ac.cn

Xiaochun Cao
SCST, Shenzhen Campus, SYSU
SKLOIS, IIE, CAS
caoxiaochun@mail.sysu.edu.cn

Qingming Huang*
SCST, UCAS
IIP, ICT, CAS
BDKM, CAS
Peng Cheng Laboratory
qmhuang@ucas.ac.cn

ABSTRACT

The Area Under ROC curve (AUC) is widely used as an evaluation metric in various applications. Due to its insensitivity towards class distribution, directly optimizing AUC performs well on the class imbalance problem. However, existing AUC optimization methods are limited to regular data such as text, images, and video. AUC optimization on graph data, which is ubiquitous and important, is seldom studied. Different from regular data, AUC optimization on graphs suffers from not only the class imbalance but also topology imbalance. To solve the complicated imbalance problem, we propose a unified topology-aware AUC optimization (TOPOAUC) framework, which could simultaneously deal with the topology and class imbalance problem in graph learning. We develop a multi-class AUC optimization work to deal with the class imbalance problem. With respect to topology imbalance, we propose a **T**opology-**A**ware **I**mportance **L**earning mechanism (TAIL), which considers the topology of pairwise nodes and different contributions of topology information to pairwise node neighbors. Extensive experiments on three real-world datasets demonstrate the effectiveness of our proposed method.

CCS CONCEPTS

• **Computing methodologies** → *Semi-supervised learning settings*; **Neural networks**; *Knowledge representation and reasoning*.

KEYWORDS

Graph Learning, AUC Optimization, Class Imbalance, Topology Imbalance

ACM Reference Format:
Junyu Chen, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2022. A Unified Framework against Topology and Class Imbalance. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548120

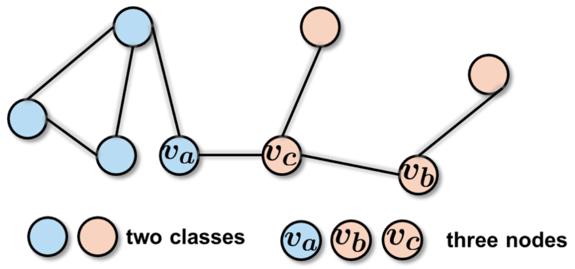


Figure 1: A simple illustration of topology imbalance. Here different colors represent different classes. We can observe that, with respect to a pair of positive-negative instance (v_a, v_b), there are two kinds of neighbors. v_c is a common neighbor for both v_a and v_b. The other nodes either belong to the neighbor v_a or that of v_b. Such nodes like v_c cannot effectively differentiate v_a and v_b, and thus should not be treated equally as other nodes during message-passing.

1 INTRODUCTION

The Area Under the ROC Curve (AUC), which measures the average classification performance of a scoring function under different thresholds, is a widely-used performance metric in various applications such as classification [44] and bipartite ranking [38]. Compared to accuracy, it is insensitive toward class distributions and costs. Therefore, AUC is more suitable for long-tailed or imbalanced datasets, where the quantity of the majority class is significantly larger than those of the minority class. Motivated by its appealing properties, in recent years, there has been a surge of studies trying to optimize AUC directly [34, 42, 43].

Despite the great success they have achieved, existing AUC optimization algorithms are limited to regular data such as texts, statistics features, and images. In real-world applications, many data such as websites, social networks, and transportation networks are usually characterized by the graph structure, where nodes represent

MM '22, October 10–14, 2022, Lisboa, Portugal.

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548120

*Corresponding authors.

instances while edges indicate the relationship between instances. In such cases, simply leveraging the existing AUC literature is insufficient to attain a desirable performance since the imbalance problem on graph data is generally more complicated than regular ones. On the one hand, similar to regular data structures, graph learning could suffer from the **class imbalance** problem in the sense that the decision boundary may be dominated by the majority class. On the other hand, since graph learning methods [9] learn from not only node attributes but also the topology of the graph, it will suffer from another kind of imbalance, i.e., **topology imbalance**. As shown in Figure 1, under this scenario, when comparing a pair of positive/negative instances (v_a, v_b), we have two kinds of neighbors. v_c is a common neighbor for both v_a and v_b. The other nodes either belong to the neighbor v_a or that of v_b. We argue that nodes like v_c cannot effectively differentiate v_a and v_b, and thus should not be treated equally as other nodes during message-passing. Moreover, the topological imbalance issue and class imbalance issue are different in nature. Thus, we have to consider them simultaneously in our setting.

Recently, ReNode [5] presented an early study targeting the topology imbalance issues. They denoted topological position by node influence conflicts, which described that the influence received at a node is from different classes of nodes after label propagation [49]. The nodes with high influence conflicts should pay less attention during training and vice versa. Remarkable success as they made, the weighting mechanism of ReNode is suited to instance-wise objective and could not be directly applied to AUC optimization, whose objective is pair-wise formulated. Besides, ReNode focuses on direct influences from labeled nodes, losing a fine-grained investigation of the neighborhood of nodes.

With the above discussions, in this paper, we care about the following challenge:

Facing the complicated imbalance problem on the graph, how to develop an effective framework against topology and class imbalance simultaneously?

To achieve this goal, **we present a unified framework, called unified topology-aware AUC optimization TOPOAUC**, which relies on two specifically tailored components. First, motivated by the great success of AUC optimization on the general imbalanced classification problem, we conduct AUC optimization on the graph to handle the class imbalance issue in graph learning. Since the task of graph learning is usually multi-class, we formulate the multi-class AUC metric as an average of binary AUC in terms of each class pair. Subsequently, targeting the topology imbalance as shown in Figure 1, we propose a **T**opology-**A**ware **I**mportance **L**earning (TAIL) mechanism to simultaneously consider the topology structures to boost the prediction performance. Concretely, instead of directly aggregating the topology influence acted on the target nodes, we investigate the fine-grained topology structure of paired nodes. Finally, we unify the AUC optimization and the TAIL to obtain our proposed TOPOAUC framework.

In summary, our contributions are three-fold:

- We present an effective framework, named as unified topology-aware AUC optimization, which could simultaneously deal with the topology and class imbalance problem in the task of graph learning.
- We present the early trial to introduce the AUC optimization problem to the task of graph learning, which is an effective way to handle the class imbalance in the graph.
- With respect to topology imbalance issues, we propose the topology-aware importance learning mechanism, which gives a fine-grained investigation of the node pair's neighborhood to pay different attention to the influences on different node neighbors.

Empirical experiments on three real-world datasets demonstrate the superiority of TOPOAUC.

2 RELATED WORK

2.1 AUC optimization

Since the early study [6] shows that maximizing AUC should not be replaced by minimizing error rate, it is necessary to explore the direct AUC optimization methods. At the early stage, most studies [3, 23, 47] pay attention to the full-batch off-line setting, which processes all training examples at each iteration in the algorithmic optimization. Later on, AUC optimization is extended to online optimization for large-scale data analysis. There are two kinds of methods for this online AUC maximization: online buffer-based methods [21, 24, 36] and online statistics-based methods [10, 14, 29]. Subsequently, a mass of studies [8, 16, 41, 45] explore another family of stochastic optimization algorithms for handling big data, which process a mini-batch of examples at each iteration for updating the model parameters.

Later on, since deep neural networks gain great success in computer vision, natural language processing, etc, the applications of AUC optimization in deep learning [20, 35, 46, 51] have begun to come under the spotlight. However, to the best of our knowledge, few works focus on non-Euclidean data such as graphs. We get the first step to studying the application of AUC optimization for graph learning.

2.2 Imbalance Classification

The imbalanced classification problem exists in many real scenarios, such as fraud detection, disease prediction, and topic classification. As for the regular data (e.g., texts and images), there is a class imbalance issue where the label distribution is highly skewed. Existing studies for this issue fall in several directions: 1) re-sampling methods [17, 19, 30] try to over-sample the minority classes and under-sample the majority classes to build balanced datasets. This method may over-fit these repeated samples drawn from the minority class, or discard the valuable information in the majority class; 2) re-weighting methods [1, 2, 50] punish training samples by the labeling sizes of each class. Concretely, they attach high penalty weights to the majority classes and low ones to the minority classes; 3) ensemble learning methods [13, 37] alleviate the problem of poor classification ability in minority classes by combining a series of weak classifiers; 4) manual-designed losses [4, 7, 27] attempt to balance majority/minority classes or simple/hard samples.

Another specific data, i.e., the graph, faces the tougher class imbalance challenge: the data has non-Euclidean properties, and there are topological connections between nodes. Most of the studies mentioned above are not suitable for graphs, which overlook the topology structure. By borrowing similar ideas from previous works, recently, some methods [31, 33, 40, 48] gain success to solve the class imbalance on graphs. However, these models ignore another imbalance problem caused by the topology property of graphs: the topology imbalance arises from the asymmetry of node connections shown in Figure 1. ReNode [5] is the first work focused on this issue, which aggregates global topological information across the whole graph to locate node positions, and calculate the values to weight each sample. However, this method has two limitations. First, it is designed for instance-wise loss functions such that it could not be directly applied to AUC optimization. Second, it overlooks the adjacency topology of target nodes. We argue that a fine-grained investigation of nodes' neighborhoods helps to better understand the topology imbalance issues for paired nodes, so as to better solve this issue.

3 PRELIMINARY

Before formally presenting our method, we first provide a brief introduction to some important backgrounds.

3.1 Problem Definition

This paper deals with the node classification task. An undirected graph is denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{L}\}$, where $\mathcal{V} = \{v_i\}_{i=1}^{n}$ is the node set of size n, $\mathcal{E} = \{e_{i,j}\}_{i,j=1,\cdots,n;i\neq j}$ is the edge set, $\mathcal{L} \subset \mathcal{V}$ is the labeled node set. Usually, we have $|\mathcal{L}| \ll |\mathcal{V}|$. Denote the adjacency matrix as $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, and $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with the self connections. Then $\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}$ is the adjacency matrix normalized by the diagonal degree matrix $\tilde{\boldsymbol{D}} = diag(\tilde{\boldsymbol{A}}\boldsymbol{1}_n)$, where $\boldsymbol{1}_n$ is a column vector with all values of one. The feature matrix of the node set is denoted by $\boldsymbol{X} \in \mathbb{R}^{n\times d_f}$, where d_f is the input feature dimension. Besides, the label matrix is denoted by $\boldsymbol{Y} \in \mathbb{R}^{n\times N_C}$ where labeled nodes are represented by the one-hot vectors, and N_C is the number of classes. Given $\mathcal{G}, \boldsymbol{X}$ and $\boldsymbol{Y}$, the target of semi-supervised node classification is to learn a classifier $\mathcal{F}$ (such as a GNN model) from the labeled node set $\mathcal{L}$, such that $\mathcal{F}$ could correctly predict the class of the unlabeled node in $\mathcal{U} = \mathcal{V} - \mathcal{L}$. The notations used in this paper refer to in the Appendix.

3.2 Label Propagation

In this subsection, we introduce the Label Propagation (LP) process [49], which is utilized to measure the influence from a node to other nodes [39] in our method. The LP process is a random walk processing starting from labeled nodes, which propagates the labels of nodes to their neighbor nodes. Along with the iterations of the propagation, each node will finally be influenced by the class information of labeled nodes. Specifically, if we denote the propagated status at step t as $\boldsymbol{Y}^t$, the status of the next propagation step $t+1$ can be formulated as follows:

$$\boldsymbol{Y}^{t+1} = \alpha\boldsymbol{Y}^0 + (1-\alpha)\hat{\boldsymbol{A}}\boldsymbol{Y}^t, \tag{1}$$

where $\alpha \in (0, 1]$ is the restart probability, with which the status are initialized with the label matrix, *i.e.*, $\boldsymbol{Y}^0 = \boldsymbol{Y}$. It could be proved that the propagation process will converge to a stable status $\boldsymbol{Y}^\star$ [11]:

$$\boldsymbol{Y}^\star = \alpha(\boldsymbol{I} - (1-\alpha)\hat{\boldsymbol{A}})^{-1}\boldsymbol{Y}. \tag{2}$$

In this way, the final status $\boldsymbol{Y}^\star$ could be viewed as a kind of pseudo label, depending on two factors: the initial label $\boldsymbol{Y}$ and the graph topology (adjacency between nodes). In light of this, we could measure the influence of the topology on nodes with the *topology influence matrix* $\boldsymbol{P} \in \mathbb{R}^{n\times n}$:

$$\boldsymbol{P} = \alpha(\boldsymbol{I} - (1-\alpha)\hat{\boldsymbol{A}})^{-1}, \tag{3}$$

where $\boldsymbol{P}_{ij}$ can be regarded as the influence of node j on node i.

4 UNIFIED TOPOLOGY AUC FRAMEWORK

We aim to simultaneously solve the class imbalance and topology imbalance issues on the graph. Considering the class imbalance, we propose to directly optimize the AUC metric, which is formulated as a sum of pairwise losses. As for the topology imbalance issue, we dive into the topology structure of node pairs, and design a topology-aware importance learning mechanism. In what follows, we will introduce them in detail.

4.1 Formulation of AUC Optimization

Since AUC is a well-defined metric that is insensitive to label distribution [12], we first consider directly leveraging AUC on the graph to tackle the class imbalance problem. Mathematically, in terms of a score function f, AUC measures the probability that a positive sample achieves a higher score than a negative one [18]:

$$\begin{aligned} \mathsf{AUC}(f) =& \mathbb{P}\left[f(x^+) > f(x^-) \middle| x^+ \sim P_{\mathcal{P}}, x^- \sim P_{\mathcal{N}}\right] \\ =& \mathbb{E}_{\boldsymbol{x}^+\sim P_{\mathcal{P}}}\mathbb{E}_{\boldsymbol{x}^-\sim P_{\mathcal{N}}}\left[\mathbb{I}[f(\boldsymbol{x}^+) > f(\boldsymbol{x}^-)]\right] \\ =& 1 - \mathbb{E}_{\boldsymbol{x}^+\sim P_{\mathcal{P}}}\mathbb{E}_{\boldsymbol{x}^-\sim P_{\mathcal{N}}}\left[\mathbb{I}[f(\boldsymbol{x}^+) \leq f(\boldsymbol{x}^-)]\right] \\ =& 1 - \mathbb{E}_{\boldsymbol{x}^+\sim P_{\mathcal{P}}}\mathbb{E}_{\boldsymbol{x}^-\sim P_{\mathcal{N}}}\left[\ell_{0,1}(f(\boldsymbol{x}^+) - f(\boldsymbol{x}^-))\right] \end{aligned} \tag{4}$$

where $\mathbb{I}[A]$ is the indicator function that equals 1 if A holds and equals 0 otherwise, $\ell_{0,1}(z)$ is the zero-one loss function that equals 1 if $z > 0$ and equals 0 otherwise. $P_{\mathcal{P}}$ and $P_{\mathcal{N}}$ are the distribution of positive samples $\mathcal{P}$ and negative samples $\mathcal{N}$, respectively. From the last equation in Equation (4), we could see that maximizing AUC is equivalent to minimizing the following AUC risk:

$$\mathcal{R}_{\mathsf{AUC}}(f) = \mathbb{E}_{\boldsymbol{x}^+\sim P_{\mathcal{P}}}\mathbb{E}_{\boldsymbol{x}^-\sim P_{\mathcal{N}}}\left[\ell_{0,1}(f(\boldsymbol{x}^+) - f(\boldsymbol{x}^-))\right]$$

However, directly minimizing the AUC risk is infeasible since it is hard to obtain the true distributions $P_{\mathcal{P}}$ and $P_{\mathcal{N}}$. Therefore, given finite datasets, one typically resorts to the following AUC empirical risk:

$$\hat{\mathcal{R}}_{\mathsf{AUC}}(f) = \sum_{\boldsymbol{x}^+\in\mathcal{P},\boldsymbol{x}^-\in\mathcal{N}} \frac{\ell_{0,1}(f(\boldsymbol{x}^+) - f(\boldsymbol{x}^-))}{|\mathcal{P}||\mathcal{N}|}. \tag{5}$$

Furthermore, note that AUC is suitable for the binary classification problem. In order to suit the multi-class situation, we extend the AUC optimization from binary to multi-class problems. Specifically, following general methods [28, 42], we turn to optimize the multi-class AUC metric defined as a sum of binary class AUC scores for each class pair:

$$\tilde{\mathsf{AUC}}(\boldsymbol{f}) = \frac{\sum_{i=1}^{N_c}\sum_{j\neq i}\mathsf{AUC}_{i|j}\left(f^{(i)}\right)}{N_C\,(N_C-1)}, \tag{6}$$

where $\boldsymbol{f} = \{f^{(1)}, \cdots, f^{(N_C)}\}$ and $f^{(i)}$ is the scoring function for predicting the i-th class, $\mathsf{AUC}_{i|j}\left(f^{(i)}\right)$ is a variant of AUC(f) in

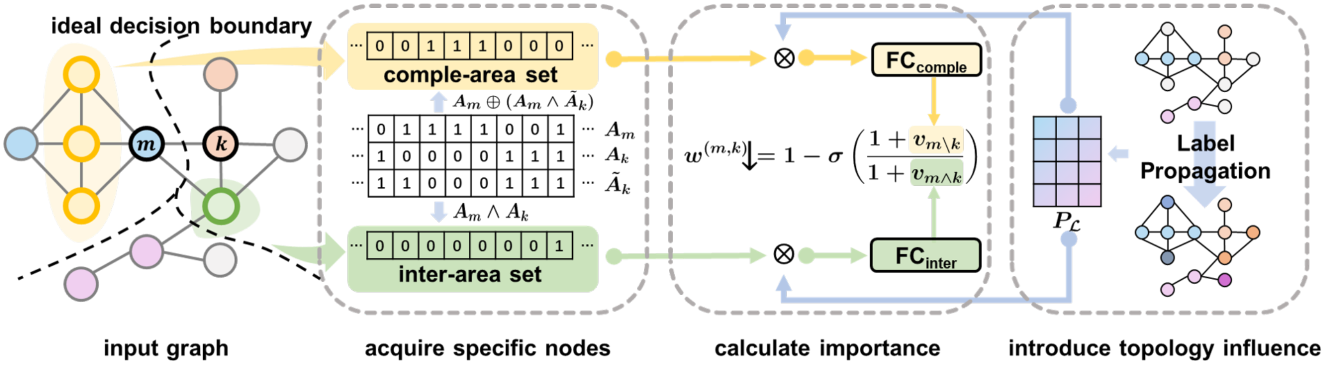


Figure 2: The illustration of our proposed Topology-Aware Importance Learning mechanism (TAIL). For the node pair (v_m, v_k) in the input graph, first, we acquire specific nodes from the inter-area and the comple-area, which are parts of two neighbor sets. Second, we introduce the topology influence by the Label Propagation. Finally, we calculate the importance of the node pair by aggregating the influence acted on the specific nodes.

Equation (4) for class pair (i, j). Then given finite datasets, we could obtain the following multi-class AUC empirical risk:

$$\hat{\mathcal{R}}_\ell = \sum_{i=1}^{N_C} \sum_{x_m \in \mathcal{L}_i} \sum_{j \neq i} \sum_{x_k \in \mathcal{L}_j} \frac{\ell\left(f^{(i)}(x_m) - f^{(i)}(x_k)\right)}{n_i n_j}, \tag{7}$$

where n_i, n_j denote the numbers of i-th and j-th class samples separately, and $\mathcal{L}_i$ and $\mathcal{L}_j$ are the sample sets for i-th and j-th class, respectively. Herein, we replace the non-differentiable 0-1 loss function to a differentiable and Bayes-consistent [15] surrogate loss function $\ell(\cdot)$. Although the above multi-class AUC optimization could handle the class imbalance on a graph, it overlooks the topology structure which is indispensable for constructing a discriminative classifier on the graph. In the next section, we will elaborate on how to inject graph information to boost the learning process.

4.2 Topology-Aware Importance Learning

As discussed in Section 1, the imbalance of the topology structure is also non-negligible for node classification on graph data. However, existing methods for topology imbalance, like ReNode [5], has two limitations. First, it is designed for instance-wise loss functions such that it could not be directly applied to AUC optimization that its objective is pairwise formulated. Second, it determines the importance weight of node i based on the topology influences $\boldsymbol{P}_{ij}$ from all node j that is from a different class with node i, overlooking node i's topology structure. We argue that a fine-grained investigation of nodes' neighborhoods helps to better understand the topology imbalance issues for paired nodes, so as to better solve this issue.

Under topology imbalance, nodes in different locations are influenced differently by labeled nodes. Therefore, after label propagation or several iterations in GNNs, some nodes may receive much conflictive class information and are hard to discriminate, while other nodes receive less conflictive class information and possess more discriminable information. Intuitively, nodes with different discriminability should be treated differently during training. Back to our AUC optimization, we need to first study whether a positive-negative node pair is easy to discriminate. To do this, we dive into the neighborhood of node pairs to seek some clues. On top of that, we design a **T**opology-**A**ware **I**mportance **L**earning (TAIL) mechanism to learn from different node pairs, which is shown in Figure 2.

Let us take a closer look at the neighboring nodes of a positive-negative node pair. For a class pair (i, j) and a node pair (v_m, v_k), where v_m is of positive class i and v_k is of negative class j, denote node v_m's adjacency neighbors as $\boldsymbol{A}_m$ (the m-th row of $\boldsymbol{A}$) and denote node v_k's adjacency neighbors as $\boldsymbol{A}_k$ (the k-th row of $\boldsymbol{A}$). We could figure out three sets of neighboring nodes for the node pair (v_m, v_k). First, the shared neighboring nodes of v_m and v_k, denoted as the **inter-area set** $\boldsymbol{A}_{m \wedge k}$, which can be determined as:

$$\boldsymbol{A}_{m \wedge k} = \boldsymbol{A}_m \wedge \boldsymbol{A}_k, \tag{8}$$

where $\wedge$ denotes the AND operation. Secondly, the neighboring nodes that are adjacent to v_m alone, denoted the set of such nodes as the **comple-area set** $\boldsymbol{A}_{m \backslash k}$, which could be determined by:

$$\boldsymbol{A}_{m \backslash k} = \boldsymbol{A}_m \oplus (\boldsymbol{A}_m \wedge \tilde{\boldsymbol{A}}_k), \tag{9}$$

where $\oplus$ is the XOR operation, and $\tilde{\boldsymbol{A}}_k$ is the k-th row in $\tilde{\boldsymbol{A}}$ (the adjacent matrix with self connections). Here, we additionally exclude the negative node v_k since it is in target node pair. The last set includes all neighboring nodes that are only adjacent to node v_k, i.e., $\boldsymbol{A}_{k \backslash m}$, which could be obtained similar to Equation (9).

Before discussing how the node sets influence differently w.r.t node pair (v_m, v_k), we first introduce topology influence which refers to the influence between nodes via topology structure. We use the well-known label propagation models to measure the influence across different nodes. Specifically, we use the topology influence matrix $\boldsymbol{P}$ as shown in Equation (3), where P_{ij} represents the influence of node j on node i. We ignore the influences from unlabeled nodes and extract the labeled topology influence matrix $\boldsymbol{P}_L \in \mathbb{R}^{n \times |\mathcal{L}|}$ from labeled nodes:

$$\boldsymbol{P}_L = \boldsymbol{P}_{[:, \boldsymbol{i}_L]}, \tag{10}$$

where $\boldsymbol{i_L} \in \mathbb{R}^n$ is the index vector of labeled nodes, where a term value equals 1 if the corresponding node is labeled, otherwise it equals 0.

Based on the labeled topology influences matrix, let us discuss the aforementioned neighboring nodes. For nodes from $\boldsymbol{A}_{m\wedge k}$, during message propagation, they directly act on both node v_m and node v_k. There is a tendency for them to smoothen the information of node v_m and node v_k. Therefore, these nodes contribute less to distinguishing node pair (v_m, v_k). And the information received by these nodes should be suppressed during learning. For nodes from $\boldsymbol{A}_{m\backslash k}$, they escape from the above tendency since they only directly act on the node v_m. Moreover, there is a high possibility that these nodes are connected to the nodes from positive class i and received more information of class i. Therefore, the information on these nodes contributes more to distinguishing this node pair and could be strengthened for model learning. We need not to consider $\boldsymbol{A}_{k\backslash m}$ because the multi-class AUC metric will calculate all class pair (i, j), $\forall i, j = 1, \cdots, N_C$, and this set of nodes will be considered in its symmetric class pair (j, i). Motivated by the above intuition, we design a topology-aware importance learning mechanism to put different importance on neighboring nodes when model learning. Let $v_{m\wedge k}$ and $v_{m\backslash k}$ be the influence on $\boldsymbol{A}_{m\wedge k}$ and $\boldsymbol{A}_{m\backslash k}$ from all labeled nodes, respectively, where:

$$\begin{aligned} v_{m\wedge k} &= \mathrm{SumPool}\left(\mathrm{FC}_{\mathrm{inter}}\left(\boldsymbol{A}_{m\wedge k}\boldsymbol{P_L}\right)\right), \\ v_{m\backslash k} &= \mathrm{SumPool}(\mathrm{FC}_{\mathrm{comple}}(\boldsymbol{A}_{m\backslash k}\boldsymbol{P_L})), \end{aligned} \tag{11}$$

Herein, SumPool$(\cdot)$ is a sum-pooling layer, $\mathrm{FC}_{\mathrm{inter}}(\cdot)$ and $\mathrm{FC}_{\mathrm{comple}}(\cdot)$ are fully connected layers for inter-area and comple-area nodes, respectively. The fully connected layers provide a learnable weighting mechanism to produce the final learning importance.

Recall that we want to decrease the contribution of the confusing influences aggregated on the inter-area nodes, and increase the contribution of the helpful influences aggregated on the comple-area nodes. To this end, the final importance of the node pair (v_m, v_k) during model learning is calculated based on $v_{m\wedge k}$ and $v_{m\backslash k}$:

$$w^{(m,k)} = 1 - \sigma\left(\frac{1 + v_{m\backslash k}}{1 + v_{m\wedge k}}\right), \tag{12}$$

where $\sigma(\cdot)$ is the Sigmoid function to scale the value to the range $[0, 1]$.

4.3 Optimization Objective

Finally, in order to tackle the topology imbalance and class imbalance simultaneously, we integrate topology-aware importance learning mechanism into AUC optimization and arrive at the following joint objective function:

$$\hat{\mathcal{R}}_\ell = \sum_{i=1}^{N_C} \sum_{v_m \in \mathcal{L}_i} \sum_{j \neq i} \sum_{v_k \in \mathcal{L}_j} \frac{w^{(m,k)}}{n_i n_j} \ell\left(f^{(i)}(\boldsymbol{x}_m) - f^{(i)}(\boldsymbol{x}_k)\right) \tag{13}$$

Here, in terms of the surrogate loss $\ell(\cdot)$, we adopt three widely used losses [42] in the experiment part (Section 5), including $\ell_{exp}(t) = exp(-t)$, $\ell_{hinge}(t) = max(1-t, 0)$ and $\ell_{sq}(t) = (1-t)^2$, where t is the input of the surrogate loss.

Table 1: Statistics of CORA, CiteSeer and PubMed.

Datasets	Nodes	Edges	Features	Classes
CORA	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3

5 EXPERIMENTS

In this section, we first introduce the experimental setup, then conduct experiments to validate the effectiveness of our proposed TOPOAUC. All experiments are run on a machine with E5-2620 CPU, TITAN RTX GPU and 256G RAM.

5.1 Experimental Setup

Datasets. We adopt three real-world citation network datasets [32] for evaluation: CORA, CiteSeer, and PubMed. These datasets contain sparse bag-of-words feature vectors for each document and a list of citation links between documents. Following [5, 26], we treat the citation links as (undirected) edges and construct a binary, symmetric adjacency matrix $\boldsymbol{A}$. The statistical information about these datasets is shown in Table 1.

Baselines. The backbone model for all methods is GCN [26] with 3-layers. We implement following baselines for comparison: Cross Entropy loss (**CE**), Re-Weight loss (**RW**) [22], **Focal** loss [27], Class Balanced loss (**CB**) [7], **ReNode** [5] and *AUC* loss [42]. Moreover, we use the topological weighting of ReNode [5] to combine with RW, Focal and CB, which are written as: **RW+ReNode**, **CB+ReNode** and **Focal+ReNode**. Besides, as mentioned in Section 4.3, we adopt three popular surrogate losses $\ell_{exp}, \ell_{hinge}, \ell_{sq}$ to compare: $\mathbf{TOPOAUC_{exp}}$**(Ours1)**, $\mathbf{TOPOAUC_{hinge}}$**(Ours2)** and $\mathbf{TOPOAUC_{sq}}$**(Ours3)**. Meanwhile, the compared pure AUC losses also adopt these surrogate losses, which are written as $\mathbf{AUC_{exp}}$**,** $\mathbf{AUC_{hinge}}$**,** $\mathbf{AUC_{sq}}$. Due to the special designed losses in graph-specific methods [31, 33, 40, 48], it's almost infeasible to replace with our pairwise TOPOAUC losses. Therefore, we pay more attention to the performance of general losses, and do not consider these graph-specific methods for comparison.

Metrics. Following existing work in evaluating the imbalance classification problem [1, 5], we adopt three metrics: macro F1 measure (M-F1), weighted F1 measure (W-F1) and mean AUC score. M-F1 calculates the unweighted mean of F1 scores for each class. On the contrary, W-F1 obtains the average weighted by the number of instances for each class. Besides, the AUC score could evaluate the rank ability of methods. On the one hand, both M-F1 and AUC are calculated separately for each class, and can better reflect the performance of minority classes. On the other hand, W-F1 could explain the actual situation's overall performance in all classes.

Setting. Following existing studies [4, 5], we adopt the transductive setting for the semi-supervised node classification problem. We select $\frac{1}{2}N_c$ classes of samples to be the majority classes and the rest classes to be the minority classes. Since the original datasets are fully-supervised, to produce a semi-supervised setting, we stratified randomly sample 5% of all the nodes as the labeled nodes for

Table 2: Performance of different imbalance ratios [10,15,20] on CORA dataset. The results are reported as mean±standard deviation over 15 repeat experiments, and the best is marked red while the second-best is marked blue.

Dataset	CORA								
Imbalance Ratio	10			15			20		
Metric	M-F1(%)	W-F1(%)	AUC(%)	M-F1(%)	W-F1(%)	AUC(%)	M-F1(%)	W-F1(%)	AUC(%)
CE	47.65±3.68	44.77±5.30	89.17±0.98	46.17±2.32	43.60±5.17	84.96±1.70	42.97±4.83	40.05±7.10	84.79±1.21
RW	50.99±4.91	48.70±7.81	89.90±1.09	47.01±5.43	43.56±4.69	89.47±2.45	43.49±2.97	39.63±4.00	88.56±2.24
RW+ReNode	51.23±4.41	49.50±6.98	90.16±1.10	48.96±4.76	47.51±7.07	89.58±1.88	45.14±3.83	42.00±5.35	88.46±2.37
Focal	52.35±4.43	48.45±7.13	90.73±0.89	47.49±3.69	44.62±4.01	90.21±1.72	44.90±4.68	41.80±5.97	88.48±1.96
Focal+ReNode	53.66±4.13	49.95±6.62	90.98±0.89	48.44±4.40	45.26±4.91	90.29±1.57	45.98±4.11	42.46±5.33	88.71±1.71
CB	52.26±4.98	50.97±7.24	90.29±1.00	47.06±3.59	43.56±4.54	89.70±1.61	44.84±4.47	40.16±4.77	88.76±2.09
CB+ReNode	52.50±5.38	51.28±6.94	90.51±0.97	48.40±5.05	47.30±7.29	89.74±1.44	46.69±4.69	42.51±6.00	88.87±1.85
AUC_{exp}	56.04±4.37	53.40±6.80	93.37±0.80	44.67±3.76	41.04±6.06	89.69±1.47	45.19±4.71	39.67±4.34	89.92±1.88
$\text{AUC}_{\text{hinge}}$	55.74±8.79	53.41±11.15	92.76±0.67	45.10±3.67	39.89±5.17	89.90±1.83	45.35±3.43	39.99±4.69	90.04±1.93
AUC_{sq}	56.79±4.89	54.53±8.11	93.43±0.91	44.21±3.76	39.46±4.48	89.82±2.00	46.43±5.63	42.05±7.07	89.81±2.12
$\text{TOPOAUC}_{\text{exp}}$(Ours1)	59.36±3.96	58.51±4.43	93.43±0.73	49.66±7.55	48.17±8.64	90.44±1.68	48.80±6.61	46.58±5.40	90.78±1.39
$\text{TOPOAUC}_{\text{hinge}}$(Ours2)	58.65±5.39	56.14±7.14	93.21±0.66	50.25±5.12	47.98±3.39	90.72±1.71	47.64±6.69	45.54±8.25	90.55±1.57
$\text{TOPOAUC}_{\text{sq}}$(Ours3)	60.90±4.85	60.06±6.14	93.51±0.57	51.32±3.53	48.02±4.46	90.53±1.84	49.70±7.71	49.44±9.27	90.42±1.68

Table 3: Performance of different imbalance ratios [10,15,20] on CiteSeer dataset.

Dataset	CiteSeer								
Imbalance Ratio	10			15			20		
Metric	M-F1(%)	W-F1(%)	AUC(%)	M-F1(%)	W-F1(%)	AUC(%)	M-F1(%)	W-F1(%)	AUC(%)
CE	49.46±8.87	52.29±9.73	80.86±2.16	36.11±2.99	37.38±3.33	76.65±1.88	30.11±3.74	30.88±4.19	74.68±2.30
RW	52.69±6.66	57.03±7.92	86.20±1.26	41.75±4.22	44.24±4.46	83.96±1.19	38.81±6.43	41.74±7.19	80.72±2.84
RW+ReNode	51.77±6.53	56.47±6.42	86.86±1.12	42.48±4.91	45.21±5.39	83.99±1.15	41.10±8.83	44.03±9.48	80.60±3.34
Focal	54.05±5.34	58.28±5.68	85.23±1.47	44.97±5.50	47.76±6.19	83.93±1.36	39.32±5.85	41.67±6.37	81.66±1.85
Focal+ReNode	54.33±5.05	58.47±5.35	86.61±1.33	44.98±5.56	47.97±6.16	83.97±1.34	39.59±7.26	42.20±7.89	81.37±1.97
CB	48.43±6.24	52.45±6.64	85.63±1.48	41.32±6.93	45.04±7.20	83.58±1.05	36.69±6.00	39.17±6.45	81.10±1.73
CB+ReNode	53.57±6.43	57.37±6.65	86.98±1.43	43.31±4.63	46.88±4.90	84.56±1.30	40.58±4.20	43.80±5.10	81.80±2.05
AUC_{exp}	46.19±9.04	49.51±9.70	86.57±1.48	45.97±6.65	48.85±7.56	85.00±0.92	41.22±9.15	43.60±9.87	82.10±2.39
$\text{AUC}_{\text{hinge}}$	41.65±8.93	44.87±9.77	86.20±1.41	47.89±7.79	50.72±8.82	84.69±1.11	41.43±9.26	43.79±9.91	82.02±2.50
AUC_{sq}	44.03±6.26	47.58±6.68	86.51±1.14	48.13±7.63	50.82±8.54	84.90±0.93	41.62±7.79	44.12±8.51	82.08±2.05
$\text{TOPOAUC}_{\text{exp}}$(Ours1)	52.52±10.02	56.12±10.98	87.20±1.33	47.28±7.11	51.17±7.64	85.22±1.17	48.68±6.86	51.92±7.58	83.24±2.16
$\text{TOPOAUC}_{\text{hinge}}$(Ours2)	50.51±8.05	54.76±8.34	87.03±1.15	49.31±5.05	52.59±5.67	84.77±1.34	48.51±7.13	51.75±7.86	83.17±2.21
$\text{TOPOAUC}_{\text{sq}}$(Ours3)	51.99±7.30	56.33±7.40	87.11±1.05	51.97±5.13	55.43±5.74	85.00±1.31	49.94±6.52	53.56±7.30	83.42±2.06

training, while the remaining 95% are regarded as unlabeled nodes. For these labeled nodes, each minority class has n_{min} samples and each majority class has $n_{maj} = r_{imb} * n_{min}$ samples, where r_{imb} is the imbalance ratio searched from $[10, 15, 20]$. The value of n_min is calculated based on the total number of training samples. For the remaining 95% nodes, we randomly select 30 nodes per class to produce the validation set, and the rest nodes serve as the test set. To demonstrate the stability of methods, we use five different random seeds $[0, 1, 2, 3, 4]$ to split datasets and three different random seeds $[0, 1, 2]$ to initialize parameters, then record the mean and the standard deviation of the three performance metrics on these 15 repeated experiments.

During training, we take Adam [25] as the model optimizer. The learning rate is selected from $[0.005, 0.0075, 0.01, 0.015]$, and it begins to decay after 10 epochs with a ratio of 0.95. The max size of epochs is 500, and the training process will be early stopped if there is no improvement in 20 epochs. Details of other hyper-parameters are referred to in the Appendix.

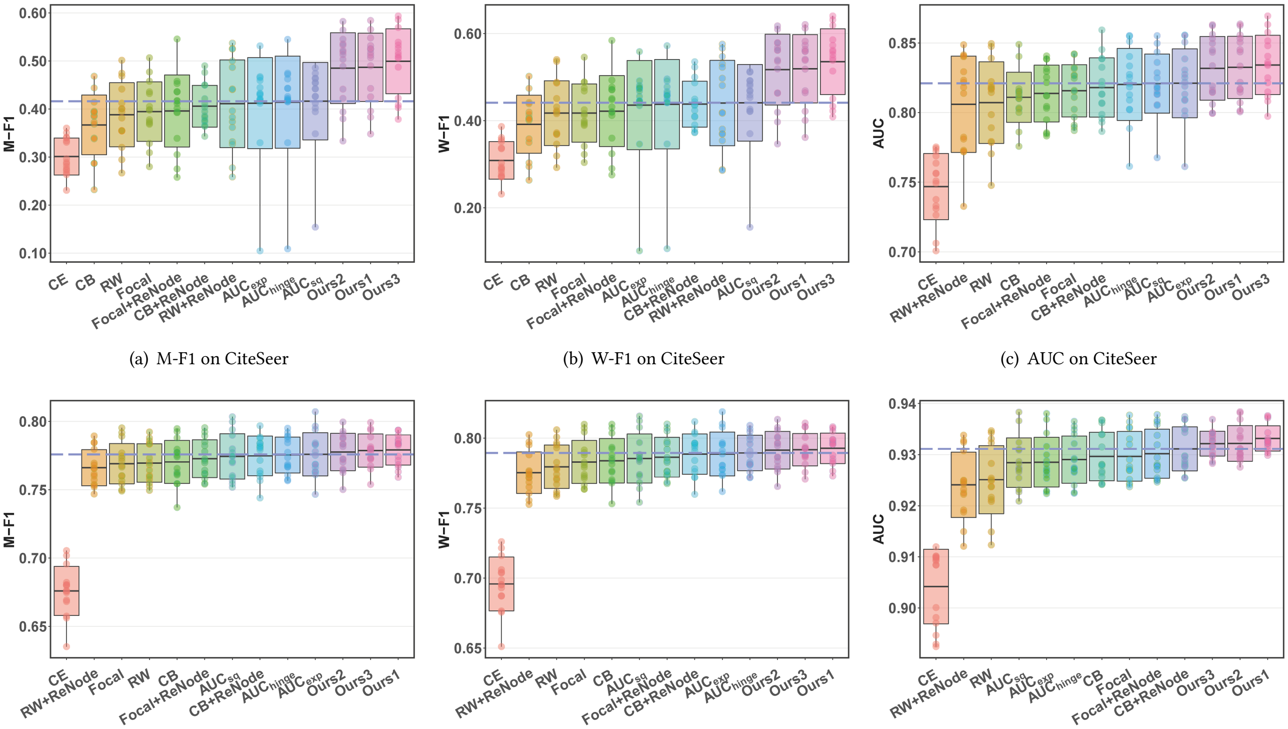
5.2 Results Analysis

The experimental results on CORA, CiteSeer, and PubMed datasets are shown in Table 2, Table 3 and Table 4, respectively. Furthermore, to illustrate the stability of our method on 15 randomly repeated experiments, we draw the results on CiteSeer and PubMed when $r_{imb} = 20$ in Figure 3.

From the observations of Table 2-4 and Figure 3, we can find that our proposed TOPOAUC outperforms all the competitors over M-F1, W-F1, and AUC metrics at different imbalance ratios in the most

Table 4: Performance of different imbalance ratios [10,15,20] on PubMed dataset.

Dataset	PubMed								
Imbalance Ratio	10			15			20		
Metric	M-F1(%)	W-F1(%)	AUC(%)	M-F1(%)	W-F1(%)	AUC(%)	M-F1(%)	W-F1(%)	AUC(%)
CE	72.68±0.76	74.56±0.84	91.94±0.30	69.59±1.86	71.53±1.96	91.08±0.60	67.59±1.74	69.59±1.85	90.42±0.70
RW	78.53±1.17	79.84±1.11	93.52±0.34	77.87±0.53	79.16±0.68	93.28±0.40	76.95±1.36	77.95±1.50	92.51±0.64
RW+ReNode	78.94±0.96	80.21±1.04	93.65±0.35	78.27±0.51	79.57±0.59	93.40±0.35	76.61±1.28	77.53±1.43	92.41±0.62
Focal	79.12±1.11	80.40±1.01	93.64±0.34	77.83±1.04	79.25±1.02	93.34±0.32	76.90±1.42	78.29±1.48	92.97±0.47
Focal+ReNode	79.20±0.76	80.52±0.75	93.71±0.26	78.17±0.67	79.60±0.67	93.43±0.26	77.27±1.33	78.64±1.37	93.02±0.46
CB	78.20±1.40	79.42±1.50	93.46±0.38	77.58±0.63	78.94±0.69	93.11±0.33	77.03±1.52	78.38±1.53	92.96±0.46
CB+ReNode	78.52±1.09	79.80±1.14	93.52±0.32	77.91±0.71	79.27±0.64	93.19±0.31	77.47±1.41	78.86±1.39	93.11±0.42
AUC$_{\text{exp}}$	79.41±0.91	80.68±0.84	93.62±0.30	78.36±0.53	79.69±0.50	93.30±0.25	77.59±1.53	78.86±1.52	92.85±0.47
AUC$_{\text{hinge}}$	79.43±0.90	80.77±0.80	93.60±0.29	77.76±0.74	79.17±0.69	93.18±0.31	77.53±1.27	78.93±1.22	92.90±0.45
AUC$_{\text{sq}}$	78.84±0.69	80.13±0.62	93.51±0.25	78.13±1.49	79.30±1.40	92.99±0.43	77.55±1.70	78.67±1.79	92.85±0.49
TOPOAUC$_{\text{exp}}$(Ours1)	79.91±0.45	81.23±0.46	94.06±0.21	78.96±0.77	80.30±0.71	93.70±0.20	77.92±1.10	79.29±1.09	93.33±0.26
TOPOAUC$_{\text{hinge}}$(Ours2)	79.48±0.74	80.88±0.71	93.88±0.21	78.31±0.76	79.74±0.65	93.51±0.21	77.75±1.32	79.13±1.30	93.22±0.34
TOPOAUC$_{\text{sq}}$(Ours3)	79.45±0.72	80.79±0.70	93.87±0.22	78.41±1.58	79.64±1.52	93.40±0.27	77.93±1.12	79.24±1.08	93.23±0.21



(a) M-F1 on CiteSeer

(b) W-F1 on CiteSeer

(c) AUC on CiteSeer

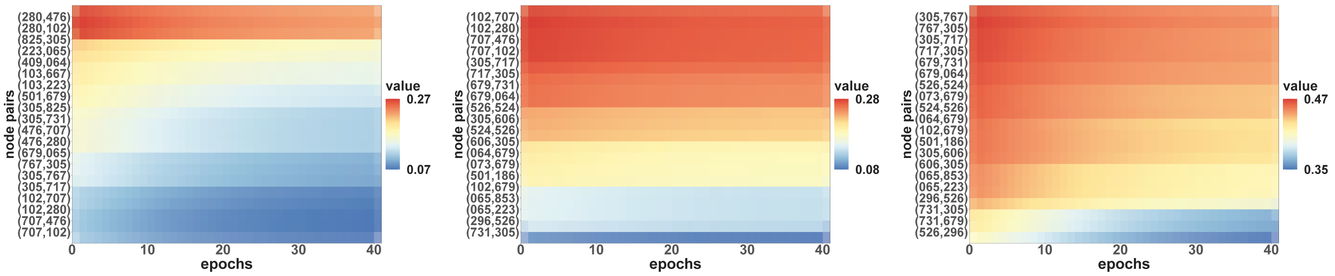
(d) M-F1 on PubMed

(e) W-F1 on PubMed

(f) AUC on PubMed

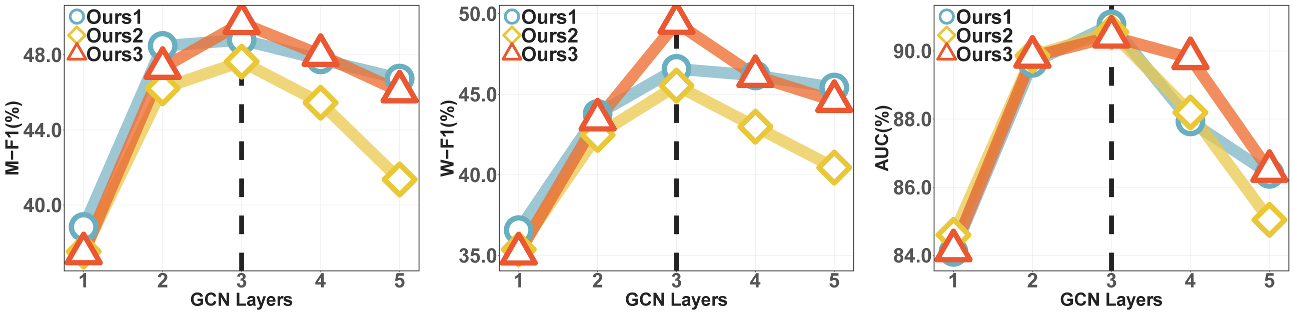
Figure 3: Illustrations of results on CiteSeer and PubMed when the imbalance ratio r_{imb} is set to 20. These boxplots are variants that show the minimum, mean-standard deviation, mean, mean+standard deviation, and maximum of the corresponding metric over 15 experiments.

cases. Moreover, we have the following observations: 1) The general class-imbalance methods (RW, Focal, and CB) are less effective than the AUC optimization methods (AUC$_{\text{exp}}$, AUC$_{\text{hing}}$ and AUC$_{\text{sq}}$) and our TOPOAUC methods. This shows that directly optimize AUC could better alleviate the class imbalance problem. 2) ReNode obtain the relatively worse performance than our TOPOAUC methods. This may be due to that ReNode fails to consider nodes' topology structure. Besides, the static weighting method of ReNode may be



(a) importance values of TOPOAUC$_{\text{exp}}$ when r_{imb} = 20 (b) importance values of TOPOAUC$_{\text{hinge}}$ when r_{imb} = 20 (c) importance values of TOPOAUC$_{\text{sq}}$ when r_{imb} = 20

Figure 4: Effectiveness of TAIL on CODA dataset. Each heatmap shows the decrease of importance values for 20 node pairs in the first 40 epochs.



(a) M-F1 results when r_{imb} = 20 (b) W-F1 results when r_{imb} = 20 (c) AUC results when r_{imb} = 20

Figure 5: Performance of model depth (number of GCN layers) on CORA dataset. Markers are denoted as the mean values of the corresponding metric in 15 repeated experiments.

insufficient to solve topology imbalance. 3) Compared with AUC optimization methods, the corresponding TOPOAUC methods achieve better results over all metrics. This shows that solving the topology imbalance issue is vital for node classification and our methods provide an effective way to solve the topology imbalance issue.

Effectiveness of TAIL. We record the changes of $w^{(m,k)}$ for our methods at an imbalance ratio of 20 in Figure 4. We could observe that corresponding importance values of almost all node pairs decrease continuously with the training process. Among them, we could see that importance values of TOPOAUC$_{\text{exp}}$ and TOPOAUC$_{\text{sq}}$ have a more obvious downward trend than TOPOAUC$_{\text{hinge}}$, which is consistent with the higher performance shown in Table 2.

Studies on the model depth. To study the effectiveness of our proposed method in different model depths, we conduct experiments on the number of GCN layers and show the results on the CORA dataset when r_{imb} = 20 in Figure 5. The results when r_{imb} = 10 and 15 are recorded in Appendix. We could observe that the best M-F1, W-F1 and AUC values are all obtained with a 3-layer model at different imbalance ratios. This may be because the model starts to suffer from over-fitting or over-smoothing as model depth increases.

6 CONCLUSION

In this paper, we start an early exploration to apply AUC optimization to graph learning. Facing the complicated imbalance graph data, we propose a unified topology-aware AUC optimization framework, called TOPOAUC, which could simultaneously handle the class and topology imbalance issues induced by the graph data. Concretely, we present a multi-class AUC optimization framework to deal with the class imbalance on the graph. Meanwhile, a topology-aware importance learning mechanism is proposed against the topology imbalance problem, which could inject graph topology information to the AUC optimization objective. Finally, empirical results on three real-world datasets demonstrate the superiority of our proposed framework for node classification.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102000, in part by National Natural Science Foundation of China: U21B2038, U2001202, U1936208, 61931008, 6212200758, and 61976202, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000, and in part by mindspore*, which is a new AI computing framework.

*https://www.mindspore.cn/

REFERENCES

[1] 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.

[2] Jonathon Byrd and Zachary Lipton. 2019. What is the effect of importance weighting in deep learning?. In *International Conference on Machine Learning*. 872–881.

[3] Toon Calders and Szymon Jaroszewicz. 2007. Efficient AUC optimization for classification. In *European Conference on Principles of Data Mining and Knowledge Discovery*. 42–53.

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Neural Information Processing Systems*. 1567–1578.

[5] Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. 2021. Topology-Imbalance Learning for Semi-Supervised Node Classification. *Advances in Neural Information Processing Systems* 34 (2021).

[6] Corinna Cortes and Mehryar Mohri. 2003. AUC optimization vs. error rate minimization. *Neural Information Processing Systems* 16 (2003), 313–320.

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition*. 9268–9277.

[8] Soham Dan and Dushyant Sahoo. 2021. Variance reduced stochastic proximal algorithm for auc maximization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 184–199.

[9] Prathamesh Dharangutte and Christopher Musco. 2021. Graph learning for inverse landscape genetics. In *AAAI Conference on Artificial Intelligence*. 14739–14747.

[10] Yi Ding, Peilin Zhao, Steven Hoi, and Yew-Soon Ong. 2015. An adaptive gradient method for online auc maximization. In *AAAI Conference on Artificial Intelligence*, Vol. 29. 2568–2574.

[11] Hande Dong, Jiawei Chen, Fuli Feng, Xiangnan He, Shuxian Bi, Zhaolin Ding, and Peng Cui. 2021. On the equivalence of decoupled graph convolution network and label propagation. In *the Web Conference*. 3651–3662.

[12] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27, 8 (2006), 861–874.

[13] Mikel Galar, Alberto Fernández, Edurne Barrenechea, and Francisco Herrera. 2013. EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition* 46, 12 (2013), 3460–3471.

[14] Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. 2013. One-pass AUC optimization. In *International Conference on Machine Learning*. 906–914.

[15] Wei Gao and Zhi-Hua Zhou. 2015. On the consistency of AUC pairwise optimization. In *International Joint Conference on Artificial Intelligence*. 939–945.

[16] Zheng-Chu Guo, Yiming Ying, and Ding-Xuan Zhou. 2017. Online regularized learning with pairwise loss functions. *Advances in Computational Mathematics* 43, 1 (2017), 127–150.

[17] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*. 878–887.

[18] James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143, 1 (1982), 29–36.

[19] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks*. 1322–1328.

[20] Siyuan He, Pengcheng Xi, Ashkan Ebadi, Stephane Tremblay, and Alexander Wong. 2021. Performance or Trust? Why Not Both. Deep AUC Maximization with Self-Supervised Learning for COVID-19 Chest X-ray Classifications. *arXiv* abs/2112.08363 (2021).

[21] Junjie Hu, Haiqin Yang, Michael R Lyu, Irwin King, and Anthony Man-Cho So. 2017. Online nonlinear AUC maximization for imbalanced data sets. *IEEE Transactions on Neural Networks and Learning Systems* 29, 4 (2017), 882–895.

[22] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5375–5384.

[23] Thorsten Joachims. 2006. Training linear SVMs in linear time. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 217–226.

[24] Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. 2014. Online and stochastic gradient methods for non-decomposable loss functions. *Neural Information Processing Systems* 27 (2014), 694–702.

[25] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[26] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*. 2980–2988.

[28] Mingrui Liu, Zhuoning Yuan, Yiming Ying, and Tianbao Yang. 2020. Stochastic AUC Maximization with Deep Neural Networks. In *International Conference on Learning Representations*.

[29] Xin Liu, Zhisong Pan, Haimin Yang, Xingyu Zhou, Wei Bai, and Xianghua Niu. 2019. An Adaptive Moment estimation method for online AUC maximization. *PLoS ONE* 14, 4 (2019), 1–16.

[30] Iman Nekooeimehr and Susana K Lai-Yuen. 2016. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. *Expert Systems with Applications* 46, C (2016), 405–416.

[31] Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. 2021. ImGAGN: Imbalanced Network Embedding via Generative Adversarial Graph Networks. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1390–1398.

[32] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine* 29, 3 (2008), 93–93.

[33] Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. 2020. Multi-class imbalanced graph convolutional network learning. In *International Joint Conference on Artificial Intelligence*. 2879–2885.

[34] Wanli Shi, Bin Gu, Xiang Li, Xiang Geng, and Heng Huang. 2019. Quadruply stochastic gradients for large scale nonlinear semi-supervised AUC optimization. In *International Joint Conference on Artificial Intelligence*. 3418–3424.

[35] Jeremias Sulam, Rami Ben-Ari, and Pavel Kisilev. 2017. Maximizing AUC with Deep Learning for Classification of Imbalanced Mammogram Datasets. In *Eurographics Workshop on Visual Computing for Biology and Medicine*. 131–135.

[36] Balázs Szörényi, Snir Cohen, and Shie Mannor. 2017. Non-parametric online auc maximization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 575–590.

[37] Aboozar Taherkhani, Georgina Cosma, and T Martin McGinnity. 2020. AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing* 404 (2020), 351–366.

[38] Robin Vogel, Aurélien Bellet, Stephan Clémen, et al. 2021. Learning Fair Scoring Functions: Bipartite Ranking under ROC-based Fairness Constraints. In *International Conference on Artificial Intelligence and Statistics*. 784–792.

[39] Hongwei Wang and Jure Leskovec. 2020. Unifying graph convolutional neural networks and label propagation. *arXiv* abs/2002.06755 (2020).

[40] Lirong Wu, Haitao Lin, Zhangyang Gao, Cheng Tan, Stan Li, et al. 2021. GraphMixup: Improving Class-Imbalanced Node Classification on Graphs by Self-supervised Context Prediction. *arXiv* abs/2106.11133 (2021).

[41] Zhenhuan Yang, Wei Shen, Yiming Ying, and Xiaoming Yuan. 2020. Stochastic AUC optimization with general loss. *Communications on Pure & Applied Analysis* 19, 8 (2020), 4191.

[42] Zhiyong Yang, Qianqian Xu, Shilong Bao, Xiaochun Cao, and Qingming Huang. 2021. Learning with Multiclass AUC: Theory and Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[43] Zhiyong Yang, Qianqian Xu, Shilong Bao, Yuan He, Xiaochun Cao, and Qingming Huang. 2021. When All We Need is a Piece of the Pie: A Generic Framework for Optimizing Two-way Partial AUC. In *International Conference on Machine Learning*. 11820–11829.

[44] Jian Yin, Chunjing Gan, Kaiqi Zhao, Xuan Lin, Zhe Quan, and Zhi-Jie Wang. 2020. A novel model for imbalanced data classification. In *AAAI Conference on Artificial Intelligence*. 6680–6687.

[45] Yiming Ying and Ding-Xuan Zhou. 2016. Online pairwise learning algorithms. *Neural Computation* 28, 4 (2016), 743–777.

[46] Zhuoning Yuan, Yan Yan, Milan Sonka, and Tianbao Yang. 2021. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In *IEEE International Conference on Computer Vision*. 3040–3049.

[47] Xinhua Zhang, Ankan Saha, and SVN Vishwanathan. 2012. Smoothing multivariate performance measures. *Journal of Machine Learning Research* 13, 1 (2012), 3623–3680.

[48] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2021. GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks. In *ACM International Conference on Web Search and Data Mining*. 833–841.

[49] Dengyong Zhou and Christopher JC Burges. 2007. Spectral clustering and transductive learning with multiple views. In *International Conference on Machine Learning*. 1159–1166.

[50] Zhi-Hua Zhou and Xu-Ying Liu. 2005. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18, 1 (2005), 63–77.

[51] Dixian Zhu, Gang Li, Bokun Wang, Xiaodong Wu, and Tianbao Yang. 2022. When AUC meets DRO: Optimizing Partial AUC for Deep Learning with Non-Convex Convergence Guarantee. *arXiv* abs/2203.00176 (2022).