# QTIAH-GNN: Quantity and Topology Imbalance-aware Heterogeneous Graph Neural Network for Bankruptcy Prediction


Yucheng Liu
University of Science and Technology of China
Hefei, China
liuzy.yucheng@foxmail.com

Zipeng Gao
University of Science and Technology of China, Hefei, China
Hefei, China
gaozp619@mail.ustc.edu.cn

Xiangyang Liu
University of Science and Technology of China
Hefei, China
liuxiangyang@mail.ustc.edu.cn

Pengfei Luo
University of Science and Technology of China
Hefei, China
pfluo@mail.ustc.edu.cn

Yang Yang*
Nanjing University of Science and Technology & The Hong Kong Polytechnic University
Nanjing & Hong Kong, China
yyang@njust.edu.cn

Hui Xiong*
The Hong Kong University of Science and Technology (Guangzhou) & The Hong Kong University of Science and Technology
Guangzhou & Hong Kong, China
xionghui@ust.hk



## ABSTRACT

The timely prediction of bankruptcy is highly desirable to guarantee an upward spiral for overall societal well-being. By extracting multifaceted information from the business interaction networks, Graph Neural Networks (GNNs) may be able to automatically make more informed predictions for bankruptcy, as compared to methods that rely heavily on abundant manpower to a large extent. Yet in real applications, bankruptcy prediction faces the key issue of quantity-imbalance: data usually comes with a long-tailed distribution wherein bankrupt corporates occupy the least of the data proportion but are our target to be identified. Apart from that, the topology-imbalance issue behind graph-structural data exacerbates prediction deterioration: feature propagation is dominated by non-bankrupt nodes through messages passing between nodes; thus, bankrupt nodes receive highly confusing information and could be easily assimilated by nearby non-bankrupt nodes. Unfortunately, the existing GNN methods are not immune to these two imbalance issues. To tackle the challenging but practically useful scenario, we propose a novel bankruptcy prediction model called the Quantity and Topology Imbalance-Aware Heterogeneous Graph Neural Network (QTIAH-GNN) to boost the final performance. Specifically, QTIAH-GNN employs the multi-hierarchy label-aware neighbor selection to conquer the topology-imbalance issue by using the class-semantic representation and the learnable parameterized similarity metric, and employs the imbalance-oriented loss to obtain the optimal tradeoff between the accuracies of the majority and minority classes. In experiments, we evaluate the proposed QTIAH-GNN on two large-scale, real-world datasets. The results show that QTIAH-GNN outperforms other state-of-the-art baselines in terms of prediction accuracy with superior efficiency and generalization ability, has stronger robustness to data imbalance, and provides meaningful model interpretation.



## CCS CONCEPTS

• **Information systems** → **Network data models**; **Enterprise applications**.

## KEYWORDS

Bankruptcy prediction; Graph Neural Network (GNN); Quantity-imbalance; Topology-imbalance

**ACM Reference Format:**
Yucheng Liu, Zipeng Gao, Xiangyang Liu, Pengfei Luo, Yang Yang, and Hui Xiong. 2023. QTIAH-GNN: Quantity and Topology Imbalance-aware Heterogeneous Graph Neural Network for Bankruptcy Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599479


## 1 INTRODUCTION

Of late, an unprecedented global public emergency is leading to an extreme world-wide financial crisis as already stock markets are falling, and some corporates, large or small, come near to bankruptcy more easily now more than ever. The term 'bankruptcy' describes the condition in which corporates struggle to fulfill their obligations and pay liabilities that have become due and thus must undergo debt reorganization and assets liquidation or, in other words, businesses that are facing serious financial distress. Corporate bankruptcy leads, on the one hand, to disastrous macroeconomic consequences for the economic system and the society as a whole and, on the other hand, to catastrophic microeconomic consequences for many stakeholders: investors, managers, governments and, of course, the corporate's employees. In consequence,


*Corresponding authors




*KDD '23, August 6–10, 2023, Long Beach, CA, USA*



ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599479

it is imperative to provide better insight into and prediction of bankruptcy events at macro and micro levels.

Although a wealth of bankruptcy prediction models have been put forward, the majority of the research mainly concentrates on the bankruptcy prediction for publicly-held corporations based on financial statements since comprehensive financial information for publicly-held corporates is broadly available [22]. Nevertheless, financial statement is backward looking and prepared on the basis of going-concern and conservatism assumptions [13], thus it only provide a snapshot of a corporate's financial situation passed by and do not reflect its latest operational status. Furthermore, non-listed companies, particularly small-to-medium enterprises, are the dominant legal forms of corporations in markets, but they do not have the accountability to publish their financial statements regularly [13, 17]. Moreover, bankruptcy is contagious, that is, corporates whose business operations have strong link with bankrupt corporate may follow in bankruptcy [22]. Hence, rich business interactions can be established among corporates no matter whether they are bankruptcy or non-bankruptcy. These interactions can be transformed into graph-structured data through modeling corporates as nodes and the corresponding interactions among them as edges, which renders effective multifaceted non-financial information for bankruptcy prediction.

With the advent of deep learning, Graph Neural Networks (GNNs), architectures designed with non-Euclidean geometric data in mind, have been developed in recent years, following the message passing scheme by which nodes representations are generated using both structural and contextual information. Accordingly, GNNs can achieve state-of-the-art results on purely supervised node classification [7, 20] on such graph-structured data by virtue of the superior nodes representations, rendering the graph-based bankruptcy prediction is possible.

In the real world, bankruptcy prediction faces issues with quantity-imbalance: data usually comes with a Zipfian distribution in which a few dominant classes claim most of the examples, while most of the minority classes are represented by relatively few examples, that is, the highly skewed distribution of bankruptcies versus non-bankruptcies in quantity. Taking the real-world dataset from tianyancha.com as an example, only 4.23% of corporates are bankruptcies while the others are non-bankruptcies. This quantity-imbalanced issue brings non-trivial challenges to existing classification models because the main component of the training loss comes from majority classes and thus the gradient is dominated by majority classes such that the model is updated towards behaving significantly better on majority classes than minority ones, leaving nodes from minority classes under-represented. Graph-structured data are not immune to this quantity-imbalance issue and, to add insult to injury, classification deterioration may be amplified by the addition of the topology-imbalance issue which caused by the asymmetric topological role of nodes from different classes. Specifically, the topology-imbalance is the situation in which minority class nodes are more likely to be neighbored by those in different classes but majority class nodes are more likely to be neighbored by those in the same classes. The most critical challenge, when learning from topology-imbalance graphs, is that the neighborhood aggregation operation in the GNNs will aggregate a number of majority class nodes' features onto minority class nodes, and, if this happens, the suspiciousness of the center nodes will be smoothed out, leading to be indistinguishable.

Taking these challenges into consideration, we propose a novel framework of bankruptcy prediction, namely Quantity and Topology Imbalance-Aware Heterogeneous Graph Neural Network (QTIAH-GNN), which can jointly model the non-financial information from both macro and micro levels and has dual resistance to quantity and topology-imbalance. Specifically, with the constructed heterogeneous tripartite graph, a type-specific transformation is first conducted to project heterogeneous node attributes, with possibly diverse dimensions of feature vectors among different types of nodes, to the same latent vector space. This provides a solid foundation for further magnifying the discrimination between bankruptcies and non-bankruptcies. To combat topology-imbalance issue, we devise a multi-hierarchy label-aware neighbor selector to sample the similar neighbors of the center node and, conversely, filter out as many nodes as possible that are of a different class relative to the central node, using the class-semantic representation that has high energy in summarizing features of all nodes in the same class and a learnable parameterized similarity metric that considers different parts of the semantics captured in the class-semantic representation. Following neighbor selection, QTIAH-GNN conducts aggregation and combination to capture the local and non-local semantics ingrained in the heterogeneous graph. As for quantity-imbalance issue, we design a class-balanced loss that is in expectation closer to the class distribution and, therefore, can achieve the optimal tradeoff between the accuracies of the majority and minority classes.

In summary, our key contribution is threefold:

• ***Problem View.*** We investigate the bankruptcy prediction and imbalance issue in a holistic way, which provides a better understanding of the imbalance issue in the field of corporate failure and serves as a methodological guide for designing bankruptcy prediction models in imbalanced datasets.

• ***Technical View.*** For the first time, QTIAH-GNN takes quantity-imbalance and topology-imbalance into consideration simultaneously, allowing us to efficiently train on all examples without modifying the original graph topology and without easy negatives overwhelming the aggregation and loss.

• ***Practical View.*** Extensive experiments on two realistic large-scale datasets show that QTIAH-GNN consistently outperforms other leading methods with a large gap. Further analysis intuitively reveals the superiority and generalization ability of QTIAH-GNN.

## 2 RELATED WORK

### 2.1 Bankruptcy Prediction

Since the publication of the most well-known bankruptcy prediction models in 1968 [1], a multitude of bankruptcy prediction models has flooded the literature. Most bankruptcy prediction models are either based on classical statistical methods or sophisticated artificial intelligence techniques. The main drawbacks of statistical methods (e.g., multiple discriminant analysis, logistic regression and logit models) lie in the fact that some restrictive requirements, such as the multivariate normality assumptions for explanatory variables, are frequently violated in the practice, which makes these methods theoretically invalid for infinite or high-dimensional samples [15]. Given these constraints, a great interest was paid to the artificial

intelligence techniques that have the powerful capability of dealing with high-dimensional nonlinear data and reflecting non-trivial relationships among economic metrics, including neural network [42], decision tree [6], support vector machines [36], genetic algorithm [9], as well as other approaches and various hybrid approaches. However, these models severely suffer from imbalance issues and their independent variables are often financial ratios drawn from financial statements, which limits their predictive power.

## 2.2 Graph Neural Network (GNN)

Graph Neural Network (GNN) is often empowered by a complex encoder, enabling to learn expressive node representations by considering the node features and graph topology information together following a message passing scheme, among which the most representative method is Graph Convolutional Network (GCN). GCN exploits features in the spectral domain efficiently by using a localized first-order approximation [41]. Recent spatial-based GNNs, including GraphSAGE [11], GAT [37], and many other variants, perform graph convolution operations directly on the graph domain and exploit a set of spatially adjacent neighbors. Beginning with homogeneous graphs, GNNs have been expanded to heterogeneous graphs. There is one specific type of Heterogeneous Graph Neural Network (HGNN) that uses random walk as a tool for sampling strongly correlated heterogeneous neighbors. For example, HetGNN [43] designs a heterogeneous neighbors sampling strategy based on random walk, followed by type-specific Bi-LSTMs to encode features for each type of neighbor nodes. MAGNN [7] applies intra-metapath and inter-metapath aggregation with the attention mechanism to capture comprehensive semantics ingrained in the heterogeneous graph.

## 2.3 Class Imbalance Leaning

Class imbalance refers to a disproportion in the number of examples belonging to each class of a long-tailed dataset [30] and is well-known as a significant data-intrinsic problem which plagues many real-world application domains, comprising fraud detection [20], financial risk analysis [21], topic classification [40], object detection [19], to name a few. Class imbalance, unlike selection bias, data loss or data corruption, is not a problem that derives from deficiencies in data acquisitions, mechanical errors, or measurements but rather occurs due to the essence of practical application and, therefore, this problem is almost inescapable [34].

In the Euclidean data space, approaches for tackling the imbalance problem can be roughly divided into two streams: data-level and algorithm-level. The data-level approaches aim to generate a balanced class distribution by over-sampling minority class samples or under-sampling majority class samples, or both [23]. These approaches relieve the class imbalance but come with a high potential risk of overfitting as no extra information is introduced or valuable information is discarded [3], and the effect of noise examples is also amplified [21]; thus may limit the generalization of the model [21], especially when the classes are extremely imbalanced [2]. On account of these disadvantages of the data-level approaches, the present works focus on algorithm-level approaches, namely, how to devise an efficient class-balanced loss [5]. An intuitive strategy is to weight samples based on the inverse of class frequency [39] or use a smoothed version, inverse square root of class frequency [24] or effective number [5]. Another line of splendid work makes attempts to have the optimal trade-off between per-class margins by designing a label-distribution-aware margin loss function[2].

In the non-Euclidean data space, traditionally successful countermeasures have had limited power with complex, structured graph data handled by deep learning models. DR-GCN is a pioneer work investigating imbalanced problem with graphs, but it cannot scale to real-world large graphs [32]. Recently, some works have been proposed to augment the graph-structured data: GraphSMOTE, follows the oversampling strategy and synthesizes new nodes by extending previous SMOTE to graph-structured data [46]; ImGAGN, simulates virtual minor nodes via the conditional Generative Adversarial Networks (GAN) [28]; GraphENS, synthesizes the whole ego network for minor class by considering neighbor structure[25]. Nevertheless, these studies only take the quantity-imbalance into consideration and neglect the topology-imbalance.

# 3 PRELIMINARIES

In this section, we first give formal definitions of some important terminologies pertinent to our work, and then present the problem formulation of graph-based bankruptcy prediction, followed by data description.

## 3.1 Definition and Problem Statement

In the graph-structured data, we define class imbalance to include quantity-imbalance and topology-imbalance.

Definition 1 (Quantity-Imbalance Ratio). *Given a set of labeled nodes $\mathcal{V}$ having m classes, $\{C_1, \cdots, C_m\}$. $|C_i|$ is the size of i-th class, referring to the number of samples belonging to that class. The quantity-imbalance ratio is defined as $QIR = \frac{min_i|C_i|}{max_i|C_i|}$, to measure the extent of quantity-imbalance. Therefore, QIR lies in the range $[0, 1]$. Specially, if $QIR = 1$, node set $\mathcal{V}$ is quantity-balanced.*

Definition 2 (Topology-Imbalance Ratio). *Suppose we have a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the labeled node set and $\mathcal{E}$ is the edge set; each node $v \in \mathcal{V}$ have a class label $y_v$; $\mathcal{N}_v^1$ denotes the set of first-order neighbors of node v. In order not to be sensitive to quantity-imbalance, inspired by [18], the topology-imbalance ratio is defined as:*

$$TIR = \frac{1}{m}\sum_{i=1}^{m}\left(max\left\{h_i - \frac{|C_i|}{|\mathcal{V}|}, 0\right\}\right)$$

$$h_i = \frac{\sum_{v \in C_i}\left|\{u|u \in \mathcal{N}_v^1, y_u = y_v = i\}\right|}{\sum_{v \in C_i}\left|\mathcal{N}_v^1\right|}.$$

*TIR is in the range of $[0, 1]$ and a value close to 1 corresponds to a strong topology-balance while a value close to 0 indicates a strong topology-imbalance.*

**Problem Statement.** ***(Graph-Based Bankruptcy Prediction).*** The graph-based bankruptcy prediction is defined on the imbalanced heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ associated with a node type mapping function $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge type mapping function $\phi : \mathcal{E} \rightarrow \mathcal{B}$. $\mathcal{A}$ and $\mathcal{B}$ denote the predefined sets of node types and edge types, respectively, with $|\mathcal{A}| + |\mathcal{B}| \geq 2$. Each node is associated with a feature vector $\mathbf{x}_i \in \mathcal{X}$. Nodes including the target

nodes (e.g., corporates) and the background nodes (e.g., groups and brands), and the target node $v$ comes with the label $y_v \in \{0, 1\}$. Graph-based bankruptcy prediction is to find the bankruptcy nodes that significantly differ from the other non-bankruptcy nodes on the imbalanced heterogeneous graph $\mathcal{G}$, which can be regarded as a supervised binary classification task on the imbalanced heterogeneous graph.

## 3.2 Data Description

Across many empirical studies, wide-ranging non-financial factors have been shown to have predictive power in corporate bankruptcy and they are functions, among other things, of the financial factors themselves [15]. These non-financial factors are mainly divided into three groups: the company's basic information (e.g., geographical location [17], age [22], employment [27], company size [15, 17], business category [22], stockholder structure [33], business sector [15], company patents [17], corporate governance indicators [15]), soft information (e.g., management quality [10], regulatory compliance [17], preferences of domain experts [17]), and external information (e.g., market position [17], the network of enterprises [35], market information [15]).

Armed with this insight, We collect a large-scale industrial dataset (BP) (ranging from 2019/12/01 to 2020/12/01) from an online business inquiry platform provided by Tianyancha inc[1] on the premise of complying with the security and privacy policies. It contains 63,180 corporates, also known as the target nodes, of which 2,672 are bankruptcy corporates, accounting for 4.23%. Through the API interface provided by Tianyancha inc, corporates that petition the court to declare bankruptcy and execute bankruptcy proceedings from 2021/01/01 to 2022/05/31 are marked as bankruptcy. Except for such nodes as corporates, the dataset used includes two other types of nodes, namely, groups and brands, also known as the background nodes, which can provide auxiliary information for bankruptcy prediction. Specifically, groups control multiple subsidiaries, which can be entity investment institutions, liability investment institutions and alternative investment institutions. Brands are projects of corporates, which are the targets of business events. These three types of nodes are connected by multiple edges, including competition edges, investment edges, shareholder edges, business edges and among others. In addition, each node is associated with its own non-financial attributes: the corporate node has 261 dimensional features, which cover corporate age, business category, registered capital, market position and so on; the group node has 37 dimensional features, which cover geographical location and so on; the brand node has 183 dimensional features, which cover brand age, business category, market information and so on. The dataset statistical information is shown in Tabel 1.

# 4 METHODOLOGY

In this section, we formally present QTIAH-GNN to resolve quantity-imbalance and topology-imbalance issues in bankruptcy prediction described in Section 1. In the QTIAH-GNN framework, four different modules are unified into a GNN, consisting of (1) topology-agnostic embedding; (2) multi-hierarchy label-aware neighbor selection; (3) aggregation and combination; (4) imbalance-oriented

[1]www.tianyancha.com

**Table 1: Statistical of BP dataset**

| Dataset | Node | Edge | QIR | TIR |
|---|---|---|---|---|
| BP | Coporate: 63,180<br>Brand: 34,588<br>Group: 4,148 | Investment: 209,264<br>Competition:3,883,124<br>Business event:63,656<br>Member:2,838<br>Business:75,013<br>Shareholder:136,676 | 0.044 | 0.021 |

model training. The overall architecture of QTIAH-GNN is shown in Figure 1.

## 4.1 Topology-agnostic Embedding

For a heterogeneous graph associated with node attributes, the dimensions of feature vectors among different types of nodes may differ greatly, even by several orders of magnitude. Even if feature vectors happen to be the same dimension, they should also be in different feature spaces [4]. To this end, as an initial step, a type-specific transformation, parametrized by a weight matrix, is applied to every node to project the features of different types of nodes into the same higher-order feature space. This projection process can be shown as follows:

$$\mathbf{h}_v^0 = \mathbf{W}_A \cdot \mathbf{x}_v^A , \tag{1}$$

where $\mathbf{x}_v^A \in \mathfrak{R}^{d_A}$ is the original feature vector of node $v$ belonging to type $A \in \mathcal{A}$, $\mathbf{h}_v^0$ is the projected feature vector of node $v$, $\mathbf{W}_A \in \mathfrak{R}^{d \times d_A}$ is the parametric weight matrix of type $A$'s nodes.Furthermore, the topology-agnostic embedding layer drops all structural information and only encodes node attributes, which offers promise for further magnifying the discrimination between bankruptcies and non-bankruptcies via topological information.

## 4.2 Multi-hierarchy Label-aware Neighbor Selection

The core idea of the neighbor aggregation mechanism in GNNs is to aggregate the feature information of neighbor nodes along the edges of the graph, which means that the class prediction of each node is no longer determined only by its own feature, but is forcefully intervened by the nodes connected to it as well. Under the topology-imbalanced scenario, minority nodes have less chance to be neighbored by nodes in the same class; thus, directly aggregating all neighbor nodes' features onto the central node will actually make the unique semantic characteristics of the minority nodes assimilated by nearby majority nodes and more difficult to be distinguished. To combat this challenge, before the neighbor aggregation, we need to sample nodes that are of the same class as the central node, and conversely, filter out as many nodes as possible that are not of the same class as the central node to avoid majority nodes dominating the feature propagation and even the oversmoothing issue.

### 4.2.1 *Virtual Label Assign.*

*Class-Semantic Representation.* The class-semantic representation aims to transform each class semantic information into a powerfully

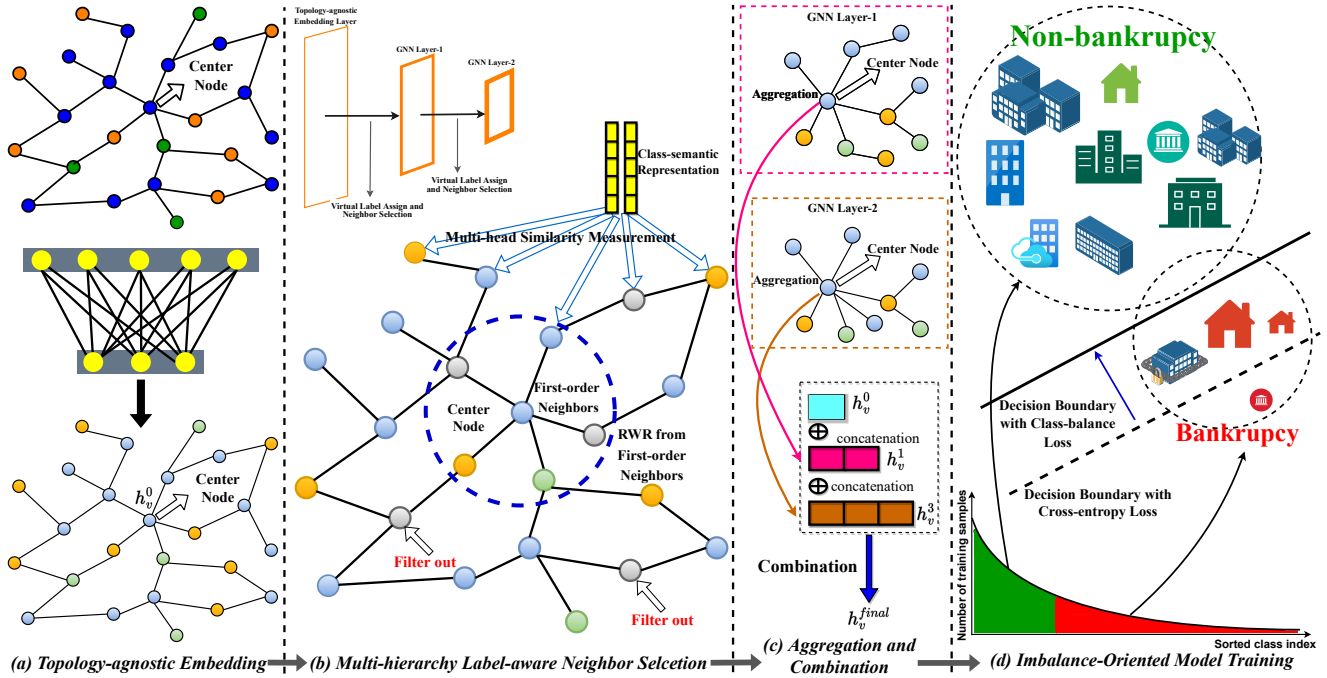



**Figure 1: The overall architecture of QTIAH-GNN. Different colors represent different types of nodes, where gray nodes represent the nodes that are filtered out. Each layer of GNN selects the neighbors of the center node based on the virtual label, but the virtual labels are not the same in each layer as node features are updated with layers, so the neighbors of the center node in each layer are distinct.**

expressive vector which has high energy in summarizing features of all nodes in the same class. We can exploit a readout function $\mathbb{F}$ (e.g., average function) to obtain the class-semantic representation vector at layer $l$ (denoted as$\mathrm{Y}^l_{C_i}$) for each class $C_i$ as follows:

$$\mathrm{Y}^l_{C_i} = \mathbb{F}\left(\left\{\mathbf{h}^l_v | v \in C_i\right\}\right). \tag{2}$$

*Multi-head Similarity Metric.* For the sake of increasing stability and expressivity, we design a multi-head weighted cosine similarity metric to measure the similarity between the node $v$'s feature vector $\mathbf{h}^l_v$ and the class-semantic representation vector $\mathrm{Y}^l_{C_i}$ as follows:

$$\begin{aligned} s^{k,l}_{vC_i} &= cos\left(\mathbf{w}^l_k \odot \mathbf{h}^l_v, \mathbf{w}^l_k \odot \mathrm{Y}^l_{C_i}\right); \\ S^l_{vC_i} &= \frac{1}{n}\sum_{k=1}^{n} s^{k,l}_{vC_i}; \end{aligned} \tag{3}$$

where $\mathbf{w}$ is a learnable weight vector which has the same dimension as the input vectors $\mathbf{h}$ and Y, and $\odot$ denotes the Hadamard product. Specifically, $s^{k,l}_{vC_i}$ computes the cosine similarity between the two input vectors $\mathbf{h}_v$ and $\mathrm{Y}_{C_i}$ at layer $l$ for the $k$-th perspective, where each perspective considers one part of the semantics captured in the class-semantic representation vector $\mathrm{Y}^l_{C_i}$ and learns to highlight different dimensions of the class-semantic representation vector$\mathrm{Y}^l_{C_i}$. Subsequently, $s^{k,l}_{vC_i}$ is computed $n$ times independently and takes their average as the final similarity measure $S^l_{vC_i}$.

Next, we assign a virtual label to each node at each layer of GNN by calculating the multi-head weighted cosine between node $v$'s feature vector $\mathbf{h}^l_v$ and each class-semantic representation vector $\mathrm{Y}^l_{C_i}$ and taking the maximum value as follows:

$$\hat{y}^l_v = \underset{i\in\{1,\cdots,m\}}{\arg\max}\ S^l_{vC_i}. \tag{4}$$

This virtual label guides subsequent node sampling and filtering.

*4.2.2 Multi-hierarchy Label-aware Neighbor Sampling.* Most commonly, the neighbor sampling method is to sample each node's first-order neighbors. If this is the case, node embedding may be insufficient due to limited neighbor sizes and noise neighbors with incorrect relations or attributes [43]. Topology-imbalance will exacerbate this problem since minority class nodes are surrounded by majority class nodes. To ensure that the minority nodes have enough neighbors of the same class, we design a multi-hierarchy label-aware neighbor sampling strategy based on random walk with restart (RWR). To be specific, we start random walks from node $v$'s first-order neighbors at each layer of GNN, the walks iteratively run to the neighbors of the current node or return to the starting node with a probability $p$. When returning to the starting node, RWR will start from a different edge than before and run until it successfully collects a fixed number (node degree) of nodes whose virtual label is the same as central node $v$. Note that we start random walks from node $v$'s first-order neighbors rather than node $v$ itself to guarantee determinacy and randomness, and the final set of neighbors $\mathcal{N}_v$

includes both first-order and higher-order neighbors. In addition, we specify the maximum length of the RWR.

*4.2.3 Multi-hierarchy Label-aware Neighbor Filtering.* To prevent easy negatives from overwhelming the aggregation, we filter out neighbors with the different virtual labels from the central node $v$ and neighbors with low similarity at each layer of GNN. The filtered neighbor set of node $v$ could be denoted as $\mathbb{N}_v^l$ and formulated as:

$$\mathbb{N}_v^l = \left\{ u \in \mathcal{N}_v | \hat{y}_u^l = \hat{y}_v^l \cap \mathcal{S}_{uC_I}^l > \rho \right\}, \tag{5}$$

where $\rho$ is the filter threshold, $I = \hat{y}_v^l$.

*4.2.4 Optimization.* If the label-aware sampling could not effectively sample the neighbors from the same classes and the label-aware filtering could not effectively filter the neighbors from different classes, it will impair the function of following GNN layers and result in sub-optimal overall performance. The neighborhood sampling and filtering is learnable due to the parameterization of the multi-head cosine similarity metric. To train the neighborhood sampling and filtering with a direct supervised signal from real label, we define the cross-entropy loss as:

$$\mathcal{L}_{NSF} = -\sum_{l=1}^{L} \sum_{v \in \mathcal{V}} y_v log \hat{y}_v^l, \tag{6}$$

where $L$ is the total number of layers of the GNN.

## 4.3 Aggregation and Combination

Following the filtering step, a message passing based graph neural network is designed to aggregate information from all selected neighbors. The aggregation operation entails encoding each node's embedding separately from the aggregated embeddings of its neighbors since they are likely to be in different classes in the topology-imbalanced scenarios.

Let $\mathbf{h}_v^l$ is the hidden features of node $v$ at the $l$-th layer, the $l$-th layer updates $\mathbf{h}_v^l$ for every $v \in \mathcal{V}$ by aggregating all the information from selected neighbors can be formulated as:

$$\mathbf{h}_v^l = \mathbf{h}_v^{l-1} \oplus AGG^l \left\{ \mathbf{h}_u^{l-1} | u \in \mathbb{N}_v^{l-1} \right\}, \tag{7}$$

where $AGG^l$ is the aggregator function at layer $l$.

Different GNN layers output nodes' embeddings with different smoothness[44] and each layer collect information with different locality—earlier layers are more local, while later layers capture increasingly more global and richer context information[47]; thus, combining the intermediate embeddings output by the different layers can further improve classification performance. To this end, we combine the intermediate embeddings with different smoothness as follows:

$$\mathbf{h}_v^{final} = \mathbf{h}_v^0 \oplus \mathbf{h}_v^1 \oplus \cdots \oplus \mathbf{h}_v^L, \tag{8}$$

to explicitly capture local and non-local dependencies via concatenation $\oplus$.

## 4.4 Imbalance-Oriented Model Training

The representation learning for the bankruptcy prediction faces the challenge of quantity-imbalance issue, where the decision boundary of the trained classifier is mainly decided by the majority classes, leading to the minority classes having higher uncertainty in the prediction space and the associated classifier confidence levels are low. This is because most classifiers adopt the standard cross-entropy loss or hinge loss as the loss function. The representation of the standard hinge loss $\mathcal{L}_{hinge}$ would be:

$$\mathcal{L}_{hinge} = \sum_{v \in \mathcal{V}} max \left\{ 0, \underset{j \neq y}{max}(p_v^j) - p_v^y + 1 \right\}, \tag{9}$$

where $p_v^j$ denotes the output of the classifier's decision function for the $j$-th class with respect to node $v \in \mathcal{V}$.

Intuitively, we can let the minority classes have larger margins so as to enlarge the classifier confidence regions of the minority classes. The margin $\eta(i)$ of the $i$-th class refers to the minimum distance of the node in the $i$-th class to the decision boundary. Inspired by the trade-off between the class margins in [2] for two classes, we employ a class-dependent margin of the form:

$$\eta(i) = \frac{C}{[n(i)]^{\frac{1}{4}}}, \tag{10}$$

where $n(i)$ denotes the number of nodes in class $i$, and $C$ is a hyperparameter. Therefore, the hinge loss can be rewritten as:

$$\mathcal{L}_{hinge} = \sum_{v \in \mathcal{V}} max \left\{ 0, \underset{j \neq y}{max}(p_v^j) - p_v^y + \eta(y_v) \right\}. \tag{11}$$

Empirically, the non-smoothness of hinge loss may pose difficulties for optimization. The smooth relaxation of above hinge loss is the following cross-entropy loss with class-dependent margins:

$$\mathcal{L}_{SM-hinge} = -\sum_{v \in \mathcal{V}} log \frac{e^{p_v^y - \eta(y_v)}}{e^{p_v^y - \eta(y_v)} + \sum_{j \neq y_v} e^{p_v^y}}. \tag{12}$$

In addition, due to the small population, the minority nodes have less engagement in the loss function of training and the classifier tends to be overwhelmed by the majority nodes while ignoring the minority nodes. That is, the representation of the minority nodes cannot be adequately learned. One simple solution seems to be the introduction of costs to re-weight the minority node errors. As a class-dependent regularization technique, the above loss function $\mathcal{L}_{SM-hinge}$ is orthogonal to the re-weighting scheme [2]. To this end, we utilize the effective number [5] to re-weight the loss function $\mathcal{L}_{SM-hinge}$, thereby yielding the class-balanced loss:

$$\mathcal{L}_{CB} = -\sum_{v \in \mathcal{V}} \frac{1-\beta}{1-\beta^{n(y_v)}} log \frac{e^{p_v^y - \eta(y_v)}}{e^{p_v^y - \eta(y_v)} + \sum_{j \neq y_v} e^{p_v^y}}. \tag{13}$$

The effective number $E_n = (1-\beta^{n(i)})/(1-\beta)$ of nodes enables us to use a hyperparameter $\beta \in [0, 1)$ to smoothly adjust the class-balanced term $(1-\beta)/(1-\beta^{n(i)})$ between no re-weighting and re-weighing by inverse class frequency: $\beta = 0$ corresponds to no re-weighting and $\beta \to 1$ corresponds to re-weighing by inverse class frequency.

Finally, the overall loss function is formulated as follows, where $\lambda$ is a weighting parameter to be tuned.

$$\mathcal{L} = \mathcal{L}_{CB} + \lambda \mathcal{L}_{NSF}. \tag{14}$$

# 5 EXPERIMENTS

In this section, we present experiments to demonstrate the efficacy of QTIAH-GNN for bankruptcy prediction, with the aim of answering the following research questions.

- **RQ1.** Does QTIAH-GNN outperform the state-of-the-art methods for bankruptcy prediction?
- **RQ2.** How do the key components benefit the prediction?
- **RQ3.** How robust is QTIAH-GNN with respect to its hyper-parameter values?
- **RQ4.** How interpretable is QTIAH-GNN?
- **RQ5.** Can QTIAH-GNN generalize well to other imbalanced heterogeneous datasets? (See Appendix A.2)

## 5.1 Experimental Setting

*5.1.1 Baselines.* We considered a wide range of representative methods for bankruptcy prediction, which can be divided into three groups: feature-based methods, imbalance-agnostic GNN methods and imbalance-aware GNN-based methods. The feature-based representatives are SVM and MLP[29], which only use node features. The imbalance-agnostic GNN representatives are GCN [41], GraphSAGE [11], GAT [37], HGT [14] and RGCN [31], which utilize both node features and topological structure. It is worth noting the former two types of methods were all equipped with the traditional countermeasures dealing with class imbalance, i.e., Over-sampling, SMOTE[2], Under-sampling and Re-weight. The over-sampling ratio, under-sampling ratio and re-weight ratio were all set as the inverse of class frequency. The imbalance-aware GNN-based methods are GraphSMOTE [46], GraphENS [25], FRAUDRE [44] and PC-GNN [20], which take advantage of the node features and topological structure but involve the class imbalanced problem as well. See Appendix A.1 for a detailed description of these baselines.

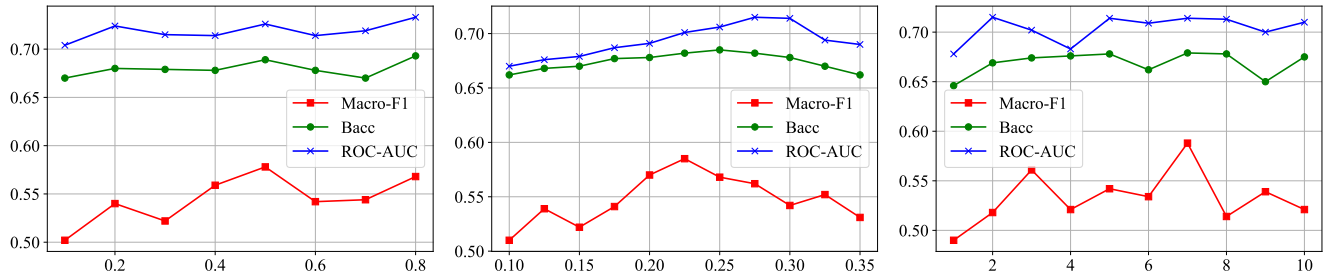
*5.1.2 Evaluation Metrics.* Since the BP dataset has imbalanced classes, like previous studies [20, 25, 28, 44], we utilized Macro-F1, Balanced Accuracy (BAcc) and ROC-AUC to digitize the performance of models. Macro-F1 is the arithmetic mean (unweighted mean) of the F1 scores calculated per class, so it treats all classes equally regardless of their support values. Balanced accuracy is a further development on the standard accuracy metric where it's adjusted to perform better on imbalanced datasets. The way it does this is by calculating the sensitivity and specificity, instead of dividing the total correct prediction over the total prediction as is the case with standard accuracy. ROC-AUC is computed based on the relative ranking of prediction probabilities of all samples, which could eliminate the influence of imbalanced classes.

*5.1.3 Implementation Details.* We implemented our model by PyTorch 1.12.0[3] with Python 3.9.12 using a NVIDIA GeForce RTX 3090 GPU with 24GB memory on a Linux machine. In the process of model training, we used the Adam optimizer [16] for parameter optimization and we set the learning rate as 0.001. In QTIAH-GNN, we set the embedding size in the topology-agnostic embedding module to 64, the number of graph convolution layers to 2, the number of heads in multi-head weighted cosine similarity metric to 5. The parameters of baselines were set up either as their default values or the same as in our model and were all tuned to be optimal to ensure fair comparisons. For the GNNs and MLP, the number of hidden layers was set as K = 2 to avoid over-smoothing. SVM and MLP were implemented based on Scikit-learn[26]. GCN, GraphSAGE and GAT were implemented based on DGL [38]. RGCN and HGT were implemented based on OpenHGNN [45]. GraphSMOTE, GraphENS, FRAUDRE and PC-GNN were implemented using their provided source code. To avoid over-fitting, 60% nodes in the graph

[2]SMOTE is the most popular oversampling-like approach, which generates synthetic samples along line segments between in-class neighboring samples to augment the minority class.
[3]https://pytorch.org/

**Table 2: Performance comparison for bankruptcy prediction**

| Models | Macro-F1 | BAcc | ROC-AUC |
|---|---|---|---|
| SVM | 0.490 ± 0.000 | 0.500 ± 0.000 | 0.674 ± 0.000 |
| SVM+*Oversampling* | 0.270 ± 0.000 | 0.391 ± 0.000 | 0.502 ± 0.001 |
| SVM+*SMOTE* | 0.485 ± 0.000 | 0.417 ± 0.001 | 0.529 ± 0.000 |
| SVM+*Undersampling* | 0.335 ± 0.000 | 0.391 ± 0.000 | 0.510 ± 0.002 |
| SVM+*Re-weight* | 0.189 ± 0.000 | 0.421 ± 0.000 | 0.347 ± 0.000 |
| MLP | 0.489 ± 0.008 | 0.500 ± 0.014 | 0.663 ± 0.003 |
| MLP+*Oversampling* | 0.409 ± 0.017 | 0.629 ± 0.013 | 0.662 ± 0.005 |
| MLP+*SMOTE* | 0.427 ± 0.012 | 0.634 ± 0.019 | 0.668 ± 0.007 |
| MLP+*Undersampling* | 0.253 ± 0.008 | 0.590 ± 0.011 | 0.669 ± 0.013 |
| MLP+*Re-weight* | 0.268 ± 0.005 | 0.583 ± 0.009 | 0.676 ± 0.018 |
| GCN | 0.529 ± 0.007 | 0.541 ± 0.009 | 0.535 ± 0.012 |
| GCN+*Oversampling* | 0.484 ± 0.056 | 0.508 ± 0.028 | 0.508 ± 0.101 |
| GCN+*SMOTE* | 0.516 ± 0.017 | 0.533 ± 0.052 | 0.506 ± 0.099 |
| GCN+*Undersampling* | 0.509 ± 0.014 | 0.532 ± 0.071 | 0.522 ± 0.080 |
| GCN+*Re-weight* | 0.463 ± 0.008 | 0.581 ± 0.058 | 0.591 ± 0.072 |
| GraphSAGE | 0.520 ± 0.003 | 0.534 ± 0.008 | 0.538 ± 0.002 |
| GraphSAGE+*Oversampling* | 0.510 ± 0.020 | 0.530 ± 0.007 | 0.529 ± 0.051 |
| GraphSAGE+*SMOTE* | 0.518 ± 0.014 | 0.524 ± 0.027 | 0.523 ± 0.019 |
| GraphSAGE+*Undersampling* | 0.520 ± 0.008 | 0.548 ± 0.006 | 0.551 ± 0.010 |
| GraphSAGE+*Re-weight* | 0.501 ± 0.002 | 0.591 ± 0.009 | 0.593 ± 0.013 |
| GAT | 0.487 ± 0.072 | 0.526 ± 0.004 | 0.516 ± 0.057 |
| GAT+*Oversampling* | 0.482 ± 0.018 | 0.511 ± 0.008 | 0.510 ± 0.011 |
| GAT+*SMOTE* | 0.501 ± 0.050 | 0.533 ± 0.013 | 0.529 ± 0.058 |
| GAT+*Undersampling* | 0.448 ± 0.005 | 0.538 ± 0.003 | 0.540 ± 0.016 |
| GAT+*Re-weight* | 0.458 ± 0.002 | 0.561 ± 0.007 | 0.563 ± 0.009 |
| RGCN | 0.503 ± 0.008 | 0.515 ± 0.012 | 0.519 ± 0.005 |
| RGCN+*Oversampling* | 0.475 ± 0.019 | 0.508 ± 0.025 | 0.507 ± 0.073 |
| RGCN+*SMOTE* | 0.482 ± 0.022 | 0.528 ± 0.017 | 0.526 ± 0.064 |
| RGCN+*Undersampling* | 0.493 ± 0.004 | 0.529 ± 0.009 | 0.538 ± 0.006 |
| RGCN+*Re-weight* | 0.470 ± 0.007 | 0.557 ± 0.003 | 0.554 ± 0.010 |
| HGT | 0.469 ± 0.005 | 0.506 ± 0.003 | 0.501 ± 0.003 |
| HGT+*Oversampling* | 0.461 ± 0.008 | 0.462 ± 0.084 | 0.482 ± 0.147 |
| HGT+*SMOTE* | 0.479 ± 0.087 | 0.469 ± 0.059 | 0.490 ± 0.103 |
| HGT+*Undersampling* | 0.475 ± 0.012 | 0.515 ± 0.007 | 0.516 ± 0.027 |
| HGT+*Re-weight* | 0.434 ± 0.006 | 0.550 ± 0.006 | 0.567 ± 0.083 |
| GraphSMOTE | 0.471 ± 0.031 | 0.609 ± 0.008 | 0.640 ± 0.022 |
| GraphENS | 0.330 ± 0.076 | 0.517 ± 0.029 | 0.514 ± 0.060 |
| FRAUDRE | 0.349 ± 0.007 | 0.553 ± 0.053 | 0.672 ± 0.018 |
| PC-GNN | 0.357 ± 0.002 | 0.653 ± 0.006 | 0.682 ± 0.004 |
| **QTIAH-GNN** | **0.542±0.009** | **0.678±0.003** | **0.714±0.016** |




(a) Training Ratio (b) The Filter Threshold (c) The Number of Heads

**Figure 2: Parameter sensitivity of QTIAH-GNN w.r.t. training ratio, the filter threshold, the number of heads in multi-head weighted cosine similarity metric.**

were used for training, 20% nodes for validation and 20% nodes for testing. The train-valid-test split was based on the stratified sampling provided by Scikit-learn to ensure that the imbalance ratio is consistent in both train, valid and test set. For all methods, we reported the average value and standard deviation for 10 repetitions.

## 5.2 Performance Comparison (RQ1)

To answer RQ1 and verify the effectiveness of QTIAH-GNN, we challenged it against existing methods in the task of bankruptcy prediction and considered different perspectives. The corresponding Macro-F1, Balanced Accuracy and ROC-AUC scores were reported in Table 2. We can find that our model consistently outperformed all the baselines by a large margin, improving the Macro-F1, Balanced Accuracy and ROC-AUC scores of the best competitors by up to 4.23%, 3.83%, and 4.69%, respectively.

The ROC-AUC scores of feature-based methods were comparable and higher than that of imbalance-agnostic GNN methods, but the Balanced Accuracy scores for feature-based methods were only around 0.500 and were lower than that of GNN-based methods. That is to say, these feature-based methods ignored the quantity-imbalance issue and classified all bankruptcies as non-bankruptcies to minimize the overall classification error during training, resulting in the inclination to learning towards majority classes and the catastrophic forgetting of previous learned instances in minority classes. These results also indicated that the relationship between corporates can enhance bankruptcy prediction, which is consistent with the above analysis.

GraphSAGE and GAT are two advanced GNN-based methods. Unlike GCN taking the full-size neighbors equally to obtain the target node embedding, GraphSAGE employed the node sampling to obtain a fixed number of neighbors, GAT dynamically assigned an aggregation weight to each neighbor, which, to some extent, can resolve the topology-imbalance issue. However, these two methods did not explore label distribution to identify whether one neighbor is bankruptcy or non-bankruptcy. Consequently, the node sampling and the attention mechanism might have little effect when the central node had very few same class neighbors. HGT and RGCN judiciously utilized the heterogeneous node features, which may help improve the embedding performance. Nevertheless, with these algorithms, no countermeasures were taken to alleviate the imbalance issue, thus it performed worse than imbalance-aware GNN-based methods as expected.

GraphSMOTE and GraphENS are two state-of-the-art data augmentation methods for graph-structured data, improved significantly in terms of Balanced Accuracy and ROC-AUC scores. The reason is that these two methods oversampled the bankruptcy nodes to equalize the number of bankruptcies and non-bankruptcies in the training set. FRAUDRE and PC-GNN are two algorithm-level methods for tackling the imbalance problem and had higher Balanced Accuracy and ROC-AUC scores, demonstrating that solving the class-imbalanced problem on graph-structured data at the algorithm level is feasible. While these four methods still had much room for improvement in terms of Macro-F1. The proposed method QTIAH-GNN can also be thought of as an algorithm-level method. Notwithstanding, it achieved better performance than all baselines, the improvements could be attributed to both awareness of quantity and topology imbalance.

We can also find that our model consistently outperformed all the combinatorial methods by a large margin, directly applying existing countermeasures to graph data may lead to sub-optimal results due to the non-Euclidean property of graphs. Oversampling-like approaches led to worse results and Re-weight gave the greatest improvement, which may indicate that the class-imbalanced problem in graph-structured data cannot be directly addressed from the data-level, requiring consideration of the graph topology structure and node features.

## 5.3 Ablation Study (RQ2)

To answer RQ2, we further conducted a series of ablation studies on some modules in the QTIAH-GNN. The four variants tested are: (1) QTIAH-V1: which do not include the topology-agnostic embedding module so the graph convolution operation is performed on the node attributes directly; (2) QTIAH-V2: which do not include the label-aware neighbor selection module so the graph convolution operation is performed only on the full-size first-order neighbors without neighbors sampling and filtering; (3) QTIAH-V3: in which only the neighbor aggregation operation is preserved and only the embedding output by last GNN layer is taken as the final node

embeddings; (4) QTIAH-V4: where only the standard cross-entropy loss is exploited rather than the imbalance-oriented loss.

As Table 3 shows, the full model was superior to the above four variants in terms of Balanced Accuracy and ROC-AUC scores. In particular, the performance of QTIAH-V2 with the full-size first-order neighbors and QTIAH-V4 with the standard cross-entropy loss greatly under-performed the QTIAH-GNN in terms of all evaluation metrics, which indicated that the neighbor selection module and the imbalance-oriented loss function effectively got the better of topology and quantity-imbalance issue.

**Table 3: Ablation Study**

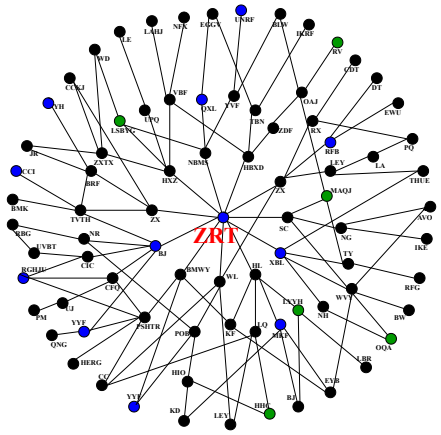
| Models | Macro-F1 | BAcc | ROC-AUC |
|---|---|---|---|
| QTIAH-V1 | 0.365±0.001 | 0.631±0.006 | 0.651±0.008 |
| QTIAH-V2 | 0.416±0.008 | 0.602±0.014 | 0.567±0.017 |
| QTIAH-V3 | **0.545±0.006** | 0.657±0.005 | 0.708±0.013 |
| QTIAH-V4 | 0.489±0.013 | 0.500±0.008 | 0.498±0.038 |
| **QTIAH-GNN** | 0.542±0.009 | **0.678±0.003** | **0.714±0.016** |

## 5.4 Sensitivity Analysis (RQ3)

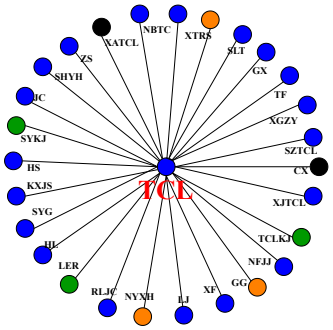
To answer Q3, we investigated the sensitivity of QTIAH-GNN to the parameters with regard to training ratio, the filter threshold, the number of heads in multi-head weighted cosine similarity metric. We varied the percentage of training ratio from 10% to 80% with step size 10%, the filter threshold from 0.1 to 0.35 with step size 0.025, and the number of heads in multi-head weighted cosine similarity metric from 1 to 10 with step size 1. The experimental results are shown in Figure 2. From experimental results, in general, we can observe that: (1) Our method maintained acceptable results with most hyper-parameters combinations and had relative stability. (2) With an increase in the filter threshold, performance initially improved but then gradually declined. The reason is that QTIAH-GNN needed a suitable threshold to filter dissimilar neighbors and larger threshold may destroy the neighbor aggregation mechanism of GNN.

## 5.5 Case Study (RQ4)

To further analyze the neighbor selection of our QTIAH-GNN model, we picked one bankrupt corporate (Zebra Fast Running Technology Co. LTD, short as ZRT) and one non-bankrupt corporate (TCL Technology Co. LTD, short as TCL) from the test set for case study. The neighbor selection visualization results are shown in Figure 3. The blue, orange and green nodes represent companies, brands and groups, respectively. The black nodes represent the nodes that were filtered out. The first row displays bankrupt corporate ZRT's neighbor selection process, we can find that the neighbor selection sampled more higher-order neighbors by RWR and filtered out more nodes with different classes or unfamiliar features since ZRT was neighbored by a large number of the different class corporates. The second row displays non-bankrupt corporate TCL's neighbor selection process, we can find that the neighbor selection only sampled the first-order neighbors and filtered out few nodes since TCL was almost neighbored by non-bankrupt corporates.




(a) Bankruptcy Corporate: ZRT




(b) Non-bankruptcy Corporate: TCL

**Figure 3: Visualization of two corporates' neighbor sampling and filtering.**

## 6 CONCLUSION

Real-world data often exhibit long-tailed distribution with heavy class imbalance, posing significant challenges for bankruptcy prediction. In this paper, our proposed model, QTIAH-GNN, aims to tackle the quantity and topology-imbalance issue in the face of bankruptcy prediction. The overall framework could be decoupled into four major components: the topology-agnostic embedding, the multi-hierarchy label-aware neighbor selection, the aggregation and combination module, the imbalance-oriented model training. These modules, unify into a GNN, and learn desirable boundaries for bankruptcies and non-bankruptcies, reducing the overwhelming training bias towards non-bankruptcies. Extensive empirical results have verified the effectiveness of our model in two large-scale real-world datasets, showed stronger robustness and generalization to the imbalance issue of graph-structured data, and provided meaningful model interpretation.

## ACKNOWLEDGMENTS

This research was supported by the National Natural Science Foundation of China (Grant No.91746301; 62276131), the National Key RD Program of China (Grant No.2022YFF0712100), and the Young Elite Scientists Sponsorship Program by CAST.

# REFERENCES


[1] Edward I. Altman. 1968. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of finance* 23, 4 (1968), 589–609.

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems* 32 (2019).

[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[4] Liyi Chen, Zhi Li, Tong Xu, Han Wu, Zhefeng Wang, Nicholas Jing Yuan, and Enhong Chen. 2022. Multi-modal siamese network for entity alignment. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 118–126.

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9268–9277.

[6] Halina Frydman, Edward I Altman, and Duen-Li Kao. 1985. Introducing recursive partitioning for financial classification: the case of financial distress. *Journal of finance* 40, 1 (1985), 269–291.

[7] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference*. 2331–2341.

[8] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. 2009. Graph-based consensus maximization among multiple supervised and unsupervised models. *Advances in Neural Information Processing Systems* 22 (2009).

[9] Niccolo Gordini. 2014. A genetic algorithm approach for SMEs bankruptcy prediction: Empirical evidence from Italy. *Expert systems with applications* 41, 14 (2014), 6433–6445.

[10] Jens Grunert, Lars Norden, and Martin Weber. 2005. The role of non-financial factors in internal credit ratings. *Journal of Banking & Finance* 29, 2 (2005), 509–531.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[12] Liancheng He, Liang Bai, and Jiye Liang. 2023. The Impact of Neighborhood Distribution in Graph Convolutional Networks. https://openreview.net/forum?id=XUqTyU9VlWp

[13] Stephen A Hillegeist, Elizabeth K Keating, Donald P Cram, and Kyle G Lundstedt. 2004. Assessing the probability of bankruptcy. *Review of accounting studies* 9, 1 (2004), 5–34.

[14] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[15] Stewart Jones. 2017. Corporate bankruptcy prediction: a high dimensional analysis. *Review of Accounting Studies* 22, 3 (2017), 1366–1422.

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

[17] Gang Kou, Yong Xu, Yi Peng, Feng Shen, Yang Chen, Kun Chang, and Shaomin Kou. 2021. Bankruptcy prediction for SMEs using transactional data and two-stage multiobjective feature selection. *Decision Support Systems* 140 (2021), 113429.

[18] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems* 34 (2021), 20887–20902.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[20] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference*. 3168–3177.

[21] Yang Liu, Xiang Ao, Qiwei Zhong, Jinghua Feng, Jiayu Tang, and Qing He. 2020. Alike and unlike: Resolving class imbalance problem in financial credit risk assessment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2125–2128.

[22] Frank Ranganai Matenda, Mabutho Sibanda, Eriyoti Chikodza, and Victor Gumbo. 2021. Bankruptcy prediction for private firms in developing economies: a scoping review and guidance for future research. *Management Review Quarterly* (2021), 1–40.

[23] Fadel M Megahed, Ying-Ju Chen, Aly Megahed, Yuya Ong, Naomi Altman, and Martin Krzywinski. 2021. The class imbalance problem. *Nature Methods* 18, 11 (2021), 1270–1277.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[25] Joonhyung Park, Jaeyun Song, and Eunho Yang. 2021. GraphENS: Neighbor-Aware Ego Network Synthesis for Class-Imbalanced Node Classification. In *International Conference on Learning Representations*.

[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine Learning research* 12 (2011), 2825–2830.

[27] Aneta Ptak-Chmielewska. 2019. Predicting micro-enterprise failures using data mining techniques. *Journal of Risk and Financial Management* 12, 1 (2019), 30.

[28] Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. 2021. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1390–1398.

[29] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533–536.

[30] Miriam Seoane Santos, Pedro Henriques Abreu, Nathalie Japkowicz, Alberto Fernández, Carlos Soares, Szymon Wilk, and João Santos. 2022. On the joint-effect of class imbalance and overlap: a critical review. *Artificial Intelligence Review* (2022), 1–69.

[31] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*. 593–607.

[32] Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. 2020. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.

[33] Andrei Shleifer and Robert W Vishny. 1997. A survey of corporate governance. *Journal of finance* 52, 2 (1997), 737–783.

[34] Robert A Sowah, Bernard Kuditchar, Godfrey A Mills, Amevi Acakpovi, Raphael A Twum, Gifty Buah, and Robert Agboyi. 2021. HCBST: An Efficient Hybrid Sampling Technique for Class Imbalance Problems. *ACM Transactions on Knowledge Discovery from Data* 16, 3 (2021), 1–37.

[35] Ellen Tobback, Tony Bellotti, Julie Moeyersoms, Marija Stankova, and David Martens. 2017. Bankruptcy prediction for SMEs using relational data. *Decision Support Systems* 102 (2017), 69–81.

[36] Tony Van Gestel, Bart Baesens, Johan AK Suykens, Dirk Van den Poel, Dirk-Emma Baestaens, and Marleen Willekens. 2006. Bayesian kernel based classification for financial distress detection. *European journal of operational research* 172, 3 (2006), 979–1003.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *International Conference on Learning Representations* (2017).

[38] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, et al. 2019. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315* (2019).

[39] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. *Advances in neural information processing systems* 30 (2017).

[40] Zheng Wang, Xiaojun Ye, Chaokun Wang, Jian Cui, and S Yu Philip. 2020. Network embedding with completely-imbalanced labels. *IEEE Transactions on Knowledge and Data Engineering* 33, 11 (2020), 3634–3647.

[41] Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[42] ZR Yang, Marjorie B Platt, and Harlan D Platt. 1999. Probabilistic neural networks in bankruptcy prediction. *Journal of business research* 44, 2 (1999), 67–74.

[43] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 793–803.

[44] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng. 2021. FRAUDRE: fraud detection dual-resistant to graph inconsistency and imbalance. In *IEEE International Conference on Data Mining*. 867–876.

[45] Tianyu Zhao, Cheng Yang, Yibo Li, Quan Gan, Zhenyi Wang, Fengqi Liang, Huan Zhao, Yingxia Shao, Xiao Wang, and Chuan Shi. 2022. Space4HGNN: A Novel, Modularized and Reproducible Platform to Evaluate Heterogeneous Graph Neural Network. *arXiv preprint arXiv:2202.09177* (2022).

[46] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. 2021. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*. 833–841.

[47] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems* 33 (2020), 7793–7804.

# A APPENDIX

## A.1 Baselines Description

We used eleven baselines which include feature-based methods, imbalance-agnostic GNN methods as well as imbalance-aware GNN-based methods. Detailed descriptions of these three types of baselines are as follows:

- **SVM**: It is a binary linear classification technique in Machine Learning, which separates the classes with largest gap (called optimal margin) between the support vectors.
- **MLP**[29]: It is a typical non-parametric neural network classifier, which maps sets of input data onto a set of outputs in a feedforward manner.
- **GCN** [41]: It performs convolutional operations in the graph Fourier domain through a localized first-order approximation.
- **GraphSAGE** [11]: It is a spatial GNN method that represents nodes inductively based on a fixed number of neighbor nodes' features.
- **GAT** [37]: It performs convolutional operations in the graph spatial domain by leveraging an attention mechanism to learn the relative weights between the neighborhood nodes.
- **HGT** [14]: It designs node- and edge-type dependent parameters to characterize the heterogeneous attention over each edge, empowering HGT to maintain dedicated representations for different types of nodes and edges.
- **RGCN** [31]: It can be regarded as an autoencoder to develop specifically to deal with highly multi-relational heterogeneous graphs.
- **GraphSMOTE** [46]: It is a data augmentation method for graphs, which follows the oversampling strategy and synthesizes new nodes by extending previous SMOTE to graph-structured data.
- **GraphENS** [25]: It is also a data augmentation method for graphs, which synthesizes the whole ego network for minor classes by considering neighbor structure.
- **FRAUDRE** [44]: It is built from a GNN, with three layers consisting of one embedding layer and two graph convolution layers, tackling the graph inconsistency and imbalance issues without modifying the original graph topology.
- **PC-GNN** [20]: It designs a label-balanced sampler to pick nodes and edges to train and a neighbor sampler to choose neighbors with a learnable parameterized distance function.

## A.2 Generalization Performance (RQ5)

To explore the generalization performance of QTIAH-GNN on other applications, we also conducted more experiments on another imbalanced heterogeneous graph dataset. Since Chameleon, Squirrel and Wisconsin are partially imbalanced datasets [12], Amazon and YelpChi contain only one type of node [20], so we adopted a subset of DBLP, a computer science bibliography website, extracted by [8], containing three types of nodes (28,702 Authors (A), 13,214 Terms (T), 20 Conferences (C)) and four types of edges (66823 AAs, 381176 ATs, 44747 CAs, 41257 CTs). The author nodes are divided into four research areas: database, data mining, machine learning and information retrieval. In this dataset, class distributions are relatively balanced, so we used an imitative imbalanced setting: three classes (data mining, machine learning, information retrieval) were merged and selected as the majority class, and the rest of the class, database, was downsampled and selected as the minority class. Ultimately, the quantity-imbalance ratio (QIR) and topology-imbalance ratio (TIR) of this dataset are 0.033 and 0.074, respectively. Also, each author was described by a bag-of-words representation of their paper keywords. The author nodes were divided into training, validation, and testing sets of 60%, 20%, and 20% nodes, respectively.

**Table 4: Performance comparison on DBLP**

| Models | Macro-F1 | BAcc | ROC-AUC |
|---|---|---|---|
| SVM | 0.480±0.000 | 0.587±0.000 | 0.721±0.000 |
| MLP | 0.589±0.058 | 0.559±0.037 | 0.821±0.179 |
| GCN | 0.517±0.024 | 0.513±0.014 | 0.842±0.036 |
| GraphSAGE | 0.714±0.063 | 0.667±0.054 | 0.945±0.004 |
| GAT | 0.701±0.022 | 0.647±0.017 | 0.939±0.007 |
| HGT | 0.728±0.007 | 0.695±0.013 | 0.873±0.009 |
| RGCN | 0.735±0.012 | 0.741±0.008 | 0.890±0.014 |
| GraphSMOTE | 0.599±0.014 | 0.579±0.013 | 0.862±0.010 |
| GraphENS | 0.653±0.070 | 0.625±0.061 | 0.922±0.009 |
| FRAUDRE | 0.564±0.073 | 0.714±0.024 | 0.793±0.037 |
| PC-GNN | 0.637±0.008 | 0.883±0.005 | 0.824±0.012 |
| **QTIAH-GNN** | **0.754±0.017** | **0.894±0.012** | **0.963±0.008** |

Table 4 reports the classification results of QTIAH-GNN and three types of baselines. QTIAH-GNN still outperformed the strongest baseline by 1-5% with respect to Macro-F1, Balanced Accuracy and ROC-AUC scores, which shows the effectiveness and generalization of QTIAH-GNN on other applications along with imbalanced data distribution.