# Enhancing Attribute-driven Fraud Detection with Risk-aware Graph Representation

Sheng Xiang, Guibin Zhang, Dawei Cheng and Ying Zhang

***Abstract*—Credit card fraud is a severe issue that causes significant losses for both cardholders and issuing banks. Existing methods utilize machine learning-based classifiers to identify fraudulent transactions from labeled transaction records. However, labeled data are often scarce compared to the billions of real transactions due to the high cost of annotation, which means that previous methods do not fully utilize the rich features of unlabeled data. What's more, contemporary methods succumb to a fallacy of unawareness of the local risk structure and the inability to capture certain risk patterns. Therefore, we propose the Risk-aware Gated Temporal Attention Network (RGTAN) for fraud detection in this work. Specifically, we first build a temporal transaction graph based on the transaction records, which consists of temporal transactions (nodes) and their interactions (edges). Then we leverage a Gated Temporal Graph Attention (GTGA) Mechanism to propagate messages among the nodes and learn adaptive representations of transactions. We also model the fraud patterns through risk propagation, taking advantage of the relations among transactions. More importantly, we devise a neighbor risk-aware representation learning layer to enhance our method's perception of multi-hop risk structures. We conduct extensive experiments on a real-world credit card transaction dataset and two public fraud detection datasets. The results show that our proposed method, RGTAN, outperforms other state-of-the-art methods on three fraud detection datasets. The risk-aware semi-supervised experiments also demonstrate the excellent performance of our model with only a small fraction of manually labeled data. Moreover, RGTAN has been deployed in a world-leading credit card issuer for credit card fraud detection, and the case study results show the effectiveness of our method in uncovering real-world fraud patterns.**

***Index Terms*—Fraud Detection, Semi-supervised Learning, Graph Neural Network.**

## I. Introduction

THE great losses caused by financial fraud have attracted continuous attention from academia, industry, and regulatory agencies. Ensuring the security of financial transactions is crucial for protecting the privacy and assets of customers, preventing fraud and identity theft, maintaining trust and confidence in the financial system, and complying with the relevant laws and regulations. However, fraudulent behaviors against online payments, such as illegal card swiping, have caused property losses to online payment users [1]. An effective financial fraud detection method can reduce the operating costs of service providers and protect the property of bank users.

Research in the area of financial fraud detection often focuses on credit card fraud, which involves unauthorized transactions typically made through credit or debit cards [2]. A common framework used in commercial systems for detecting such fraud is depicted in Figure 1 [3]. Initially, fraud can be identified through straightforward methods like rule-based systems, which check against blacklists and expenditure limits. However, these systems can be compromised as fraudsters learn to exploit their vulnerabilities. To address these shortcomings, predictive models have been developed to identify fraudulent patterns and generate a risk score for each transaction. This allows domain experts to prioritize their attention on transactions that pose the highest risk.

A considerable amount of research has focused on developing predictive models for identifying fraudulent transactions in the existing literature (e.g., [4]–[6]). These models generally fall into two main groups: (1) *Rule-based methods*, where domain experts craft complex rules to pinpoint suspicious activities, such as the association rule method suggested in [7] for detecting frequent fraudulent patterns; and (2) *Machine learning-based methods*, which rely on analyzing vast amounts of historical data to create static predictive models. For instance, the study in [8] utilized neural networks to extract features and develop supervised classifiers for fraud detection, while [5] explored automated feature engineering using convolutional neural networks (CNN). Additionally, novel approaches utilizing graph machine learning have emerged [9]–[11], where transaction data is represented as graphs, employing sophisticated graph embedding techniques to enhance fraud detection capabilities.

Cutting-edge fraud detection methods [9], [10], [12]–[14] effectively identify transaction patterns, either temporal or graph-based, and considerably enhance credit card fraud detection performance. Nonetheless, these techniques typically encounter one or more of the following significant drawbacks: (1) they overlook unlabeled data, which often contains valuable information about fraud patterns, for example, the 4-vertex-motif [15]; (2) they neglect the relevance of categorical attributes like card and merchant types, which are prevalent in actual operational settings; (3) they demand extensive time

Manuscript received on 22 March. 2023; revised on 28 April. 2024; accepted on day Mon year. Date of publication Day Mon year; date of current version Day Mon Year. This work was supported by the National Key R&D Program of China (Grant no. 2022YFB4501704), the National Natural Science Foundation of China (Grant no. 62102287, 62472317), and the Shanghai Science and Technology Innovation Action Plan Project (Grant no. 22YS1400600 and 22511100700). *(Corresponding author: Dawei Cheng.)*

Sheng Xiang is with the Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia.E-mail: sheng.xiang@uts.edu.au

Guibin Zhang is with the Department of Computer Science and Technology, Tongji University, Shanghai, China.E-mail: bin2003@tongji.edu.cn

Dawei Cheng is with the Department of Computer Science and Technology, Tongji University, Shanghai, China.E-mail: dcheng@tongji.edu.cn

Ying Zhang is with the School of Computer Science and Information Technology and School of Statistics and Mathmatics, Zhejiang Gongshang University, Hangzhou, China. E-mail: ying.zhang@zjgsu.edu.cn

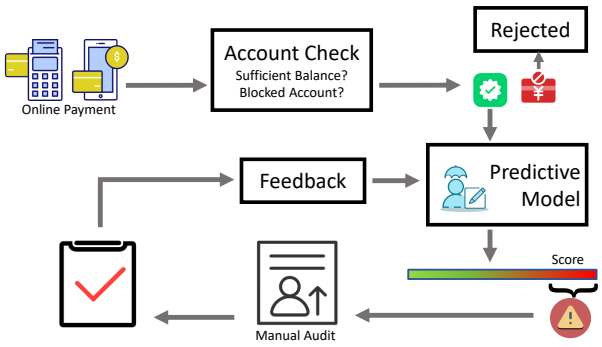


Fig. 1. The framework of credit card fraud detection. The card issuer assesses each transaction with an online predictive model once it has passed account checking.

for feature engineering, particularly concerning temporal and categorical attributes.

In our preliminary work [16], we proposed a gated temporal attention network to address the above challenges and gained competitive results. To understand the relationships in credit card transactions involving temporal data, we use a temporal transaction graph to model time-related patterns. Labeling transactions is labor-intensive and expensive, with fewer than 10% of billions of real-life transactions being labeled, yet many contain undetected fraud patterns. It's essential to utilize features from unlabeled data effectively. To address underutilization, we introduce risk embedding to integrate feature and label information, maximizing the use of risk data. Moreover, given the prevalence and relevance of categorical attributes in practical settings, we have developed an attribute learning layer to preprocess transaction attributes.

However, an increasing number of criminals are organized like enterprises, which can be far-reaching and move quickly from place to place, to conduct conspiracy frauds to covet money from innocent consumers [17]. To fight against human brain-armed criminal behavior, existing graph neural-based methods still face significant challenges in capturing these complicated fraud patterns. Therefore, in this paper, we substantially improved our previous work by proposing a risk-aware graph network to represent the high-order adjacent fraud patterns via a cross-attentional mechanism on multi-hop neighbors. In practice, we aggregate degree and risk information from multi-hop neighbor transactions, which is then processed by a convolutional embedding layer and a structural attention layer, which can extract the local risk information and make our model aware of higher-order risk structures. Our work makes significant contributions in several key areas:

- We construct a temporal transaction graph to represent credit card activities and approach credit card fraud detection as a semi-supervised node classification challenge.
- We introduce an innovative attribute-driven temporal graph neural network tailored for detecting credit card fraud. This includes a gated temporal attention network designed to efficiently process both temporal and attribute-based data.
- Our network incorporates a risk-aware representation learning layer that utilizes degree and risk information from multi-hop neighborhood connections to enhance the local risk structure representations.
- Comprehensive testing on three different datasets confirms our RGTAN's enhanced performance in fraud detection. The semi-supervised testing highlights our method's ability to utilize the vast amount of unlabeled data along with a small portion of labeled data to identify more fraudulent transactions compared to existing baselines. Furthermore, case studies in real-world scenarios validate the effectiveness of our method in recognizing actual fraud patterns.

A preliminary version of this manuscript appeared in [16]. To further capture the high-order adjacent fraud patterns, this journal version proposed a risk-aware gated temporal attention network in Section 4 (new section) to enhance the capacity of the existing graph neural model. In the preliminary submission [16], historical fraud labels and attribute features are concatenated directly as encodings for downstream tasks. While in this extension, we proposed a risk-aware graph network to represent the high-order adjacent fraud patterns via a cross-attentional mechanism on multi-hop neighbors, which could overcome the 1-WL test capacity limitations of the existing graph neural model [18]. We thoroughly evaluated the new proposed substantial improvement approach, compared with the preliminary work and the state-of-the-art baselines in Section 5 (updated section). The experimental results prove the superior performance of our new contribution in detecting complicated inter-connected fraud patterns. In addition, we added empirical studies on real-world application scenarios after system deployment and reported our knowledge discovery in Section 6 (new section).

The rest of the paper is organized as follows. In Section II, we conduct a survey on the previous works regarding credit card fraud detection, graph-based methods, and graph structure learning. In Section III, we present the Graph Temporal Graph Attention (GTGA) mechanism designed for extracting temporal fraud patterns as well as the attribute embedding layer. In Section IV, we detailedly introduce risk embedding and neighbor risk-aware embedding, as well as how they equip the network with awareness of risk structures. Comprehensive experimental results for our proposed method are presented in Section V. Section VI studies two typical risk patterns and validates the risk-aware capacity of our model. Section VII concludes the paper.

## II. Related Works

### A. Credit Card Fraud Detection

Numerous machine learning techniques have been explored in literature to address credit card fraud detection [19]–[21]. Studies have implemented Bayesian Belief Networks (BBN) and Artificial Neural Networks (ANN) on datasets like those from Europay International [22]. Comparisons between neural network models and decision trees have been made in [23], while [24] utilized both decision trees and support vector machines (SVM) on datasets from a national bank. The work

in [5] revealed that convolution models, which extract spatial patterns, can outperform traditional neural networks in terms of accuracy. Additionally, the use of graph-based methods for fraud detection is on the rise [25]–[27]. For instance, CARE-GNN was developed to improve fraud detection on relational graphs [12], and PC-GNN was aimed at addressing imbalances in supervised learning on graphs [10]. AO-GNN applied reinforcement learning to optimize edge pruning for better handling of label imbalances [28]. The H2-FDetector utilized a mix of homophilic and heterophilic connections and incorporated a prototype-based approach to enhance fraudster identification [29]. Works like [3], [14] focused on joint feature learning from spatial and temporal data but were limited to single transaction/cardholder scenarios, which overlooks unlabeled transaction data. Our approach diverges significantly from these methods by adopting a semi-supervised model that leverages an attribute-driven graph neural network to concurrently learn from both labeled and unlabeled data in detecting fraud patterns.

### B. Graph-based Semi-supervised Learning

Recent studies have highlighted the advantages of utilizing unlabeled node attributes in graph neural networks across various predictive tasks, including text classification [30], time series forecasting [31], molecular feature prediction [32], and language processing [33]. For example, graph convolutional networks (GCN) have been used for property prediction in sparsely labeled citation networks [34]. GraphSAGE [35] was developed to create embeddings for new data, while graph attention networks and random walks were applied to social networks to integrate unlabeled and labeled data for message passing [9]. Moreover, SPC-GNN [36] implemented a self-paced labeling enhancement strategy to boost performance in semi-supervised node classification tasks. Nonetheless, these approaches often encounter challenges such as scaling to graphs with millions of nodes, propagating and learning categorical attribute embeddings, particularly risk-related ones, and leveraging graph structural data effectively. In contrast, our method tackles fraud detection by using a message-passing model that synergistically handles categorical attributes and structural risk data. We introduce an attribute-driven, semi-supervised graph neural network approach to enhance the detection of fraud patterns and significantly improve the precision of credit card fraud detection efforts.

### C. Graph Structure Learning

Graph structure learning (GSL) is a research area that aims to learn more effective graph structures and representations for downstream tasks [37] and involves inferring optimal graph structures and representations from data that are generated by or correlated with the graph [38]. Most approaches in this realm are inspired by [39], which employs *persistent homology* to calculate topological features(e.g., cycle, path, connected components). [40] proposes a new kernel and an optimization framework to learn the topological summaries of data and achieve competitive results in graph classification. [41] augment the subtree features of the Weisfeiler–Lehman graph kernel with topological information so as to improve the performance of graph-level classification. [42] enables deep neural networks to capture topological structure via inputting features obtained from persistent homology. PEGN in [43] designs a persistence layer to enrich graph representations, and [44] further makes graph neural networks topological-aware via a topological graph layer (TOGL). A subgraph isomorphism counting layer is raised in GSN to capture higher-order structural information [18]. It is worth paying attention to such structural information in the realm of fraud detection. [15] utilized a HGAR attention mechanism to select risk pattern candidates(i.e. 4-vertex-motif structures). However, the aforementioned approaches are not readily applicable to the domain of fraud detection. Purely structure-aware methods fail to leverage label information, thus hindering their ability to detect fraud patterns that are intimately associated with fraud labels. In this paper, we innovatively introduce the idea of graph structural learning into fraud detection. Specifically, we devise a risk-aware learning layer, which adopts the idea of 'structure-ware', to model high-order adjacent fraud patterns which are proven to be conducive to improving the expressiveness and performance of graph neural networks.

## III. Gated Temporal Graph Attention

In this section, we first introduce the framework of our proposed Gated Temporal Graph Attention (GTGA) mechanism. After that, we present the process of feature engineering and the gated temporal attention networks. The optimization strategy and learning objective are defined at the end.

### A. Model Architecture

The architecture of our proposed model is depicted in Figure 2. This section explains components such as *(b)Attribute Embedding* and *(d) Gated Temporal Graph Attention*. Initially, the raw attributes from transaction records undergo a process of attribute embedding using a look-up and feature learning layer, which includes a multi-layer perceptron (MLP) for feature aggregation. Attributes related to the card encompass aspects like card type, cardholder category, credit limit, and available balance. Transaction-related attributes cover elements such as channel ID, currency ID, and transaction volume, whereas merchant-related attributes include merchant category, terminal type, location, industry sector, and fee percentage.

Subsequently, a gated temporal attention network is employed to assimilate and prioritize the significance of past transaction embeddings. This is followed by the application of a two-layer MLP that calculates the likelihood of fraud based on these learned embeddings. The entire model is designed for end-to-end optimization using the standard stochastic gradient descent algorithm.

### B. Attribute Embedding and Feature Learning

This subsection details our approach to preprocessing transaction attributes. Each transaction record $\mathbf{r} = (r_1, r_2, \cdots, r_N)$ includes attributes for cards $f_c^i$, transactions $f_r^i$, and merchants

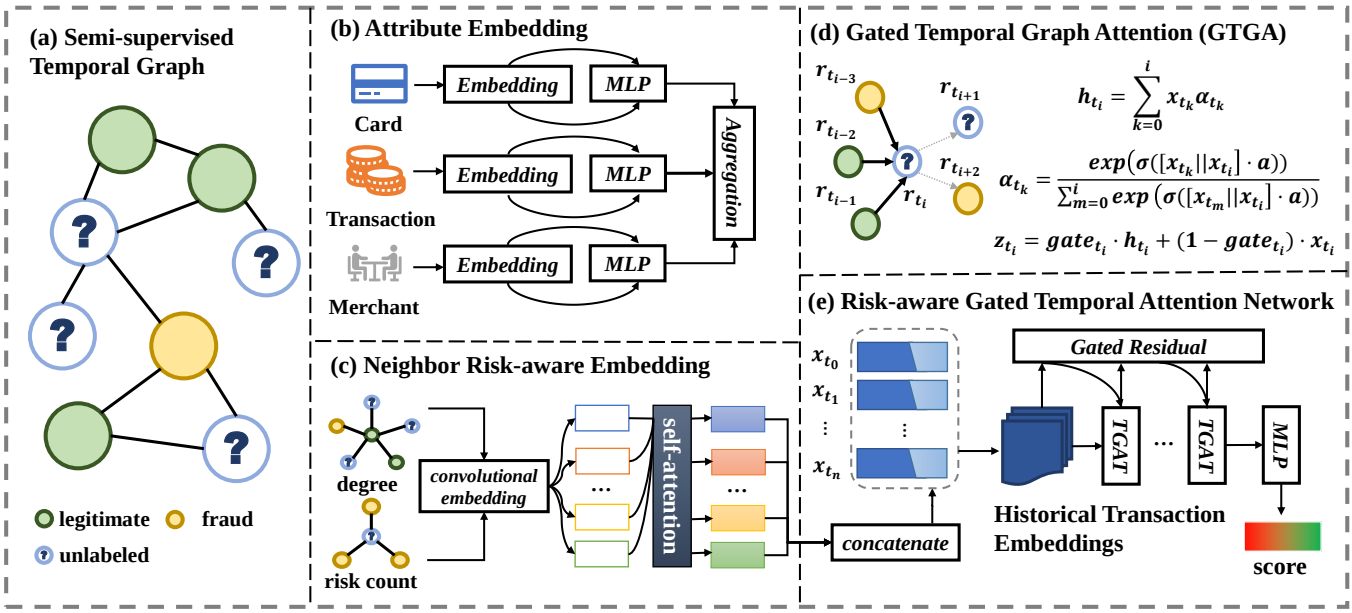


Fig. 2. The illustration of the proposed graph neural network model. Raw transaction records are processed by attributed embedding and attribute aggregation to combine each semantic representation. Degree and risk information is collected from the multi-hop neighborhood and concatenated into node features after convolutional embedding and self-attention operations. Afterward, the learned representations are fed into a risk-aware gated temporal attention network (RGTAN) for representation learning. The transaction representation is then fed into a multi-layer perceptron for fraud detection. Attentional weights are jointly optimized in an end-to-end mechanism with graph neural networks and fraud detection networks.

$f_m^i$ represented as $r_i = f_c^i, f_r^i, f_m^i$. Unlike previous methods referenced in [3], [14], we retain all cards and merchants, regardless of their number of authorized transactions, to preserve all potential fraud indicators in our dataset. We also tested data with different label ratios, which can be found in Section V-C. We then convert these attributes into a numerical tensor $\mathbf{X}num \in \mathbb{R}^{N\times d}$, with $N$ being the transaction count and $d$ is the feature dimensions. Additionally, we separately process card, transaction, and merchant categories into $\mathbf{X}cat \in \mathbb{R}^{N\times d}$ using attribute embedding layers, calculated as below:

$$\begin{aligned} e_{attr} =& \text{onehot}(f_{attr}) \odot \mathbf{E}_{attr}, \\ x_{cat,i} =& \text{MLP}_i(\sum_{\forall j \in table_i} e_j), i \in \{card, trans, mchnt\}, \end{aligned} \quad (1)$$

where each column $j$ in table $i$ is represented as $j \in table_i$. Here, $e_{attr} \in \mathbb{R}^{1\times d}$ is the embedding for an attribute $attr$, onehot$(\cdot)$ is used for one-hot encoding, $f_{attr}$ is a single attribute from a transaction, and $\mathbf{E}_{attr} \in \mathbb{R}^{m\times d}$ is the embedding matrix for the attribute $attr$, where $m$ represents the maximum possible variations of $attr$.

Following the creation of the attribute embeddings for the card, transaction, and merchant categories, we use add-pooling to combine these embeddings into a single categorical embedding per transaction with $x_{cat}^{(u)} = \sum_i x_{cat,i}^{(u)}$, where $i \in card, trans, mchnt$ and $x_{cat}^{(u)} \in \mathbb{R}^{1\times d}$ represents the category embedding vector for the $u$-th transaction record. Our approach effectively reduces the space complexity from $O(Nac)$ to $O(Na + acd)$, with $N$ indicating the transaction count, $c$ the category count, and $a$ the average number of unique values per category. This reduction is particularly beneficial in large-scale applications with numerous categorical attributes. Moreover, the heterogeneous nature of these attributes allows our feature learning layer to model and map them into a unified spatial dimension, enhancing our attribute-driven graph learning framework.

### C. Gated Temporal Graph Attention Mechanism

To discern temporal fraud patterns, we construct a temporal transaction graph and utilize it to aggregate messages that update each transaction's embedding. Specifically, we generate directed temporal edges, positioning prior transactions as sources and subsequent ones as targets, as depicted in Figure 2(c). Message aggregation is then conducted via Temporal Graph Attention, and the quantity of temporal edges for each node is a tunable hyper-parameter, details of which will be explored in the experimental section.

**Temporal Graph Attention.** Starting with attribute embedding and feature engineering, we engage a series of transaction embeddings $\mathbf{X} = x_{t_0}, x_{t_1}, ...x_{t_n}$ to deduce each transaction's temporal embedding. Initially, we integrate categorical and numerical attributes for the RGTAN input with $x_{t_i} = x_{num}^{(t_i)} + x_{cat}^{(t_i)}$. Setting $\mathbf{H}_0 = \mathbf{X}$ as the initial embedding matrix at the first GTGA layer, we apply multi-head attention to evaluate the significance of each neighboring node and update embeddings accordingly, as outlined in the following equation:

$$\mathbf{H} = \text{Concat}(\text{Head}1, ..., \text{Head}h_{att})\mathbf{W}o, \quad (2)$$

where $hatt$ represents the number of attention heads, $\mathbf{W}o \in \mathbb{R}^{d\times d}$ are learnable parameters, and $\mathbf{H}$ embodies the updated embeddings with $\mathbf{H} = ht_0, h_{t_1}, ..., h_{t_n}$. Each head in the attention mechanism operates as follows:

$$\begin{aligned} \text{Head} &= \sum_{x_i \in \mathcal{X}} \sigma( \sum_{x_t \in \mathcal{N}(x_i)} \alpha_{x_t,x_i} x_t), \\ \alpha_{x_t,x_i} &= \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[x_t||x_i]))}{\sum_{x_j \in \mathcal{N}(x_t)} \exp(\text{LeakyReLU}(\mathbf{a}^T[x_t||x_j]))}, \end{aligned} \tag{3}$$

where $\mathcal{N}(x_i)$ indicates the temporal neighbors of the transaction $i$, $\alpha_{x_t,x_i}$ quantifies the importance of each temporal edge in the attention process, and $\mathbf{a} \in \mathbb{R}^{2d}$ is the weight vector for each head. We control the number of neighbors $|\mathcal{N}(x_t)|$ using a neighbor sampling and truncation strategy to manage space consumption effectively, especially during periods of high-frequency transactions. It's also ensured that only past transactions from the same cardholder are considered to maintain the integrity of the temporal pattern modeling.

**Attribute-driven Gated Residual.** To enhance our method's interpretability and effectiveness further, after aggregating embeddings, we employ them along with raw attributes to ascertain the significance of the combined embeddings post-temporal graph attention. This process is described by:

$$\begin{aligned} \text{gate}t_i &= \sigma([xcat, t_i||x_{num,t_i}||h_{t_i}]\beta_{t_i}), \\ z_{t_i} &= \text{gate}t_i \cdot ht_i + (1 - \text{gate}t_i) \cdot xt_i, \end{aligned} \tag{4}$$

where $\text{gate}t_i$ represents the gating variable for transaction $t_i$, $\sigma$ is the sigmoid function, $\beta t_i \in \mathbb{R}^{3d \times 1}$ denotes the gating vector, and $z_{t_i}$ is the output vector of each GTGA layer, used as input for subsequent layers. If additional GTGA layers are stacked, the output from the $k$-th gating mechanism serves as input to the $k+1$-th GTGA layer. In this stacking framework, the bottom-up $k$-th GTGA layer weighs the importance of the $k$-th order neighbor transactions. In addition, the bottom-up $k$-th attribute-driven gated residual mechanism weighs the importance of each transaction's $k$-th order neighbor transaction embedding and its own embedding. Algorithm 1 shows the detailed computation process of message passing in one GTGA layer.

### D. Fraud Risk Prediction

Upon acquiring the collective embeddings of transaction data, we apply a dual-layer MLP to determine the potential fraud risk. The prediction model is described by the following equation:

$$\hat{\mathbf{y}} = \sigma(\text{PReLU}(\mathbf{H}\mathbf{W}_0 + \mathbf{b}_0)\mathbf{W}_1 + \mathbf{b}_1), \tag{5}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{N \times 1}$ represents the predicted risk outcomes for all transactions, with $\mathbf{W}$ and $\mathbf{b}$ as the adjustable parameters of the MLP. The loss function $\mathcal{L}$ is then computed using binary cross-entropy, detailed as follows:

$$\begin{aligned} \mathcal{L} = -\frac{1}{N} \sum_{i=0}^{N} [\mathbf{y}_i \cdot \log p(\hat{\mathbf{y}}_i|\mathbf{X}, \mathbf{A}) + \\ (1 - \mathbf{y}_i) \cdot \log(1 - p(\hat{\mathbf{y}}_i|\mathbf{X}, \mathbf{A}))], \end{aligned} \tag{6}$$

where $\mathbf{y}$ indicates the actual labels of the transactions. This network model is optimized using conventional stochastic gradient descent (SGD)-based methods.

## IV. Risk-aware Gated Temporal Attention Network

In this section, we present the details of risk propagation and neighbor risk-aware embedding. Practically, manually annotated labels are adopted as categorical features and integrated into the original features after embedding transformations. Risk information from multi-hop neighborhoods is encoded into features by adding the degree and risk count from multi-hop neighbors. Both are jointly passed and updated through GTGA, which we call *risk-aware message passing*. To avoid possible label leakage, we leverage a masked fraud detection strategy.

**Algorithm 1:** Steps of computation in a GTGA layer

**Input :** $G(V, E)$: the given transaction graph
$\mathbf{H}_k$: the embedding matrix from the $k^{\text{th}}$ layer
$x_{cat}$: categorical features
$x_{num}$: numerical features

**Output:** $\mathbf{H}_{k+1}$: updated embedding feature matrix as input of $(k+1)^{\text{th}}$ layer

**for** $i \leftarrow 1$ **to** $N$ **do**
  **for** $h \leftarrow 1$ **to** $h_{att}$ **do**
    $\text{Head}_h^i \leftarrow \sigma(\sum_{x_t \in \mathcal{N}(x_i)} \alpha_{x_t,x_i} x_t)$;
    $\alpha \leftarrow \frac{\exp(\sigma(\mathbf{a}^T[x_t||x_i]))}{\sum_{x_j \in \mathcal{N}(x_i)} \exp(\sigma(\mathbf{a}^T[x_t||x_j]))}$;
  $h_i \leftarrow \text{Concat}(\text{Head}_1^i, ..., \text{Head}_{h_{att}}^i)\mathbf{W}_o$;
  $v_i \leftarrow \text{Concat}(x_{cat,i}, x_{num,i}, h_i)$;
  $\text{gate}_i \leftarrow \sigma(v_i\beta_i)$;
  $z_i \leftarrow \text{gate}_i \cdot h_i + (1 - \text{gate}_i) \cdot \mathbf{H}_{k,i}$;
$\mathbf{H}_{k+1} \leftarrow [z_1||...||z_N]$

### A. Risk Propagation Representation

The manual-annotated labels are expensive in real-world fraud detection practice. With labeled risk information, we can effectively model more fraud patterns, such as risk propagation. Drawing inspiration from the integration of label propagation with feature propagation [45], we introduce a concept we term *risk embedding*. Specifically, we utilize the manually annotated label of each transaction as a risk feature, categorizing unlabeled data as 'unlabeled' and the remainder as 'fraud' or 'legitimate'. This label is incorporated as a categorical attribute within our transaction data. Previous solutions have not employed this attribute due to the potential risk of label leakage, which we address with specific strategies discussed later. We then embed these partially observed risk attributes into the same dimensional space as other node features, resulting in risk embedding vectors for labeled nodes and zero vectors for unlabeled ones. These are then combined with other node features for input with $x_{t_i} = x_{num}^{(t_i)} + x_{cat}^{(t_i)} + \tilde{y}^{(t_1)}\mathbf{W}_r$, where $\mathbf{W}_r$ represents the adjustable parameters for risk embedding. Research by [45] has demonstrated that by aligning partially-labeled $\hat{\mathbf{Y}}$ with node features $\mathbf{X}$ in the same space and combining them, a single graph neural network can effectively facilitate both attribute and label propagation. Thus, our fraud detection framework effectively models both

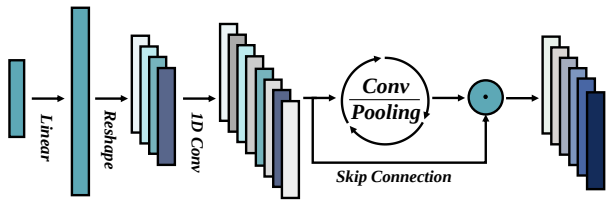


Fig. 3. Convolutional Embedding. The final output of the convolutional embedding layers is represented as $\mathbb{R}^{N\times r\times d}$, with each output channel corresponding to a previous neighbor risk feature.

temporal fraud patterns and risk propagation by incorporating the transaction label as a categorical attribute of transactions.

### *B. Neighbor Risk-aware Attentional Embedding*

Fraud transactions often fabricate noisy information to make them difficult to be recognized and therefore result in redundant link information around fraudulent nodes, which can to some extent weaken the power of neighborhood aggregation [10]. For example, fraudulent transactions might be deliberately connected to numerous legitimate transactions so that spammers could hide among legitimate users. Such risk patterns are relatively difficult to be captured by vanilla graph-based methods. In this paper, we leverage the degree information of multi-hop neighbor nodes and count of risky neighbors as node features, which can inherently reflect the local risk structure, to alleviate the problems described above. Risk-aware representations are further learned by a convolutional embedding layer and structural attention layer.

**Convolutional Embedding.** For given transaction records $\mathbf{r} = (r_1, r_2, \cdots, r_N)$, each record $r_i$ have neighbor degree feature $f^i_{degree,k}$ and risk neighbor count feature $f^i_{risk,k}$, where $k$ denotes $k$-hop neighborhood. Besides, to avoid future label leakage, we merely gather information from past transaction records belonging to the same cardholder of $r_i$. The above two features can be formulated as follows:

$$\begin{aligned} f^i_{degree,k} &= \sum_{u\in\mathcal{N}^i_k} \text{Degree}(u), \\ f^i_{risk,k} &= \sum_{u\in\mathcal{N}^i_k, \mathbf{y}_u=1} \mathbf{y}_u \end{aligned} \tag{7}$$

where $\mathcal{N}^i_k$ denotes the filtered $k$-hop neighbor nodes of $r_i$, $\text{Degree}(u)$ counts the in degree of $r_u$ and $\mathbf{y} \in \mathbb{R}^N$ denotes labels of all transactions. After obtaining the neighbor risk information for each transaction record, we stack these numerical features and construct neighborhood risk-aware representation through a convolutional embedding layer. [46] has proven such embedding numerical features can be conducive to many backbone structures. We construct the neighbor risk features into tensor format $\mathbf{X}_{nei} \in \mathbb{R}^{N\times r}$ where $r$ denotes the number of neighbor risk features, which later will be transformed into the risk-aware embedding matrix $\mathbf{X}_{rsk} \in \mathbb{R}^{N\times r\times d}$ as Figure 3 shows.

Specifically, $\mathbf{X}_{nei} \in \mathbb{R}^{N\times r}$ is expanded via a linear layer and reshaped into $c$ channels, which we denote as $\mathbf{X}_{conv} \in \mathbb{R}^{N\times c\times L}$ with $L$ referring to the length of each channel. Thereafter, multiple 1D convolutional layers are leveraged to extract the risk information representations. We denote each channel of record $i$ at layer $l$ as $\mathbf{X}^{i,j,l}_{conv}(1 <= j <= r)$, and each channel vector is convolved by a filter vector $\mathbf{w}$ of length $H$ with zero-padding after the batch norm operation.

$$\mathbf{X}^{i,j,l+1}_{conv}[m] = \sigma(\sum_{h=0}^{H-1} \mathbf{w}[h]\mathbf{X}^{i,j,l}_{conv}[m-h+P] + b), \quad m \in [0, L-1] \tag{8}$$

where $b$ is the bias term and $P$ is the padding size that satisfies $P = \frac{H-1}{2}$. At the final convolutional layer, the number of channels and the vector length are adjusted to $r$ and $d$, via an output convolutional layer and an adaptive pooling layer respectively, with each channel corresponding to a previous neighbor risk feature. The skip connection strategy is leveraged to alleviate the problem of gradient vanishing and exploit spatial correlations and translations between risk features. $\mathbf{X}^L_{conv} \in \mathbb{R}^{N\times r\times d}$ are adopted as the risk-aware embedding.

**Algorithm 2:** Neighbor risk-aware embedding

**Input :** $G(V, E)$: the given transaction graph
**Output:** $\mathbf{X}_{rsk}$: neighbor risk-aware embeddings

**for** $i \leftarrow 1$ **to** $N$ **do**
  **for** $k \leftarrow 1$ **to** $K$ **do**
    $f^i_{de,k} = \sum_{u\in\mathcal{N}^i_k} \text{Degree}(u)$,
    $f^i_{ri,k} = \sum_{u\in\mathcal{N}^i_k} \mathbf{y}_u$ **if** $\mathbf{y}_u = 1$; **then**
  $f^i_{nei} \leftarrow [f^i_{de,1}, ..., f^i_{de,K}, f^i_{ri,1}, ..., f^i_{ri,K}]$
  $h^i_0 \leftarrow \texttt{Reshape}(f^i_{nei}\mathbf{W}_0$,(c,L))
  **for** $l \leftarrow 1$ **to** $L$ **do**
    **for** $c \leftarrow 1$ **to** $C$ **do**
      **for** $m \leftarrow 1$ **to** $L$ **do**
        $h^i_{l,c}[m] \leftarrow \sigma^{H-1}_{s=0}(\mathbf{w}^{l,c}[s]h^i_{l-1,c}[m-s+P]+b) + h^i_{l-1,c}$
  $\{f_1, f_2, ...f_r\} \leftarrow \{h_{L,1}, h_{L,2}, ...h_{L,r}\}$
  **for** $h \leftarrow 1$ **to** $num$ **do**
    $\text{Head}^i_h \leftarrow \sigma(\sum_{t\in[1,r]} \alpha_{f_t,f_i} x_t)$;
    $\alpha_{f_t,f_i} \leftarrow \frac{\exp(\sigma(\mathbf{a}^T[f_t||f_i]))}{\sum_{j\in[1,r]}\exp(\sigma(\mathbf{a}^T[f_j||f_i]))}$;
  $\mathbf{H}^i \leftarrow \text{Concat}(\text{Head}^i_1, ..., \text{Head}^i_{num})\mathbf{W}_1$;
$\mathbf{X}_{rsk} \leftarrow \{\mathbf{H}^1, \mathbf{H}^2, ..., \mathbf{H}^N\}$

**Structural Attention.** Afterwards, we introduce a structure-aware self-attention module to separately calculate the importance of each neighbor risk-aware feature and update the embeddings, which can further exploit the neighborhood risk information and help learn fraud patterns. For a certain transaction record $r_i$, we denote its embedding matrix as $\mathbf{X}^i \in \mathrm{R}^{r\times d} = \{f_1, f_2, ..., f_r\}$. The calculation process can be formulated as follows:

$$\mathbf{H}^i = \text{Concat}(\text{Head}_1, ..., \text{Head}_{num})\mathbf{W}_o, \tag{9}$$

where $num$ denotes the number of attention heads, $\mathbf{W}_o \in \mathrm{R}^{d\times d}$ denotes learnable parameters, $\mathbf{H}$ denotes the updated risk-aware embeddings and each attention head is formulated as follows:

$$\begin{aligned} \text{Head} &= \sum_{i\in[1,r]} \sigma(\sum_{m\in[1,r]} \alpha_{f_m,f_i} f_i), \\ \alpha_{f_m,f_i} &= \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[f_m||f_i]))}{\sum_{n\in[1,r]}\exp(\text{LeakyReLU}(\mathbf{a}^T[x_t||x_n]))} \end{aligned} \tag{10}$$

where $\alpha_{f_m,f_i}$ denotes the importance between feature $f_m$ and $f_i$ in each attention head, and $\mathbf{a} \in \mathrm{R}^{2d}$ denots the weight vector of each head. The output embedding matrix $\mathbf{H}^i \in \mathbb{R}^{r\times d}$ is then reduced in the last dimension and then concatenated to the input node features, which can be formulated as follows:

$$\begin{aligned} x_{t_i} = \text{CONCAT}(&x_{num}^{(t_i)} + x_{cat}^{(t_i)} + \tilde{y}^{(t_1)}\mathbf{W}_r, \\ &\text{SQUEEZE}(H^i\mathbf{W}_{proj})) \end{aligned} \tag{11}$$

where $\mathbf{W}_{proj} \in \mathbb{R}^{d\times 1}$ and SQUEEZE denotes squeezing matrix in the last dimension. Therefore, our fraud detection model is enhanced by the integration of neighbor risk-aware embeddings and is capable of better modeling temporal fraud patterns. Algorithm 2 demonstrates the detailed computation of risk-aware embeddings.

### C. Loss Function and Model Optimization

Unlike traditional credit card fraud detection solutions, our novel RGTAN model employs a semi-supervised approach by propagating transaction attributes, risk embeddings, and neighbor risk-aware features across both labeled and unlabeled transactions to train our model. Employing an unmasked objective in our fraud detection model would lead to label leakage during training, causing the model to rely solely on observed labels and ignore complex hidden fraud patterns that are crucial for predicting future fraudulent activities.

To mitigate this, we adopt a training strategy that learns from the risk information of neighboring transactions rather than direct labels of individual transactions. Specifically, we employ a masked training approach where each training step involves randomly sampling a set of center nodes along with their corresponding neighbors. We transform $\hat{\mathbf{Y}}$ into $\tilde{\mathbf{Y}}$ by setting the risk embeddings of the center nodes to zero and maintaining the others. Additionally, to better capture fraud patterns from neighboring transactions, we integrate risk information from these neighbors directly into node features. To prevent potential label leakage, we apply a multi-hop masking strategy, setting the risk feature from $k$-hop neighbors ($f_{risk}^k$) of center nodes to zero during training. The objective function is structured as follows:

$$\begin{aligned} \mathcal{L} = -\frac{1}{|V|}\sum_{i=0}^{|V|}[&\mathbf{y}_i \cdot \log p(\hat{\mathbf{y}}_i|\mathbf{X},\tilde{\mathbf{Y}},\mathbf{A}) + \\ &(1-\mathbf{y}_i)\cdot\log(1-p(\hat{\mathbf{y}}_i|\mathbf{X},\tilde{\mathbf{Y}},\mathbf{A}))], \end{aligned} \tag{12}$$

where $|V|$ indicates the number of center nodes with masked labels. This strategy ensures our training avoids self-loop risk information leakage. During inference, all observed labels $\hat{\mathbf{Y}}$ are used as input categorical attributes to predict transaction risks outside the training dataset. Ultimately, our model's optimization goal is to capture fraud patterns by leveraging attribute information from neighboring transaction nodes, including risk and neighbor risk-aware embeddings, alongside the attribute information of the nodes themselves, excluding direct risk details.

TABLE I
STATISTICS OF THE THREE FRAUD DETECTION DATASETS.

| Dataset | YelpChi | Amazon | FFSD |
|---|---|---|---|
| #Node | 45,954 | 11,948 | 1,820,840 |
| #Edge | 7,739,912 | 8,808,728 | 31,619,440 |
| #Fraud | 6,677 | 821 | 33,858 |
| #Legitimate | 39,277 | 11,127 | 141,861 |
| #Unlabeled | - | - | 1,645,121 |

## V. EXPERIMENTS

This section outlines the datasets employed in our study, compares the performance of our RGTAN against other leading graph-based fraud detection models across two supervised and one semi-supervised datasets, discusses the results of ablation studies on two variants of our model, and highlights findings from real-world case studies where our approach notably excels in identifying typical fraud patterns.

### A. Experiment Settings

*1) Datasets:* To our knowledge, there are no publicly available semi-supervised datasets specifically for credit card fraud detection. Thus, we have compiled a dataset from a leading global financial institution, referred to as the Financial Fraud Semi-supervised Dataset (**FFSD**), which includes real-world credit card transactions over a ten-month period. Labels for these transactions were derived from consumer reports and validations by financial experts. Transactions confirmed as fraudulent are labeled as 1, and all others as 0.

Furthermore, we conducted experiments on two publicly accessible supervised fraud detection datasets. The **YelpChi** dataset [47] comprises hotel and restaurant reviews from Yelp, structured into a graph where nodes represent reviews equipped with 32-dimensional features, and edges represent relationships among these reviews. The **Amazon** dataset [48] consists of musical instrument reviews, where nodes are user reviews featuring 25-dimensional attributes, and edges delineate the interactions among these reviews. Essential statistics for these datasets are presented in Table I.

*2) Compared Methods.:* To demonstrate the efficacy of our proposed GTAN, we benchmark against several well-established methods:

- *GEM.* This heterogeneous GNN model is adapted from [49] with a learning rate set to 0.1 and neighbor hops limited to 5.
- *FdGars.* A graph convolutional network for fraudster detection from [50], with adjustments including a learning rate of 0.01 and a hidden dimension of 256.

TABLE II
FRAUD DETECTION PERFORMANCE OF VARIOUS METHODS ON THREE DATASETS: YELPCHI, AMAZON, AND FFSD. THE EVALUATION METRICS USED ARE THE AREA UNDER THE ROC CURVE (AUC), MACRO AVERAGE OF F1 SCORE (F1-MACRO), AND AVERAGE PRECISION (AP). AMONG THE METHODS, RGTAN STANDS OUT WITH THE HIGHEST AUC AND AP SCORES ON ALL THREE DATASETS. RGTAN'S EXCELLENT PERFORMANCE ON YELPCHI DATASET WITH AUC SCORE OF 0.9498, F1-MACRO SCORE OF 0.8492*, AND AP SCORE OF 0.8241 IS PARTICULARLY NOTEWORTHY.

| Dataset | YelpChi | | | Amazon | | | FFSD | | |
|---|---|---|---|---|---|---|---|---|---|
| | AUC | F1 | AP | AUC | F1 | AP | AUC | F1 | AP |
| GEM | 0.5270 | 0.1060 | 0.1807 | 0.5261 | 0.0941 | 0.1159 | 0.5383 | 0.1490 | 0.1889 |
| Player2Vec | 0.7003 | 0.4121 | 0.2473 | 0.6185 | 0.2451 | 0.1291 | 0.5278 | 0.2147 | 0.2041 |
| FdGars | 0.7332 | 0.4420 | 0.2709 | 0.6556 | 0.2713 | 0.1438 | 0.6965 | 0.4089 | 0.2449 |
| Semi-GNN | 0.5161 | 0.1023 | 0.1811 | 0.7063 | 0.5492 | 0.2254 | 0.5473 | 0.4485 | 0.2758 |
| GraphSAGE | 0.5364 | 0.4508 | 0.1712 | 0.7502 | 0.5795 | 0.2624 | 0.6527 | 0.5370 | 0.3844 |
| GraphConsis | 0.7060 | 0.6041 | 0.3331 | 0.8782 | 0.7819 | 0.7336 | 0.6579 | 0.5466 | 0.3876 |
| CARE-GNN | 0.7934 | 0.6493 | 0.4268 | 0.9115 | 0.8531 | 0.8219 | 0.6623 | 0.5771 | 0.4060 |
| PC-GNN | 0.8174 | 0.6682 | 0.4810 | 0.9581 | 0.9153 | 0.8549 | 0.6795 | 0.6077 | 0.4487 |
| GTAN | 0.9241 | 0.7988 | 0.7513 | 0.9630 | **0.9213** | 0.8838 | 0.7616 | 0.6764 | 0.5767 |
| RGTAN | **0.9498*** | **0.8492*** | **0.8241*** | **0.9705*** | 0.9198 | **0.8925*** | **0.7680*** | **0.6800*** | **0.5786*** |

- *Player2Vec.* This attributed heterogeneous information network model from [51] follows the parameter settings of the FdGars.
- *Semi-GNN.* A semi-supervised attentive network focusing on financial fraud, sourced from [9], with a learning rate of 0.001.
- *GraphSAGE.* An inductive learning approach from [35], where the embedding dimension is set at 128.
- *GraphConsis.* This model addresses inconsistency issues in GNNs and is based on [13] using its default settings.
- *CARE-GNN.* Focused on relational graph fraud detection, this model from [12] uses its standard parameters.
- *PC-GNN.* Designed to tackle class imbalance, this model from [10] also follows default settings.
- GTAN. Our attribute-driven semi-supervised attention network based on the framework in [16], using standard parameters.
- **RGTAN.** Our risk-aware gated temporal attention network includes three variants (RGTAN-A, RGTAN-R, RGTAN-N) that test different aspects of our model by excluding the temporal graph attention, risk embedding, and risk-aware representations, respectively. The model incorporates 2-hop neighborhood risk-aware embeddings, with settings including a batch size of 128, learning rate of 0.002, input dropout of 0.2, four attention heads, hidden dimension $d$ of 256, and uses the Adam optimizer over 25 epochs with early stopping.

*3) Evaluation Metrics:* Our performance metrics for credit card and opinion fraud detection involve the area under the ROC curve (AUC), macro average of the F1 score (F1-macro), and average precision (AP). These metrics are computed as follows:

The calculations for these metrics start by determining the True Positives $N_{TP}$ (correctly identified positive instances), False Positives $N_{FP}$ (incorrectly identified positive instances), and False Negatives $N_{FN}$ (missed negative instances). The F1-macro score and AP are then computed using $F1macro = \frac{1}{l}\sum i = 1^{l}\frac{2\times P_i\times R_i}{P_i+R_i}$ and $AP = \sum_{i=1}^{l}(R_i - R_{i-1})P_i$, where $P_i = N_{TP}/(N_{TP} + N_{FP})$ and $R_i = N_{TP}/(N_{TP} + N_{FN})$. AUC, representing the area under the ROC curve, is also reported for our experiments.

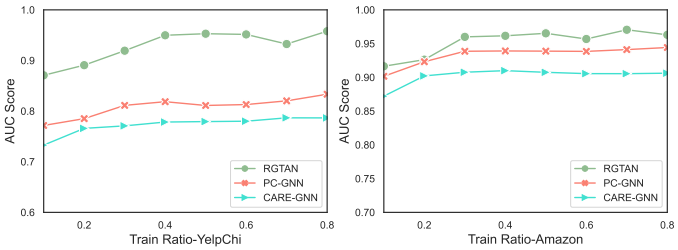


Fig. 4. The result of semi-supervised experiments with different ratios of labeled training data. The left is the performance of CRAE-GNN, PC-GNN and RGTAN on YelpChi dataset, with the training ratio ranging from 0.1 to 0.8, which generally displays an upward trend with more data used for training. The right is the compared performance on Amazon dataset and roughly exhibits the same trend.

### B. Fraud Detection Experiment

In the YelpChi and Amazon datasets, the ratio of training to testing data was established at 2:3. For the FFSD dataset, transactions from the initial seven months serve as the training set, while transactions from the subsequent three months (August, September, and October of 2021) are analyzed for fraud detection. Each method undergoes ten trials, and the mean outcomes are summarized in Table II. The statistical significance of enhancements is indicated by $^*$, validated through a paired t-test where the $p$-value is below 0.01.

Table II first outlines the performances of traditional graph-based models such as GEM, Player2Vec, FdGars, Semi-GNN, and GraphSASE in its initial five rows. The analysis reveals GEM's underwhelming performance, underscoring its limitations in addressing complex fraud scenarios due to its shallow architecture. Both Player2Vec and FdGars exhibited better results, likely owing to their increased model capacities. Semi-GNN and GraphSAGE, performing comparably and outstripping the earlier three, underscore the benefits of employing deep graph-based learning models for fraud detection.

Further down, the inclusion of transaction graphs within the learning framework allows PC-GNN to outperform the previous models mentioned, yielding more robust results. The

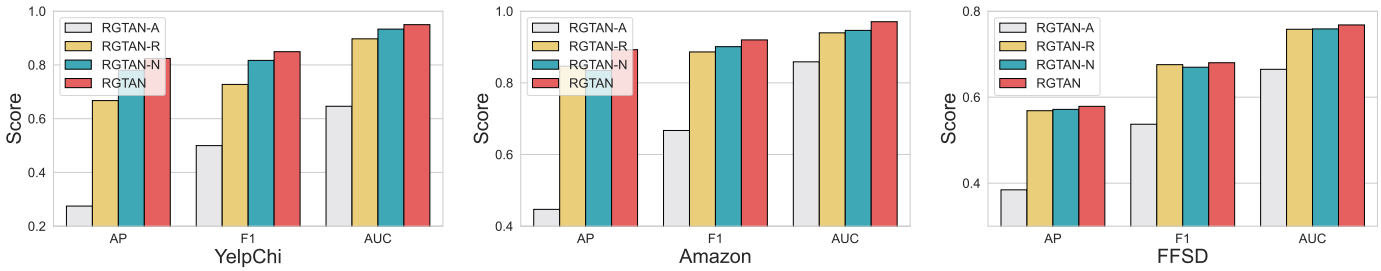


Fig. 5. The ablation study results on three datasets. Gray bars represent the RGTAN-A variant, yellow bars represent the RGTAN-R variant, blue bars represent the RGTAN-N model and red bars represent the RGTAN model. The removal of GTGA component across three datasets in RGTAN-A lead to a dramatic performance drop. Discarding either risk embedding or neighbor risk-aware embedding results in a certain degree of performance deterioration.

effectiveness of employing graph features in detecting fraudulent transactions is strongly demonstrated. GTAN outperforms the previous methods across all three datasets, validating the expressiveness of the temporal gated attention mechanism. The last row of Table II presents the results of our proposed method, RGTAN, which successfully outperforms all baselines with at least 2.5%, 0.75%, and 0.64% AUC improvements across the three datasets, respectively. Furthermore, RGTAN outperforms other baselines by at least 7.3%, 0.9%, and 0.2% AP improvements across the three datasets, respectively, strongly demonstrating the effectiveness of our risk-aware embeddings to capture multi-hop risk structure and higher-order risk patterns.

### C. Risk-aware Semi-supervised Experiment

To assess the effectiveness of semi-supervised learning, we conducted experiments with varying ratios of labeled and unlabeled data in the training set. To streamline the presentation of our findings, we focus on two of the most competitive baselines, namely PC-GNN and CARE-GNN, and use them as a point of comparison for the subsequent semi-supervised experiments. Specifically, we vary the proportion of training nodes from 10% to 80% in increments of 10%, while keeping the remaining nodes as the test set for each experiment. Our experiments are conducted on fully annotated datasets, YelpChi and Amazon, to enable us to explore a wider range of labeled data ratios. The results of our experiments are displayed in Figure 4.

Our analysis of the YelpChi dataset reveals that RGTAN consistently outperforms the other models under different training ratios. Even in scenarios with a limited number of labeled data (i.e., 10% training ratio), RGTAN performs well. Moreover, as the number of labeled data increases, the performance of RGTAN steadily improves despite some minor fluctuations. Similarly, our experiments on the Amazon dataset show that RGTAN consistently achieves the best performance across different training ratios. Compared to the YelpChi dataset, the RGTAN model exhibits less sensitivity to changes in the training ratio on the Amazon dataset, with no more than a 5% variation in AUC. These results suggest that RGTAN can perform well even with a small proportion of labeled data (as low as 10%).

In conclusion, our experiments demonstrate the robustness of the RGTAN model to changes in training ratio and its consistent superiority over PC-GNN and CARE-GNN in semi-supervised learning. These findings underscore the effectiveness of the RGTAN model in this domain.

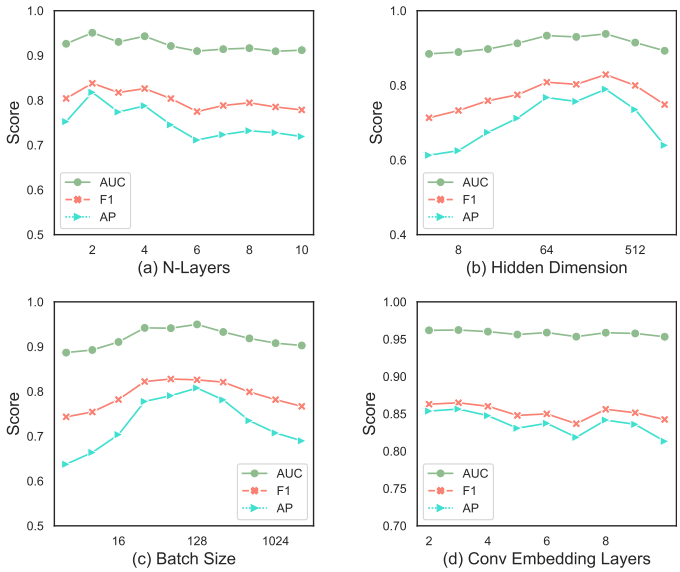


Fig. 6. Parameter sensitivity analysis with respect to (a) the number of GNN layers; (b) the number of temporal edges per node; (c) hidden dimension; and (d) the batch size. (a) shows that the performance of our model reaches the highest when the number of GNN layers is set to 2, and it slightly decreases when the number of layers continues to increase possibly due to the over-smoothing problem. (b) demonstrates that our model is robust to the choice of hidden dimension, with the overall AUC fluctuating by less than 5%. (c) illustrates that RGATN achieves the optimal performance when the batch size is 128, yet it is not sensitive to the choice of batch size, with the AUC under all settings varying by less than 3%. (d) exhibits high robustness to the setting of convolutional embedding layers, with the overall AUC fluctuating by less than 1%.

### D. Ablation Study

To test the effectiveness of each key component in our model, we evaluate three variants: RGTAN-A, RGTAN-R and RGTAN-N. RGTAN-A ablates the GTGA component and aggregates messages from neighboring nodes with equal weights, which means it fails to utilize the temporal graph attention mechanism to adaptively adjust the weights of neighbor nodes based on their importance for fraud detection. RGTAN-R ablates the risk embedding component and only uses the original node attributes $\mathbf{X}$ without considering the risk propagation process. This variant does not capture the credit

card fraud patterns from transaction risk propagation and only models the transaction embeddings from other attributes that are not related to risk propagation. In the RGTAN-N model, neighbor risk-aware embedding layers are removed to test its usability and validity. This variant RGTAN-N is incapable of capturing the high-order adjacent fraud patterns.

Figure 5 compares the performance of our model (RGTAN) and its three variants that ablate different components: RGTAN-A, RGTAN-R and RGTAN-N. The grey bars indicate that RGTAN-A, which removes the temporal graph attention mechanism, has the lowest scores among the four models. This demonstrates that the temporal graph attention mechanism is crucial for reweighting the temporal transaction neighbors and capturing their importance for fraud detection. The yellow bars show that RGTAN-R, which removes the risk embedding component, also performs worse than RGTAN, indicating that the risk embedding is conducive to modeling credit card fraud patterns from transaction risk propagation. The blue bars represent RGTAN-N, which removes the neighbor risk-aware embedding layers. RGTAN-N gains the second-highest scores among the four models, which also validates the effectiveness of neighbor risk-aware embedding in enhancing the capacity of RGTAN to detect fraud patterns. In summary, removing either component deteriorates the performance of RGTAN, which proves that the temporal graph attention mechanism, risk embedding and neighborhood risk representation are effective in graph-based credit card fraud detection.

### E. Parameter Sensitivity Experiment

In this section, we explore the sensitivity of our model's parameters by adjusting the number of temporal graph attention layers, hidden dimensions, batch size, and convolutional embedding layers. The findings from our tests on the YELP dataset are shown in Figure 6.

Our analysis starts with evaluating the impact of varying the number of temporal graph attention layers, testing from 1 to 10 layers as seen in Figure 6(a). The model's performance is consistent up to 10 layers of GNN. Increasing the number of hidden layers allows our model to incorporate temporal information from broader neighborhoods. Optimal results are observed with two GNN layers, where both AUC and AP scores are maximized, setting our standard layer depth at 2. Beyond this, adding more GTGA layers slightly reduces performance, potentially due to the over-smoothing effect on transaction embeddings as discussed in [52].

Next, we assess how changes in the hidden dimension, ranging from 4 to 1024, influence the model's effectiveness in Figure 6(b). The model exhibits stable performance throughout various hidden dimensions, peaking at a dimension of 256. Furthermore, Figure 6(c) reveals that the ideal batch size for our RGTAN model is 64. Despite minimal sensitivity to changes in the hidden dimension and batch size—showing less than 3% variation in AUC for dimensions between 16 and 1024—we opt for a batch size of 128 to enhance training efficiency. Lastly, the stability of our risk-aware representation learning mechanism is tested by altering the number of convolutional embedding layers from 2 to 10, as depicted in Figure 6(d). The performance of our RGTAN model remains robust, with less than 1% fluctuation in overall AUC.

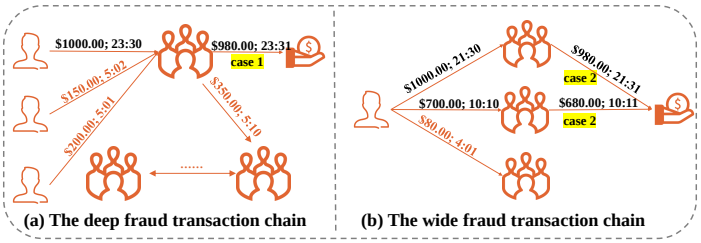


Fig. 7. The case studies on two typical fraud patterns: (a) fraudulent transactions can be hidden by a deep transaction chain; (b) over multiple links, multiple fraudulent transactions let money separately enter the same cardholder.

## VI. Case Studies

With the development of rule-based and machine learning-based fraud detection systems, today's credit card fraud is not simply an individual risk. For example, one of the cardholder's credit cards fell to the ground and was picked up by others. Instead, today's credit card fraud patterns have multi-hop temporal connections that form a *transaction chain*. As shown in Figure 7, according to the practical experience in credit card fraud detection, there are two typical types of fraud *transaction chains*. According to Figure 7(a), the money flows between multi-hop neighbors, and the end of this deep transaction chain is hidden by a long series of *'legitimate'* transactions. In Figure 7(b), the money flows through multiple pipelines, and the end of this wide transaction chain is hidden by multi-source *'legitimate'* transactions. In these cases, cardholders and rule-based fraud detection systems can only report the beginning of the transaction chains, while the end transactions of such fraud chains are difficult to detect by traditional machine learning algorithms due to their incapability to model relations among transaction chains. Therefore, detecting such cases requires a large number of effort from reviewers in card issuers.

We conduct case studies in a world-leading card issuer to validate the performance of detecting typical fraud patterns. Specifically, we select all transactions matching end-of-chain fraud patterns (i.e., case 1 and case 2 illustrated in Figure 7) from manually annotated cases. Then, an equal number of legitimate transactions are randomly selected among all transactions as the legitimate samples of the two cases, respectively. Then, we calculate the AUC and AP of the predicted results in these two groups of transactions, respectively. Table III reports the performance of 5 methods in detecting the end of chains of two typical types of fraud. Based on the first four rows, GTAN far outperforms all previous baselines with 16.0% and 14.9% AUC improvements and 19.5% and 17.4% AP improvements. According to the last two rows of Table III, the results show that our method RGTAN outperforms GTAN with 2.3% and 2.0% AUC improvements and 2.2% and 2.1% AP improvements. This demonstrates the effectiveness of our proposed RGTAN for identifying real-world human brain-armed credit card fraud patterns. The high capability in detecting complex

TABLE III
EVALUATION RESULTS ON DETECTING THE END OF TWO TYPES OF *fraud transaction chains*. OUR PROPOSED MODEL OUTPERFORMS OTHER BASELINES SIGNIFICANTLY IN DETECTING THESE TWO FRAUD PATTERNS.

| | Case 1 | | Case 2 | |
|---|---|---|---|---|
| | AUC | AP | AUC | AP |
| GraphConsis | 0.6178 | 0.3625 | 0.6626 | 0.4002 |
| CARE-GNN | 0.6274 | 0.3576 | 0.6715 | 0.4128 |
| PC-GNN | 0.6431 | 0.4251 | 0.6933 | 0.4656 |
| GTAN | 0.7932 | 0.6108 | 0.8321 | 0.6298 |
| RGTAN | **0.8167*** | **0.6325*** | **0.8518*** | **0.6504*** |

fraud patterns may be the main source of performance gain of our substantial improvement approach.

## VII. CONCLUSION AND FUTURE WORK

In this study, we tackled the significant practical challenge of credit card fraud detection. We developed an effective semi-supervised method that utilizes temporal transaction graphs and employs attribute-driven gated temporal attention networks due to the intensive and expensive nature of fraud transaction labeling. Our model introduces an attribute representation and risk propagation mechanism to accurately identify fraud patterns, considering the widespread categorical attributes and manually annotated labels. We introduced the use of neighborhood risk-aware representations to enhance the RGTAN's ability to discern local risk factors, highlighting the relevance of adjacent risk structures in detecting fraud. Our comprehensive testing demonstrated that our proposed methods outperform existing baselines across three datasets dedicated to fraud detection. The semi-supervised tests highlighted our model's exceptional ability to detect fraud using only a small portion of labeled data. Additionally, our case studies, which examined transaction chain propagation, revealed underlying fraud patterns, affirming our model's capacity to identify real-world fraudulent activities. Our methodology is progressing towards becoming an essential element of a real-world credit card fraud analysis system used by leading global card issuers, benefiting over a hundred million users. Despite its strong detection capabilities, the model still faces challenges with the computational complexity of analyzing high-order fraud patterns. Future work will focus on refining the detection of risk-aware fraud patterns more effectively and efficiently.

## REFERENCES


[1] S. Bhattacharyya, S. Jha, K. K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, pp. 602–613, 2011.

[2] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[3] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, "Graph neural network for fraud detection via spatial-temporal attention," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[4] R. Patidar, L. Sharma *et al.*, "Credit card fraud detection using neural network," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 1, no. 32-38, 2011.

[5] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *International Conference on Neural Information Processing*. Springer, 2016, pp. 483–490.

[6] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decision Support Systems*, vol. 95, pp. 91–101, 2017.

[7] K. Seeja and M. Zareapoor, "Fraudminer: A novel credit card fraud detection model based on frequent itemset mining," *The Scientific World Journal*, vol. 2014, 2014.

[8] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, 2017.

[9] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, and Y. Qi, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 598–607.

[10] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: a gnn-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021*, 2021, pp. 3168–3177.

[11] Y. Gao, X. Wang, X. He, H. Feng, and Y. Zhang, "Rumor detection with self-supervised learning on texts and social graph," *Frontiers of Computer Science*, vol. 17, pp. 1–15, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248240133

[12] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 315–324.

[13] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[14] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, "Spatio-temporal attention-based neural network for credit card fraud detection," in *AAAI*, 2020, pp. 362–369.

[15] D. Cheng, Y. Tu, Z. Ma, Z. Niu, and L. Zhang, "Risk assessment for networked-guarantee loans using high-order graph attention representation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 5822–5828.

[16] S. Xiang, M. Zhu, D. Cheng, E. Li, R. Zhao, Y. Ouyang, L. Chen, and Y. Zheng, "Semi-supervised credit card fraud detection via attribute-driven graph representation," in *AAAI*, 2023.

[17] A. Reurink, "Financial fraud: a literature review," *Journal of Economic Surveys*, vol. 32, no. 5, pp. 1292–1325, 2018.

[18] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein, "Improving graph neural network expressivity via subgraph isomorphism counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 657–668, 2022.

[19] B. Baesens, S. Höppner, and T. Verdonck, "Data engineering for fraud detection," *Decision Support Systems*, vol. 150, p. 113492, 2021, interpretable Data Science For Decision Making. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923621000026

[20] R. Saia and S. Carta, "Evaluating the benefits of using proactive transformed-domain-based techniques in fraud detection tasks," *Future Generation Computer Systems*, vol. 93, pp. 18–32, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18306423

[21] S. Carta, G. Fenu, D. Reforgiato Recupero, and R. Saia, "Fraud detection for e-commerce transactions by employing a prudential multiple consensus model," *Journal of Information Security and Applications*, vol. 46, pp. 13–22, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212618304216

[22] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," in *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, 2002, pp. 261–270.

[23] M. J. Zaki, W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[24] Y. G. Şahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," 2011.

[25] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: A systematic literature review of graph-based anomaly detection approaches," *Decision Support Systems*, vol. 133, p. 113303, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923620300580

[26] R. Van Belle, S. Mitrović, and J. De Weerdt, "Representation learning in graphs for credit card fraud detection," in *Mining Data for Financial*

*Applications: 4th ECML PKDD Workshop, MIDAS 2019, Würzburg, Germany, September 16, 2019, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, 2019, p. 32–46. [Online]. Available: https://doi.org/10.1007/978-3-030-37720-5_3

[27] R. Zhang, D. Cheng, J. Yang, Y. Ouyang, X. Wu, Y. Zheng, and C. Jiang, "Pre-trained online contrastive learning for insurance fraud detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 20, pp. 22 511–22 519, Mar. 2024. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/30259

[28] M. Huang, Y. Liu, X. Ao, K. Li, J. Chi, J. Feng, H. Yang, and Q. He, "Auc-oriented graph neural network for fraud detection," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1311–1321.

[29] F. Shi, Y. Cao, Y. Shang, Y. Zhou, C. Zhou, and J. Wu, "H2-fdetector: a gnn-based fraud detector with homophilic and heterophilic connections," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1486–1494.

[30] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3917–3925.

[31] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognition*, vol. 121, p. 108218, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132032100399X

[32] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang, "Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification," *arXiv preprint arXiv:1801.09927*, 2018.

[33] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[35] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017.

[36] M. Gong, H. Zhou, A. Qin, W. Liu, and Z. Zhao, "Self-paced co-training of graph neural networks for semi-supervised node classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[37] Y. Zhu, W. Xu, J. Zhang, Y. Du, J. Zhang, Q. Liu, C. Yang, and S. Wu, "A survey on graph structure learning: Progress and opportunities."

[38] Q. Sun, J. Li, H. Peng, J. Wu, X. Fu, C. Ji, and S. Y. Philip, "Graph structure learning with variational information bottleneck," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4165–4174.

[39] H. Edelsbrunner and J. L. Harer, *Computational topology: an introduction*. American Mathematical Society, 2022.

[40] Q. Zhao and Y. Wang, "Learning metrics for persistence-based summaries and applications for graph classification," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[41] B. Rieck, C. Bock, and K. Borgwardt, "A persistent weisfeiler-lehman procedure for graph classification," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5448–5458.

[42] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl, "Deep learning with topological signatures," *Advances in neural information processing systems*, vol. 30, 2017.

[43] Q. Zhao, Z. Ye, C. Chen, and Y. Wang, "Persistence enhanced graph neural network," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2896–2906.

[44] M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt, "Topological graph neural networks," *arXiv preprint arXiv:2102.07835*, 2021.

[45] Y. Shi, Z. Huang, W. Wang, H. Zhong, S. Feng, and Y. Sun, "Masked label prediction: Unified massage passing model for semi-supervised classification," in *IJCAI*, 2021.

[46] Y. Gorishniy, I. Rubachev, and A. Babenko, "On embeddings for numerical features in tabular deep learning," 2022. [Online]. Available: https://arxiv.org/abs/2203.05556

[47] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, 2015, pp. 985–994.

[48] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 897–908.

[49] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 2077–2085.

[50] J. Wang, R. Wen, C. Wu, Y. Huang, and J. Xion, "Fdgars: Fraudster detection via graph convolutional networks in online app review system," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 310–316.

[51] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, "Key player identification in underground forums over attributed heterogeneous information network embedding framework," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 549–558.

[52] L. Zhao and L. Akoglu, "Pairnorm: Tackling oversmoothing in gnns," *ArXiv*, vol. abs/1909.12223, 2020.

**Sheng Xiang** is a PhD candidate in the Center for Artificial Intelligence, major in Computer Science, University of Technology, Sydney (UTS). He received his BSc degree in Bioinformatics Engineering from Shanghai Jiao Tong University. His research interests include graph machine learning in finance, graph generative algorithm, bipartite graph processing, and dynamic graphs.

**Guibin Zhang** is currently an undergraduate in the Department of Computer Science and Technology, College of Electronic and Information Engineering, major in Data Science, Tongji University, Shanghai, China. His research interest include data mining, graph representation learning and knowledge modeling.

**Dawei Cheng** is an associate professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. Before that, he was a postdoctoral associate at MoE Key Laboratory of Artificial Intelligence, department of computer science, Shanghai Jiao Tong University. He received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China. His research fields include graph learning, big data computing, data mining and machine learning.

**Ying Zhang** is a Professor and ARC Future Fellow (2017- 2021) at Australia Artificial Intelligence Institute (AAII), the University of Technology, Sydney (UTS). He received his BSc and MSc degrees in Computer Science from Peking University, and PhD in Computer Science from the University of New South Wales. His research interests include query processing and analytics on large-scale data with focus on graphs and high dimensional data.