PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Conspiracy Spoofing Detection via Structure-Augmented Generative Graph Model

Sheng Xiang[1] , **Ziwen Xu**[1] , **Yidong Jiang**[1] , **Dawei Cheng**[1*] and **Hui Zhao**[2]
[1]Tongji University, Shanghai, China
[2]CITIC Bank, Beijing, China
{xiangsheng, 2253886, 2253899, dcheng}@tongji.edu.cn, zhaohui3@citicbank.com,

Abstract

Detecting spoofing in financial trading is a critical data mining task. While traditional machine learning models focus on individual node features, graph-based methodologies have shown superior performance by integrating relational data and structure information. However, spoofing transactions often exhibit distribution shifts, making historical data less effective. Instead, certain consistent trading patterns, such as motif structures, remain robust for analysis across different distributions. Therefore, this paper introduces the Structure-Augmented Generative Graph Model (SAG^{2}M) to detect conspiracy spoofing through substructure frequency-augmented detection. Our approach extracts subgraph pattern frequencies among neighboring nodes using an enumeration algorithm, and encodes these motif frequencies into a structure-augmented generative framework. A temporal and heterogeneous graph generation and aggregation scheme is then applied to collect neighborhood node information, effectively uncovering conspiracy spoofing patterns. Experiments on benchmark datasets demonstrate that SAG^{2}M outperforms existing models in detection accuracy. Case studies further highlight its superior effectiveness in identifying complex fraudulent behaviors.

IJCAI-ECAI 2026 PREPRINT

1 Introduction

Spoofing, a deceptive trading practice, manipulates market perceptions to gain unfair advantages. For example, as shown in Figure 1, traders may place large non-executable orders (e.g., sell orders before buying or buy orders before selling) to distort supply and demand, thereby manipulating prices [Charoenwong *et al.*, 2024]. This practice distorts market dynamics, undermines fairness and transparency, and causes significant economic losses annually. Detecting spoofing is critical for fair trading in financial markets such as stock trading and commodity exchanges [Du *et al.*, 2024; Charoenwong *et al.*, 2023]. Recently, as computing power grows, algorithmic trading and high-frequency trading enable spoofing to be executed at unprecedented speeds, making manual spoofing detection difficult. The expansion of e-commerce and digital banking further exacerbates this issue, highlighting the need for advanced fraud detection systems [Motie and Raahemi, 2024; See *et al.*, 2024].

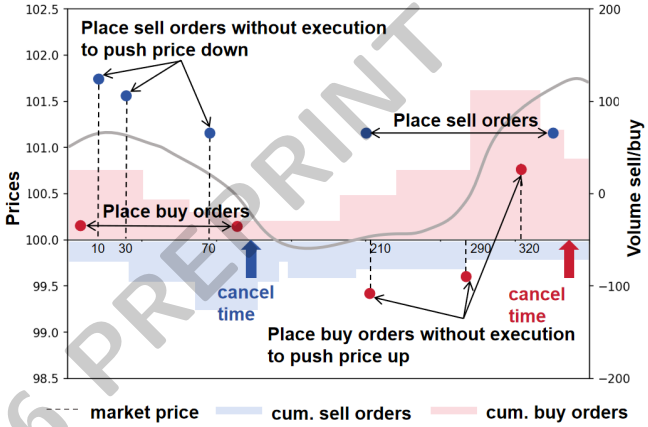

Figure 1: The typical example of spoofing. A trader places orders and cancels them before execution with the intent of gaining illegal profit by creating a false impression of demand or supply.

Traditional spoofing detection relied on rule-based methods, which generated alerts based on predefined behaviors [Whitrow *et al.*, 2009]. These methods lacked adaptability to evolving fraud patterns. Machine learning approaches emerged as data-driven solutions that continuously improve over time [Bhattacharyya *et al.*, 2011]. Deep learning techniques, particularly Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs), have advanced the field by identifying complex fraud patterns [Fu *et al.*, 2016; Cheng *et al.*, 2020; Xiang *et al.*, 2023; Liu *et al.*, 2023a; Li *et al.*, 2024]. Graph-based methods have further enhanced fraud detection by considering both individual node features and network relationships [Cheng *et al.*, 2025]. These methods capture intricate patterns often missed by traditional approaches [Vatter *et al.*, 2023]. For example, graph attention networks and spatial graph networks have been applied to financial fraud detection [Liu *et al.*, 2023b]. Recent models like BWGNN, GAGA, PC-GNN, GTAN, and

*Corresponding Author.

BSL address challenges such as spectral heterophily and label scarcity [Tang *et al.*, 2022b; Wang *et al.*, 2023; Liu *et al.*, 2021; Xiang *et al.*, 2023; Yu *et al.*, 2024].

However, existing graph-based methods face three main challenges in spoofing detection: (i) newly emerging transactions often lack labeled connections, limiting the effectiveness of label propagation; (ii) current methods struggle to capture complex substructures associated with spoofing; and (iii) they fail to simultaneously account for temporal dynamics and heterogeneity in distinguishing spoofing from benign transactions. All the baselines fall in at least one of them. To address these challenges, we propose a novel Structure-Augmented Generative Graph Model (SAG^{2}M). Our approach includes a GPU-friendly preprocessing step to extract subgraph pattern frequencies, generating node orbit data that enhances GNN generalization. A structure-augmented generative encoder captures neural representations of node orbit frequencies and temporal-heterogeneous aggregation mechanisms propagate these representations across nodes. Extensive experiments on seven datasets show that SAG^{2}M outperforms state-of-the-art models. Additionally, a case study demonstrates our model's ability to adaptively select key node orbits and generate meaningful subgraph representations for improved detection performance.

Our contributions can be summarized as follows:

- We present the first work addressing financial spoofing detection through structure-augmented generative graph learning with subgraph patterns enhancement.
- We develop a structure-augmented generative encoder to generate discriminative neural representations for neighbor substructures and propose temporal-heterogeneous graphs for representation aggregation.
- Extensive experiments on public datasets and real-world case studies demonstrate our model's superior performance in conspiracy spoofing detection.

2 Related Works

This section reviews existing methodologies relevant to our work, focusing on: (i) spoofing detection methods and (ii) deep graph learning for fraud detection.

2.1 Spoofing Detection

Spoofing has become a critical issue in financial markets [Chanthati, 2024]. Existing spoofing detection methods can be categorized into statistical model-based approaches, which utilize mathematical metrics to identify anomalies [Leangarun *et al.*, 2016], and MDP-based methods that simulate hidden states for detecting irregularities through state transitions [Cao *et al.*, 2014; Wang and Charoenwong, 2024]. Agent-based methods have been used to model spoofing strategies in limit order books [Wang and Wellman, 2017], while rule-based systems remain widely adopted despite their limited adaptability [Mendonça and De Genaro, 2020]. Some approaches focus on detecting price manipulation through order book volume imbalances [Tao *et al.*, 2022]. Deep learning techniques, including CNNs and GNNs, have been explored for conspiracy spoofing detection [Arora and Bhatia, 2020; Kang *et al.*, 2023]. However, these models face challenges such as handling distribution shift [Wu *et al.*, 2023], limited expressiveness for complex trading patterns, and difficulty in modeling the dynamic nature of financial transactions.

2.2 Deep Graph Learning for Fraud Detection

Deep graph learning has emerged as a powerful approach for fraud detection, leveraging the relational structure of data to uncover hidden patterns [Wu *et al.*, 2024]. These methods span a variety of techniques, each addressing specific challenges in fraud detection. Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs) have been proposed to enable automatic feature engineering for fraud detection tasks [Cheng *et al.*, 2020]. Graph attentive networks, introduced in [Liu *et al.*, 2023b], further enhanced fraud detection by assigning attention weights to critical nodes and edges. To tackle challenges like camouflaged fraud, reinforcement learning-enhanced GNNs were developed [Dou *et al.*, 2020], demonstrating their ability to adaptively learn fraud strategies. Label imbalance, a common issue in fraud detection datasets, has been addressed through methods such as PC-GNN [Liu *et al.*, 2021], which incorporates both positive and negative sampling to improve performance. Given the high cost of acquiring labeled data, weakly supervised methods have also been proposed [Xiang *et al.*, 2023], reducing reliance on extensive labeled datasets while maintaining detection efficacy [Ma *et al.*, 2024; Zou *et al.*, 2024]. Additionally, the heterogeneous nature of graph data has spurred the development of heterogeneous graph neural networks (HGNNs). These include methods addressing spectral heterophily [Tang *et al.*, 2022b] and spatial heterophily [Wang *et al.*, 2023], which leverage unique graph properties to improve detection performance. These approaches have achieved notable success on publicly available datasets, particularly in domains like review fraud detection.

However, these methods have not been specifically tailored to address the unique challenges of conspiracy spoofing detection, such as distribution shifts caused by data isolation and dynamic trading patterns. To bridge this gap, we propose the Structure-Augmented Generative Graph Model (SAG^{2}M). Our model introduces a novel structure-augmented generative encoder alongside an end-to-end GNN architecture, specifically designed to capture the structural and dynamic complexities of spoofing detection.

3 Methodology

In graph-based fraud detection, GNNs often struggle with capturing complex patterns due to limited expressive power [Chen *et al.*, 2020]. Additionally, new transactional data frequently lacks connections with historical data, hindering feature and label propagation effectiveness [Wang *et al.*, 2023; Xiang *et al.*, 2023]. To address these challenges, we propose a structure-augmented generative graph model. This section introduces the framework and its key components.

3.1 Problem Definition

In this paper, we focus on three key concepts for detecting spoofing behavior in financial trading:

IJCAI-ECAI 2026 PREPRINT

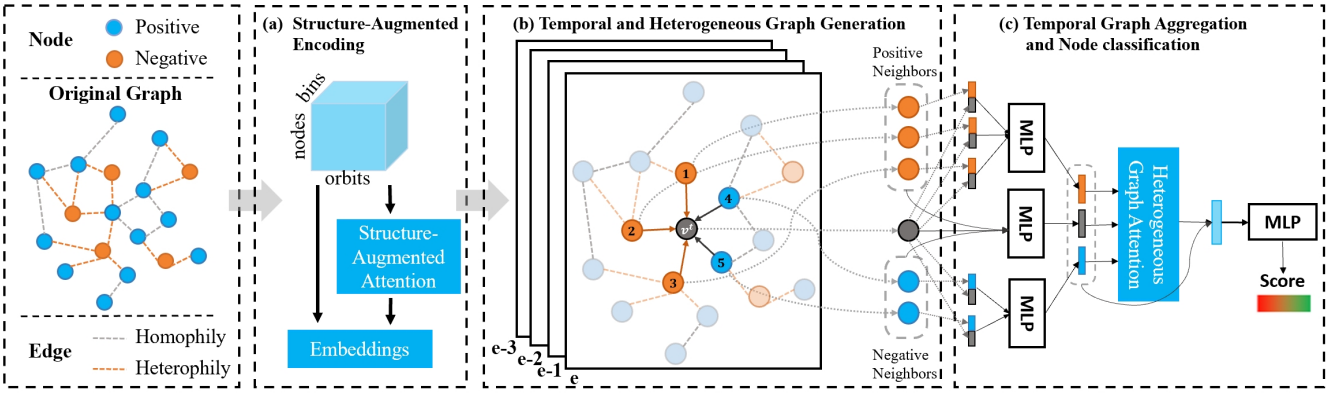

Figure 2: The proposed Structure-Augmented Generative Graph Model (SAG^{2}M) architecture for Conspiracy Spoofing Detection. The first part is the structure encoding of an input graph, which builds the embedding of the node structure orbits. The second part is the temporal and heterogeneous graph generation. The third part combines the temporal graph aggregation and heterogeneous graph attention layers, where information from temporal and heterogeneous neighbors is aggregated, and adaptive attention weights are calculated for different neighbor types. Then, we leverage a multi-layer perception to give the prediction of whether this transaction is spoofing.

- **Transactional Graphs:** Represented as $G = (V, E, X)$, where V is nodes (traders/accounts), E is edges (transactional relationships), and X denotes node attributes (e.g., order size).
- **Node Orbit Features:** Characterize roles of nodes in graph structures. Formally, the orbit of node v is defined as $\mathcal{O}(v) = \{\text{roles of } v \text{ across subgraphs}\}$.
- **Spoofing Detection Task:** Given historical transactional graph G and labels Y, predict fraud likelihood for new order o. Formally, estimate $\hat{y}_o = f(G, x_o, Y)$, where $\hat{y}_o$ indicates fraud legitimacy.

3.2 Model Architecture

Our Structure-Augmented Generative Graph Model (SAG^{2}M) consists of four components: (i) **Structure-Augmented Generative Encoding:** Captures structural properties using orbit-based embeddings through a Structure Reconstruction mechanism; (ii) **Temporal and Heterogeneous Graph Generation:** Extracts temporal snapshots (e.g., $e-3, e-2, e-1, e$) and considers node/edge heterogeneity; (iii) **Temporal Graph Aggregation:** Uses Mean pooling followed by MLP to aggregate temporal neighbor information; (iv) **Heterogeneous Graph Attention and Node Classification:** Incorporates attention mechanism to weight important neighbors, producing final classification scores via MLP. Now we will go to each component.

3.3 Node Orbit Counting Algorithm

To enhance GNN expressive power for new accounts disconnected from historical data, we propose a 4-node orbit counting approach. Given input graph G, we compute orbit features based on subgraph patterns (e.g., triangles and 4-cliques). Our three-stage GPU-accelerated algorithm has complexity $O(d^3)$ and produces an attribute matrix $\mathcal{O} \in \mathbb{R}^{n \times r}$, where n is nodes and r is orbit types.

Algorithm 1: Node Orbit Counting Algorithm

Input : $G(V, E)$: the given graph
Output: O: node orbit counts for each node
for $e(u, v) \in E$ **do**
  $\mathcal{T}[e] \leftarrow$ Enumerate triangles involved by edge e;
for $v \in V$ **do**
  $\mathcal{C}[v] \leftarrow$ Enumerate 4-cliques involved by node v;
for $v \in V$ **do**
  Enumerate all 4-node subgraph frequencies;
  $\mathcal{O}[v] \leftarrow$ Build equations of orbit counting for node v;
return $\mathcal{O}$;

Triangle Count per Edge

We propose an efficient algorithm for enumerating triangles related with each edge in a graph $G(V, E)$. Neighbors of nodes are sorted by their unique identifiers to enable optimized traversal. For an edge $e(x, y)$, let N_x and N_y denote the neighbor lists of nodes x and y. Using a two-pointer technique, we identify common neighbors between N_x and N_y to count triangles involving $e(x, y)$. The pseudocode is shown in Algorithm 2. The algorithm processes each edge in linear time relative to the degrees of its endpoints. For an edge $e(x, y)$, the time complexity is $O(\deg(x) + \deg(y))$. Summing over all edges gives an overall complexity of $O(E \cdot \bar{d})$, where $\bar{d}$ is the average degree of all nodes.

4-Clique Count per Node

This algorithm efficiently counts 4-cliques involving a node x using sorted neighbor lists and a temporary global storage structure. For each neighbor $y > x$ of x, we identify common neighbors z stored in neigh_tri. Pairs of common neighbors z_1 and z_2 are checked for connectivity to detect 4-cliques. The pseudocode is provided in Algorithm 3. For a node x, the algorithm runs in $O(\deg(x)^3)$. Summing over all nodes gives an overall complexity of $O\left(\sum_{x \in V} \deg(x)^3\right)$.

IJCAI-ECAI 2026 PREPRINT

Algorithm 2: Enumerate Triangles for Each Edge

Input : $G(V, E)$: the given graph, with neighbors sorted by ID
Output: T: number of triangles for each edge
for $e(x, y) \in E$ **do**
  $T[e] \leftarrow 0$;
  $N_x \leftarrow \text{neighbors}(x)$;
  $N_y \leftarrow \text{neighbors}(y)$;
  $i \leftarrow 0, j \leftarrow 0$;
  while $i < |N_x|$ ***and*** $j < |N_y|$ **do**
    if $N_x[i] < N_y[j]$ **then**
      $i \leftarrow i + 1$;
    else if $N_x[i] > N_y[j]$ **then**
      $j \leftarrow j + 1$;
    else
      $T[e] \leftarrow T[e] + 1$;
      $i \leftarrow i + 1, j \leftarrow j + 1$;
return T ;

Algorithm 3: Enumerate 4-Cliques for Each Node

Input : $G(V, E)$: the given graph, with neighbors sorted by ID
Output: C: number of 4-cliques for each node
for $x \in V$ **do**
  $N_x \leftarrow \text{neighbors}(x)$;
  for $y \in N_x$ ***where*** $y > x$ **do**
    $n\_tri \leftarrow 0$;
    for $z \in$ *neighbors*(y) ***where*** $z > y$ **do**
      if $z \in N_x$ **then**
        $n\_tri \leftarrow n\_tri + 1$;
        neigh_tri$[n_tri] \leftarrow z$;
    for $i \leftarrow 1$ ***to*** n_tri **do**
      for $j \leftarrow i + 1$ ***to*** n_tri **do**
        $z_1 \leftarrow$ neigh_tri$[i]$;
        $z_2 \leftarrow$ neigh_tri$[j]$;
        if $(z_1, z_2) \in E$ **then**
          $C[x] \leftarrow C[x] + 1$;
          $C[y] \leftarrow C[y] + 1$;
          $C[z_1] \leftarrow C[z_1] + 1$;
          $C[z_2] \leftarrow C[z_2] + 1$;
return C ;

Full Graphlet and Node Orbit

By leveraging edge triangle counts $T[e]$ and node 4-clique counts $C[v]$, we derive graphlet frequencies and node orbit distributions. For each node x in a 3-node graphlet xyz, four types of orbits are defined: (i) x is a side node in a path ($T_{|e(x,y)|} = 0$); (ii) x is a side node in a triangle; (iii) x is the center node in a path ($T_{|e(x,y)|} = 0$); (iv) x is the center node in a triangle. These classifications extend to 4-node orbits, capturing structural roles within graphlets. Edge triangle counts identify paths vs. triangles, while 4-clique counts detect fully connected subgraphs.

GPU-based Enumeration

Graphlet enumeration algorithms are parallelizable on GPUs with two key considerations: (i) **Safe write operations in node 4-clique counting**: Each thread writes to a preallocated memory region for its assigned node x and neighbors, avoiding interference; (ii) **Memory constraints for center-orbit paths**: Enumerating squares (4-cycles) must be handled directly without precomputed path counts due to limited per-thread memory and unsafe parallel writes. The proposed algorithms efficiently utilize GPU cores by localizing computations to individual edges or nodes. This enables scalable performance in large graphs and facilitates orbit label computation for identifying spoofing transactions.

3.4 Structure-Augmented Encoder

We propose a structure-augmented encoder that transforms node orbit features $\mathcal{O}$ into embeddings $\mathbf{H}_{orb} \in \mathbb{R}^{n \times d}$ using binning encoding with $\mathbf{X}_{orb} = \text{BinningEncoding}(\mathcal{O}, b)$, and attention mechanisms, which is:

$$\mathbf{H}_{orb} = \sum_{i=1}^{r} \alpha_i \cdot \mathbf{X}_{orb}[i] \tag{1}$$

where attention weights α_i are formulated as follows:

$$\alpha_i = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T \mathbf{X}_{orb}[i]))}{\sum_{j=1}^{r} \exp(\text{LeakyReLU}(\mathbf{a}^T \mathbf{X}_{orb}[j]))} \tag{2}$$

The embeddings are concatenated with original graph attributes for input to the aggregation network.

3.5 Temporal and Heterogeneous Aggregation

Our aggregation process includes: (i) **Graph Construction:** Edges are labeled as pos or neg based on a threshold score. Pseudo-labels are determined by model predictions; (ii) **Temporal Aggregation:** Historical transaction features are aggregated using: $\mathbf{H}_{temp}(i) = Mean\left(\sum_{j \in \mathbb{N}(i)} \mathbf{X}(i)\right)$; (iii) **Heterogeneous Aggregation:** Information from positive, negative, and all neighbors is aggregated using attention mechanisms:$\mathbf{H}_{het} = \text{HeteroAttention}(\mathbf{H}_{pos}, \mathbf{H}_{neg}, \mathbf{H}_{all})$; The coefficients β_{pos}, β_{neg}, and β_{all} are computed as:

$$\beta_t = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_t]))}{\sum_{s \in \{pos, neg, all\}} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_s]))} \tag{3}$$

The final node representation is:

$$\mathbf{h}_{het}^i = \beta_{pos} \cdot \mathbf{h}_{pos} + \beta_{neg} \cdot \mathbf{h}_{neg} + \beta_{all} \cdot \mathbf{h}_{all} \tag{4}$$

3.6 Optimization Objective

The optimization objective combines binary classification loss and a generative auto-encoder loss:

$$\begin{aligned} \mathcal{L}_{all} = &- \frac{1}{N} \sum_{i=1}^{N} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \\ &+ MSELoss(MLP(\mathbf{H}_{orb}), \mathbf{X}_{orb}) \end{aligned} \tag{5}$$

where $\mathcal{L}_{all}$ is optimized using Adam, y_i denotes the ground truth label, and MLP denotes the multi-layer perceptron to reconstruct features.

IJCAI-ECAI 2026 PREPRINT

Dataset	#Node	#Edge	Fraud(%)	#Feat
Amazon	11,944	4,398,392	6.88%	25
YelpChi	45,954	3,846,979	14.53%	32
Reddit	10,984	168,016	3.33%	64
Weibo	8,405	407,963	4.13%	400
T-Finance	39,357	21,222,543	4.58%	10
T-Social	5,781,065	73,105,508	3.02%	10
FFSD	1,820,840	31,619,440	2.34%	126
Spoofing	40,072	7,241,465	25.13%	44

Table 1: Dataset Statistics

4 Experiments

4.1 Experimental Settings

Datasets. We collect partially labeled records from our collaborated partners forming the new dataset called **Spoof Detection**, which is designed for detecting spoofing behaviours. The ground truth labels are obtained from cases reported by traders and confirmed by financial domain experts. If a transaction is reported by a trader or identified by financial experts as fraudulent, we label it as 1; otherwise, it is labeled as 0. Besides, we also experimented on seven public fraud detection datasets: **Amazon**, **YelpChi**, **Reddit**, **Weibo**, **T-Finance**, **T-Social**, and **FFSD**. The **Amazon** dataset [McAuley and Leskovec, 2013] contains product reviews for musical instruments, with nodes as users and edges as review relationships. The **YelpChi** dataset [Rayana and Akoglu, 2015] includes hotel and restaurant reviews, where nodes are reviews and edges are their connections. The **Reddit** dataset [Kumar *et al.*, 2019] features posts across subreddits, with nodes as users, edges as interactions, and features derived from Linguistic Inquiry and Word Count categories. The **Weibo** dataset [Zhao *et al.*, 2020] represents Tencent Weibo users as nodes, with features including location and bag-of-words representations, while edges indicate associations between users and hashtags. The **T-Finance** and **T-Social** [Tang *et al.*, 2022a] datasets focus on anomaly detection in transaction and social networks, respectively, with nodes as accounts and edges as transactions or friendships. The **FFSD** dataset [Xiang *et al.*, 2023] models financial fraud with nodes as transactions and edges representing relationships. The datasets' basic statistics are shown in Table 1.

Compared Methods. To highlight the effectiveness of our proposed SAG^{2}M, we compared it with several state-of-the-art methods including: MLP, a Multi-Layer Perceptron with hidden size 32; RF, a Random Forest model using default parameters; GCN [Kipf and Welling, 2017], a Graph Convolutional Network with hidden size 64 and learning rate 0.01; GAT [Veličković *et al.*, 2018], a Graph Attention Network with similar hyperparameters as GCN; CARE-GNN [Dou *et al.*, 2020] and PC-GNN [Liu *et al.*, 2021], both using default parameters for fraud detection on relational graphs and class imbalance problems respectively; BWGNN [Tang *et al.*, 2022b], a model using spectral analysis for anomaly detection; GTAN [Xiang *et al.*, 2023], a semi-supervised GNN with Gated Temporal Attention Networks; PMP [Zhuo *et al.*, 2024], a Partitioning Message Passing framework with learning rate 0.01 and batch size 256; DGA-GNN [Duan *et al.*, 2024], a Dynamic Grouping Aggregation GNN using decision tree binning with hidden dimension 192. Our proposed SAG^{2}M uses threshold $h = 0.1$, embedding dimension $d = 128$, and number of bins $b = 4$.

Evaluation Metrics. We employ three evaluation metrics to assess the experimental results on spoof detection and other public fraud detection datasets: area under the ROC curve (AUC), macro average of F1 score (F1-macro), and averaged precision (AP). We count the number of True Positive N_{TP} (i.e., correct identification of positive labels), False Positive N_{FP} (i.e., incorrect identification of positive labels), and False Negatives N_{FN} (i.e., incorrect identification of negative labels). Then, F1 score and AP are formulated as follows: $F1_{\text{macro}} = \frac{1}{l}\sum_{i=1}^{l}\frac{2\times P_i \times R_i}{P_i+R_i}$, where $P_i = \frac{N_{TP}}{N_{TP}+N_{FP}}$ and $R_i = \frac{N_{TP}}{N_{TP}+N_{FN}}$. And similarly, the AP score is formulated as follows: $AP = \sum_{i=1}^{l}(R_i - R_{i-1})P_i$.

4.2 Performance Comparison

In this section, we compare the performance of our proposed model against several baseline methods across three domains: Finance, Review, and Social. We utilize metrics such as Area Under the ROC Curve (AUC) and Precision to assess the effectiveness of each method. Table 2 displays the detailed results of these comparisons. In the experimental setup, each model was evaluated over ten iterations, with average results presented in Table 2. Significant improvements, marked with *, were validated through a paired t-test with a p-value threshold of 0.01. The first 11 rows of Table 2 showcases the performance of machine learning baselines and traditional GNN-based models such as CARE-GNN, PC-GNN, BW-GNN, GTAN, and DGA-GNN across the three domains. It becomes evident that MLP and RF struggle to handle complex financial fraud scenarios due to their limited ability to utilize relational information. In contrast, GNN-based methods, such as PC-GNN and BW-GNN, yield improved outcomes due to their enhanced architectural advancement, though they still lag behind more advanced models. GTAN and DGA-GNN further refine this progression, with DGA-GNN consistently delivering competitive results due to its dynamic grouping capabilities. The addition of our model, listed as "Ours" in Table 2, consistently surpasses all baseline methods. Notably, our approach achieves at least 0.21%, 1.12%, and 0.35% improvements in AUC for T-Finance, FFSD, and Reddit datasets, respectively. Furthermore, the precision enhancements across these datasets emphasize our model's robustness in identifying fraudulent activities with higher accuracy. This indicates the superior capability of our model in capturing intricate patterns and relationships in graph-structured data, thereby offering a substantial advantage in fraud detection tasks.

4.3 Ablation Study

To evaluate the contribution of each component in our model, we conducted an ablation study by removing specific features from the model: (i) **Without Heterogeneous Aggregation:** This variant omits the heterogeneous aggregation mechanism, which leverages mean-aggregation to collect different types of neighbor data; (ii) **Without Structure-Augmented Encoder:** We removed the structure-augmented encoder to

IJCAI-ECAI 2026 PREPRINT

Domain	Finance						Review				Social					
Method	T-Finance		FFSD		Spoofing		Amazon		YelpChi		Reddit		Weibo		T-Social	
	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.	AUC	Prec.
MLP	93.2	69.2	66.8	44.4	78.7	62.0	93.6	79.7	74.1	37.0	59.4	5.60	87.0	28.9	76.5	10.9
RF	93.7	80.3	75.6	47.7	82.5	61.1	95.1	85.8	90.7	69.1	57.6	4.97	85.9	25.5	78.9	41.4
GCN	91.3	76.3	66.5	34.1	73.6	60.1	89.5	63.5	63.9	27.4	59.8	4.74	91.3	32.6	87.9	58.3
GAT	91.8	78.7	64.1	30.1	73.4	58.6	88.7	66.0	75.7	39.2	61.5	5.01	87.6	26.3	93.7	62.7
CARE-GNN	91.6	73.2	70.4	47.5	78.5	61.7	91.2	82.2	79.3	42.7	66.8	6.28	89.2	29.8	85.1	41.1
PC-GNN	91.4	67.4	71.2	45.2	78.3	50.9	95.8	85.5	81.7	48.1	62.6	3.45	90.0	14.1	89.0	50.8
BW-GNN	95.6	85.3	68.8	43.3	82.9	70.1	97.0	87.5	83.4	47.6	65.4	14.3	78.8	36.8	95.5	74.9
GTAN	96.4	88.8	80.3	63.2	81.7	66.7	96.3	83.9	92.4	75.1	65.1	4.97	91.1	30.2	98.6	90.3
DGA-GNN	97.7	90.2	69.4	43.8	90.3	78.4	97.7	92.0	97.7	92.6	63.3	6.64	92.5	32.8	99.3	91.9
PMP	97.4	89.9	65.1	33.1	80.5	61.5	97.8	90.5	94.1	78.1	54.7	3.91	81.8	18.3	**99.4**	93.4
Ours	**97.9***	**90.3***	**81.5***	**64.5***	**90.4***	**78.6***	**98.1***	**92.3***	**98.0***	**93.0***	**67.1***	**14.4***	**92.7***	**37.2***	99.1	**93.4**

Table 2: Performance comparison for 8 datasets with baseline methods, where we have omitted the percentage sign (%).

Model Variant	AUC	Precision
Full Model	**0.9792**	**0.9025**
w/o Heterogeneous Aggregation	0.9435	0.8764
w/o Structure-Augmented Encoder	0.9387	0.8702
w/o Node Orbit Features	0.9123	0.8490

Table 3: Ablation study results, showing the impact of different model components, where the higher is better.

assess its impact on the model's ability to leverage graph structures effectively; (iii) **Without Node Orbit Features:** This configuration excludes the node orbit features, which are pivotal in capturing subgraph patterns like triangles and 4-cliques. The experimental results on T-Finance dataset, as shown in Table 3, illustrate the significant impact of each component, demonstrating that the full model achieves superior performance by integrating these features.

It is well-known that subgraph enumeration algorithms are typically complex and time-consuming. To further evaluate our approach, we conducted a runtime comparison on synthetic Erdős-Rényi graph datasets [Erdos and Rényi, 1984]. Specifically, we compared our GPU-friendly improved version ("Ours") with the original C++ implementation ("Orca") [Hočevar and Demšar, 2014] in terms of time consumption across graph sizes ranging from 100 to 1 million nodes. As shown in Figure 3, our method significantly outperforms the baseline in all tested scenarios. Notably, the maximum speedup achieved by our approach over Orca is 11x on Stage 1, 24x on Stage 2, 32x on Stage 3, and 28x on All Stages combined. These substantial improvements highlight the efficiency of our GPU-accelerated implementation, which effectively exploits parallelism to handle large-scale transaction graphs. Furthermore, our method maintains consistent scaling performance as the graph size increases, demonstrating its robustness for graphs with millions of nodes. By leveraging a GPU-friendly design, our approach reduces the computational overhead associated with subgraph enumeration, making it feasible for practical applications involving large graphs. The results underscore the superiority of our proposed approach in terms of runtime efficiency, which is suitable for real-world spoofing detection systems.

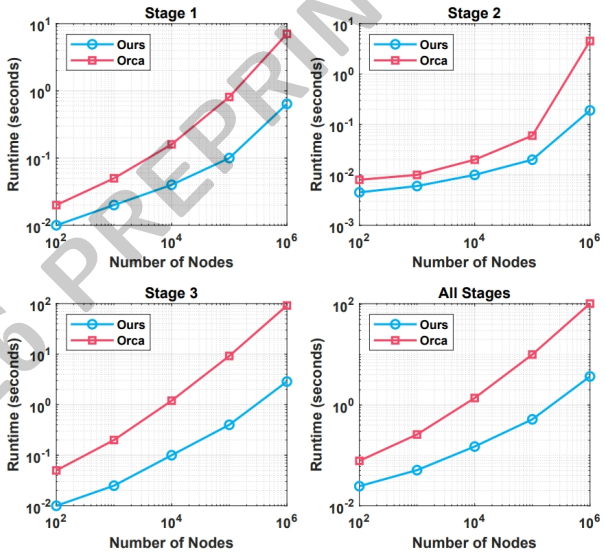

Figure 3: Runtime comparison between our GPU-friendly implementation (Ours) and the original C++ implementation (Orca) across four different stages of graphlet enumeration. The maximum speedup achieved by our method is 11x, 24x, 32x, and 28x for Stage 1, Stage 2, Stage 3, and All Stages, respectively.

4.4 Parameter Sensitivity

To examine the sensitivity of our model to its hyperparameters, we conduct a comprehensive analysis by varying key parameters including embedding size, learning rate, and threshold. Figure 4 illustrates the impact of these parameters on the model's performance across different configurations using the spoofing detection dataset. Specifically, in Figure 4(a), we observe the effect of varying the threshold parameter ranging from 0 to 0.1. The results reveal significant fluctuations in performance metrics as the threshold increases. While the AUC metric remains relatively stable and maintains high performance across different thresholds, both the F1 score and AP exhibit sharp declines once the threshold exceeds 0.1. This suggests that higher thresholds may lead to a loss of heterogeneous information crucial for accurate detection. In Figure 4(b), we analyze the model's sensitivity to hidden size

IJCAI-ECAI 2026 PREPRINT

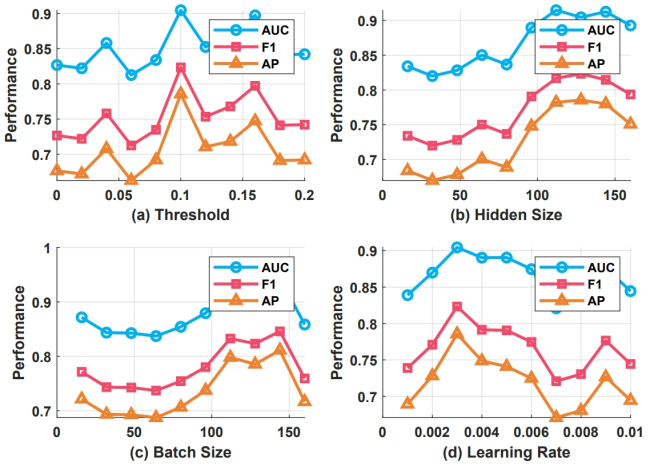

Figure 4: Parameter sensitivity analysis for the SAG^{2}M model. The optimal parameter settings were selected as the final experimental settings.

variations from 20 to 128. The performance metrics initially increase as the hidden size grows, reaching optimal values around 100-128 dimensions. Beyond this range, the F1 score and AP begin to slightly decline, indicating that while larger hidden sizes enable the model to capture more complex patterns, they also risk overfitting or diminishing returns. The trends in batch size, shown in Figure 4(c), reveal a clear peak in performance at a batch size of 128. Both the AUC and F1 scores improve significantly as the batch size increases from 50 to 128, after which performance plateaus or slightly declines. astly, Figure 4(d) evaluates the effect of learning rate adjustments. A lower learning rate of around 0.002 to 0.004 appears optimal, as indicated by higher values in all performance metrics. The AP and F1 scores decrease substantially with higher learning rates, suggesting that too large a step may lead to instability in training or skipping over optimal solutions. Based on these analyses, we choose our best parameter settings according to these results.

4.5 Case Studies

In this section, we perform an in-depth case analysis using the spoofing detection dataset to evaluate the influence of node orbit features on detecting fraudulent activities. By examining attention values associated with different node orbits, we identify key subgraph patterns that distinguish spoofing behaviors from legitimate transactions. Figure 5(a) illustrates the distribution of attention weights across various node orbits. Each bar represents the importance assigned to a specific node orbit in the detection process, with higher weights indicating greater significance. Figure 5(b) provides visualizations of subgraph structures, including triangles and 4-cliques. Notably, node orbits associated with 4-cliques and triangles exhibit substantially higher attention weights compared to other structures. This suggests that these subgraph patterns play a pivotal role in identifying intricate structural anomalies characteristic of conspiracy spoofing behaviors. The prominence of these orbits underscores their importance as critical indicators, reflecting the underlying complexity

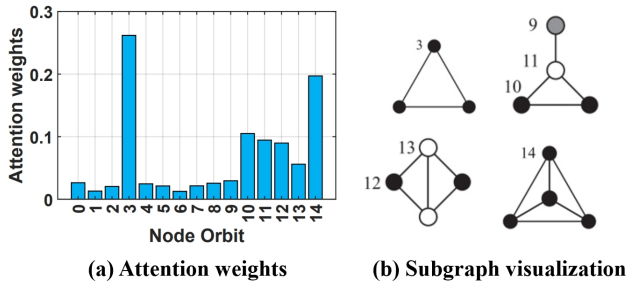

Figure 5: Visualization of node orbit attention values in the spoof detection dataset. Higher attention values are assigned to 4-cliques and triangle-related subgraph patterns.

of fraudulent behavior within trading networks. The results demonstrate the potential of SAG^{2}M to enhance fraud detection accuracy by leveraging complex graph topology structures. By focusing on these key subgraph patterns, our approach offers significant advantages over traditional graph-based detection methods that may overlook such nuanced structural features in detecting sophisticated fraudulent activities such as conspiracy spoofing.

5 Conclusion

In this paper, we proposed a structure augmented generative graph mode for spoofing detection task, which is a critical data mining task in financial trading. The structure distribution shifts between new spoofing transactions and historical transactions often limit the utility of graph-based detection methods. To overcome these challenges, we introduced the Structure-Augmented Generative Graph Model (SAG^{2}M), which leverages a novel preprocessing and subgraph frequency-enhanced detection approach. Our method focuses on extracting subgraph pattern frequencies among node neighbors. Considering real-world implementation, we use a parallelizable enumeration to efficiently compute the node orbit data. Then, the generative encoding of structure information allows for the capture of intricate structural patterns to provide a comprehensive representation of each transaction. Furthermore, the temporal and heterogeneous aggregation scheme effectively collects and analyzes neighborhood node interactions to identify conspiracy spoofing patterns. The experimental results, conducted on diverse datasets such as Amazon, YelpChi, and T-Finance, demonstrate that SAG^{2}M consistently outperforms existing models in detection accuracy. Our findings underscore the superiority of SAG^{2}M in enhancing the detection of spoofing transactions by harnessing complex graph structures and sophisticated aggregation techniques. The case study on conspiracy spoofing detection further validates the model's efficacy, illustrating its potential for real-world applications in financial fraud detection. Future work could explore extending the scalability of node orbit counting algorithms where relational data play a critical role in spoofing detection.

IJCAI-ECAI 2026 PREPRINT

Acknowledgments

This work is sponsored by the National Science Fundation of China (62522213, 62472317), Shanghai Pujiang Program (25PJA139), and the Fundamental Research Funds for the Central Universities.

References

[Arora and Bhatia, 2020] Shefali Arora and MP S Bhatia. Fingerprint spoofing detection to improve customer security in mobile financial applications using deep learning. *Arabian journal for science and engineering*, 45(4):2847–2863, 2020.

[Bhattacharyya *et al.*, 2011] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, 2011.

[Cao *et al.*, 2014] Yi Cao, Yuhua Li, Sonya Coleman, Ammar Belatreche, and Thomas Martin McGinnity. Adaptive hidden markov model with anomaly states for price manipulation detection. *IEEE transactions on neural networks and learning systems*, 26(2):318–330, 2014.

[Chanthati, 2024] NSR Chanthati. How the power of machine–machine learning, data science and nlp can be used to prevent spoofing and reduce financial risks. *Global Journal of Engineering and Technology Advances*, 20(2):100–119, 2024.

[Charoenwong *et al.*, 2023] Ben Charoenwong, Robert M Kirby, and Jonathan Reiter. Risk-free interest rates in decentralized finance. In *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*, pages 466–473. IEEE, 2023.

[Charoenwong *et al.*, 2024] Ben Charoenwong, Zachary T Kowaleski, Alan Kwan, and Andrew G Sutherland. Regtech: Technology-driven compliance and its effects on profitability, operations, and market structure. *Journal of Financial Economics*, 154:103792, 2024.

[Chen *et al.*, 2020] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395, 2020.

[Cheng *et al.*, 2020] Dawei Cheng, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. Graph neural network for fraud detection via spatial-temporal attention. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3800–3813, 2020.

[Cheng *et al.*, 2025] Dawei Cheng, Yao Zou, Sheng Xiang, and Changjun Jiang. Graph neural networks for financial fraud detection: a review. *Frontiers of Computer Science*, 19(9):1–15, 2025.

[Dou *et al.*, 2020] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[Du *et al.*, 2024] Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. Financial sentiment analysis: Techniques and applications. *ACM Comput. Surv.*, 56(9), April 2024.

[Duan *et al.*, 2024] Mingjiang Duan, Tongya Zheng, Yang Gao, Gang Wang, Zunlei Feng, and Xinyu Wang. Dga-gnn: Dynamic grouping aggregation gnn for fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11820–11828, 2024.

[Erdos and Rényi, 1984] Paul L. Erdos and Alfréd Rényi. On the evolution of random graphs. *Transactions of the American Mathematical Society*, 286:257–257, 1984.

[Fu *et al.*, 2016] Kang Fu, Dawei Cheng, Yi Tu, and Liqing Zhang. Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*, 2016.

[Hočevar and Demšar, 2014] Tomaž Hočevar and Janez Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 2014.

[Kang *et al.*, 2023] Le Kang, Tai-Jiang Mu, and Xiaodong Ning. Conspiracy spoofing orders detection with transformer-based deep graph learning. In *International Conference on Advanced Data Mining and Applications*, 2023.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[Kumar *et al.*, 2019] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[Leangarun *et al.*, 2016] Teema Leangarun, Poj Tangamchit, and Suttipong Thajchayapong. Stock price manipulation detection based on mathematical models. *International journal of trade, economics and finance*, 7(3):81–88, 2016.

[Li *et al.*, 2024] Enxia Li, Jin Ouyang, Sheng Xiang, Lu Qin, and Ling Chen. Relation-aware heterogeneous graph neural network for fraud detection. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 240–255. Springer, 2024.

[Liu *et al.*, 2021] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, WWW '21, page 3168–3177, New York, NY, USA, 2021. Association for Computing Machinery.

[Liu *et al.*, 2023a] Charles Z Liu, Sheng Xiang, Dawei Cheng, Junyi Liu, Ying Zhang, and Lu Qin. Transformed graph attention for credit rating. In *2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA)*, pages 1011–1016. IEEE, 2023.

IJCAI-ECAI 2026 PREPRINT

[Liu *et al.*, 2023b] Yajing Liu, Zhengya Sun, and Wensheng Zhang. Improving fraud detection via hierarchical attention-based graph neural network. *Journal of Information Security and Applications*, 72:103399, 2023.

[Ma *et al.*, 2024] Jiacheng Ma, Sheng Xiang, Qiang Li, Liangyu Yuan, Dawei Cheng, and Changjun Jiang. Parallel graph learning with temporal stamp encoding for fraudulent transactions detections. *IEEE Transactions on Big Data*, 2024.

[McAuley and Leskovec, 2013] Julian McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[Mendonça and De Genaro, 2020] Luisa Mendonça and Alan De Genaro. Detection and analysis of occurrences of spoofing in the brazilian capital market. *Journal of Financial Regulation and Compliance*, 28(3):369–408, 2020.

[Motie and Raahemi, 2024] Soroor Motie and Bijan Raahemi. Financial fraud detection using graph neural networks: A systematic review. *Expert Systems with Applications*, 240:122156, 2024.

[Rayana and Akoglu, 2015] Shebuti Rayana and Leman 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*, KDD '15, page 985–994, New York, NY, USA, 2015. Association for Computing Machinery.

[See *et al.*, 2024] Kenneth See, Hanzholah Shobri, and Nicholas Garcia. Pssimpy: A design science approach to constructing and implementing a large-value payment system simulator. 2024.

[Tang *et al.*, 2022a] Jianheng Tang, Jiajin Li, Zi-Chao Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, 2022.

[Tang *et al.*, 2022b] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, pages 21076–21089. PMLR, 2022.

[Tao *et al.*, 2022] Xuan Tao, Andrew Day, Lan Ling, and Samuel Drapeau. On detecting spoofing strategies in high-frequency trading. *Quantitative Finance*, 22(8):1405–1425, 2022.

[Vatter *et al.*, 2023] Jana Vatter, Ruben Mayer, and Hans-Arno Jacobsen. The evolution of distributed systems for graph neural networks and their origin in graph processing and deep learning: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[Wang and Charoenwong, 2024] Yiyao Wang and Ben Charoenwong. The bigtech-bank lending ecosystem. *Available at SSRN 4785114*, 2024.

[Wang and Wellman, 2017] Xintong Wang and Michael Paul Wellman. Spoofing the limit order book: An agent-based model. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Wang *et al.*, 2023] Yuchen Wang, Jinghui Zhang, Zhengjie Huang, Weibin Li, Shikun Feng, Ziheng Ma, Yu Sun, Dianhai Yu, Fang Dong, Jiahui Jin, et al. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023*, pages 406–416, 2023.

[Whitrow *et al.*, 2009] Christopher Whitrow, David J. Hand, Piotr Juszczak, David John Weston, and Niall M. Adams. Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, 18:30–55, 2009.

[Wu *et al.*, 2023] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. Energy-based out-of-distribution detection for graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[Wu *et al.*, 2024] Jiasheng Wu, Xin Liu, Dawei Cheng, Ouyang Yi, Xian Wu, and Yefeng Zheng. Safeguarding fraud detection from attacks: A robust graph learning approach. In *International Joint Conference on Artificial Intelligence*, 2024.

[Xiang *et al.*, 2023] Sheng Xiang, Mingzhi Zhu, Dawei Cheng, Enxia Li, Ruihui Zhao, Yi Ouyang, Ling Chen, and Yefeng Zheng. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *AAAI*, 2023.

[Yu *et al.*, 2024] Hang Yu, Zhengyang Liu, and Xiangfeng Luo. Barely supervised learning for graph-based fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16548–16557, 2024.

[Zhao *et al.*, 2020] Tong Zhao, Chuchen Deng, Kaifeng Yu, Tianwen Jiang, Daheng Wang, and Meng Jiang. Error-bounded graph anomaly loss for gnns. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1873–1882, New York, NY, USA, 2020. Association for Computing Machinery.

[Zhuo *et al.*, 2024] Wei Zhuo, Zemin Liu, Bryan Hooi, Bingsheng He, Guang Tan, Rizal Fathony, and Jia Chen. Partitioning message passing for graph fraud detection. In *The Twelfth International Conference on Learning Representations*, 2024.

[Zou *et al.*, 2024] Yao Zou, Sheng Xiang, Qijun Miao, Dawei Cheng, and Changjun Jiang. Subgraph patterns enhanced graph neural network for fraud detection. In *International Conference on Database Systems for Advanced Applications*, pages 375–384, 2024.

IJCAI-ECAI 2026 PREPRINT