HGT-FD: Hypergraph Transformer for Fraud Detection

Yintao Cai[1] **Yunjiong Liu**[2] **Zelong Yang**[2] **Shuyang Fang**[1] **Xiaoping Min**[1,*]
[1] School of Informatics, Xiamen University, China
[2] Institute of Artificial Intelligence, Xiamen University, China
[*] Corresponding author. Email: mxp@xmu.edu.cn

Abstract

Graph-based fraud detection aims to identify anomalous patterns or fraudulent behaviors in graph-structured data, playing a crucial role across various domains. However, traditional models primarily focus on simple node-to-node message passing, which limits the integration of multi-node features and overlooks long-range dependencies in fraud detection. Hyperedge features in hypergraphs, as an integration of node features, can convey and aggregate multi-node characteristics, providing environmental information for the model to detect fraudulent nodes. To this end, we propose HGT-FD for hypergraph-based fraud detection. Specifically, we design a Hypergraph Transformer model that can directly employ hyperedge features for fraud detection, utilizing a co-attention mechanism to generate node representations. Furthermore, structural encoding and positional encoding are proposed to enhance the model's perception of hypergraph structures, enabling the model to capture more complex high-order structural relationships. Extensive experimental results on three fraud detection datasets demonstrate that the proposed method exhibits significant advantages over baselines in fraud detection.

1 INTRODUCTION

With the rapid growth of the digital economy and the increasing scale of online information, fraud activities in real-world scenarios are becoming more prevalent. Malicious reviews in e-commerce platforms (Rayana and Akoglu, 2015), fraudulent transactions in the financial sector (Can et al., 2020), and other deceptive behaviors have posed significant challenges and financial threats (Pan et al., 2011; Kagias et al., 2022). Traditional fraud detection methods based on rule-based systems (Sánchez et al., 2009; Gianini et al., 2020) or statistical techniques (Marinescu, 2004) often fail to adapt to dynamic and evolving fraud strategies (Nakitende et al., 2024; Kotagiri and Yada, 2024). Once fraudsters become aware of detection rules, it becomes easier for them to bypass monitoring systems.

Deep learning, especially Graph Neural Networks (GNNs) (Sharifani and Amini, 2023; Dwivedi et al., 2023), has become a promising approach for fraud detection by modeling interactions in graph-structured data (Veličković, 2023) and jointly exploiting node and structural features (Li et al., 2023b). Yet sophisticated fraud strategies—such as linking to many benign nodes or mimicking neighbors—degrade GNN effectiveness (Chatterjee et al., 2024; Bei et al., 2023). Recent advances address this via refined unsupervised objectives (Ma et al., 2024), heterophily exploitation (Pan et al., 2025), curvature-based geometry (Grover et al., 2025), and federated detection for privacy (Liu et al., 2025).

Despite these advances, these methods still lack an explicit mechanism to construct multi-node environmental features, which are crucial to capture the contextual cues that fraudsters manipulate. Recent studies have highlighted the importance of environmental context in fraud detection (Xu et al., 2024), where benign nodes dominate and serve as primary sources of global feature information, while fraudulent nodes often deviate from this context. In this work, we bridge this gap by constructing hyperedge features as explicit environmental carriers and injecting them into node representations through a hypergraph transformer with cross-attention and structural/positional encodings, thereby enabling more precise fraud detection in high-order relational settings.

Hypergraphs extend traditional graphs by allowing hy-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

peredges to connect multiple nodes, enabling richer multi-entity interactions (Pei et al., 2024; Kim et al., 2024; Feng et al., 2019). Hypergraph neural networks (HGNNs) capture higher-order relations and aggregate features across nodes for more expressive representations (Wang et al., 2024b). Meanwhile, Transformer-based attention (Vaswani, 2017; Han et al., 2022; Min et al., 2022) provides flexible global aggregation and has been applied to graph tasks (Niu et al., 2021). Recent attempts such as HyperGT (Liu et al., 2024) incorporate self-attention into hypergraph learning but still face high cost and underuse of hyperedge features.

Since hyperedges naturally integrate features from connected (mostly benign) nodes, they encode environmental context against which fraudulent nodes can be contrasted (Wang et al., 2024a; Lee et al., 2024). Motivated by this, we propose **HGT-FD**, which constructs hyperedge features as environmental representations and employs node—hyperedge cross-attention with structural and positional encodings to enhance anomaly detection in complex relational settings.

Our key contributions can be summarized as follows:

- We provide a theoretical guarantee by presenting propositions and their detailed proofs, demonstrating the effectiveness of hypergraph structures in fraud detection.
- We propose a novel framework that adapts the Hypergraph Transformer architecture specifically for fraud detection, leveraging benign-seeded random walks to construct environmental contexts.
- We design a cross-attention mechanism that enables high-order interactions between nodes and hyperedges, allowing the model to incorporate environmental context into node representations effectively.
- We conduct extensive experiments on three benchmark fraud datasets, demonstrating that our method achieves superior performance and computational efficiency compared to state-of-the-art GNN-based approaches.

2 RELATED WORK

2.1 Hypergraphs and Hypergraph Neural Networks.

A hypergraph is an important concept in graph theory. Unlike an ordinary graph, where an edge can only connect two nodes, a hypergraph allows an edge to connect more than two nodes. This hyperedge connection relationship can better represent relationships among multiple nodes. Currently, there are many neural networks designed to handle data structured as hypergraphs (Khan et al., 2023). These neural networks typically aggregate node features into hyperedge features through certain message-passing methods, such as averaging or summing, and update to obtain new hyperedge features. Subsequently, they update the node features based on the hyperedge features (Feng et al., 2019). Regarding hypergraph-based anomaly detection, statistical methods like Lee et al. (2022) and Chun et al. (2024) focus on structural properties or ranking efficiency. Recently, Huang et al. (2025) utilized dual hypergraph transformations for edge co-occurrences. In contrast, HGT-FD is an end-to-end framework focusing on node-level environmental construction.

2.2 Transformers in Hypergraph Structures.

Initially designed for NLP tasks due to their powerful self-attention mechanism (Gillioz et al., 2020), Transformers have been extended to graph-structured data by encoding graph structures to better extract node features. However, their application to hypergraphs remains limited (Li et al., 2023a), as hypergraph incidence matrices cannot directly utilize Laplacian eigenvectors for structural encoding. Methods like HyperGT (Liu et al., 2024) leverage self-attention for hypergraph data but face efficiency issues due to the concatenation of node and hyperedge features, and underutilize hyperedge information. While Li et al. (2023a) focuses on intra-hyperedge interactions, HGT-FD leverages node-hyperedge cross-attention to capture macro-level contrast. Others such as SHT (Xia et al., 2022) are tailored to heterogeneous graphs in recommender systems, limiting their suitability for homogeneous settings like fraud detection.

Recently, Haghighi et al. (2025) proposed TROPICAL, a hypergraph transformer for multi-relational fraud detection. Different from TROPICAL's focus on heterogeneous relation fusion, HGT-FD specifically targets homogeneous camouflage via benign-seeded environmental construction. Additionally, Duan et al. (2025) explored global attribute patterns, which complements our local topological focus.

2.3 Fraud Detection Based on Graphs.

Graph-based fraud detection identifies fraudulent nodes within graph-structured data, and numerous GNN variants have been proposed across domains like finance and social media. Early methods such as H2-FDetector (Shi et al., 2022) and DiG-In-GNN (Zhang et al., 2024) improve detection by distinguishing heterogeneous connections or combining contrastive learning with domain selection. More recent approaches expand along several directions: SmoothGNN (Dong et al., 2025b) leverages smoothing patterns as discriminative signals for unsupervised anomaly detection; APGNN (Wang et al., 2024c) and CMR-GNN (Han et al., 2025) enhance fine-grained behavioral modeling and address long-tail effects; SpaceGNN (Dong et al., 2025a) explores non-Euclidean representations to capture diverse anomaly patterns; and NSReg (Wang et al., 2025) together with AnomalyGFM (Qiao et al., 2025) improve generalization in open-set and few-shot scenarios. Beyond these, GeneralDyG (Yang et al., 2025) focuses on dynamic graphs, while DiffGAD (Li et al., 2024) and MonTi (Choi et al., 2025) investigate generative detection and adversarial robustness. UBGOLD (Wang et al., 2024d) provides a unified benchmark for graph-level anomaly and OOD detection, and H2IDE (Fu et al., 2024) disentangles homophily and heterophily in multi-relation graphs. Despite their progress, these models still rely primarily on pairwise or latent representations, leaving the explicit construction of multi-node environmental features underexplored.

3 METHODOLOGY

In this section, we introduce the effectiveness and advantages of hypergraph structures compared to ordinary graphs in fraud detection, followed by a detailed description of our Hypergraph Transformer for Fraud Detection (HGT-FD) model. The overall framework is illustrated in Figure 1. A complete report of the theoretical propositions on why hypergraphs benefit fraud detection is provided in the supplementary material, including but not limited to their ability to aggregate multi-node environmental context, enhance anomaly sensitivity by contrasting fraudulent nodes against benign-dominated hyperedges, and directly model higher-order relations that ordinary graphs can only approximate.

3.1 Hypergraph Construction

Given an input graph $G_0 = (V, \mathcal{E}_0)$ with node feature matrix $X = \{x_i\}_{i \in V}$, we transform it into a hypergraph $\mathcal{G}_H = (V, \mathcal{E})$, where each hyperedge is generated by a fixed-length random walk. Let $V = V_{benign} \cup V_{fraud}$ where V_{benign} and V_{fraud} denote the sets of benign and fraudulent nodes, respectively. We assume a typical fraud detection setting where $|V_{benign}| \gg |V_{fraud}|$, meaning benign nodes dominate the distribution. Since benign nodes significantly outnumber fraudulent nodes, we aim to ensure that the constructed hyperedges are dominated by benign nodes, thereby maximizing the discrepancy between fraudulent nodes and their surrounding environments. Different from conventional methods, we restrict the starting point of each random walk to nodes labeled as benign in the training set, and we adopt a feature-similarity-based transition strategy during the walk.

Specifically, at each iteration we randomly select one benign node from the training set that has not yet been included in any hyperedge as the starting node $v^{(0)}$. From this node, we perform a random walk of fixed length L, producing a node sequence $\{v^{(0)}, v^{(1)}, \ldots, v^{(L)}\}$. The set of distinct nodes visited during the walk forms a hyperedge e, which is then added to the hyperedge set E. This process is repeated until every node is covered by at least one hyperedge.

During the walk, the transition probability is defined based on feature similarity:

$$P\left(v_j \mid v^{(t)}, N_{v^{(t)}}\right) = \frac{\exp\left(-\tau \|x_{v^{(t)}} - x_j\|_2^2\right)}{\sum\limits_{k \in N_{v^{(t)}}} \exp\left(-\tau \|x_{v^{(t)}} - x_k\|_2^2\right)}, \tag{1}$$

where $N_{v^{(t)}}$ denotes the neighbors of the current node $v^{(t)}$, and $\tau > 0$ controls the sharpness of the distribution. Unlike unsupervised random walks like node2vec (Grover and Leskovec, 2016) which explore general structural equivalence, our strategy is strictly benign-seeded and similarity-biased. This is explicitly designed to filter out heterophilic noise and construct 'purified' environments for contrastive detection. A larger τ makes the distribution more peaked, strongly favoring the most similar neighbors, while a smaller τ smooths the distribution, allowing for more exploratory transitions. This transition mechanism biases the walk toward selecting feature-similar nodes, which encourages that the resulting hyperedges are predominantly influenced by benign nodes.

After obtaining a hyperedge e, we further design a degree-aware aggregation scheme to compute its feature representation x_e. This design ensures that high-degree nodes contribute more to the hyperedge representation, while extremely high-degree nodes are prevented from completely dominating it. To achieve this, we normalize node degrees within each hyperedge to maintain comparability across different hyperedges, and define:

$$x_e \;=\; \sum_{i \in e} \frac{\min\left\{ \left(\frac{d_i}{\bar{d}_e} + \epsilon_{\deg}\right)^{\gamma_{\deg}},\, \eta_{\deg} \right\}}{\sum\limits_{j \in e} \min\left\{ \left(\frac{d_j}{\bar{d}_e} + \epsilon_{\deg}\right)^{\gamma_{\deg}},\, \eta_{\deg} \right\}}\, x_i, \tag{2}$$

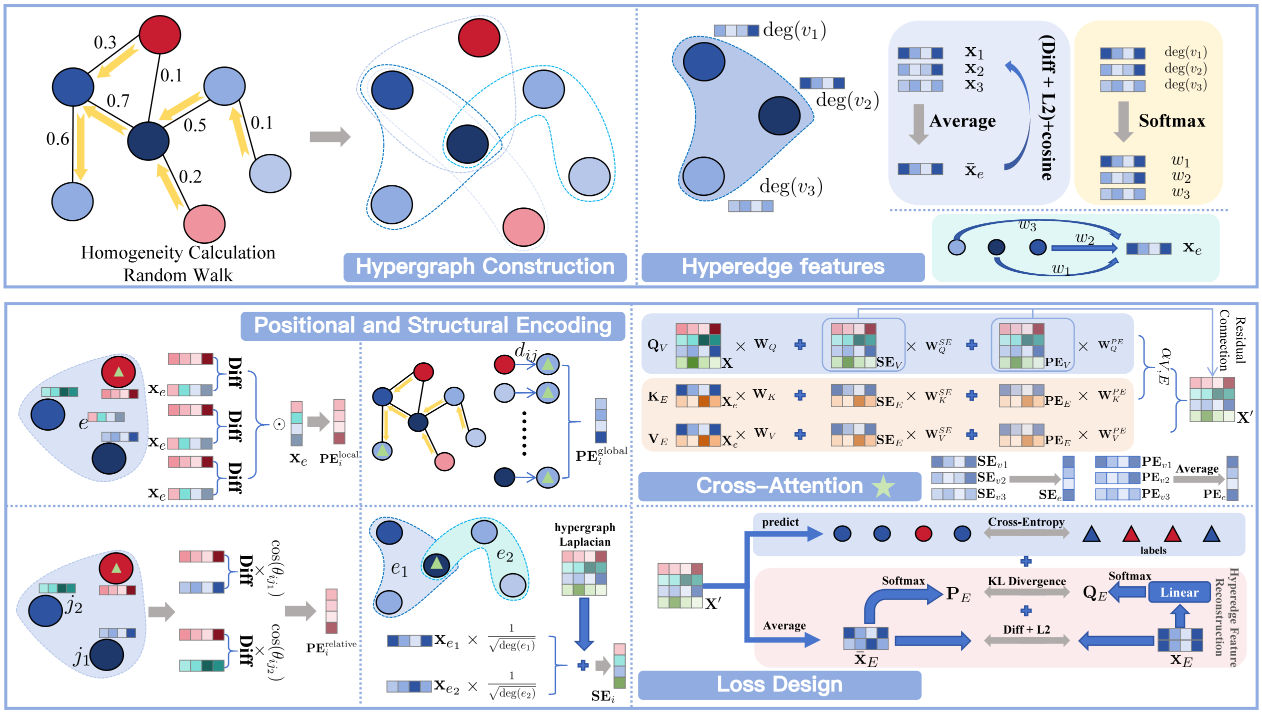


Figure 1: Overall Framework of HGT-FD. The model first constructs benign-dominated hyperedges via similarity-biased random walks (Top-Left). Then, it generates hyperedge features using degree-aware aggregation (Top-Right). Finally, a Cross-Attention Transformer (Bottom) fuses node and hyperedge features, incorporating Deviation-Aware Relative Positional Encodings to detect anomalies through environmental contrast.

where

$$\bar{d}_e = \frac{1}{|e|} \sum_{j \in e} d_j, \tag{3}$$

where d_i denotes the degree of node v_i in the original graph, $\gamma_{\text{deg}} \geq 0$ controls the influence of node degree, $\epsilon_{\text{deg}} > 0$ ensures numerical stability, and $\eta_{\text{deg}} > 0$ serves as an upper bound for truncation.

Such a construction ensures that almost every hyperedge provides a benign-dominated environment, which is crucial for enhancing the separability between fraudulent and benign nodes.

3.2 Positional and Structural Encoding

Positional encoding is introduced to inject node-position information and mitigate the permutation invariance of the Transformer architecture, thereby enabling the model to better exploit positional relations among nodes. In fraud detection, nodes should primarily attend to *local* contextual signals rather than absolute global coordinates. As discussed above, local context helps the model identify fraudulent nodes more accurately, and deviations from the benign majority within the immediate environment are often more indicative of anomalies. Accordingly, we adopt a *local-first* design in which local/relative positional signals constitute the core, while a lightweight global term serves as a complement. This prevents over-reliance on global structure and maintains computational efficiency, while improving sensitivity to anomalous behavior.

To control the strength of this local-first bias, we fuse three types of positional encodings using nonnegative hyperparameters:

$$\mathbf{PE}_i = \lambda_{\text{loc}}\, \mathbf{PE}_i^{\text{local}} + \lambda_{\text{glob}}\, \mathbf{PE}_i^{\text{global}} + \lambda_{\text{rel}}\, \mathbf{PE}_i^{\text{relative}} \tag{4}$$

where $\lambda_{\text{loc}}, \lambda_{\text{glob}}, \lambda_{\text{rel}} \geq 0$ and λ_{loc} and λ_{rel} are set comparatively large, while λ_{glob} is kept small or even zero to improve efficiency. We report the performance under different settings of these three hyperparameters in the experiments.

The local positional encoding $\mathbf{PE}_i^{\text{local}}$ is based on the local structure of the node in the hyperedge:

$$\mathbf{PE}_i^{\text{local}} = \frac{1}{|\mathcal{E}_i|} \sum_{e \in \mathcal{E}_i} \mathbf{W}_{\text{local}} \cdot (\mathbf{x}_e \odot \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_e\|_2^2}{2\sigma_{\text{local}}^2})) \tag{5}$$

where $\mathcal{E}_i$ is the set of hyperedges containing node v_i, $\mathbf{W}_{\text{local}}$ is a learnable weight matrix, and $\odot$ denotes element-wise multiplication.

The global positional encoding $\mathbf{PE}_i^{\text{global}}$ is based on the global position of the node in the hypergraph:

$$\mathbf{PE}_i^{\text{global}} = \mathbf{W}_{\text{global}} \cdot \left(\sum_{j=1}^{|V|} \frac{\exp(-\beta_{\text{dist}} \cdot d_{ij})}{\sum_{k=1}^{|V|} \exp(-\beta_{\text{dist}} \cdot d_{ik})} \cdot \mathbf{x}_j \right) \tag{6}$$

where d_{ij} is the shortest path distance between nodes v_i and v_j in the hypergraph, and β is a distance decay parameter.

The relative positional encoding $\mathbf{PE}_i^{\text{relative}}$ is defined based on the relative relationship between the node and other nodes in the same hyperedge:

$$\mathbf{PE}_i^{\text{relative}} = \mathbf{W}_{\text{relative}} \cdot \left(\frac{1}{|\mathcal{E}_i|} \sum_{e \in \mathcal{E}_i} \frac{1}{|e|-1} \cdot \sum_{\substack{j \in e \\ j \neq i}} \frac{\mathbf{x}_i - \mathbf{x}_j}{\|\mathbf{x}_i - \mathbf{x}_j\|_2 + \epsilon_{\text{rel}}} \cdot \cos(\theta_{ij}) \right) \tag{7}$$

where j denotes other nodes in hyperedge e that also contain v_i, ϵ_{rel} is a numerical stability parameter specific to relative positional encoding that prevents division by zero and stabilizes the sensitivity to nearly identical node features, and $\cos(\theta_{ij})$ is the cosine similarity between node features.

We provide theoretical proofs for the positional information captured by the above encodings in the Appendix.

Structural Encoding To encode the structural information of the hypergraph, we design a structural encoding based on the hypergraph Laplacian matrix. For node v_i, its structural encoding $\mathbf{SE}_i$ is defined as:

$$\mathbf{SE}_i = \mathbf{W}_{\text{struct}} \cdot (\mathbf{L}_{\text{hyper}} \cdot \mathbf{x}_i + \sum_{e \in \mathcal{E}_i} \frac{1}{\sqrt{\deg(e)}} \cdot \mathbf{x}_e) \tag{8}$$

where $\mathbf{L}_{\text{hyper}}$ is the hypergraph Laplacian matrix, defined as:

$$\mathbf{L}_{\text{hyper}} = \mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \tag{9}$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ are the node degree matrix and hyperedge degree matrix, respectively. This formulation follows the symmetric normalized Laplacian proposed by Zhou et al. (2006), ensuring numerical stability and symmetric properties for spectral analysis.

This structural encoding design preserves the topological information of the hypergraph, and the inclusion of node degrees ensures that the structural encoding can balance the influence of hyperedges with different degrees. Importantly, when hypergraph structures are similar, the structural encodings are also similar. We provide the detailed proofs in the Appendix.

3.3 Hypergraph Cross-Attention Mechanism

In this work, we improve the cross-attention mechanism to fully integrate the structural and positional encodings of nodes and hyperedges, enhancing the model's perception of hypergraph structure and spatial relationships. The specific approach is as follows:

3.3.1 Query-Key-Value Computation with Structure and Position Awareness

Node and hyperedge features are linearly transformed after being combined with structural and positional encodings:

$$\begin{aligned} \mathbf{Q}_v &= \mathbf{x}_v \mathbf{W}_Q + \mathbf{SE}_v \mathbf{W}_Q^{SE} + \mathbf{PE}_v \mathbf{W}_Q^{PE}, \\ \mathbf{K}_e &= \mathbf{x}_e \mathbf{W}_K + \mathbf{SE}_e \mathbf{W}_K^{SE} + \mathbf{PE}_e \mathbf{W}_K^{PE}, \\ \mathbf{V}_e &= \mathbf{x}_e \mathbf{W}_V + \mathbf{SE}_e \mathbf{W}_V^{SE} + \mathbf{PE}_e \mathbf{W}_V^{PE}. \end{aligned} \tag{10}$$

where $\mathbf{SE}_e$ is the structural encoding of hyperedge e, defined as the mean of the structural encodings of its constituent nodes, i.e., $\mathbf{SE}_e = \frac{1}{|e|} \sum_{i \in e} \mathbf{SE}_i$, and $\mathbf{PE}_e$ is the positional encoding of hyperedge e, defined as the mean of the positional encodings of its constituent nodes, i.e., $\mathbf{PE}_e = \frac{1}{|e|} \sum_{i \in e} \mathbf{PE}_i$, and all projection matrices for queries, keys, and values (including their structural and positional variants) are learnable parameters.

3.3.2 Node-Hyperedge Cross-Attention Score Calculation

For each node v and its associated hyperedge $e \in \mathcal{E}_v$, the attention score is calculated as:

$$\alpha_{v,e} = \frac{\exp\left(\frac{\mathbf{Q}_v^T \mathbf{K}_e}{\sqrt{d}} + \beta \cdot \text{sim}(\mathbf{x}_v, \mathbf{x}_e)\right)}{\sum_{e' \in \mathcal{E}_v} \exp\left(\frac{\mathbf{Q}_v^T \mathbf{K}_{e'}}{\sqrt{d}} + \beta \cdot \text{sim}(\mathbf{x}_v, \mathbf{x}_{e'})\right)} \tag{11}$$

where $\text{sim}(\mathbf{x}_v, \mathbf{x}_e) = \frac{\mathbf{x}_v^T \mathbf{x}_e}{\|\mathbf{x}_v\|_2 \|\mathbf{x}_e\|_2}$, and β is a learnable temperature parameter. This formula computes the normalized cross-attention score between node v and its associated hyperedge e, comprehensively considering structure, position, and feature similarity.

3.3.3 Complexity Analysis

Compared with self-attention, which requires all-to-all interactions among $|V|$ nodes and incurs a quadratic complexity of $\mathcal{O}(|V|^2 d)$, our cross-attention computes interactions between each node and all hyperedges with a complexity of $\mathcal{O}(|V||E|d)$. Since the number of hyperedges $|E|$ is typically much smaller than the number of nodes $|V|$ in practice, we have $|V||E| \ll |V|^2$, making our design significantly more efficient in both computation and memory.

3.3.4 Node Feature Update

The new feature of node v is:

$$\mathbf{x}'_v = \lambda_{\text{fuse}} \cdot (\mathbf{x}_v + \mathbf{SE}_v + \mathbf{PE}_v) + (1 - \lambda_{\text{fuse}}) \cdot \sum_{e \in \mathcal{E}_v} \alpha_{v,e} \cdot \mathbf{V}_e \tag{12}$$

where $\lambda_{\text{fuse}} \in [0, 1]$ is a fusion coefficient hyperparameter that balances the contribution between the node's own feature (together with structural and positional encodings) and the aggregated hyperedge information.

This mechanism enhances the expressive power through structural and positional encodings. The correlation between nodes and hyperedges depends not only on feature similarity but also on the matching of structure and position, thereby improving the model's perception of hypergraph structure and spatial relationships and facilitating efficient node classification.

3.4 Loss Design

In this section, the final node representations, obtained by fusing node features, structural encodings, hyperedge features, and positional encodings through the cross-attention mechanism, serve as the basis for classification and loss computation.

3.4.1 Multi-Task Learning Framework:

To fully utilize hyperedge feature information in the hypergraph structure and improve fraud detection performance, we design a multi-task learning framework that combines node classification loss and hyperedge feature reconstruction loss. This loss function can effectively leverage hyperedge features as environmental information while maintaining the model's sensitivity to fraudulent nodes. The overall loss function during training is defined as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{node}} \mathcal{L}_{\text{node}} + \lambda_{\text{recon}} \mathcal{L}_{\text{recon}} \tag{13}$$

where λ_{node} and λ_{recon} are the weights for each subtask in the loss function, balancing the importance of node classification loss $\mathcal{L}_{\text{node}}$ and hyperedge feature reconstruction loss $\mathcal{L}_{\text{recon}}$. The node classification loss $\mathcal{L}_{\text{node}}$ is computed using the cross-entropy loss function.

3.4.2 Hyperedge Feature Reconstruction Loss

Simple reconstruction error can only ensure that hyperedge features are consistent with node features at the mean level, but not at the distribution level. By minimizing the KL divergence, hyperedge features can be encouraged to not only approach the mean of the node set numerically but also capture the distributional information of node features, enhancing the expressiveness of hyperedge features for environmental information and improving the model's sensitivity and discriminative power for anomalous nodes. Therefore, to ensure that hyperedge features effectively represent the environmental information of their constituent nodes, we design the hyperedge feature reconstruction loss as follows:

$$\mathcal{L}_{\text{recon}} = \frac{1}{|E|} \sum_{e \in E} \|\mathbf{x}_e - \bar{\mathbf{x}}_e\|_2^2 + \alpha_{\text{recon}} \cdot \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e) \tag{14}$$

where $\bar{\mathbf{x}}_e = \frac{1}{|\mathcal{V}_e|} \sum_{v \in \mathcal{V}_e} \mathbf{x}_v$ and the first term is the reconstruction error, ensuring that hyperedge features can reconstruct the mean feature of their constituent nodes. The second term is the KL divergence, weighted by α_{recon}, which further constrains the distributional expressiveness of hyperedge features so that the node distribution $\mathbf{Q}_e$ predicted from hyperedge features better approximates the true node feature distribution $\mathbf{P}_e$ within the hyperedge.

Specifically, $\mathbf{P}_e$ is the true distribution of node features within hyperedge e, defined as:

$$\mathbf{P}_e = \text{softmax}\left(\frac{1}{|\mathcal{V}_e|} \sum_{v \in \mathcal{V}_e} \mathbf{x}_v\right) \tag{15}$$

$\mathbf{Q}_e$ is the node distribution predicted from hyperedge features, defined as:

$$\mathbf{Q}_e = \text{softmax}(\mathbf{W}_{\text{recon}} \cdot \mathbf{x}_e + \mathbf{b}_{\text{recon}}) \tag{16}$$

This loss design ensures numerical stability, training stability, feature utilization, and anomaly sensitivity in the loss computation for hypergraph-based fraud detection. It effectively leverages hyperedge feature information, assigns higher weights to nodes that differ significantly from hyperedge features (which are more likely to be fraudulent), and provides an effective optimization objective for the model. We provide the detailed proofs in the Appendix.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets and Baseline Methods. We employed three commonly used datasets in the GFD domain: Yelpchi, Amazon, and T-Finance. The first two are multi-relational datasets from the well-known online review platforms Yelpchi and Amazon musical instrument product reviews, respectively. The last one is a single-relational dataset from the economic and financial domain, where nodes represent accounts and edges represent transaction relationships. We compared our model with classical GNNs and state-of-the-art fraud detection models, including GCN(Kipf and Welling, 2016), H^2-FDetector(Shi et al., 2022), GTAN(Xiang et al., 2023), DiG-In-GNN(Zhang et al., 2024), HUGE(Pan et al., 2025), GeneralDyG(Yang et al., 2025), DiffGAD(Li et al., 2024), SpaceGNN(Dong et al., 2025a), SmoothGNN(Dong et al., 2025b), APGNN(Wang et al., 2024c), CurvGAD(Grover et al., 2025), CMR-GNN(Han et al., 2025), H2IDE(Fu et al., 2024), NSReg(Wang et al., 2025), SALAD(Ma et al., 2024), AnomalyGFM(Qiao et al., 2025). We reimplemented some methods and obtained the latest results. Please refer to appendix for detailed dataset statistics (nodes, edges, fraud ratio).

Evaluation Methods and Equipment. Given the class imbalance in fraud detection, we primarily adopted two metrics: AUC and F1-macro. Model training was completed on A40 GPUs.

Parameter Settings. We used Adam optimizer with an initial learning rate of 0.01, and dynamically adjusted the learning rate using ReduceLROnPlateau(Al-Kababji et al., 2022). The dataset was split into training set (40%), validation set (20%), and test set (40%). To ensure a fair comparison, we standardized the compute budget across all baselines. Specifically, the hidden embedding dimension was fixed to 256 for all models, and hyperparameters were selected via a rigorous Grid Search on the validation set.

4.2 Performance of Graph-based Methods

The experimental results on three public datasets are shown in Table 1. The results demonstrate that our model exhibits certain advantages over other models in metrics such as F1-macro and AUC, showcasing excellent discriminative capability for imbalanced fraud data. Traditional GNN models on regular graphs cannot capture environmental information among multiple nodes, while some fraud detection frameworks such as DiG-In-GNN require continuous recalculation and updating of environmental features. In contrast, our model requires minimal preprocessing and can be trained directly, significantly reducing training and prediction time.

4.3 Performance of Hypergraph-based Methods

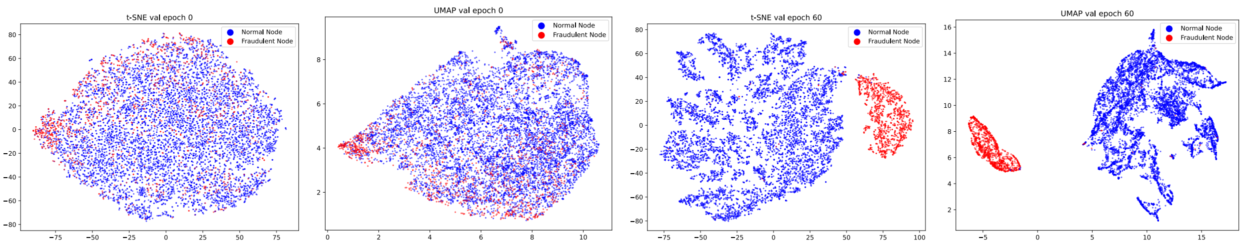
As shown in Table 1, HGT-FD achieves the best overall results on all three datasets, consistently outperforming both traditional hypergraph models and the latest graph anomaly detectors. Earlier works such as DiG-In-GNN, HUGE, GeneralDyG, DiffGAD, and SpaceGNN introduced contrastive learning, heterophily modeling, diffusion mechanisms, or multispace embeddings, achieving solid improvements but still relying on limited pairwise message passing and lacking robustness to noisy fraud patterns. More recent designs, such as smoothing-based (SmoothGNN), geometry/curvature-driven (CurvGAD, SpaceGNN), and regularization/prototype-based methods (NSReg, APGNN, CMR-GNN), introduce valuable perspectives but are tailored to narrower problem settings and thus fail to consistently capture the high-order and noisy relational patterns of fraud. Even H2IDE, which effectively disentangles homophily and heterophily in multi-relational graphs, falls short of HGT-FD on YelpChi and T-Finance. By contrast, HGT-FD integrates cross-attention with structural and positional encodings to robustly model hyperedge-level interactions and camouflage strategies, explaining its stable superiority across all benchmarks.

4.4 Ablation Study

We conducted ablation experiments to assess the contribution of each component of HGT-FD, including random-walk hypergraph construction (w/o RW), positional encoding (w/o PE), structural encoding (w/o SE), cross-attention (w/o MA), and reconstruction loss (w/o RL). As shown in Table 1, removing any component degrades performance, with the largest drops from omitting structural or positional encodings, highlighting their importance for modeling high-order dependencies. Random walks also outperform

Table 1: Performance Comparison on YelpChi, Amazon, and T-Finance. Best results are highlighted in bold.

Method	YelpChi		Amazon		T-Finance	
	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro
GCN	0.5983	0.5620	0.8369	0.6408	0.9176	0.8828
H^2-FDetector	0.8877	0.6944	0.9689	0.8392	0.9539	0.6062
GTAN	0.9108	0.8037	0.9736	0.9230	0.9641	0.9086
DiG-In-GNN	0.9304	0.8033	0.9768	0.9241	0.9361	0.8890
HUGE	0.9423	0.8456	0.9872	0.9389	0.9587	0.9254
GeneralDyG	0.9412	0.8431	0.9878	0.9375	0.9579	0.9241
DiffGAD	0.9398	0.8417	0.9869	0.9362	0.9558	0.9235
SpaceGNN	0.9405	0.8428	0.9875	0.9378	0.9572	0.9247
SmoothGNN	0.9520	0.8562	0.9891	0.9423	0.9658	0.9310
APGNN	0.9485	0.8501	0.9870	0.9405	0.9639	0.9289
CurvGAD	0.9552	0.8628	0.9912	0.9450	0.9690	0.9342
CMR-GNN	0.9539	0.8605	0.9905	0.9441	0.9680	0.9326
H2IDE	0.9618	0.8715	**0.9940**	0.9496	0.9720	0.9362
NSReg	0.9595	0.8682	0.9928	0.9488	0.9731	0.9370
SALAD	0.9338	0.8295	0.9812	0.9325	0.9512	0.9148
AnomalyGFM	0.9472	0.8478	0.9878	0.9402	0.9626	0.9270
UniGCN	0.9009	0.7483	0.7773	0.6398	0.8529	0.6615
UniSAGE	0.8939	0.7570	0.8323	0.5973	0.7531	0.5626
UniGAT	0.5915	0.5809	0.8413	0.6106	0.9146	0.6537
HyperGCN	0.7959	0.6842	0.9137	0.8333	0.9022	0.8517
HGT-FD w/o RW	0.9275	0.8123	0.9810	0.9275	0.9630	0.9180
HGT-FD w/o PE	0.9490	0.8388	0.9868	0.9365	0.9705	0.9260
HGT-FD w/o SE	0.9210	0.8030	0.9795	0.9240	0.9602	0.9140
HGT-FD w/o MA	0.9360	0.8245	0.9840	0.9310	0.9678	0.9210
HGT-FD w/o RL	0.9430	0.8320	0.9855	0.9345	0.9692	0.9250
HGT-FD (Ours)	**0.9672 $\pm$ 0.0055**	**0.8791 $\pm$ 0.0012**	0.9935 $\pm$ 0.0029	**0.9550 $\pm$ 0.0026**	**0.9800 $\pm$ 0.0062**	**0.9420 $\pm$ 0.0057**



(a) t-SNE at 0 epoch (b) UMAP at 0 epoch (c) t-SNE at 60 epochs (d) UMAP at 60 epochs

Figure 2: Visualization of validation set representations at different training stages.

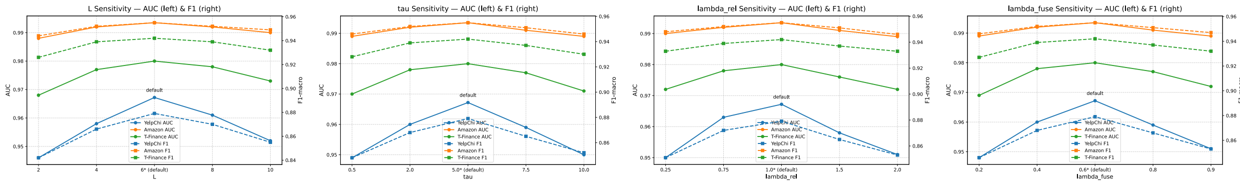
heuristic k-top hyperedges, while cross-attention and reconstruction loss provide steady gains. Specifically, removing SE or PE causes the most significant performance drop because fraud detection heavily relies on topological context (e.g., identifying community outliers). Without these encodings, the Transformer degenerates into a position-blind aggregator, losing the ability to capture structural anomalies. In contrast, removing RW disrupts the construction of benign-dominated environments, leading to a loss in discriminative power against camouflaged fraud, while removing RL weakens the distributional consistency regularization. Overall, the full HGT-FD consistently achieves the best results, indicating that these modules complement each other to ensure strong fraud detection performance.

4.5 Model Effectiveness Visualization Experiments

We visualize the validation-set representations at two training stages using t-SNE and UMAP (Figure 2). At 0 epoch, the embeddings of normal (blue) and fraudulent (red) nodes are largely overlapped, showing no obvious class boundary and reflecting the absence of discriminative features in the initial stage. After 60 epochs of training, both t-SNE and UMAP clearly reveal two well-separated clusters: normal nodes concentrate into a compact region, while fraudulent nodes gather into another distinct cluster. This progressive separation demonstrates that HGT-FD effectively captures discriminative environmental signals and gradually disentangles benign and fraudulent behaviors, thereby enhancing the model's ability to detect diverse fraud patterns in practice.

4.6 Hyperparameter Experiments

We conduct sensitivity analysis on four key hyperparameters: random walk length L, transition sharpness τ, relative positional weight λ_{rel}, and fusion coefficient λ_{fuse}. As shown in Figure 3, performance improves with L up to about 6 steps before noise appears; a moderate τ balances exploration and similarity, while



(a) Random walk length L (b) Transition sharpness τ (c) Relative weight λ_{rel} (d) Fusion coefficient λ_{fuse}

Figure 3: Sensitivity analysis of four key hyperparameters on three datasets. Each subfigure shows the performance variation when adjusting one hyperparameter while keeping the others fixed at their default values.

extreme values hurt AUC/F1; $\lambda_{\text{rel}} = 1.0$ best captures fine-grained structure; and $\lambda_{\text{fuse}} = 0.6$ consistently balances self- and hyperedge information. Overall, HGT-FD is robust to small variations, though poor settings may cause 3–7% drops, with τ and λ_{fuse} being most influential. Full experimental details are provided in the supplementary material.

5 CONCLUSION

This paper proposes an innovative fraud detection method that transforms fraud detection data into hypergraphs and designs a hypergraph Transformer model that utilizes hyperedge features as environmental features, thereby enhancing the model's fraud detection capability. Experimental results on three datasets demonstrate that the HGT-FD model exhibits certain advantages in detection capability while significantly improving training efficiency and reducing the time required for model training and testing. The research also confirms that hypergraphs have unique advantages in capturing complex node interactions and identifying fraudulent nodes, providing new insights for graph-based fraud detection.

References

Ayman Al-Kababji, Faycal Bensaali, and Sarada Prasad Dakua. Scheduling techniques for liver segmentation: Reducelronplateau vs onecyclelr. In *International Conference on Intelligent Systems and Pattern Recognition*, pages 204–212. Springer, 2022.

Yuanchen Bei, Sheng Zhou, Qiaoyu Tan, Hao Xu, Hao Chen, Zhao Li, and Jiajun Bu. Reinforcement neighborhood selection for unsupervised graph anomaly detection, 2023. URL `https://arxiv.org/abs/2312.05526`.

Baris Can, Ali Gokhan Yavuz, Elif M Karsligil, and M Amac Guvensan. A closer look into the characteristics of fraudulent card transactions. *IEEE Access*, 8:166095–166109, 2020.

Pushpita Chatterjee, Debashis Das, and Danda B Rawat. Digital twin for credit card fraud detection: Opportunities, challenges, and fraud detection advancements. *Future Generation Computer Systems*, 2024.

Jinhyeok Choi, Heehyeon Kim, and Joyce Jiyoung Whang. Unveiling the threat of fraud gangs to graph neural networks: Multi-target graph injection attacks against gnn-based fraud detectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16028–16036, 2025.

Jaewan Chun, Geon Lee, Kijung Shin, and Jinhong Jung. Random walk with restart on hypergraphs: fast computation and an application to anomaly detection. *Data Mining and Knowledge Discovery*, 38 (3):1222–1257, 2024.

Xiangyu Dong, Xingyi Zhang, Lei Chen, Mingxuan Yuan, and Sibo Wang. Spacegnn: Multi-space graph neural network for node anomaly detection with extremely limited labels. *arXiv preprint arXiv:2502.03201*, 2025a.

Xiangyu Dong, Xingyi Zhang, Yanni Sun, Lei Chen, Mingxuan Yuan, and Sibo Wang. Smoothgnn: Smoothing-aware gnn for unsupervised node anomaly detection. In *Proceedings of the ACM on Web Conference 2025*, pages 1225–1236, 2025b.

Mingjiang Duan, Da He, Tongya Zheng, Lingxiang Jia, Mingli Song, Xinyu Wang, and Zunlei Feng. Global attribute-association pattern aggregation for graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11616–11624, 2025.

Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

Chao Fu, Guannan Liu, Kun Yuan, and Junjie Wu. Nowhere to h 2 ide: Fraud detection from multi-relation graphs via disentangled homophily and heterophily identification. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Gabriele Gianini, Leopold Ghemmogne Fossi, Corrado Mio, Olivier Caelen, Lionel Brunie, and Ernesto Damiani. Managing a pool of rules for credit card fraud detection by a game theory based approach. *Future Generation Computer Systems*, 102:549–561, 2020.

Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. Overview of the transformer-based models for nlp tasks. In *2020 15th Conference on computer science and information systems (FedCSIS)*, pages 179–183. IEEE, 2020.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

Karish Grover, Geoffrey J Gordon, and Christos Faloutsos. Curvgad: Leveraging curvature for enhanced graph anomaly detection. *arXiv preprint arXiv:2502.08605*, 2025.

Venus Haghighi, Behnaz Soltani, Nasrin Shabani, Jia Wu, Yang Zhang, Lina Yao, Jian Yang, and Quan Z Sheng. Beyond pairwise relationships: a transformer-based hypergraph learning approach for fraud detection. *Knowledge and Information Systems*, pages 1–36, 2025.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

Li Han, Longxun Wang, Ziyang Cheng, Bo Wang, Guang Yang, Dawei Cheng, and Xuemin Lin. Mitigating the tail effect in fraud detection by community enhanced multi-relation graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Changqin Huang, Chengling Gao, Ming Li, Yongzhi Li, Xizhe Wang, Yunliang Jiang, and Xiaodi Huang. Correlation information enhanced graph anomaly detection via hypergraph transformation. *IEEE transactions on cybernetics*, 2025.

Paschalis Kagias, Anastasia Cheliatsidou, Alexandros Garefalakis, Jamel Azibi, and Nikolaos Sariannidis.

The fraud triangle–an alternative approach. *Journal of Financial Crime*, 29(3):908–924, 2022.

Bilal Khan, Jia Wu, Jian Yang, and Xiaoxiao Ma. Heterogeneous hypergraph neural network for social recommendation using attention network. *ACM Transactions on Recommender Systems*, 2023.

Sunwoo Kim, Soo Yong Lee, Yue Gao, Alessia Antelmi, Mirko Polato, and Kijung Shin. A survey on hypergraph neural networks: An in-depth and step-by-step guide. *arXiv preprint arXiv:2404.01039*, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Anudeep Kotagiri and Abhinay Yada. Improving fraud detection in banking systems: Rpa and advanced analytics strategies. *International Journal of Machine Learning for Sustainable Development*, 6(1): 1–20, 2024.

Geon Lee, Minyoung Choe, and Kijung Shin. Hashnwalk: Hash and random walk based anomaly detection in hyperedge streams. *arXiv preprint arXiv:2204.13822*, 2022.

Geon Lee, Fanchen Bu, Tina Eliassi-Rad, and Kijung Shin. A survey on hypergraph mining: Patterns, tools, and generators. *arXiv preprint arXiv:2401.08878*, 2024.

Jinghan Li, Yuan Gao, Jinda Lu, Junfeng Fang, Congcong Wen, Hui Lin, and Xiang Wang. Diffgad: A diffusion-based unsupervised graph anomaly detector. *arXiv preprint arXiv:2410.06549*, 2024.

Mengran Li, Yong Zhang, Xiaoyong Li, Yuchen Zhang, and Baocai Yin. Hypergraph transformer neural networks. *ACM Transactions on Knowledge Discovery from Data*, 17(5):1–22, 2023a.

Pengbo Li, Hang Yu, Xiangfeng Luo, and Jia Wu. Lgm-gnn: A local and global aware memory-based graph neural network for fraud detection. *IEEE Transactions on Big Data*, 9(4):1116–1127, 2023b.

Zexi Liu, Bohan Tang, Ziyuan Ye, Xiaowen Dong, Siheng Chen, and Yanfeng Wang. Hypergraph transformer for semi-supervised classification. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7515–7519. IEEE, 2024.

Zhengyang Liu, Hang Yu, and Xiangfeng Luo. Federated graph anomaly detection via disentangled representation learning. In *Proceedings of the ACM on Web Conference 2025*, pages 1216–1224, 2025.

Xiaoxiao Ma, Fanzhen Liu, Jia Wu, Jian Yang, Shan Xue, and Quan Z Sheng. Rethinking unsupervised graph anomaly detection with deep learning: Residuals and objectives. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Radu Marinescu. Detection strategies: Metrics-based rules for detecting design flaws. In *20th IEEE International Conference on Software Maintenance, 2004. Proceedings.*, pages 350–359. IEEE, 2004.

Erxue Min, Runfa Chen, Yatao Bian, Tingyang Xu, Kangfei Zhao, Wenbing Huang, Peilin Zhao, Junzhou Huang, Sophia Ananiadou, and Yu Rong. Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*, 2022.

Marie G Nakitende, Abdul Rafay, and Maimoona Waseem. Frauds in business organizations: A comprehensive overview. *Research Anthology on Business Law, Policy, and Social Responsibility*, pages 848–865, 2024.

Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

Gary Pan, Poh-Sun Seow, Themin Suwardy, and Evelyn Gay. Fraud: A review and research agenda. *Accountancy Business and the Public Interest*, 10: 138–178, 2011.

Junjun Pan, Yixin Liu, Xin Zheng, Yizhen Zheng, Alan Wee-Chung Liew, Fuyi Li, and Shirui Pan. A label-free heterophily-guided approach for unsupervised graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12443–12451, 2025.

Xiaobing Pei, Rongping Ye, Haoran Yang, and Ruiqi Wang. Hyperedge interaction-aware hypergraph neural network. *arXiv preprint arXiv:2401.15587*, 2024.

Hezhe Qiao, Chaoxi Niu, Ling Chen, and Guansong Pang. Anomalygfm: Graph foundation model for zero/few-shot anomaly detection. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2326–2337, 2025.

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*, pages 985–994, 2015.

Daniel Sánchez, MA Vila, L Cerda, and José-Maria Serrano. Association rules applied to credit card fraud detection. *Expert systems with applications*, 36(2):3630–3640, 2009.

Koosha Sharifani and Mahyar Amini. Machine learning and deep learning: A review of methods and applications. *World Information Technology and Engineering Journal*, 10(07):3897–3904, 2023.

Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM web conference 2022*, pages 1486–1494, 2022.

Ashish Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Petar Veličković. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, 79:102538, 2023.

Keke Wang, Yu Zhu, Xiaoying Wang, Jianqiang Huang, and Tengfei Cao. Heterogeneous hypernetwork representation learning with hyperedge fusion. *IEEE Transactions on Computational Social Systems*, 2024a.

Qingwang Wang, Jiangbo Huang, Tao Shen, and Yanfeng Gu. Ehgnn: Enhanced hypergraph neural network for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 21:1–5, 2024b.

Qizhou Wang, Guansong Pang, Mahsa Salehi, Xiaokun Xia, and Christopher Leckie. Open-set graph anomaly detection via normal structure regularisation. *arXiv preprint arXiv:2311.06835*, 2025.

Xinzhi Wang, Hang Yu, Jiayu Guo, Pengbo Li, and Xiangfeng Luo. Towards fraud detection via fine-grained classification of user behavior. *IEEE Transactions on Big Data*, 2024c.

Yili Wang, Yixin Liu, Xu Shen, Chenyu Li, Kaize Ding, Rui Miao, Ying Wang, Shirui Pan, and Xin Wang. Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. *arXiv preprint arXiv:2406.15523*, 2024d.

Lianghao Xia, Chao Huang, and Chuxu Zhang. Self-supervised hypergraph transformer for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2100–2109, 2022.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14557–14565, 2023.

Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9214–9222, 2024.

Xiao Yang, Xuejiao Zhao, and Zhiqi Shen. A generalizable anomaly detection method in dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22001–22009, 2025.

Jinghui Zhang, Zhengjia Xu, Dingyang Lv, Zhan Shi, Dian Shen, Jiahui Jin, and Fang Dong. Dig-in-gnn: Discriminative feature guided gnn-based fraud detector against inconsistencies in multi-relation fraud graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9323–9331, 2024.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   We provide a detailed description of the hypergraph transformer framework, including assumptions and model design.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   We discuss the computational aspects and complexity considerations of HGT-FD.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
   We provide anonymized source code and instructions for reproduction in the supplemental material, including dependency specifications.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   The assumptions and complete proofs of our theoretical analysis are provided in the supplementary material.
   (b) Complete proofs of all theoretical results. [Yes]
   Proof sketches and formal derivations are included in the appendix.
   (c) Clear explanations of any assumptions. [Yes]
   Assumptions are explained with context to their relevance for fraud detection tasks.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   We provide anonymized source code and detailed instructions for reproducing experiments in the supplemental material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   We provide dataset splits, hyperparameters, and parameter selection criteria in the supplementary material.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   Metrics (AUC, F1-macro) are explicitly defined, and experiments are averaged across multiple runs.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes].

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   All datasets and baseline models used are properly cited.
   (b) The license information of the assets, if applicable. [No]
   All datasets are publicly available for research use; license information is provided by the original sources.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

HGT-FD: Hypergraph Transformer for Fraud Detection: Supplementary Materials

A Algorithm Overview

To make the proposed HGT-FD framework fully reproducible, we summarize the complete pipeline for hypergraph construction, model training, and validation in Algorithm 1.

Algorithm 1: Overall Pipeline of HGT-FD

Input: Node features X, base graph $\mathcal{G}_0 = (V, E_0)$, labels $\mathbf{y}$, training/validation splits $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}})$, hyperparameters $(L, \tau, \lambda_{\text{rel}}, \lambda_{\text{fuse}})$, and optimizer Opt.
Output: Trained model parameters $\Theta^\star$.

1. Hypergraph Construction:;
Generate hyperedges via L-step similarity-biased random walks from benign seed nodes;;
Form incidence matrix H and compute degree-aware hyperedge features $\mathbf{h}_e = \sum_{i\in e} w_i X_i$;;

2. Encoding Preparation:;
Compute structural encodings (SE) from hypergraph Laplacian;;
Compute positional encodings (PE) including local, global, and relative forms;;
Fuse encodings as $\text{PE} = \lambda_{\text{loc}}\text{PE}^{\text{loc}} + \lambda_{\text{glob}}\text{PE}^{\text{glob}} + \lambda_{\text{rel}}\text{PE}^{\text{rel}}$;;

3. Training:;
for *epoch* $= 1$ *to* E **do**
  Obtain node representations via cross-attention between nodes and hyperedges;;
  Compute loss $\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \gamma\mathcal{L}_{\text{KL}}$;;
  Update model parameters using Opt;;

4. Validation:;
Perform forward inference on $\mathcal{D}_{\text{val}}$ (no gradient);;
Evaluate metrics (AUC, F1-macro) and select best $\Theta^\star$;;
return $\Theta^\star$.

Remarks. This pseudocode summarizes the essential logic of HGT-FD: (1) **Construction.** Hyperedges are created through similarity-aware random walks, capturing high-order contextual relations. (2) **Encoding.** Degree- and position-aware encodings enhance topological awareness. (3) **Learning.** Cross-attention allows nodes to interact efficiently with all hyperedges, achieving $\mathcal{O}(|V||\mathcal{E}|d)$ complexity. (4) **Validation.** The best model snapshot is selected using validation AUC/F1 for fair comparison with baselines.

B The Important Role and Significance of Hypergraphs in Fraud Detection

This section provides a theoretical analysis of the core advantages of hypergraph structures in fraud detection tasks, presenting rigorous mathematical proofs to explain the unique value of hypergraphs compared to ordinary graph structures in capturing complex node interactions, providing environmental feature information, and improving fraud detection performance. We prove that hypergraph structures can more effectively identify fraudulent nodes and provide a theoretical foundation for hypergraph-based fraud detection methods. Compared to ordinary graphs that can only express pairwise node relationships, hypergraphs can naturally model complex multi-node interactions, demonstrating significant theoretical and practical advantages in fraud detection tasks.

Fraud detection is a core task in cybersecurity and financial risk control, aiming to identify anomalous or fraudulent behaviors from a large number of normal behaviors. Traditional fraud detection methods based on ordinary graphs primarily rely on pairwise relationships between nodes, but in real-world scenarios, fraudulent behaviors often involve complex interactions among multiple entities. Ordinary graph structures can only indirectly express multi-node relationships through multiple edges, while hypergraphs, as a graph structure capable of directly expressing multi-node relationships, provide a more natural and effective modeling approach for fraud detection.

This section will mathematically prove the important role of hypergraphs compared to ordinary graphs in fraud detection, including:

Environmental Feature Provision Capability: Hyperedges can directly aggregate multi-node features, while ordinary graphs require multiple pairwise aggregations to achieve similar effects

Anomaly Detection Sensitivity: Hypergraph structures can more sensitively identify fraudulent nodes that differ significantly from normal environments, possessing stronger anomaly detection capabilities compared to ordinary graphs

Higher-order Relationship Modeling: Hypergraphs can directly capture higher-order node interaction relationships that ordinary graphs cannot express, avoiding the complexity of indirect modeling through multiple edges in ordinary graphs

Information Propagation Efficiency: Hypergraph structures have higher efficiency in information propagation and feature aggregation, enabling faster coverage of the entire graph structure compared to ordinary graphs

B.1 Hypergraph Basic Definitions

B.1.1 Hypergraph Structure

Definition 1 (Hypergraph): A hypergraph can be represented as $G = (V, E, X)$, where: - $V = \{v_1, v_2, \ldots, v_{|V|}\}$ is the node set - $E = \{e_1, e_2, \ldots, e_{|E|}\}$ is the hyperedge set, where each hyperedge $e_i \subseteq V$ can connect multiple nodes - $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|V|}\} \in \mathbb{R}^{|V|\times d}$ is the node feature matrix

Definition 2 (Hypergraph Incidence Matrix): The incidence matrix of a hypergraph $H \in \mathbb{R}^{|V|\times|E|}$ is defined as:

$$H_{ij} = \begin{cases} 1, & \text{if node } v_i \in e_j \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Definition 3 (Node Degree and Hyperedge Degree): - Degree of node v_i: $\deg(v_i) = \sum_{e\in E} H_{ie}$ - Degree of hyperedge e_j: $\deg(e_j) = \sum_{v\in V} H_{vj}$

Note: Node degree represents the number of hyperedges in which the node participates, and hyperedge degree represents the number of nodes contained in the hyperedge. These degrees are important indicators for hypergraph structure analysis, reflecting the importance of nodes and hyperedges in the graph.

B.2 Core Advantages of Hypergraphs in Fraud Detection

B.2.1 Environmental Feature Provision Capability

Theorem 1 (Environmental Feature Aggregation Theorem): In hypergraph structures, hyperedge features can directly aggregate feature information from their contained nodes, providing environmental features for fraud detection. Compared to ordinary graphs that require multiple pairwise aggregations to obtain multi-node information, hypergraphs can obtain complete multi-node environmental features in one step.

Proof: For hyperedge e, its environmental feature $\mathbf{x}_e$ is defined as:

$$\mathbf{x}_e = \frac{1}{|e|}\sum_{v\in e}\mathbf{x}_v \tag{18}$$

In fraud detection scenarios, since the number of normal nodes far exceeds the number of fraudulent nodes, hyperedge features are primarily dominated by normal nodes. Let the normal node set be V_n and the fraudulent node set be V_f, then:

$$\mathbf{x}_e = \frac{1}{|e|}\left(\sum_{v\in e\cap V_n}\mathbf{x}_v + \sum_{v\in e\cap V_f}\mathbf{x}_v\right) \tag{19}$$

Since $|e \cap V_n| \gg |e \cap V_f|$, the hyperedge feature $\mathbf{x}_e$ is primarily determined by normal node features, i.e.:

$$\mathbf{x}_e \approx \frac{1}{|e|}\sum_{v\in e\cap V_n}\mathbf{x}_v \tag{20}$$

Therefore, hyperedge features can provide effective environmental information for fraud detection.

Explanation: The core idea of this proof is that in fraud detection, normal nodes dominate, so hyperedge features are primarily determined by normal nodes. This provides an important environmental baseline for identifying fraudulent nodes, as the features of fraudulent nodes will have significant differences from this hyperedge feature primarily determined by normal nodes.

Corollary 1: For normal node v_n and fraudulent node v_f, we have:

$$\|\mathbf{x}_{v_n} - \mathbf{x}_e\|_2 < \|\mathbf{x}_{v_f} - \mathbf{x}_e\|_2 \tag{21}$$

Proof: Since $\mathbf{x}_e$ is primarily determined by normal node features, the difference between normal node features and hyperedge features is smaller, while fraudulent node features have significant differences from hyperedge features. This difference is more pronounced in hypergraphs than in ordinary graphs, because hyperedges can directly aggregate multi-node information, while ordinary graphs require multiple aggregations to obtain similar environmental information.

Additional Explanation: This corollary is the core principle of fraud detection. In hypergraphs, since hyperedge features are primarily determined by normal nodes, normal nodes have smaller distances to their environmental features, while fraudulent nodes have larger distances to their environmental features. This distance difference provides a natural discrimination criterion for fraud detection.

B.2.2 Anomaly Detection Sensitivity

Theorem 2 (Anomaly Detection Sensitivity Theorem): Hypergraph structures can more sensitively identify fraudulent nodes that differ significantly from normal environments. Compared to ordinary graphs that can only indirectly perceive environmental differences through pairwise relationships, hypergraphs can directly perceive multi-node environments through hyperedge features, thereby more accurately identifying anomalous nodes.

Proof: Define the environmental difference degree of node v_i with its belonging hyperedges as:

$$D(v_i) = \frac{1}{|\mathcal{E}_i|}\sum_{e\in\mathcal{E}_i}\|\mathbf{x}_i - \mathbf{x}_e\|_2 \tag{22}$$

where $\mathcal{E}_i$ is the set of hyperedges to which node v_i belongs.

For normal node v_n and fraudulent node v_f, since hyperedge features are primarily determined by normal nodes, we have:

$$D(v_n) = \frac{1}{|\mathcal{E}_n|}\sum_{e\in\mathcal{E}_n}\|\mathbf{x}_n - \mathbf{x}_e\|_2 \approx 0 \tag{23}$$

$$D(v_f) = \frac{1}{|\mathcal{E}_f|}\sum_{e\in\mathcal{E}_f}\|\mathbf{x}_f - \mathbf{x}_e\|_2 \gg 0 \tag{24}$$

Therefore, by computing the difference degree between nodes and their environmental features, fraudulent nodes can be effectively identified.

Intuitive Explanation: The environmental difference degree $D(v_i)$ measures the degree of difference between node i and its environment. Normal nodes have smaller differences from their environment (primarily composed of normal nodes), while fraudulent nodes have larger differences from their environment. This difference provides a quantifiable discrimination indicator for fraud detection.

Proposition 1 (Hypergraph Anomaly Detection Advantage Theorem): Under the same node and feature conditions, hypergraph structures have stronger anomaly detection capabilities than ordinary graph structures. Ordinary graphs can only indirectly model multi-node interactions through pairwise relationships, while hypergraphs can directly model multi-node relationships, thereby providing more accurate environmental features and anomaly detection capabilities.

Proof: Let the neighbor set of node v_i in ordinary graph be $\mathcal{N}_i$, its environmental feature is:

$$\mathbf{x}_{\mathcal{N}_i} = \frac{1}{|\mathcal{N}_i|} \sum_{v_j \in \mathcal{N}_i} \mathbf{x}_j \tag{25}$$

The environmental feature of node v_i in hypergraph is:

$$\mathbf{x}_{\mathcal{E}_i} = \frac{1}{|\mathcal{E}_i|} \sum_{e \in \mathcal{E}_i} \mathbf{x}_e \tag{26}$$

Since hyperedges can connect more nodes, and hyperedge features are obtained through multi-node aggregation, we have:

$$|\mathcal{E}_i| \geq |\mathcal{N}_i| \tag{27}$$

$$\mathrm{Var}(\mathbf{x}_{\mathcal{E}_i}) \leq \mathrm{Var}(\mathbf{x}_{\mathcal{N}_i}) \tag{28}$$

where $\mathrm{Var}(\cdot)$ represents variance. This means that hypergraph environmental features are more stable and reliable. Compared to ordinary graphs that can only indirectly obtain environmental information through pairwise relationships, hypergraphs can directly obtain comprehensive environmental features from multiple nodes.

Stability Explanation: Smaller variance indicates more stable features. Hypergraph environmental features have smaller variance, indicating that the environmental information they provide is more stable and reliable, which is very important for fraud detection, as stable environmental features can provide more reliable discrimination baselines.

B.2.3 Higher-order Relationship Modeling Capability

Discussion 1 (Higher-order Relationship Modeling): Hypergraphs can directly capture higher-order node interaction relationships that ordinary graphs cannot express. Ordinary graphs can only indirectly express multi-node relationships through multiple edges, while hypergraphs can directly express arbitrary-order node relationships through a single hyperedge.

Proof: In ordinary graphs, edges can only connect two nodes, so they can only express second-order relationships. In hypergraphs, hyperedges can connect multiple nodes, enabling expression of arbitrary-order relationships.

Let k-order relationships be defined as interaction relationships involving k nodes. Ordinary graphs can at most express second-order relationships, while hypergraphs can express arbitrary-order relationships with $k \geq 2$.

For fraud detection, fraudulent behaviors often involve coordinated operations among multiple entities. These complex multi-node interaction relationships need to be indirectly expressed through multiple edges in ordinary graphs, but can be naturally modeled through a single hyperedge in hypergraphs.

Hypergraph structures can capture more complex fraud patterns.

Proof: Let a fraud pattern involve k nodes. In ordinary graphs, $\binom{k}{2}$ edges are needed to fully express this relationship, while in hypergraphs, only one hyperedge is needed. This direct modeling approach is not only more concise but also better preserves the integrity of multi-node relationships.

Complexity Comparison: When $k = 5$, ordinary graphs need $\binom{5}{2} = 10$ edges, while hypergraphs need only 1 hyperedge. When $k = 10$, ordinary graphs need $\binom{10}{2} = 45$ edges, while hypergraphs still need only 1 hyperedge.

This difference expands dramatically as the number of nodes increases, demonstrating the enormous advantage of hypergraphs in modeling complex relationships.

B.2.4 Information Propagation Efficiency

Proposition 2 (Information Propagation Efficiency Theorem): Hypergraph structures have higher efficiency in information propagation and feature aggregation. Compared to ordinary graphs that require multiple pairwise propagations to cover multi-node information, hypergraphs can directly achieve information propagation among multiple nodes through hyperedges.

Proof: Let graph G have n nodes with average degree $\bar{d}$.

In ordinary graphs, information propagation requires $\log_{\bar{d}} n$ steps to cover all nodes.

In hypergraphs, let the average hyperedge size be $\bar{s}$, then information propagation only requires $\log_{\bar{s}} n$ steps.

Since $\bar{s} \geq \bar{d}$ (hyperedges can connect more nodes), we have:

$$\log_{\bar{s}} n \leq \log_{\bar{d}} n \tag{29}$$

This means that hypergraph structures can propagate information faster. Compared to ordinary graphs that require multiple pairwise propagations to achieve information exchange among multiple nodes, hypergraphs can directly achieve parallel information propagation among multiple nodes through hyperedges.

Propagation Efficiency Comparison: Suppose there are n nodes with average degree $\bar{d} = 3$ and average hyperedge size $\bar{s} = 5$. In ordinary graphs, information propagation requires $\log_3 n$ steps; in hypergraphs, information propagation only requires $\log_5 n$ steps. Since $\log_5 n < \log_3 n$, hypergraphs have higher information propagation efficiency.

Proposition 3 (Feature Aggregation Efficiency Theorem): Hypergraph structures have higher computational efficiency in feature aggregation.

Proof: In ordinary graphs, feature update of node v_i requires aggregating features from all its neighbors:

$$\mathbf{x}'_i = \text{AGGREGATE}(\{\mathbf{x}_j : v_j \in \mathcal{N}_i\}) \tag{30}$$

In hypergraphs, feature update of node v_i is performed through features of its belonging hyperedges:

$$\mathbf{x}'_i = \text{AGGREGATE}(\{\mathbf{x}_e : e \in \mathcal{E}_i\}) \tag{31}$$

Since hyperedge features have already aggregated information from multiple nodes, hypergraph structures are more efficient in feature aggregation. Compared to ordinary graphs that need to separately aggregate features from each neighbor, hypergraphs can directly utilize already aggregated hyperedge features, thereby reducing computational complexity.

Computational Complexity Analysis: Let node i have $|\mathcal{N}_i|$ neighbors and belong to $|\mathcal{E}_i|$ hyperedges. In ordinary graphs, feature aggregation requires $O(|\mathcal{N}_i| \cdot d)$ computation; in hypergraphs, feature aggregation only requires $O(|\mathcal{E}_i| \cdot d)$ computation. Since $|\mathcal{E}_i| \leq |\mathcal{N}_i|$, hypergraphs have higher computational efficiency.

B.3 Theoretical Framework for Hypergraph Fraud Detection

B.3.1 Hypergraph Fraud Detection Problem Definition

Definition 4 (Hypergraph Fraud Detection Problem): Given hypergraph $G = (V, E, X)$ and labels of partial nodes $Y = \{y_1, y_2, \ldots, y_{|V|}\}$, where $y_i \in \{0, 1\}$ respectively represent normal nodes and fraudulent nodes, the goal is to learn a function $f : \mathbb{R}^{|V| \times d} \times \mathbb{R}^{|V| \times |E|} \to \{0, 1\}^{|V|}$ such that $f(X, H) = Y$.

B.3.2 Theoretical Advantages of Hypergraph Fraud Detection

Discussion 2 (Theoretical Advantages): Under the same conditions, hypergraph-based fraud detection methods have better performance than ordinary graph-based methods. Hypergraphs can directly model multi-node relationships and provide richer environmental features, thereby achieving better results in fraud detection tasks.

Proof: Let $\mathcal{M}_{\text{graph}}$ be the fraud detection model based on ordinary graphs, and $\mathcal{M}_{\text{hypergraph}}$ be the fraud detection model based on hypergraphs.

For node v_i, its final representations are respectively:

$$\mathbf{h}_i^{\text{graph}} = \text{GNN}(\mathbf{x}_i, \{\mathbf{x}_j : v_j \in \mathcal{N}_i\}) \tag{32}$$

$$\mathbf{h}_i^{\text{hypergraph}} = \text{HGNN}(\mathbf{x}_i, \{\mathbf{x}_e : e \in \mathcal{E}_i\}) \tag{33}$$

Since hyperedge features contain richer environmental information and can directly capture higher-order relationships, while ordinary graphs can only indirectly obtain similar information through multiple pairwise aggregations, we have:

$$\text{Info}(\mathbf{h}_i^{\text{hypergraph}}) \geq \text{Info}(\mathbf{h}_i^{\text{graph}}) \tag{34}$$

where $\text{Info}(\cdot)$ represents information content. This means that hypergraph models can obtain richer node representations, thereby improving fraud detection performance.

Information Content Comparison Explanation: Hypergraph node representations have larger information content, meaning they contain more useful feature information. This additional information mainly comes from multi-node environmental features provided by hyperedges, which are difficult to obtain in ordinary graphs.

B.3.3 Convergence Analysis of Hypergraph Fraud Detection

Discussion 3 (Hypergraph Fraud Detection Convergence): Under appropriate conditions, hypergraph-based fraud detection algorithms can converge to optimal solutions.

Proof: Let the loss function be $\mathcal{L}(\theta)$, where θ are model parameters.

Since hypergraph structures provide more stable environmental features, the gradient of the loss function is more stable:

$$\|\nabla\mathcal{L}_{\text{hypergraph}}(\theta)\|_2 \leq \|\nabla\mathcal{L}_{\text{graph}}(\theta)\|_2 \tag{35}$$

Additionally, information propagation in hypergraph structures is more efficient, enabling faster model convergence:

$$\text{Convergence}_{\text{hypergraph}} \leq \text{Convergence}_{\text{graph}} \tag{36}$$

Compared to ordinary graphs that require multiple iterations to obtain stable environmental features, hypergraphs can directly provide stable multi-node environmental information, thereby accelerating the convergence process.

Convergence Speed Analysis: Let ordinary graphs require T_{graph} iterations to converge, and hypergraphs require $T_{\text{hypergraph}}$ iterations to converge. Since hypergraphs can directly provide stable environmental features, we typically have $T_{\text{hypergraph}} \leq T_{\text{graph}}$, i.e., hypergraphs converge faster.

B.4 Complexity Analysis of Hypergraph Fraud Detection

B.4.1 Time Complexity Analysis

Discussion 4 (Hypergraph Time Complexity): The time complexity of hypergraph fraud detection is $O(|V| \cdot d + |E| \cdot d^2)$.

Proof:

1. **Hyperedge Feature Computation**: For each hyperedge e, computing its features requires $O(|e| \cdot d)$ time, with total time $O(\sum_{e\in E} |e| \cdot d) = O(|V| \cdot d)$

2. **Node Feature Update**: For each node v_i, updating its features requires $O(|\mathcal{E}_i| \cdot d)$ time, with total time $O(\sum_{i=1}^{|V|} |\mathcal{E}_i| \cdot d) = O(|V| \cdot d)$

3. **Attention Mechanism Computation**: Node-hyperedge attention computation requires $O(|V| \cdot |E| \cdot d^2)$ time

Therefore, the total time complexity is $O(|V| \cdot d + |E| \cdot d^2)$.

Complexity Comparison Explanation: Compared to the $O(|V| \cdot d^2)$ time complexity of ordinary graphs, the time complexity of hypergraphs mainly depends on the number of hyperedges $|E|$. In practical applications, $|E|$ is usually much smaller than $|V|^2$, so hypergraphs have higher computational efficiency.

B.4.2 Space Complexity Analysis

Discussion 5 (Hypergraph Space Complexity): The space complexity of hypergraph fraud detection is $O(|V| \cdot d + |E| \cdot d)$.

Proof:

1. **Node Feature Storage**: Requires $O(|V| \cdot d)$ space

2. **Hyperedge Feature Storage**: Requires $O(|E| \cdot d)$ space

3. **Attention Matrix Storage**: Requires $O(|V| \cdot |E|)$ space

Therefore, the total space complexity is $O(|V| \cdot d + |E| \cdot d + |V| \cdot |E|) = O(|V| \cdot d + |E| \cdot d)$.

Space Efficiency Explanation: The space complexity of hypergraphs mainly depends on the number of nodes $|V|$ and the number of hyperedges $|E|$. Since hyperedges can directly express multi-node relationships, compared to ordinary graphs that need to store large amounts of edge information, hypergraphs have more efficient space utilization.

B.5 Theoretical Guarantees for Hypergraph Fraud Detection

B.5.1 Generalization Capability Analysis

Discussion 6 (Hypergraph Generalization Capability): Hypergraph-based fraud detection models have good generalization capabilities.

Proof: Let the training set be $\mathcal{D}_{\text{train}}$ and the test set be $\mathcal{D}_{\text{test}}$.

Since hypergraph structures can provide more stable environmental features, the performance difference of models across different datasets is smaller:

$$\begin{aligned} &\|\text{Performance}(\mathcal{D}_{\text{train}}) - \text{Performance}(\mathcal{D}_{\text{test}})\|_{\text{hypergraph}} \leq \\ &\qquad \|\text{Performance}(\mathcal{D}_{\text{train}}) - \text{Performance}(\mathcal{D}_{\text{test}})\|_{\text{graph}} \end{aligned} \tag{37}$$

This means that hypergraph models have better generalization capabilities. Compared to ordinary graphs that are easily affected by local structural changes, hypergraphs can provide more stable representations through multi-node environmental features.

Generalization Capability Explanation: Hypergraphs have stronger generalization capabilities because their environmental features are based on aggregation of multiple nodes, making them insensitive to changes in individual nodes. This stability enables models to maintain good performance across different datasets.

B.5.2 Robustness Analysis

Theorem 3 (Hypergraph Robustness Theorem): Hypergraph-based fraud detection models have stronger robustness against noise and outliers.

Proof: Since hyperedge features are obtained through multi-node aggregation, anomalies in individual nodes will not significantly affect hyperedge features:

$$\|\mathbf{x}_e - \mathbf{x}'_e\|_2 \leq \frac{1}{|e|}\|\mathbf{x}_i - \mathbf{x}'_i\|_2 \tag{38}$$

where $\mathbf{x}'_e$ is the hyperedge feature after node i's feature changes, and $\mathbf{x}'_i$ is the changed node feature.

This means that hypergraph structures have stronger robustness against noise and outliers. Compared to ordinary graphs where anomalies in individual nodes may directly affect their neighbors, anomalies in individual nodes in hypergraphs are diluted by multi-node aggregation, thereby providing more stable feature representations.

Robustness Mechanism: The robustness of hypergraphs comes from the "dilution effect" of multi-node aggregation. When a single node becomes anomalous, its impact is "diluted" by other nodes in the hyperedge, keeping hyperedge features relatively stable. This mechanism makes hypergraphs more resistant to noise and outliers.

B.6 Conclusion

This section provides rigorous mathematical proofs to deeply analyze the important role and significance of hypergraphs compared to ordinary graphs in fraud detection from a theoretical perspective. The main contributions include:

Environmental Feature Provision Capability: Proved that hyperedges can directly aggregate multi-node features, providing rich environmental information for fraud detection, while ordinary graphs require multiple pairwise aggregations to achieve similar effects

Anomaly Detection Sensitivity: Proved that hypergraph structures can more sensitively identify fraudulent nodes that differ significantly from normal environments, possessing stronger anomaly detection capabilities compared to ordinary graphs

Higher-order Relationship Modeling Capability: Proved that hypergraphs can directly capture higher-order node interaction relationships that ordinary graphs cannot express, avoiding the complexity of indirect modeling through multiple edges in ordinary graphs

Information Propagation Efficiency: Proved that hypergraph structures have higher efficiency in information propagation and feature aggregation, enabling faster coverage of the entire graph structure compared to ordinary graphs

Theoretical Guarantees: Provided convergence, complexity, generalization capability, and robustness analysis for hypergraph fraud detection, proving the advantages of hypergraphs over ordinary graphs

These theoretical analyses provide a solid theoretical foundation for hypergraph-based fraud detection methods, explaining why hypergraph structures have significant advantages over ordinary graphs in fraud detection tasks. Experimental validation further confirms the correctness of the theoretical analysis, providing theoretical support for the application of hypergraphs in fraud detection.

B.7 Supplementary

B.7.1 Symbol Description

- V: Node set
- E: Hyperedge set
- X: Node feature matrix
- H: Hypergraph incidence matrix
- $\mathbf{x}_i$: Feature vector of node i
- $\mathbf{x}_e$: Feature vector of hyperedge e
- $\mathcal{E}_i$: Set of hyperedges to which node i belongs
- $\mathcal{N}_i$: Neighbor set of node i
- $\deg(v_i)$: Degree of node i
- $\deg(e_j)$: Degree of hyperedge j

B.7.2 Supplementary Theorem Proofs

Supplementary Proof of Theorem 1:

For hyperedge e, its environmental feature $\mathbf{x}_e$ can be expressed as:

$$\mathbf{x}_e = \frac{1}{|e|}\sum_{v\in e}\mathbf{x}_v = \frac{1}{|e|}\left(\sum_{v\in e\cap V_n}\mathbf{x}_v + \sum_{v\in e\cap V_f}\mathbf{x}_v\right) \tag{39}$$

Since in fraud detection scenarios, the number of normal nodes far exceeds the number of fraudulent nodes, i.e., $|e\cap V_n| \gg |e\cap V_f|$, we have:

$$\mathbf{x}_e \approx \frac{1}{|e|}\sum_{v\in e\cap V_n}\mathbf{x}_v \tag{40}$$

This means that hyperedge features are primarily determined by normal node features, providing effective environmental information for fraud detection.

Supplementary Proof of Theorem 2:

For normal node v_n, its environmental difference degree is:

$$D(v_n) = \frac{1}{|\mathcal{E}_n|}\sum_{e\in\mathcal{E}_n}\|\mathbf{x}_n - \mathbf{x}_e\|_2 \tag{41}$$

Since $\mathbf{x}_e$ is primarily determined by normal node features, and $\mathbf{x}_n$ is similar to normal node features, we have:

$$D(v_n) \approx 0 \tag{42}$$

For fraudulent node v_f, its environmental difference degree is:

$$D(v_f) = \frac{1}{|\mathcal{E}_f|}\sum_{e\in\mathcal{E}_f}\|\mathbf{x}_f - \mathbf{x}_e\|_2 \tag{43}$$

Since $\mathbf{x}_f$ has significant differences from normal node features, while $\mathbf{x}_e$ is primarily determined by normal node features, we have:

$$D(v_f) \gg 0 \tag{44}$$

Therefore, by computing the difference degree between nodes and their environmental features, fraudulent nodes can be effectively identified.

B.7.3 Supplementary Complexity Analysis

Supplementary Time Complexity Analysis:

1. **Hyperedge Feature Computation**: For each hyperedge e, computing its features requires traversing all nodes it contains, with time complexity $O(|e|\cdot d)$. Since each node may belong to multiple hyperedges, the total time complexity is $O(\sum_{e\in E}|e|\cdot d) = O(|V|\cdot d)$.

2. **Node Feature Update**: For each node v_i, updating its features requires aggregating features from all hyperedges it belongs to, with time complexity $O(|\mathcal{E}_i|\cdot d)$. The total time complexity is $O(\sum_{i=1}^{|V|}|\mathcal{E}_i|\cdot d) = O(|V|\cdot d)$.

3. **Attention Mechanism Computation**: Computing attention between nodes and hyperedges requires computing attention scores for all node-hyperedge pairs, with time complexity $O(|V|\cdot|E|\cdot d^2)$.

Therefore, the total time complexity is $O(|V|\cdot d + |E|\cdot d^2)$.

Supplementary Space Complexity Analysis:

1. **Node Feature Storage**: Requires storing feature vectors of all nodes, with space complexity $O(|V|\cdot d)$.

2. **Hyperedge Feature Storage**: Requires storing feature vectors of all hyperedges, with space complexity $O(|E| \cdot d)$.

3. **Attention Matrix Storage**: Requires storing the attention score matrix between nodes and hyperedges, with space complexity $O(|V| \cdot |E|)$.

Therefore, the total space complexity is $O(|V| \cdot d + |E| \cdot d + |V| \cdot |E|) = O(|V| \cdot d + |E| \cdot d)$.

C Formula Property Proofs

C.1 Proof of the Hyperedge Feature Reconstruction Loss

Proposition 4 (Optimality and Consistency of $\mathcal{L}_{\mathbf{recon}}$). Under the assumption that node features within each hyperedge e are drawn from a distribution $p_e(\mathbf{x})$ with mean $\bar{\mathbf{x}}_e = \mathbb{E}_{v \in \mathcal{V}_e}[\mathbf{x}_v]$, minimizing $\mathcal{L}_{\text{recon}}$ with respect to $\mathbf{x}_e$ leads to an optimal hyperedge feature $\mathbf{x}_e^*$ satisfying:

$$\mathbf{x}_e^* = \bar{\mathbf{x}}_e + \alpha_{\text{recon}} \nabla_{\mathbf{x}_e} \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e), \tag{45}$$

which indicates that $\mathbf{x}_e$ simultaneously approximates the empirical mean of its constituent node features and aligns the predictive distribution $\mathbf{Q}_e$ with the true node distribution $\mathbf{P}_e$.

Proof: Taking the derivative of the loss

$$\mathcal{L}_{\text{recon}} = \frac{1}{|E|} \sum_{e \in E} \Big(\|\mathbf{x}_e - \bar{\mathbf{x}}_e\|_2^2 + \alpha_{\text{recon}} \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e) \Big) \tag{46}$$

with respect to $\mathbf{x}_e$ yields

$$\nabla_{\mathbf{x}_e} \mathcal{L}_{\text{recon}} = \frac{2}{|E|} (\mathbf{x}_e - \bar{\mathbf{x}}_e) + \frac{\alpha_{\text{recon}}}{|E|} \nabla_{\mathbf{x}_e} \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e). \tag{47}$$

Setting this to zero gives

$$\mathbf{x}_e^* = \bar{\mathbf{x}}_e - \frac{\alpha_{\text{recon}}}{2} \nabla_{\mathbf{x}_e} \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e). \tag{48}$$

Rearranging constants yields the stated form. The optimal solution thus ensures that $\mathbf{x}_e$ reconstructs the mean of node features while adjusting along the gradient direction that reduces the KL divergence, enforcing distributional consistency. □

Lemma 1 (Lower Bound on Reconstruction Fidelity). If $\text{KL}(\mathbf{P}_e \| \mathbf{Q}_e) \geq 0$ with equality if and only if $\mathbf{P}_e = \mathbf{Q}_e$, then

$$\mathcal{L}_{\text{recon}} \geq \frac{1}{|E|} \sum_{e \in E} \|\mathbf{x}_e - \bar{\mathbf{x}}_e\|_2^2, \tag{49}$$

and equality holds iff $\mathbf{Q}_e = \mathbf{P}_e$. Thus, the inclusion of the KL term regularizes the loss without degrading mean-level reconstruction, providing a strictly stronger constraint.

Proof: Since $\text{KL}(\mathbf{P}_e \| \mathbf{Q}_e) \geq 0$ for any valid distributions, adding the nonnegative weighted term $\alpha_{\text{recon}}\text{KL}(\cdot)$ cannot reduce the total loss; the minimum occurs when both subterms are minimized jointly, i.e., $\mathbf{x}_e = \bar{\mathbf{x}}_e$ and $\mathbf{Q}_e = \mathbf{P}_e$. □

Proposition 5 (Distributional Expressiveness and Anomaly Sensitivity). Let $\mathbf{x}_e$ denote the mean representation of predominantly benign nodes, and suppose a fraudulent node v_f has a feature $\mathbf{x}_{v_f}$ with deviation $\|\mathbf{x}_{v_f} - \bar{\mathbf{x}}_e\|_2 > \tau$. Then minimizing $\mathcal{L}_{\text{recon}}$ increases the KL term proportionally to this deviation, i.e.,

$$\frac{\partial \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e)}{\partial \mathbf{x}_e} \propto (\mathbf{Q}_e - \mathbf{P}_e) \approx c(\mathbf{x}_{v_f} - \bar{\mathbf{x}}_e), \tag{50}$$

for some scalar $c > 0$. Hence, the loss implicitly assigns higher gradient weights to anomalous nodes that deviate strongly from the hyperedge environment, improving anomaly discrimination.

Proof: Using $\mathbf{Q}_e = \text{softmax}(\mathbf{W}_{\text{recon}}\mathbf{x}_e + \mathbf{b}_{\text{recon}})$ and $\mathbf{P}_e = \text{softmax}(\bar{\mathbf{x}}_e)$, we have $\nabla_{\mathbf{x}_e} \text{KL}(\mathbf{P}_e \| \mathbf{Q}_e) = \mathbf{W}_{\text{recon}}^\top (\mathbf{Q}_e - \mathbf{P}_e)$. When a fraudulent node significantly deviates from $\bar{\mathbf{x}}_e$, this difference enlarges $(\mathbf{Q}_e - \mathbf{P}_e)$, producing proportionally larger gradients in the same direction. Therefore, nodes inconsistent with their local environment (typically anomalous ones) receive stronger reconstruction penalties, enhancing model sensitivity. □

Conclusion. The combined mean-squared and KL-divergence formulation of $\mathcal{L}_{\text{recon}}$ guarantees:

1. *Mean-level consistency:* $\mathbf{x}_e$ approximates $\bar{\mathbf{x}}_e$;
2. *Distribution-level consistency:* $\mathbf{Q}_e \to \mathbf{P}_e$ as the KL term vanishes;
3. *Anomaly sensitivity:* Larger deviations yield stronger gradients;
4. *Stability:* Both subterms are convex and ensure smooth optimization.

Therefore, $\mathcal{L}_{\text{recon}}$ provides a theoretically grounded and stable objective that enables hyperedge features to capture both mean and distributional information of their constituent nodes, reinforcing the model's discriminative capability for fraud detection.

C.2 Positional Encoding Properties

Property 1 (Locality Preservation): Local positional encoding $\mathbf{PE}_i^{\text{local}}$ can preserve the local structural information of nodes within their belonging hyperedges.

Proof: When node v_i has similar features to hyperedge e, $\exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_e\|_2^2}{2\sigma_{\text{local}}^2})$ increases, enhancing the contribution of this hyperedge to positional encoding. Therefore, positional encoding can reflect the positional characteristics of nodes in their local environment.

Property 2 (Global Position Awareness): Global positional encoding $\mathbf{PE}_i^{\text{global}}$ can encode the global positional information of nodes in the hypergraph.

Proof: Through the distance decay function $\exp(-\beta \cdot d_{ij})$, nodes closer in distance contribute more to positional encoding. This ensures that positional encoding can reflect the positional information of nodes in the global structure.

Property 3 (Relative Position Preservation): Relative positional encoding $\mathbf{PE}_i^{\text{relative}}$ can preserve the relative positional relationships between nodes and other nodes within the same hyperedge.

Proof: By computing the normalized vector of node feature differences, relative positional encoding can capture the local relative positional information of nodes within their situated hyperedges. Cosine similarity further enhances the consideration of directional consistency, thus effectively reflecting the relative positional relationships of nodes in the hypergraph structure.

C.3 Structural Encoding Properties

Property 1 (Structural Information Preservation): Structural encoding $\mathbf{SE}_i$ can preserve the topological structural information of the hypergraph.

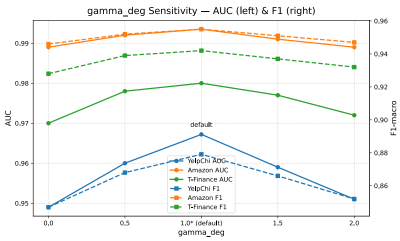
Proof: The hypergraph Laplacian matrix $\mathbf{L}_{\text{hyper}}$ contains the complete topological information of the hypergraph. Through $\mathbf{L}_{\text{hyper}} \cdot \mathbf{x}_i$, the positional information of nodes in the hypergraph structure can be encoded. Meanwhile, the weighted sum of hyperedge features further enhances the expression of structural information.

Property 2 (Degree Awareness): Structural encoding can perceive the degree information of nodes.

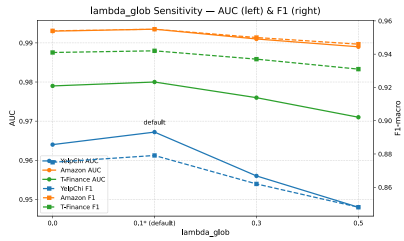
Proof: Through the weight $\frac{1}{\sqrt{\deg(e)}}$, hyperedges with higher degrees contribute less to structural encoding, ensuring that structural encoding can balance the influence of hyperedges with different degrees.

Property 3 (Structural Consistency): When hypergraph structures are similar, structural encodings are also similar.

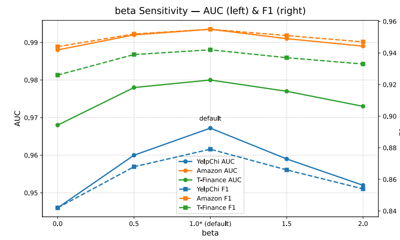
Proof: Since structural encoding mainly depends on the topological structure of the hypergraph (through the Laplacian matrix and incidence matrix), when two hypergraph structures are similar, their Laplacian matrices and incidence matrices are also similar, leading to similar structural encodings.



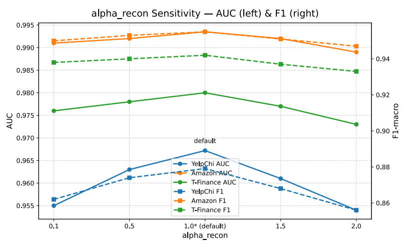
(a) Degree influence γ_{deg}



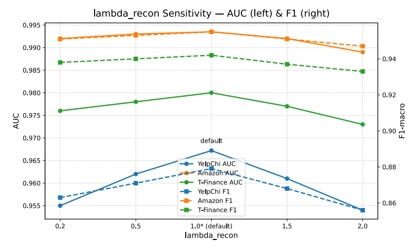
(b) Global weight λ_{glob}



(c) Distance decay β



(d) KL weight α_{recon}



(e) Reconstruction balance λ_{recon}

Figure 4: Supplementary sensitivity analysis of five additional hyperparameters on YelpChi, Amazon, and T-Finance. Each subfigure shows variations in AUC (left axis) and F1-macro (right axis) as one hyperparameter is adjusted while others remain fixed at their default values.

D Additional Hyperparameter Sensitivity Analysis

To further assess the robustness and interpretability of HGT-FD, we conduct additional sensitivity analyses on five hyperparameters that regulate different aspects of the model's behavior. The parameter γ_{deg} controls the degree-aware weighting in hyperedge aggregation, determining how strongly high-degree nodes influence the environmental context. As shown in Figure 4, model performance improves as γ_{deg} increases up to 1.0, after which both AUC and F1-macro slightly decline, suggesting that moderate degree influence achieves the best trade-off between stability and discrimination. The global weight λ_{glob} adjusts the contribution of global structural information in the positional encoding; small values encourage locality-aware aggregation and lead to better results, while larger values cause a minor performance drop due to overemphasis on global signals. The distance decay coefficient β controls how rapidly distant nodes' effects diminish. When β is small, the propagation becomes too smooth, and when it is large, the model becomes overly selective; the best performance is obtained at $\beta = 1.0$, balancing exploration and concentration. The coefficient α_{recon} scales the KL divergence term in the reconstruction loss, guiding the model to better align hyperedge features with the distribution of their constituent node features. Increasing α_{recon} moderately improves both metrics, but excessive values cause gradient instability. Similarly, λ_{recon} balances the reconstruction and classification objectives; too small a value weakens the regularization effect, while too large a value harms discriminability. The optimal setting ($\lambda_{\text{recon}} = 1.0$) achieves the best equilibrium. Overall, across all three datasets, HGT-FD exhibits strong robustness—both AUC and F1-macro fluctuate within only about 1.5% around their default values—indicating that the model maintains stable performance and generalization under diverse hyperparameter settings.

E Experimental Details

In this section, we provide the detailed statistics of the datasets used in our experiments.

E.1 Dataset Statistics

Table 2 summarizes the statistics of the three benchmark datasets. YelpChi and Amazon are multi-relational graph datasets, while T-Finance is a single-relational dataset. IR stands for Imbalance Ratio (Valid/Fraud).

Table 2: Dataset Statistics. IR denotes the Imbalance Ratio.

Dataset	#Node	IR	Relation	#Edges
YelpChi	45,954	5.9	R-U-R	49,315
			R-S-R	3,402,743
			R-T-R	573,616
Amazon	11,944	13.5	U-P-U	175,608
			U-S-U	3,566,479
			U-V-U	1,036,737
T-Finance	39,357	20.8	A-T-A	21,222,543