FINFRAUDBENCH: A HETEROGENEOUS GRAPH BENCHMARK FOR FINANCIAL FRAUD DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The increasing complexity of digital financial systems has reshaped financial fraud detection from isolated transaction classification into relational risk reasoning over interconnected financial entities. This shift has motivated graph-based fraud detection, where models identify fraudulent nodes by exploiting dependencies among customers, cards, merchants, categories, and locations. However, despite rapid progress in graph-based methods, existing public benchmarks remain misaligned with real-world financial systems in two important aspects. First, they often simplify financial ecosystems into homogeneous or single-node-type multi-relational graphs, failing to preserve the multi-entity and multi-relational nature of financial data. Second, they rarely provide large-scale heterogeneous financial graph datasets with realistic operating conditions such as extreme class imbalance and limited label availability, making it difficult to assess the practical effectiveness of current methods. To address these gaps, we present **FinFraudBench**, a heterogeneous graph benchmark for financial fraud detection. FinFraudBench contains two heterogeneous graph datasets (**CreditCard-Fraud** and **BankTrans-Fraud**) with up to 8.99M nodes and 89.23M directed typed edges. Each dataset preserves six financial entity types, fourteen directed edge types, and natural fraud rates that mirror deployment constraints. With these datasets, we establish a standardized evaluation protocol covering both ranking and imbalance-sensitive classification metrics, and evaluate representative baselines. Extensive experiments yield empirical insights into current methods' limitations and suggest promising avenues for future research. FinFraudBench is available at `https://anonymous.4open.science/r/FinFraudBench-B002`.

1 INTRODUCTION

Financial fraud detection is a central task in modern financial systems, where fraud can cause substantial economic loss and erode trust in digital payment services Bolton & Hand (2002); Phua et al. (2010). Fraud is rarely an isolated property of a single record. A suspicious transaction often becomes easier to identify when considered together with related financial entities and their interactions, such as the customer behind the transaction, the payment instrument, the receiving merchant, and the associated context. This relational nature has motivated graph-based fraud detection, where models can capture dependencies that are difficult to represent from tabular records alone in fraud analysis Akoglu et al. (2015); Ma et al. (2023); Liu et al. (2022).

Existing public graph benchmarks for fraud detection, however, often simplify this financial ecosystem before learning begins. We distinguish three graph formulations in this paper. Here, each node represents an entity instance, such as a transaction, user, account, card, merchant, or location. In a homogeneous graph, transactions, users, or accounts are usually selected as the only node type, and all edges describe one broad relation, such as a transaction between users or a payment flow between transactions Weber et al. (2019); Huang et al. (2022). A multi-relational graph still keeps a single target node type, but its relation types are often induced by typed meta paths in the original heterogeneous ecosystem, such as two target nodes connected through a shared device, shared account, shared card, or shared merchant Shi et al. (2017); Dong et al. (2017); Rayana & Akoglu (2015); McAuley & Leskovec (2013); Liu et al. (2020); Dou et al. (2020). A heterogeneous graph contains multiple node types and multiple relation types, allowing financial entities and their interactions to be represented directly Shi et al. (2017); Hu et al. (2020b). Homogeneous and multi-relational

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

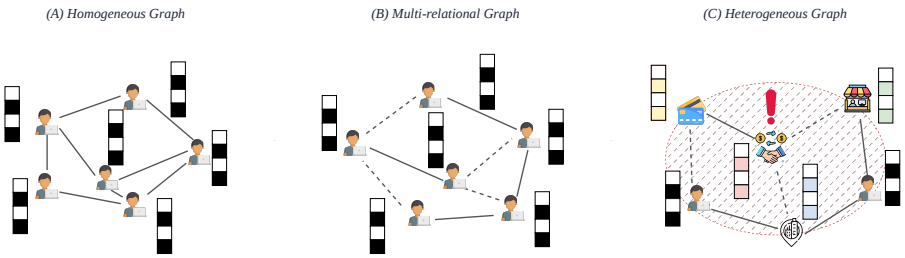


Figure 1: Comparison of three graph formulations for financial fraud detection. Homogeneous graphs use one node type and one relation type; multi-relational graphs keep a single target node type while distinguishing multiple relation types; heterogeneous graphs preserve multiple entity types, multiple relation types, and type-specific attributes.

formulations are useful abstractions and have supported important progress in graph-based fraud detection, but both lose part of the fraud entity-relation structure.

This representation gap matters for benchmark design. Financial transactions naturally involve multiple entity types, multiple relation semantics, and type-specific attribute spaces. When typed entities are folded into features, converted into target–target edges, or omitted, benchmarks no longer directly test entity- and relation-aware fraud modeling. Figure 1 clearly illustrates this distinction among homogeneous graphs, multi-relational graphs, and heterogeneous graphs considered in this work. To address this gap, we present FinFraudBench, a benchmark consisting of two heterogeneous graph datasets constructed from public financial fraud data sources Shenoy (2020); ComputingVictor (2024); the original sources are transaction-level tables rather than pre-existing graphs. To the best of our knowledge, FinFraudBench is the first benchmark to construct comparatively large heterogeneous graphs specifically for financial fraud detection from public transaction-level sources. Both datasets use a unified schema with transaction nodes as prediction targets and additional typed financial entities as contextual information. FinFraudBench releases the heterogeneous graph as the canonical representation and supports deterministic projections for feature-only, homogeneous, or multi-relational methods.

FinFraudBench also provides a reproducible evaluation protocol. Each dataset is organized into labeled training transactions, unlabeled training transactions, validation transactions, a full test split, and balanced mini-test subsets, supporting controlled comparison under limited labels and extreme class imbalance in realistic deployment settings. We further reproduce representative baselines across non-GNN, homogeneous GNN, multi-relation, fraud-oriented, and heterogeneous families. The results reveal a clear family-level pattern: methods designed for heterogeneous graphs are strongest overall, multi-relation and fraud-oriented methods remain competitive, and target-only non-GNN baselines are limited by their inability to use relational financial context encoded by the benchmark graph structure during practical financial fraud detection tasks.

The main contributions of this work are summarized as follows:

- **Heterogeneous financial fraud graph benchmarks.** We construct two heterogeneous graph datasets from public transaction-level fraud sources. The datasets preserve multiple financial entity types, multiple relation types, and type-specific attributes, while also supporting deterministic simplified views for methods that require homogeneous or multi-relational graph inputs.
- **Reproducible limited-label evaluation protocol.** We provide unified graph schemas, labeled and unlabeled training transactions, validation splits, balanced mini-test subsets, and standardized metrics to support controlled benchmarking under label scarcity and extreme class imbalance in a unified evaluation setting.
- **Empirical insights for heterogeneous fraud detection.** Our experiments show that heterogeneous graph structures preserve rich signals encoded in typed financial entities and relations, which are essential for fraud modeling. These results point to a promising direction: extending fraud-oriented designs to heterogeneous architectures, so that typed structural information and domain-specific fraud cues can be jointly exploited.

2 PRELIMINARIES AND RELATED WORK

2.1 PRELIMINARIES

Homogeneous graphs. We define the three graph formulations used in this paper. A homogeneous attributed graph is written as $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V}$ is the node set, $\mathcal{E}$ is the edge set, and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a shared node feature matrix. It assumes one node type and one edge type.

Multi-relational graphs. A multi-relational graph is a single-node-type graph with multiple relation types, written as $G = (\mathcal{V}, \mathcal{E}, \mathcal{R}, \mathbf{X}, \phi)$, where $\mathcal{R}$ is the relation-type set and $\phi : \mathcal{E} \to \mathcal{R}$ assigns each edge to a relation type. Equivalently, $\mathcal{E} = \bigcup_{r \in \mathcal{R}} \mathcal{E}_r$ with $\mathcal{E}_r = \{e \in \mathcal{E} : \phi(e) = r\}$. This formulation captures relation heterogeneity while retaining a single node type.

Heterogeneous graphs. A heterogeneous graph is written as $G = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \{\mathbf{X}^a\}_{a \in \mathcal{A}}, \tau, \phi)$, where $\mathcal{A}$ is the node-type set, $\mathcal{R}$ is the relation-type set, $\tau : \mathcal{V} \to \mathcal{A}$ assigns each node to a node type, and $\phi : \mathcal{E} \to \mathcal{R}$ assigns each edge to a relation type. For each node type $a \in \mathcal{A}$, nodes of that type are collected as $\mathcal{V}_a = \{v \in \mathcal{V} : \tau(v) = a\}$ and have their own feature matrix $\mathbf{X}^a \in \mathbb{R}^{|\mathcal{V}_a| \times d_a}$. In our constructed benchmark, this stricter heterogeneous graph setting captures both relation-level and entity-level heterogeneity.

Fraud detection task. Given a graph G in one of the above formulations, we formulate financial fraud detection as binary node classification over a target node type. Let $a^\star$ be the target type, $\mathcal{V}_{a^\star}$ be its nodes, and $\mathcal{V}_L \subseteq \mathcal{V}_{a^\star}$ be the labeled training nodes. Unlabeled target nodes may remain in the graph as relational context but do not contribute to the supervised loss. Given labels $y_v \in \{0, 1\}$, a model assigns each target node a fraud score

$$s_v = f_\theta(G, v), \quad \hat{y}_v = \mathbf{1}[s_v \geq \gamma], \tag{1}$$

where γ is a decision threshold. The supervised objective is

$$\theta^\star = \arg\min_\theta \sum_{v \in \mathcal{V}_L} \ell\big(s_v, y_v\big). \tag{2}$$

In FinFraudBench, the target type is transaction, while customers, cards, merchants, categories, and locations provide relational context.

2.2 RELATED WORK

Graph-based fraud detection. Fraud detection has been studied using tabular transaction features, anomaly detection, and graph-based learning Bolton & Hand (2002); Phua et al. (2010); Chandola et al. (2009); Breunig et al. (2000); Liu et al. (2008); Akoglu et al. (2015); Ma et al. (2023). General GNN backbones provide message-passing, convolutional, inductive, attention-based, and scalable neighborhood aggregation mechanisms Scarselli et al. (2009); Gilmer et al. (2017); Kipf & Welling (2016); Hamilton et al. (2017); Veličković et al. (2017); Chiang et al. (2019); Zeng et al. (2019); Wu et al. (2021). In fraud detection, graph-based methods represent suspicious behaviors as relational structures among transactions, accounts, users, reviews, devices, and payment flows, while fraud-oriented GNNs further introduce assumptions about noisy neighborhoods, camouflage, label scarcity, partitioned propagation, or attribute association Liu et al. (2018; 2020); Dou et al. (2020); Chen et al. (2024); Zhuo et al. (2024); Duan et al. (2025); Li et al. (2026). These works motivate relational modeling for fraud detection, but their public evaluation settings are usually built on homogeneous graphs or single-node-type multi-relational graphs.

Heterogeneous graph learning. Heterogeneous graph learning broadly studies graphs with typed nodes, typed edges, or both. To avoid ambiguity, we use multi-relational graphs for the single-node-type setting with multiple relation types, and heterogeneous graphs for the setting with multiple node types and multiple relation types. Heterogeneous information network research formalizes entity types, relation types, and meta-path semantics Shi et al. (2017); Dong et al. (2017), while neural models learn relation-specific transformations, meta-path-based attention, learned composite relations, and node- and edge-type-dependent attention Schlichtkrull et al. (2018); Wang et al. (2019); Fu et al. (2020); Yun et al. (2019); Hu et al. (2020b); Yang et al. (2023). These methods are relevant to financial fraud detection because financial risk data naturally involve different entity roles and relation semantics, yet public fraud benchmarks rarely expose these elements together with type-specific attributes.

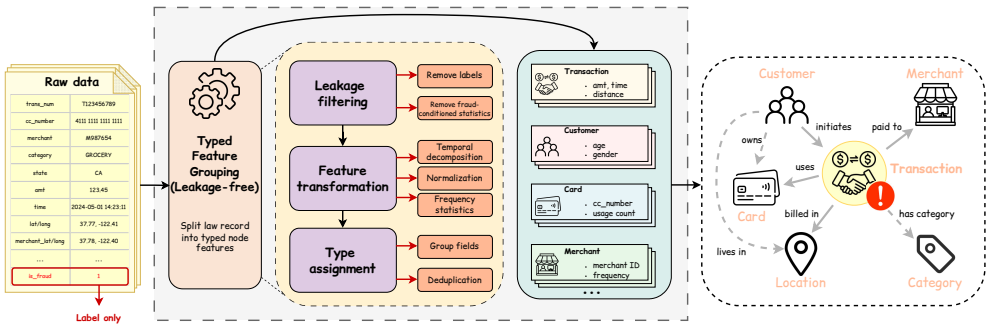


Figure 2: Overview of the graph construction pipeline. Public transaction-level tables are converted into heterogeneous graphs by mapping transactions and recurring financial objects to typed nodes, deriving semantic relations between observed entities, constructing type-specific attributes, and assigning fraud labels only to transaction nodes.

Fraud datasets and benchmarks. Public benchmarks strongly influence how fraud detection models are designed and evaluated. YelpChi and Amazon are widely used in graph-based fraud detection; although not financial datasets, they test whether models handle noisy and suspicious relational patterns Rayana & Akoglu (2015); McAuley & Leskovec (2013); Zhang et al. (2020). In common graph-fraud settings, however, they are usually treated as single-node-type multi-relational graphs, where all nodes share the same prediction target type and edge types encode alternative relations between target nodes Liu et al. (2020); Dou et al. (2020). Financial fraud benchmarks such as Elliptic and DGraph are closer to our domain, but they are representative homogeneous graphs: Elliptic connects Bitcoin transaction nodes through payment flows, while DGraph models financial users with temporal interaction edges, strong class imbalance, and many unlabeled nodes Weber et al. (2019); Huang et al. (2022). Recent efforts such as H^2GB construct large-scale heterogeneous graphs across domains, highlighting the value of realistic structural properties, but they are not specifically designed for fraud detection Lin et al. (2025). Benchmark studies further show that dataset structure shapes the modeling assumptions methods are expected to exploit Hu et al. (2020a); Dwivedi et al. (2023); Lv et al. (2021); Lin et al. (2025). The lack of public financial fraud benchmarks with heterogeneous graph structure therefore limits the development and evaluation of models that reason over multiple financial entity types, relation types, and type-specific attributes. FinFraudBench fills this gap by providing a unified benchmark with two heterogeneous graph datasets constructed from public transaction-level sources and standardized splits for node-level fraud detection.

3 Benchmark Construction

3.1 Source Data and Construction Principles

We construct CreditCard-Fraud and BankTrans-Fraud from two public tabular fraud datasets Shenoy (2020); ComputingVictor (2024). Both raw sources are transaction-level tables: each row corresponds to one transaction and contains a fraud label together with non-label fields describing the transaction itself, such as amount and time, and the observable financial context around it, such as customer- or payment-instrument identifiers, merchant and category information, and geographic or billing attributes. These fields are provided as columns in transaction records rather than as separate entity or relation tables. Our goal is to convert such tabular records into heterogeneous graphs while avoiding label leakage in graph or feature construction. Figure 2 summarizes this construction pipeline for both datasets.

The construction follows three principles. First, recurring financial objects are represented as typed nodes rather than being kept only as transaction attributes. Second, relation types follow the semantic roles between typed entities, so different entity associations are preserved as distinct edge types. Third, labels remain defined only on transaction nodes, while non-transaction nodes provide unlabeled context for limited-label fraud detection.

Table 1: Graph-scale and label statistics of the constructed datasets.

Dataset	#Trans. nodes	#Total nodes	#Context nodes	#Directed edges	#Fraud trans.	Fraud rate
CreditCard-Fraud	1.85M	1.86M	2.76K	18.53M	9,651	0.5210%
BankTrans-Fraud	8.91M	8.99M	72.14K	89.23M	13,332	0.1495%

3.2 Graph Schema, Attributes, and Labels

Both datasets share a transaction-centered heterogeneous schema with six node types and seven semantic edge types. Transaction nodes are the prediction targets, while customer, card, merchant, category, and location nodes provide typed context, where location denotes state or location fields in the source records. The seven edge types include five transaction-context associations and two context-context associations, customer–card ownership and customer–location association. Adding reverse edges yields fourteen directed edge types.

Each source-table row is mapped to one transaction node, with entity-valued fields deduplicated within their own node types. The induced typed edges preserve intermediate entities that link transactions, rather than projecting shared contexts into direct transaction–transaction edges. Node attributes are built from non-label source fields and leakage-free deterministic features, with type-specific dimensions ranging from 7 to 21. Fraud labels are attached only to transaction nodes; all other node types provide unlabeled relational context. A compact schema and feature summary is provided in Appendix A.

4 Dataset Statistics and Analysis

4.1 Graph Scale and Heterogeneity

Table 1 summarizes the graph-scale and label statistics of the two datasets. CreditCard-Fraud contains 1.86M nodes and 18.53M directed typed edges, while the larger BankTrans-Fraud contains 8.99M nodes and 89.23M directed typed edges. Both datasets share the same schema of six node types and fourteen directed edge types, but differ substantially in scale and fraud prevalence. Although transaction nodes dominate graph size, the non-transaction context nodes are central to the relational structure: they represent customers, cards, merchants, categories, and locations, and link transactions through shared financial entities and contexts instead of collapsing these connections into direct transaction–transaction edges.

4.2 Label Imbalance and Splits

Figure 3 visualizes the main transaction split proportions. Each dataset is organized into labeled training transactions, unlabeled training transactions, validation transactions, a full test split, and five balanced mini-test subsets. The labeled training portion is intentionally small overall, about 2% of transactions in each dataset: CreditCard-Fraud contains 37.0K labeled and 1.22M unlabeled training transactions, while BankTrans-Fraud contains 178.3K labeled and 6.78M unlabeled training transactions. Unlabeled training transactions remain in the graph as context, so semi-supervised and transductive methods can use their node attributes and local neighborhoods without observing their labels. The full test set preserves the natural class distribution for realistic evaluation, while balanced mini-test subsets contain equal numbers of fraudulent and non-fraudulent transactions for broader and more stable comparison under extreme class imbalance in deployment-like settings.

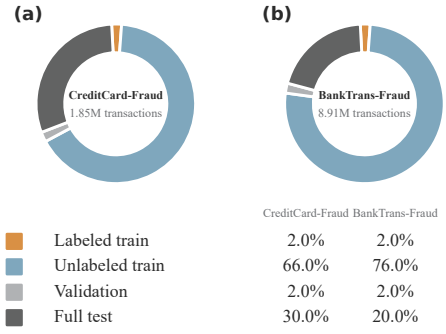


Figure 3: Transaction split proportions of the constructed fraud graphs.

5 BENCHMARK TASKS AND EVALUATION PROTOCOL

The benchmark evaluates transaction-level fraud detection under the splits defined in Section 4.2. The canonical input is the heterogeneous graph constructed in Section 3, and deterministic projections support methods that require feature-only, homogeneous, or multi-relational inputs.

5.1 TASK SETUP

Given the full graph or one of its projections, the task is binary node classification on transaction nodes. Models may use the structure and attributes available in their input view, but supervised labels are provided only for labeled training transactions. Validation labels are used only for model selection and early stopping.

5.2 METRICS AND REPORTING PROTOCOL

Our main experiments use the balanced mini-test subsets defined in Section 4.2 to compare ranking and classification behavior under equal fraud/non-fraud weight. We report six complementary metrics for each method.

AUROC and AUPRC. AUROC measures whether fraudulent transactions receive higher risk scores than legitimate transactions across thresholds Fawcett (2006). AUPRC reflects the precision–recall trade-off for the fraud class and is especially important under class imbalance Davis & Goadrich (2006); Saito & Rehmsmeier (2015). Together, they evaluate score ranking before threshold selection in deployment settings.

Accuracy and Macro-F1. Accuracy is usually misleading on naturally imbalanced fraud data, but is informative on our balanced mini-test subsets because both classes contribute equally. Macro-F1 averages the fraud and non-fraud F1 scores and is more sensitive to failures on either class.

Fraud F1 and Fraud Recall. Fraud F1 summarizes positive-class detection while penalizing excessive false positives. Fraud Recall measures the fraction of detected fraud cases and captures missed-fraud risk, but should be interpreted with AUPRC and Fraud F1.

Reporting protocol. In the main benchmark tables, we report mean performance over five balanced mini-test subsets. Unless otherwise stated, model selection uses validation AUPRC. AUROC and AUPRC are ranking metrics, while accuracy, Macro-F1, Fraud F1, and Fraud Recall are threshold-dependent classification metrics.

6 EXPERIMENTS

We evaluate representative baselines that cover different modeling assumptions on the same benchmark. We organize methods into five comparison families. MLP and the target-only LLM prompting probes are non-GNN baselines. GCN and GraphSAGE represent homogeneous graph learning on collapsed graph views Kipf & Welling (2016); Hamilton et al. (2017). R-GCN represents relation-aware message passing around transaction nodes Schlichtkrull et al. (2018); ConsisGAD is grouped with multi-relation methods because its evaluated implementation explicitly uses relation-specific graph channels on multi-relation fraud graphs Chen et al. (2024). PMP Zhuo et al. (2024) and GAAP Duan et al. (2025) represent fraud-oriented graph learning methods with assumptions tailored to graph fraud or anomaly detection. HAN, SeHGNN, and HGT represent heterogeneous graph learning methods with typed relation, meta-path, semantic, or node- and edge-type-aware message passing Wang et al. (2019); Yang et al. (2023); Hu et al. (2020b). The two LLM probes use Qwen2-7B-Instruct Yang et al. (2024): zero-shot prompting Brown et al. (2020); Kojima et al. (2022) and 4-shot in-context prompting Brown et al. (2020); Min et al. (2022). These probes serve as additional non-GNN references in our benchmark.

6.1 OVERALL BENCHMARK RESULTS

Table 2 reports method-level results averaged over five balanced mini-test subsets, while Figure 4 summarizes the best method in each family for every dataset and metric. We report AUROC,

Table 2: Overall benchmark results on balanced mini-test subsets. Results are averaged over five mini-test subsets and reported as mean ± standard deviation. The best result for each dataset and metric is highlighted with boldface and gray shading for visual emphasis.

Dataset	Method	Model family	AUROC	AUPRC	Acc.	Macro-F1	Fraud F1	Fraud Recall
CreditCard-Fraud	MLP	Non-GNN	0.880±0.006	0.905±0.005	0.754±0.005	0.739±0.006	0.678±0.009	0.517±0.010
	LLM-ZS	Non-GNN	0.659±0.005	0.688±0.009	0.551±0.003	0.452±0.004	0.219±0.006	0.126±0.004
	LLM-ICL-4	Non-GNN	0.807±0.042	0.850±0.036	0.684±0.012	0.650±0.016	0.541±0.026	0.374±0.025
	GCN	Homogeneous GNN	0.820±0.011	0.863±0.007	0.542±0.003	0.420±0.007	0.155±0.012	0.084±0.007
	GraphSAGE	Homogeneous GNN	0.905±0.006	0.928±0.004	0.768±0.010	0.756±0.012	0.700±0.017	0.542±0.020
	R-GCN	Multi-relation	0.887±0.007	0.916±0.004	0.688±0.007	0.654±0.009	0.547±0.014	0.377±0.013
	ConsisGAD	Multi-relation	0.911±0.004	0.932±0.003	0.653±0.006	0.605±0.008	0.468±0.014	0.305±0.012
	PMP	Fraud-oriented	0.868±0.008	0.903±0.005	0.699±0.007	0.670±0.009	0.571±0.013	0.400±0.012
	GAAP	Fraud-oriented	**0.924±0.004**	**0.934±0.003**	0.716±0.008	0.691±0.010	0.605±0.014	0.437±0.014
	HAN	Heterogeneous	0.903±0.004	0.927±0.002	0.686±0.008	0.652±0.011	0.542±0.017	0.372±0.016
	SeHGNN	Heterogeneous	0.903±0.005	0.928±0.003	0.758±0.006	0.743±0.007	0.681±0.011	0.517±0.014
	HGT	Heterogeneous	0.915±0.004	**0.934±0.004**	**0.832±0.008**	**0.827±0.008**	**0.799±0.011**	**0.669±0.014**
BankTrans-Fraud	MLP	Non-GNN	0.938±0.004	0.946±0.004	0.672±0.003	0.633±0.004	0.513±0.007	0.346±0.006
	LLM-ZS	Non-GNN	0.809±0.007	0.813±0.007	0.517±0.004	0.372±0.008	0.070±0.013	0.036±0.007
	LLM-ICL-4	Non-GNN	0.873±0.019	0.860±0.012	0.541±0.032	0.418±0.063	0.151±0.111	0.086±0.068
	GCN	Homogeneous GNN	0.948±0.003	0.949±0.003	0.726±0.005	0.706±0.007	0.629±0.010	0.464±0.011
	GraphSAGE	Homogeneous GNN	0.926±0.002	0.930±0.003	0.546±0.004	0.429±0.008	0.170±0.014	0.093±0.009
	R-GCN	Multi-relation	0.962±0.004	0.972±0.003	0.854±0.008	0.851±0.009	0.829±0.011	0.708±0.016
	ConsisGAD	Multi-relation	0.966±0.003	0.975±0.002	0.822±0.006	0.816±0.007	0.784±0.009	0.644±0.013
	PMP	Fraud-oriented	0.959±0.004	0.968±0.003	0.816±0.009	0.810±0.010	0.775±0.013	0.634±0.018
	GAAP	Fraud-oriented	0.949±0.005	0.954±0.004	0.643±0.004	0.591±0.006	0.445±0.010	0.286±0.008
	HAN	Heterogeneous	0.969±0.003	0.977±0.002	0.881±0.004	0.880±0.004	0.865±0.006	0.763±0.009
	SeHGNN	Heterogeneous	0.969±0.002	**0.978±0.001**	**0.884±0.003**	**0.882±0.003**	**0.869±0.004**	**0.768±0.007**
	HGT	Heterogeneous	**0.973±0.002**	**0.978±0.001**	0.779±0.006	0.768±0.007	0.717±0.010	0.559±0.012

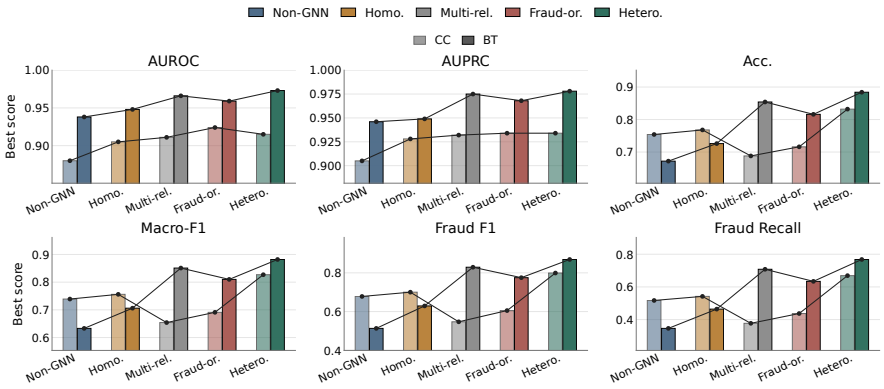


Figure 4: Family-level best results on balanced mini-test subsets. For each metric and model family, bars report the best-performing method within that family in Table 2; CC denotes CreditCard-Fraud and BT denotes BankTrans-Fraud.

AUPRC, accuracy, Macro-F1, Fraud F1, and Fraud Recall, grouping methods into non-GNN, homogeneous GNN, multi-relation, fraud-oriented, and heterogeneous families.

6.2 Representation Analysis

Figure 5 complements Figure 4 with a compact family-level mean-score view across the two datasets. The radar profiles show that heterogeneous methods are strongest overall, while multi-relation and fraud-oriented methods remain competitive and non-GNN baselines are consistently weaker.

Transaction features provide a strong but incomplete baseline. MLP reaches Macro-F1 scores of 0.739 and 0.633 on the two datasets, confirming that target-node attributes carry substantial fraud signals. However, it cannot use the customer, card, merchant, category, and location nodes exposed by the bench-

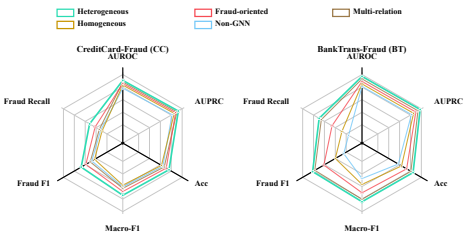


Figure 5: Family-level mean-score radar across benchmark metrics.

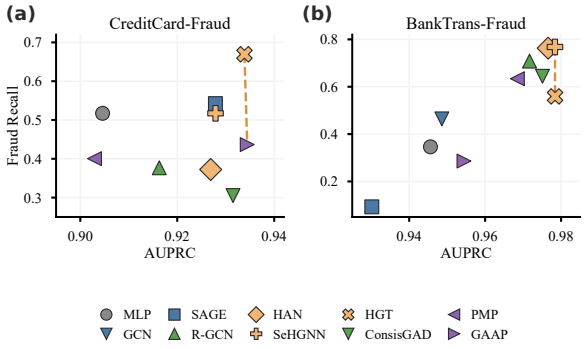


Figure 6: AUPRC–Fraud Recall frontier for trained non-LLM baselines, showing that stronger ranking performance does not always lead to higher fraud recall at the selected decision threshold. Higher values indicate better performance on both axes.

mark. As shown in Table 2, homogeneous propagation is also not consistently beneficial: GCN trails MLP on CreditCard-Fraud, and GraphSAGE improves Macro-F1 there but falls below MLP on the larger graph. This suggests that the main challenge is not merely adding edges, but learning from the heterogeneous relational structure preserved in FinFraudBench.

Heterogeneous graph models are strongest overall. HGT achieves the best accuracy, Macro-F1, Fraud F1, and Fraud Recall on CreditCard-Fraud, while GAAP slightly leads AUROC and AUPRC. On BankTrans-Fraud, HGT obtains the best AUROC and AUPRC, while SeHGNN leads the classification metrics. R-GCN also improves substantially on the larger graph, indicating that relation semantics remain useful even without full node-type heterogeneity. Overall, the results support the benchmark motivation that relation semantics, node types, and type-specific attributes provide complementary fraud signals Schlichtkrull et al. (2018); Wang et al. (2019); Yang et al. (2023); Hu et al. (2020b).

Relation-aware and fraud-oriented biases remain useful but are not sufficient alone. GAAP obtains the best AUROC and AUPRC on CreditCard-Fraud, while ConsisGAD and PMP remain competitive on BankTrans-Fraud Chen et al. (2024); Zhuo et al. (2024); Duan et al. (2025). Their gap to the strongest heterogeneous models on threshold-dependent classification metrics suggests that relation channels or fraud-oriented assumptions should be combined with explicit entity-type modeling. Several methods obtain high AUPRC but differ substantially in Fraud Recall; Figure 6 illustrates this metric mismatch and supports reporting both ranking metrics and threshold-dependent classification metrics. The LLM prompting baselines are weaker than most graph learning baselines, although 4-shot in-context prompting consistently improves over zero-shot prompting on both datasets, providing a target-only non-GNN reference point for the benchmark.

6.3 Scalability and Stability Analysis

Scalability is part of the benchmark rather than a separate engineering detail. CreditCard-Fraud contains 18.53M directed typed edges, and BankTrans-Fraud contains 89.23M, so practical methods must combine predictive quality with sampling or batching strategies. This is especially relevant for graph-based baselines, which can use unlabeled training transactions as context but must do so without full-batch propagation. The balanced mini-test protocol yields stable estimates for most trained baselines across the five subsets, while the two LLM prompting probes show larger standard deviations and are therefore less stable than the graph-learning methods.

7 Discussion

7.1 Key Findings

The experiments suggest that FinFraudBench is useful not only for ranking baselines, but also for diagnosing which parts of the released schema a method can exploit. Transaction features, relation

channels, entity types, type-specific attributes, and fraud-oriented assumptions contribute complementary signals for multi-entity financial fraud detection.

Heterogeneous graph design gives the strongest overall signal. Across experiments, heterogeneous methods provide the most consistent gains. This suggests that FinFraudBench exposes useful signals through entity types, relation semantics, and type-specific attributes. Homogeneous or single-node-type multi-relational projections simplify this structure and can discard part of the original fraud entity-relation information.

Fraud-oriented designs can build on heterogeneous graphs. Multi-relation and fraud-oriented methods remain competitive on several metrics, showing that relation channels and fraud-oriented inductive biases are useful for financial fraud detection. However, their remaining gap to the strongest heterogeneous methods suggests that these cues are most promising when combined with entity types and type-specific attributes in heterogeneous financial graphs.

7.2 Limitations and Future Work

FinFraudBench is constructed from public transaction-level tables, so its graph signals are limited by the fields exposed in those sources and may not fully reflect richer real-world fraud contexts. Future work can extend the benchmark with more complete operational signals and develop fraud-oriented models designed directly for heterogeneous fraud graphs under limited labels and extreme class imbalance in realistic deployment settings.

8 Conclusion

We presented FinFraudBench, a benchmark with two heterogeneous graph datasets for financial fraud detection constructed from public transaction-level fraud sources. The proposed graphs preserve multiple financial entity types, relation types, type-specific attributes, limited labels, and realistic fraud imbalance. Our baseline reproduction provides family-level insights into how model designs use heterogeneous fraud signals. The results show that heterogeneous GNNs perform best. Multi-relation and fraud-oriented methods remain competitive but less consistent, highlighting the value of preserving heterogeneous financial structure.

Data Availability

The original tabular datasets used in this work are publicly available on Kaggle: `https://www.kaggle.com/datasets/kartik2112/fraud-detection` and `https://www.kaggle.com/datasets/computingvictor/transactions-fraud-datasets`.

Ethics and Reproducibility Statement

FinFraudBench is built from public transaction-level data and does not involve human-subject experiments or new data collection. The benchmark is intended for fraud detection research rather than operational decision making without additional validation. All preprocessing, split construction, and reported experiments are described in the paper and appendix. The code repository has been released openly, and the constructed datasets are available on Hugging Face with fixed random splits to support reproduction of the main results.

AI Use Statement

We used AI-assisted tools to support literature search and survey, manuscript writing, language polishing, and formatting during paper preparation. These tools were not used to design the proposed benchmark, construct the datasets, define the experimental protocol, or draw scientific conclusions. All scientific claims, dataset construction, experimental analysis, and final wording were carefully reviewed and approved by the authors.

REFERENCES

Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688, 2015.

Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical science*, 17(3): 235–255, 2002.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *The twelfth international conference on learning representations*, 2024.

Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 257–266, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330925. URL https://doi.org/10.1145/3292500.3330925.

ComputingVictor. Financial transactions dataset: Analytics. Kaggle dataset, 2024. URL https://www.kaggle.com/datasets/computingvictor/transactions-fraud-datasets. Accessed: 2026-07-18.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 233–240, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pp. 135–144, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098036. URL https://doi.org/10.1145/3097983.3098036.

Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 315–324, 2020.

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, pp. 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.

Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2005.10.010. URL https://www.sciencedirect.com/science/article/pii/S016786550500303X. ROC Analysis in Pattern Recognition.

Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*, pp. 2331–2341, 2020.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

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

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020a.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pp. 2704–2710, 2020b.

Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. Dgraph: A large-scale financial dataset for graph anomaly detection. *Advances in Neural Information Processing Systems*, 35:22765–22777, 2022.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Yuan Li, Jun Hu, Bryan Hooi, Bingsheng He, and Cheng Chen. Dgp: A dual-granularity prompting framework for fraud detection with graph-enhanced llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15171–15179, 2026.

Junhong Lin, Xiaojie Guo, Shuaicheng Zhang, Yada Zhu, and Julian Shun. When heterophily meets heterogeneity: Challenges and a new large-scale graph benchmark. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5607–5618, 2025.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pp. 413–422. IEEE, 2008.

Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, et al. Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. *Advances in Neural Information Processing Systems*, 35:27021–27035, 2022.

Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 1569–1572, 2020.

Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 2077–2085, 2018.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 1150–1160, 2021.

Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12012–12038, 2023. doi: 10.1109/TKDE.2021.3118815.

Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pp. 897–908, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320351. doi: 10.1145/2488388.2488466. URL https://doi.org/10.1145/2488388.2488466.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759/.

Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.

Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 985–994, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783370. URL https://doi.org/10.1145/2783258.2783370.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Kartik Shenoy. Credit card transactions fraud detection dataset. *Kaggle. Available online: https://www. kaggle. com/datasets/kartik2112/fraud-detection (accessed on 29 July 2023)*, 2020.

Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2017. doi: 10.1109/TKDE.2016.2598561.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pp. 2022–2032, 2019.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin

Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10816–10824, 2023.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

Shijie Zhang, Hongzhi Yin, Tong Chen, Quoc Viet Nguyen Hung, Zi Huang, and Lizhen Cui. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pp. 689–698, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401165. URL https://doi.org/10.1145/3397271.3401165.

Wei Zhuo, Zemin Liu, Bryan Hooi, Bingsheng He, Guang Tan, Rizal Fathony, and Jia Chen. Partitioning message passing for graph fraud detection. *arXiv preprint arXiv:2412.00020*, 2024.

A Additional Dataset Details

A.1 Type-Specific Feature Fields

Table 3 lists representative non-label feature groups used to build node attributes. The two graphs share the same node and relation schema, but the feature dimension of the same node type can differ across datasets because the two source tables expose different raw fields. In all cases, both graphs avoid label leakage in graph and feature construction.

Table 3: Representative non-label feature groups by node type.

Node type	Representative feature groups
transaction	amount, time fields, location fields, distance or transaction context
customer	profile encodings, location fields, transaction counts and amount statistics
card	billing/context encodings, transaction counts, amount statistics, unique entity counts
merchant	category encoding, transaction counts, amount statistics, unique customer/card/state counts
category	transaction counts, amount statistics, unique merchant/customer/card counts
location	transaction counts, amount statistics, unique entity counts, location/population summaries

A.2 Detailed Transaction Splits

Table 4 reports the split statistics behind the compact split summary in the main paper. The unlabeled training split keeps transaction nodes available as graph context while hiding their labels from supervised training.

Table 4: Detailed transaction split statistics.

Dataset	Split	#Transactions	#Fraud	#Benign	Fraud rate
CreditCard-Fraud	train_labeled	37.05K	214	36.83K	0.5776%
	train_unlabeled	1.22M	7.08K	1.22M	0.5789%
	validation	37.05K	214	36.83K	0.5776%
	full_test	555.72K	2.15K	553.57K	0.3860%
BankTrans-Fraud	train_labeled	178.30K	268	178.03K	0.1503%
	train_unlabeled	6.78M	10.13K	6.77M	0.1495%
	validation	178.30K	268	178.03K	0.1503%
	full_test	1.78M	2.67K	1.78M	0.1495%

A.3 Complete Balanced Mini-Test Results

Tables 5 and 6 report the complete balanced mini-test results, including Fraud Precision, which is omitted from the main table to keep the main comparison compact.

Table 5: Complete balanced mini-test results on CreditCard-Fraud. Values are means and standard deviations across five mini-test subsets.

Method	AUROC	AUPRC	Acc.	Macro-F1	Fraud F1	Fraud Prec.	Fraud Recall
MLP	0.8798 ± 0.0057	0.9046 ± 0.0052	0.7539 ± 0.0050	0.7393 ± 0.0059	0.6775 ± 0.0087	0.9822 ± 0.0020	0.5172 ± 0.0103
GCN	0.8199 ± 0.0107	0.8632 ± 0.0067	0.5416 ± 0.0033	0.4204 ± 0.0065	0.1554 ± 0.0115	0.9861 ± 0.0099	0.0844 ± 0.0068
GraphSAGE	0.9046 ± 0.0059	0.9279 ± 0.0044	0.7681 ± 0.0102	0.7556 ± 0.0118	0.7003 ± 0.0167	0.9890 ± 0.0044	0.5422 ± 0.0195
R-GCN	0.8870 ± 0.0072	0.9163 ± 0.0044	0.6876 ± 0.0066	0.6542 ± 0.0088	0.5466 ± 0.0137	0.9957 ± 0.0030	0.3768 ± 0.0128
HAN	0.9025 ± 0.0038	0.9269 ± 0.0024	0.6858 ± 0.0080	0.6515 ± 0.0108	0.5422 ± 0.0170	0.9978 ± 0.0023	0.3724 ± 0.0160
SeHGNN	0.9026 ± 0.0048	0.9279 ± 0.0026	0.7575 ± 0.0059	0.7426 ± 0.0073	0.6807 ± 0.0111	0.9959 ± 0.0041	0.5172 ± 0.0137
HGT	0.9150 ± 0.0035	0.9339 ± 0.0036	0.8317 ± 0.0077	0.8271 ± 0.0082	0.7990 ± 0.0106	0.9911 ± 0.0042	0.6694 ± 0.0140
ConsisGAD	0.9110 ± 0.0042	0.9315 ± 0.0031	0.6526 ± 0.0058	0.6049 ± 0.0084	0.4676 ± 0.0135	1.0000 ± 0.0000	0.3052 ± 0.0115
PMP	0.8683 ± 0.0075	0.9029 ± 0.0053	0.6992 ± 0.0068	0.6697 ± 0.0086	0.5709 ± 0.0129	0.9949 ± 0.0048	0.4004 ± 0.0120
GAAP	0.9237 ± 0.0037	0.9344 ± 0.0033	0.7155 ± 0.0077	0.6915 ± 0.0096	0.6054 ± 0.0142	0.9873 ± 0.0062	0.4366 ± 0.0142
LLM-ZS	0.6591 ± 0.0046	0.6877 ± 0.0086	0.5510 ± 0.0032	0.4519 ± 0.0039	0.2189 ± 0.0059	0.8412 ± 0.0249	0.1258 ± 0.0036
LLM-ICL-4	0.8067 ± 0.0417	0.8495 ± 0.0360	0.6839 ± 0.0120	0.6501 ± 0.0164	0.5414 ± 0.0260	0.9841 ± 0.0051	0.3739 ± 0.0247

Table 6: Complete balanced mini-test results on BankTrans-Fraud. Values are means and standard deviations across five mini-test subsets.

Method	AUROC	AUPRC	Acc.	Macro-F1	Fraud F1	Fraud Prec.	Fraud Recall
MLP	0.9384 ± 0.0043	0.9456 ± 0.0041	0.6718 ± 0.0030	0.6329 ± 0.0042	0.5133 ± 0.0068	0.9926 ± 0.0032	0.3462 ± 0.0063
GCN	0.9476 ± 0.0032	0.9486 ± 0.0030	0.7257 ± 0.0052	0.7055 ± 0.0065	0.6285 ± 0.0099	0.9732 ± 0.0051	0.4642 ± 0.0111
GraphSAGE	0.9263 ± 0.0019	0.9302 ± 0.0026	0.5460 ± 0.0043	0.4286 ± 0.0082	0.1696 ± 0.0144	0.9915 ± 0.0093	0.0928 ± 0.0086
R-GCN	0.9615 ± 0.0039	0.9717 ± 0.0028	0.8542 ± 0.0082	0.8510 ± 0.0087	0.8292 ± 0.0112	1.0000 ± 0.0000	0.7084 ± 0.0164
HAN	0.9687 ± 0.0028	0.9766 ± 0.0021	0.8813 ± 0.0043	0.8796 ± 0.0044	0.8654 ± 0.0055	0.9995 ± 0.0012	0.7630 ± 0.0088
SeHGNN	0.9689 ± 0.0019	0.9784 ± 0.0012	0.8838 ± 0.0032	0.8822 ± 0.0034	0.8686 ± 0.0041	0.9995 ± 0.0007	0.7680 ± 0.0065
HGT	0.9729 ± 0.0020	0.9784 ± 0.0009	0.7793 ± 0.0062	0.7680 ± 0.0071	0.7170 ± 0.0100	0.9989 ± 0.0016	0.5592 ± 0.0116
ConsisGAD	0.9657 ± 0.0031	0.9751 ± 0.0019	0.8221 ± 0.0064	0.8163 ± 0.0071	0.7835 ± 0.0095	1.0000 ± 0.0000	0.6442 ± 0.0129
PMP	0.9592 ± 0.0036	0.9685 ± 0.0027	0.8163 ± 0.0090	0.8099 ± 0.0099	0.7752 ± 0.0134	0.9978 ± 0.0009	0.6340 ± 0.0179
GAAP	0.9493 ± 0.0049	0.9544 ± 0.0042	0.6427 ± 0.0039	0.5907 ± 0.0060	0.4447 ± 0.0099	0.9973 ± 0.0037	0.2862 ± 0.0083
LLM-ZS	0.8087 ± 0.0072	0.8127 ± 0.0074	0.5174 ± 0.0041	0.3720 ± 0.0076	0.0697 ± 0.0131	0.9587 ± 0.0504	0.0362 ± 0.0070
LLM-ICL-4	0.8725 ± 0.0186	0.8598 ± 0.0120	0.5406 ± 0.0321	0.4179 ± 0.0628	0.1513 ± 0.1111	0.9561 ± 0.0144	0.0856 ± 0.0678

A.4 Mini-Test Stability

Figure 7 reports the coefficient of variation across the five balanced mini-test subsets for each metric and trained non-LLM baseline. The figure complements the averaged results in the main paper by showing that the balanced mini-test protocol gives stable estimates for most graph-learning baselines.

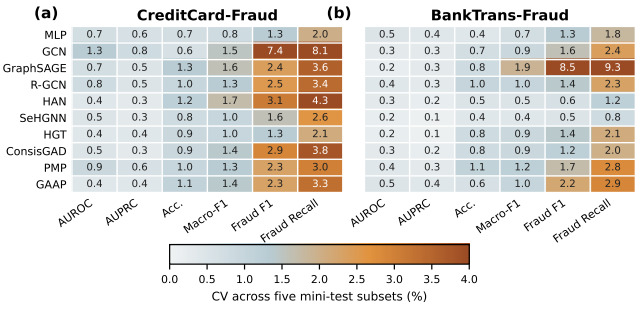


Figure 7: Stability of balanced mini-test evaluation. Heatmaps report the coefficient of variation across five mini-test subsets; lighter cells indicate more stable estimates.