Explainable Graph Neural Networks for Detecting Coordinated Fraud in Digital Transaction Networks

[1]Kismat Chhillar, [2]Anil Kewat
[1]Assistant Professor, [2]Assistant Professor
[1]Department of Mathematical Sciences and Computer Applications
[1]Bundelkhand University, Jhansi, India

***Abstract*—Coordinated fraud in digital transaction networks involves interconnected accounts that distribute illicit activity across multiple transfers, making suspicious behaviour difficult to identify through isolated transaction analysis. This paper proposes an explainable graph neural network framework for detecting such activity by jointly examining account characteristics, transaction behaviour, and relationships among participants. Digital transactions are represented as a directed graph in which nodes correspond to accounts and edges represent fund transfers, with features capturing transaction amounts, frequency, timing, and account activity. A graph neural network learns neighbourhood representations to estimate account-level fraud risk, while suspicious connections among high-risk accounts are examined to identify candidate fraud groups. An explanation module highlights the account features and transaction subgraphs that contribute most strongly to individual predictions, providing investigators with interpretable evidence for reviewing flagged activity. The proposed evaluation compares the framework with conventional machine learning and graph-based baselines using precision, recall, F1-score, and precision–recall area under the curve. Explanation quality is assessed through fidelity, sparsity, and stability, while chronological data partitioning limits information leakage. The study investigates whether relational learning can improve the identification of coordinated fraudulent behaviour while producing concise and useful explanations. The framework is intended to support transparent fraud screening and human investigation, with particular attention to incomplete coordination labels, changing transaction patterns, and the computational demands of large transaction networks.**

***Index Terms*—Coordinated Fraud, Graph Neural Networks, Explainable AI, Digital Transactions, Fraud Detection.**

I. Introduction

Digital payment services, online banking, and electronic commerce depend on the reliable processing of transactions between large numbers of interconnected participants. Protecting these services requires fraud detection systems that can identify suspicious activity while limiting unnecessary interruptions to legitimate users. This task becomes difficult when fraudulent transactions are rare, account behaviour changes over time, and confirmed fraud labels become available only after investigation. Dal Pozzolo et al. [1] demonstrate the importance of considering class imbalance, evolving transaction streams, and delayed supervision when designing realistic fraud detection systems. These challenges motivate approaches that examine both the characteristics of individual transactions and the wider relationships through which suspicious activity develops. Within this context, the present study investigates an explainable graph neural network framework for identifying potentially coordinated fraud and presenting the evidence underlying its predictions.

1.1.Digital Transaction Networks and Coordinated Fraud

Coordinated fraud involves multiple accounts whose combined activity serves a fraudulent objective, even when individual transactions appear relatively ordinary. For example, several accounts may transfer small amounts to a shared beneficiary, distribute incoming funds across intermediaries, or participate in

repeated transfer sequences within a short period. Analysing these activities separately can obscure the relationships that make the overall pattern suspicious. A graph representation provides a practical way to preserve such connections by representing participants or transactions as nodes and their interactions as edges. Research by Weber et al. [2] on illicit Bitcoin transaction classification illustrates the value of examining financial activity through transaction graphs and comparing relational methods with conventional classifiers. Although money laundering and coordinated payment fraud are distinct problems, this work provides a relevant methodological foundation for investigating connected financial behaviour. In the present study, coordination is treated as a pattern requiring supporting evidence, since a connection between accounts alone does not establish fraudulent collaboration.

1.2.Limitations of Individual Transaction Analysis and the Role of Graph Learning

Conventional fraud detection commonly relies on transaction attributes and account-level summaries, including transfer amount, transaction frequency, and deviations from previous behaviour. These features remain useful, but models operating on independent records may require substantial manual feature engineering to capture relationships extending across several accounts. Graph neural networks offer an alternative by learning account representations through the aggregation of information from connected neighbours. GraphSAGE, introduced by Hamilton et al. [3], provides an inductive framework for generating representations from node features and sampled neighbourhoods, making it relevant to networks where new accounts continually appear. Graph attention networks, proposed by Veličković et al. [4], further allow neighbouring nodes to receive different learned weights during aggregation. These mechanisms motivate the use of graph learning to examine whether an account's surroundings add predictive information beyond its own transaction history. Nevertheless, fraud networks present additional difficulties because suspicious accounts may deliberately resemble legitimate users or interact with legitimate participants. CARE-GNN addresses feature and relation camouflage, demonstrating why indiscriminate neighbourhood aggregation can weaken fraud detection [5]. The proposed study therefore evaluates the contribution of relational information rather than assuming that graph connectivity will necessarily improve performance.

1.3.Need for Explainable Fraud Detection

A fraud risk score can help prioritize investigations, but investigators also need to understand which account characteristics and transaction relationships influenced that score. In coordinated fraud screening, useful explanations should reveal a manageable set of relevant accounts and transfers, allowing an analyst to examine the surrounding evidence without reviewing an entire transaction network. The xFraud framework developed by Rao et al. [6] establishes an important precedent by combining graph-based transaction fraud prediction with an explanation component intended to support business analysis. More generally, GNNExplainer identifies compact subgraphs and subsets of node features that influence a trained graph model's predictions [7]. PGExplainer extends this direction through a parameterized explanation mechanism that can generate explanations across multiple instances [8]. These approaches provide a basis for translating complex graph predictions into inspectable evidence. However, an explanation of model behaviour does not establish the actual cause of a transaction or prove fraudulent intent. Its value in the proposed framework lies in making the model's reasoning accessible for further investigation and helping analysts identify potentially misleading predictions.

1.4.Research Objective and Contributions

The objective of this paper is to develop and evaluate a compact framework that connects account-level fraud prediction, candidate fraud-group identification, and graph-based explanations. Existing explainable fraud detection research means that combining a GNN with an explainer is not, by itself, a sufficient novelty claim. Instead, the study focuses on whether explanations can expose transaction relationships relevant to suspected coordination while remaining faithful to the model's predictions. The taxonomic survey by Yuan et al. [9] provides a foundation for selecting graph explanation methods and considering their evaluation requirements. Building on this foundation, the paper makes three contributions: first, it constructs a directed transaction graph that combines account activity with transaction relationships available at prediction time; second, it uses a compact GNN to estimate account risk and examines suspicious connections among

flagged accounts to identify candidate groups; and third, it generates feature and subgraph explanations and evaluates their fidelity, sparsity, and stability alongside detection performance. Chronological evaluation and comparisons with non-graph baselines are included to assess the practical contribution of relational learning. Where verified coordination labels are unavailable, candidate groups are presented as investigative findings or explicitly simulated scenarios, rather than confirmed fraud rings.

The remainder of this paper is organized as follows. Section 2 reviews conventional fraud detection, graph-based learning, and graph explainability methods. Section 3 presents the proposed methodology, including transaction graph construction, GNN-based risk prediction, candidate fraud-group identification, and explanation generation. Section 4 describes the dataset, preprocessing procedures, chronological evaluation protocol, baseline models, and performance measures. Section 5 presents and discusses the detection results, ablation analysis, explanation quality, and an illustrative transaction-subgraph case study, together with the study's limitations. Finally, Section 6 concludes the paper and section 7 identifies directions for future research.

II. Related Work

Research on digital transaction fraud detection has developed from statistical classification and manually designed rules toward models that learn behavioural sequences and relationships among participants. West and Bhattacharya [10] reviewed computational intelligence approaches to financial fraud detection, highlighting the variety of fraud settings and the importance of selecting methods appropriate to the available data. Their review provides a useful starting point for distinguishing transaction classification from the investigation of connected fraudulent activity. For the present study, the literature is organized into three areas: conventional fraud detection, graph-based fraud detection, and explainability. Together, these areas establish the methodological foundations for assessing account risk, identifying suspicious relationships, and presenting predictions in a form that investigators can examine. researchoutput.

2.1.Conventional Fraud Detection Approaches

Conventional fraud detection methods estimate risk from transaction attributes, account histories, and manually constructed behavioural indicators. Bhattacharyya et al. [11] compared Random Forest, Support Vector Machine, and logistic regression for credit card fraud detection, providing evidence for the usefulness of supervised classification and establishing relevant non-graph baselines. Ngai et al. [12] organized financial fraud detection research through a classification framework covering fraud categories and data mining techniques, showing that method selection must account for differences in problem formulation and available information. Moving beyond independent transaction records, Jurgovsky et al. [13] investigated sequence classification for credit card fraud detection, demonstrating the relevance of modelling the order of transactions within behavioural histories. Carcillo et al. [14] examined the integration of unsupervised anomaly information with supervised classifiers, offering a way to supplement labelled examples with information about unusual activity. Class imbalance represents another methodological concern: SMOTE, introduced by Chawla et al. [15], generates synthetic minority examples through feature-space interpolation, although applying this strategy to transaction graphs would additionally require preserving meaningful account relationships and temporal constraints. Evaluation must also reflect the operational importance of the fraud class. Saito and Rehmsmeier [16] explain why precision–recall analysis is particularly informative when positive examples are rare, supporting its inclusion alongside threshold-based measures. Collectively, these studies establish strong approaches to behavioural classification and evaluation, but coordinated fraud requires an additional examination of relationships across accounts. Sequence models capture transaction order, while ordinary tabular classifiers capture engineered attributes; neither formulation automatically represents the wider network through which multiple participants interact.

2.2.Graph-Based Fraud Detection

Graph-based methods incorporate relationships directly into the learning process, allowing predictions to depend on both an entity's attributes and its surrounding network. Kipf and Welling [17] introduced a graph

convolutional approach to semi-supervised node classification, establishing a widely used foundation for combining node features with graph structure. Schlichtkrull et al. [18] extended graph convolution to multi-relational data through relational graph convolutional networks, providing a methodological basis for distinguishing different relationship types. Although these foundational models were not developed specifically for transaction fraud, their representation mechanisms are relevant to networks containing transfers, shared devices, and other account connections. Fraud-specific research subsequently addressed difficulties that arise when connected entities differ in their characteristics or labels. Liu et al. [19] proposed GraphConsis to handle context, feature, and relation inconsistencies through context representations, consistency-based neighbour sampling, and relation attention. Wang et al. [20] developed SemiGNN to combine labelled and unlabelled users within multi-view financial networks, using hierarchical attention to integrate information from neighbours and network views. Temporal change is addressed by EvolveGCN, in which Pareja et al. [21] use recurrent mechanisms to evolve graph convolution parameters across graph snapshots. Meanwhile, Liu et al. [22] introduced PC-GNN to address imbalanced graph learning through the selection of training nodes and their neighbourhoods. These contributions show that graph fraud detection requires careful treatment of relationship meaning, neighbourhood quality, temporal evolution, and minority-class representation. However, accurate classification of individual entities does not by itself validate the recovery of a coordinated fraud group. The present study therefore treats account-risk prediction and candidate-group identification as related but distinct tasks, with group-level conclusions constrained by the availability of verified coordination labels.

2.3.Explainability Methods and Research Gap

Explainability research provides techniques for examining which input information influences a prediction, but different methods answer different investigative questions. Ribeiro et al. [23] introduced LIME, which approximates a classifier locally using an interpretable surrogate, while Lundberg and Lee [24] proposed SHAP as a framework for additive feature attribution. These approaches provide useful foundations for explaining account attributes, although standard feature-based explanations require adaptation to represent transaction relationships and graph structure. Attention weights also require careful interpretation. Jain and Wallace [25], through experiments on natural language processing models, showed that attention distributions can differ substantially while producing similar predictions; this finding motivates testing explanation faithfulness rather than assuming that attention alone supplies a reliable explanation in a transaction model. Graph-specific methods address structural information more directly. Yuan et al. [26] proposed SubgraphX, which searches for explanatory subgraphs using Monte Carlo tree search and Shapley-value-based scoring. Huang et al. [27] introduced GraphLIME, which uses a nonlinear local surrogate and feature selection to identify attributes relevant to a node prediction. Schlichtkrull et al. [28] developed GraphMask to identify edges that can be removed from a trained GNN's computation without substantially changing its predictions, providing another perspective on the structural information used by a model. These methods offer complementary explanations at the feature, edge, and subgraph levels, but transferring them to coordinated fraud screening requires domain-specific validation. An influential transaction may explain a model's output without establishing deliberate cooperation between account holders, and a concise explanation may omit contextual information needed by an investigator. The research opportunity addressed here is therefore a focused evaluation of how account-level predictions and explanatory transaction subgraphs can jointly support candidate-group investigation. The proposed framework examines detection performance together with explanation fidelity, sparsity, and stability, while keeping model explanations distinct from confirmed evidence of fraudulent coordination.

Table 1: Literature Review

Ref.	Author(s)	Method / Focus	Main Contribution	Relevance and Research Gap
[10]	West and Bhattacharya	Review of intelligent financial	Reviews computational intelligence methods	Provides background but does not develop a framework for

		fraud detection	across financial fraud settings.	explaining coordinated transaction fraud.
[11]	Bhattacharyya et al.	Random Forest, SVM, and logistic regression	Compares supervised classifiers for credit card fraud detection.	Establishes conventional baselines; inter-account relationships require explicit feature engineering.
[12]	Ngai et al.	Classification framework and literature review	Categorizes financial fraud research by fraud type and data mining technique.	Provides a foundation for method selection rather than a coordinated fraud detection model.
[13]	Jurgovsky et al.	Transaction sequence classification	Examines sequential learning for modelling transaction behaviour.	Captures transaction order but does not explicitly represent relationships across multiple accounts.
[14]	Carcillo et al.	Combined supervised and unsupervised learning	Supplements supervised fraud classification with anomaly-based information.	Supports detection of unusual behaviour; coordinated network patterns need additional modelling.
[15]	Chawla et al.	SMOTE	Generates synthetic minority-class examples to address imbalance.	Relevant to rare fraud examples, but graph applications must preserve relational and temporal validity.
[16]	Saito and Rehmsmeier	Precision–recall evaluation	Explains the usefulness of precision–recall analysis for imbalanced classification.	Supports fraud evaluation but does not address detection architecture or explanation quality.
[17]	Kipf and Welling	Graph Convolutional Networks	Combines node attributes and graph structure for semi-supervised classification.	Provides a graph-learning foundation; fraud-specific neighbourhood and imbalance issues need adaptation.
[18]	Schlichtkrull et al.	Relational Graph Convolutional Networks	Learns representations from graphs containing multiple relationship types.	Supports heterogeneous account connections but does not specifically investigate coordinated fraud.
[19]	Liu et al.	GraphConsis	Addresses context, feature, and relation inconsistencies during neighbourhood aggregation.	Improves fraud-oriented graph learning; candidate-group recovery and explanations require separate evaluation.
[20]	Wang et al.	SemiGNN	Integrates labelled and unlabelled users through multi-view graphs and hierarchical attention.	Supports financial fraud detection under limited labels; attention-based interpretations require fidelity checks.
[21]	Pareja et al.	EvolveGCN	Evolves graph convolution parameters to model	Supports temporal network modelling but does not provide a dedicated coordinated fraud explanation mechanism.

			changing graph structures.	
[22]	Liu et al.	PC-GNN	Addresses imbalanced graph classification through node and neighbourhood selection.	Relevant to minority fraud detection; group identification and explanation quality remain distinct tasks.
[23]	Ribeiro et al.	LIME	Explains individual predictions using a local interpretable surrogate.	Useful for feature-level interpretation; transaction graph structure requires adaptation.
[24]	Lundberg and Lee	SHAP	Provides an additive framework for attributing predictions to input features.	Supports account-feature explanations; structural dependencies need graph-aware treatment.
[25]	Jain and Wallace	Evaluation of attention as explanation	Shows that attention weights do not necessarily faithfully explain predictions in tested NLP models.	Motivates validating attention-based explanations rather than assuming their reliability.
[26]	Yuan et al.	SubgraphX	Identifies explanatory subgraphs using Monte Carlo tree search and Shapley-based scoring.	Relevant to suspicious transaction subgraphs; computational cost and fraud-domain usefulness require evaluation.
[27]	Huang et al.	GraphLIME	Explains node predictions through local nonlinear feature selection.	Supports account-attribute interpretation but does not directly identify coordinated transaction groups.
[28]	Schlichtkrull et al.	GraphMask	Identifies graph edges that can be removed without substantially changing predictions.	Supports analysis of influential connections; retained edges explain model behaviour rather than prove fraud.

III. Proposed Methodology

The proposed methodology combines transaction graph construction, graph neural network-based risk prediction, candidate fraud-group identification, and explanation generation within a unified analytical framework. Its purpose is to identify accounts whose suspiciousness becomes apparent through their relationships with other participants, while providing interpretable evidence for investigating the resulting alerts. The framework first converts transaction records into a directed graph containing account attributes and transaction information. A compact graph neural network then learns account representations by combining individual behaviour with information from connected accounts. Predicted account risks are subsequently examined alongside transaction timing and structural patterns to identify potentially coordinated groups. Finally, an explanation module highlights the features and transaction relationships that influence each prediction. All processing follows a chronological design so that predictions use only information available at the time of assessment. The following subsections describe the four stages of the proposed framework.

3.1.Transaction Graph Construction

Digital transaction records are represented as a directed, attributed multigraph ($G_t = (V_t, E_t, X_t, Z_t)$), constructed using transactions observed within a predefined historical window ending at assessment time (t). Here, (V_t) denotes accounts, (E_t) denotes individual transfers between accounts, (X_t) contains account features, and (Z_t) contains transaction attributes. Multiple edges between the same accounts preserve repeated transfers, while edge direction distinguishes the sender from the recipient. Account features include incoming and outgoing transaction counts, total transferred amounts, average transaction value, number of unique counterparties, account age where available, and the ratio between incoming and outgoing funds. Transaction attributes include amount, timestamp, direction, and elapsed time relative to the assessment point. Data preparation removes duplicate records, resolves account identifiers, handles missing values, and applies suitable transformations to highly skewed monetary attributes. Feature scaling parameters are estimated from training data and subsequently applied to validation and test observations. Labels are assigned according to the dataset's documented meaning: transaction-level labels are not automatically treated as account-level or group-level ground truth. Where account labels must be derived, the aggregation rule and observation horizon are explicitly stated. Historical graph windows are constructed independently for each assessment period, preventing future transfers or subsequently confirmed fraud outcomes from entering earlier predictions. This representation retains the transaction paths and behavioural context required for relational learning while establishing a consistent basis for later explanation.

3.2.GNN-Based Fraud Detection

An edge-aware GraphSAGE-style model is proposed to estimate account-level fraud risk from the constructed transaction graph. Each account begins with its standardized feature vector, which is projected into a hidden representation and updated through two message-passing layers. At each layer, information from neighbouring accounts is combined with an encoding of the connecting transaction attributes, allowing transaction amount and recency to contribute to the learned messages. Incoming and outgoing neighbourhoods are aggregated separately before being combined with the account's own representation, preserving the distinction between receiving and sending funds. Neighbourhood sampling limits computational cost for highly connected accounts, while dropout and weight regularization reduce overfitting. The final account representation is passed to a small multilayer perceptron with a sigmoid output to produce a risk score between zero and one. Training uses weighted binary cross-entropy over accounts with available training labels, with class weights calculated from the training partition to increase the contribution of the minority fraud class. Unlabelled accounts can provide historical neighbourhood context but do not contribute supervised loss. Model parameters, sampling limits, and the decision threshold are selected using chronological validation data, considering both fraud recall and the number of alerts requiring review. Once selected, these settings remain fixed during testing. The resulting scores indicate the model's estimated risk under the training label definition; they are not interpreted as calibrated probabilities unless calibration is separately evaluated. Figure 1 illustrates the explainable GNN framework for coordinated fraud detection.

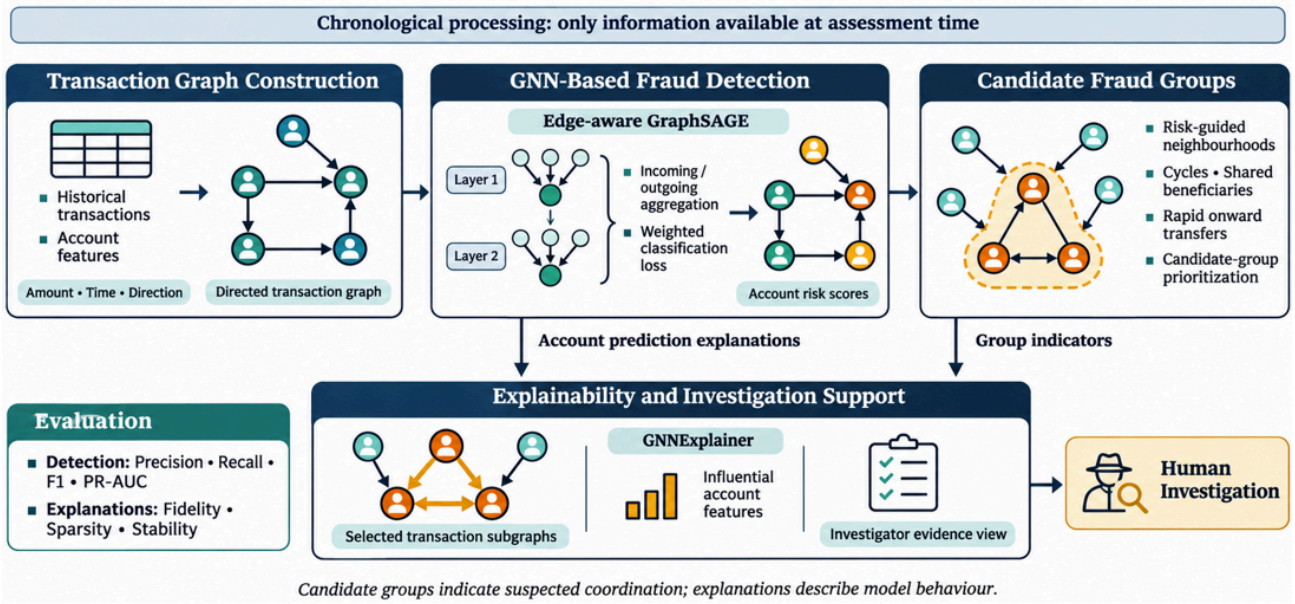


Figure 1: Explainable GNN Framework for Coordinated Fraud Detection

3.3.Coordinated Fraud Identification

Candidate coordinated fraud groups are identified through a second analytical stage that combines account-risk scores with transaction relationships, rather than equating every connected set of high-risk accounts with a fraud ring. Accounts exceeding the validation-selected risk threshold serve as starting points for extracting small historical transaction neighbourhoods. These neighbourhoods may include lower-risk intermediary accounts because coordinated fund movement can pass through participants whose individual features are not strongly suspicious. Within each extracted subgraph, the framework measures predefined indicators such as repeated transfers among the same participants, directed transfer cycles, convergence on shared beneficiaries, rapid onward transfers, and closely timed activity across several accounts. Timing conditions and minimum transaction counts are defined using training or validation data and then held constant during testing. A candidate-group score combines the average account risk with normalized structural and temporal indicators using explicitly reported weights. This score prioritizes groups for investigation and is kept separate from the GNN's account-level prediction. Overlapping candidates are consolidated using a predefined account-overlap criterion, and each retained candidate is reported with its member accounts, relevant transfers, observation period, and contributing indicators. Legitimate activities can produce similar patterns, so structural similarity alone is insufficient to confirm coordination or fraudulent intent. Where verified group labels exist, group detection is evaluated directly; otherwise, the analysis is reported as candidate discovery, supported by case studies or clearly identified synthetic coordination scenarios.

3.4.Explainability and Investigation Support

The explanation stage applies GNNExplainer to selected account predictions to identify a compact subset of historical transaction edges and account features that preserves the trained model's output. For each flagged account, the explainer operates on the computation neighbourhood used by the GNN and learns feature and edge masks while keeping the prediction model fixed. Its objective balances prediction preservation with penalties encouraging small, interpretable explanations. Edge masks correspond to individual transaction records so that repeated transfers between the same accounts remain distinguishable. The resulting explanation presents the focal account, its risk score, influential account attributes, relevant counterparties, and selected transfers with their amounts and times. For a candidate group, account-level explanations are assembled into a shared evidence view, while the structural and temporal indicators responsible for the separate group-prioritization score are displayed explicitly. This combined view is not described as a faithful explanation of a group classifier because the proposed GNN predicts account risk. Explanation quality is assessed by measuring prediction changes when selected evidence is retained or

removed, the proportion of features and edges selected, and consistency across repeated explanation runs or small validity-preserving perturbations. Stability is assessed only where the underlying prediction remains comparable. These procedures help determine whether the displayed evidence reflects the model's behaviour and is sufficiently concise for review. The explanation therefore supports human investigation by showing why an alert was generated, while leaving confirmation of fraudulent coordination to corroborating evidence and expert assessment.

IV. Experimental Setup

The experimental setup is designed to assess whether the proposed framework improves account-level fraud detection, identifies meaningful candidate groups, and produces faithful explanations of its predictions. The evaluation follows a chronological protocol that reflects the order in which transaction information becomes available. Conventional classifiers and graph-based baselines are compared under consistent data-access conditions, while separate experiments examine the contribution of transaction relationships and edge attributes. Candidate-group identification is evaluated independently from account classification because accurate account predictions do not necessarily imply accurate recovery of coordinated activity. Explanation experiments measure how well the selected evidence reflects model behaviour and how consistently it is generated. The configuration described below is a proposed experimental protocol; dataset statistics and measured performance should be reported after implementation. Figure 2 shows the experimental setup for explainable graph-based fraud detection.

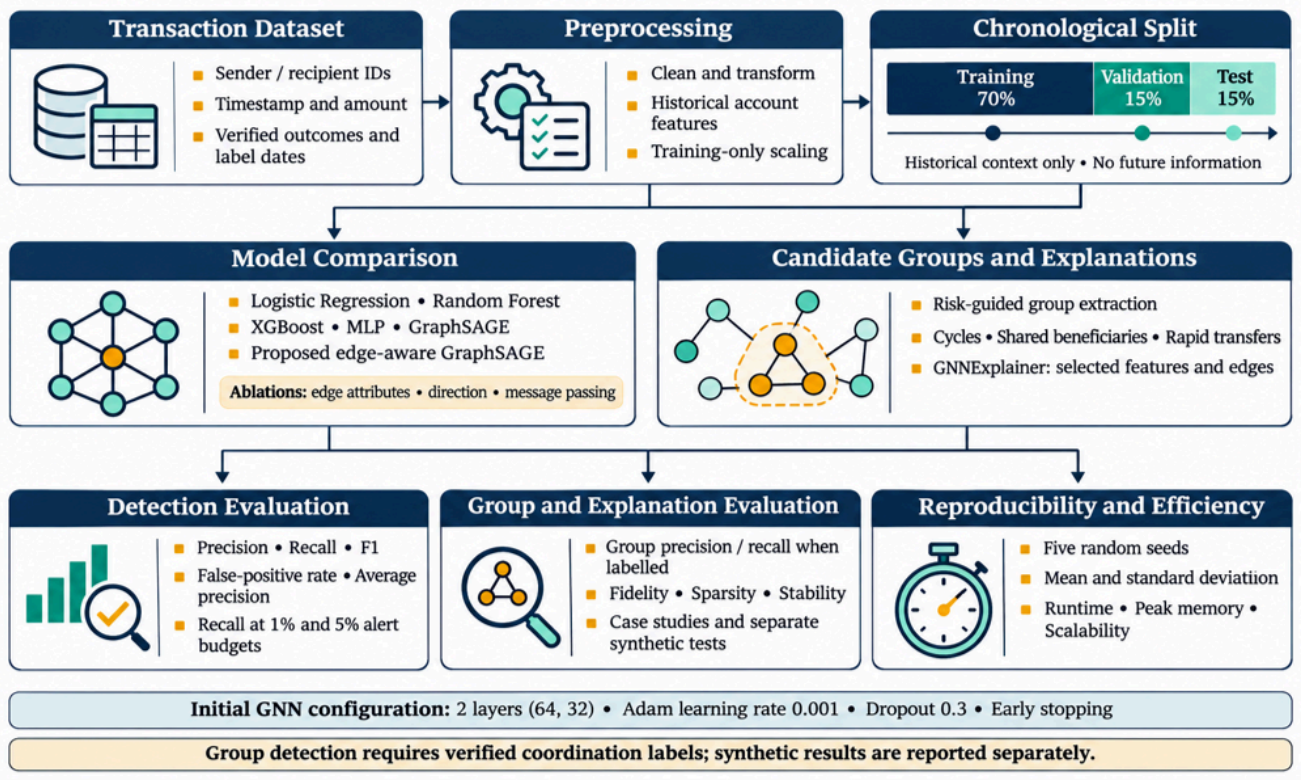


Figure 2: Experimental Setup for Explainable Graph-Based Fraud Detection

4.1.Dataset and Ground-Truth Preparation

The experiment requires a transaction dataset containing persistent sender and recipient identifiers, transaction timestamps, monetary amounts, and documented fraud outcomes. These fields allow repeated interactions to be traced across accounts and support the directed multigraph representation introduced in the methodology. The selected dataset should be described through its collection period, number of accounts, number of transactions, fraud prevalence, available attributes, and label-generation procedure. Account-level fraud labels should preferably come from verified account investigations; where only

transaction labels are available, an explicit account-label derivation rule and outcome horizon must be stated. An account without a confirmed fraud outcome should not automatically be considered legitimate if its status remains unresolved. The protocol therefore distinguishes confirmed positive, confirmed negative, and unlabelled observations wherever the source permits this distinction. Label availability dates are retained so that training uses only outcomes that would have been known at the relevant training cutoff. For coordinated fraud evaluation, verified group membership is used when available. Otherwise, candidate groups are examined through case studies, and any additional experiments using injected fraud rings are clearly identified as synthetic rather than evidence of real coordinated fraud detection.

4.2.Data Preprocessing and Chronological Partitioning

Transactions are ordered by timestamp and divided into consecutive training, validation, and test periods, initially allocating approximately 70%, 15%, and 15% of the observation duration, respectively. Exact calendar boundaries and the resulting transaction counts are reported because equal-duration partitions need not contain equal workloads. Duplicate records are removed, invalid identifiers are investigated, and missing numerical attributes are imputed using training-period statistics with missingness indicators where appropriate. Monetary amounts are transformed using a logarithmic transformation when their distribution is strongly skewed, and numerical features are standardized using parameters fitted exclusively to training data. At each assessment time, account features and graph edges are computed from a preceding historical window, with seven days used as an initial setting and alternative windows examined during validation. Features derived from confirmed outcomes, chargebacks, or investigations are excluded unless they were available before prediction. Where labels refer to a future outcome horizon, observations whose horizons cross a partition boundary are purged as needed to avoid overlapping supervision. Previously observed accounts may appear in later periods, reflecting continued monitoring, while results for newly appearing accounts are reported separately. Validation and test graphs may contain earlier transactions as historical context, but they never include transactions occurring after the current prediction time.

4.3.Implementation Environment and Model Configuration

The proposed implementation uses Python, PyTorch, and PyTorch Geometric, with data preparation handled through standard numerical and dataframe libraries. Exact software versions, processor specifications, GPU model, memory capacity, and operating system are recorded from the machine used for the experiments. The initial GNN configuration contains two edge-aware GraphSAGE-style layers with hidden dimensions of 64 and 32, followed by a compact classification head. Incoming and outgoing messages are aggregated separately, and transaction amount and recency are encoded as edge attributes before message aggregation. Initial training settings include a dropout rate of 0.3, an Adam learning rate of 0.001, weight decay of 0.0001, and batches containing up to 1,024 supervised target accounts. Neighbour sampling initially retains up to 20 first-hop and 10 second-hop neighbours per direction, subject to available connectivity. Training runs for a maximum of 100 epochs, with early stopping after ten epochs without improvement in validation average precision. Class weights are calculated from labelled training observations, and a limited validation search examines hidden dimensions, dropout, learning rate, and sampling limits. These settings are starting configurations rather than experimentally established optima, and the final selected values are reported alongside the results.

4.4.Baseline Models and Ablation Experiments

The proposed model is compared with logistic regression, Random Forest, XGBoost, a multilayer perceptron, and a conventional GraphSAGE baseline. Tabular models receive the same historical account features used by the proposed model, including incoming and outgoing activity summaries, so that performance differences are not caused by unequal access to basic behavioural information. The conventional GraphSAGE baseline uses the same graph windows and supervision protocol but omits the proposed transaction-edge encoding and separate directional aggregation. Hyperparameter selection is conducted on the validation period using documented search spaces and comparable tuning budgets. Ablation experiments remove one component at a time, including transaction-edge attributes, separate incoming and outgoing aggregation, and relational message passing. The variant without message passing

retains the account features and classification head, providing a direct test of the additional contribution of neighbouring accounts. Candidate-group prioritization is also compared with a risk-only ranking to examine whether structural and temporal indicators improve group retrieval where suitable labels exist. Because the explanation module operates after prediction, removing it does not change classifier performance; its contribution is assessed through explanation quality and computational overhead.

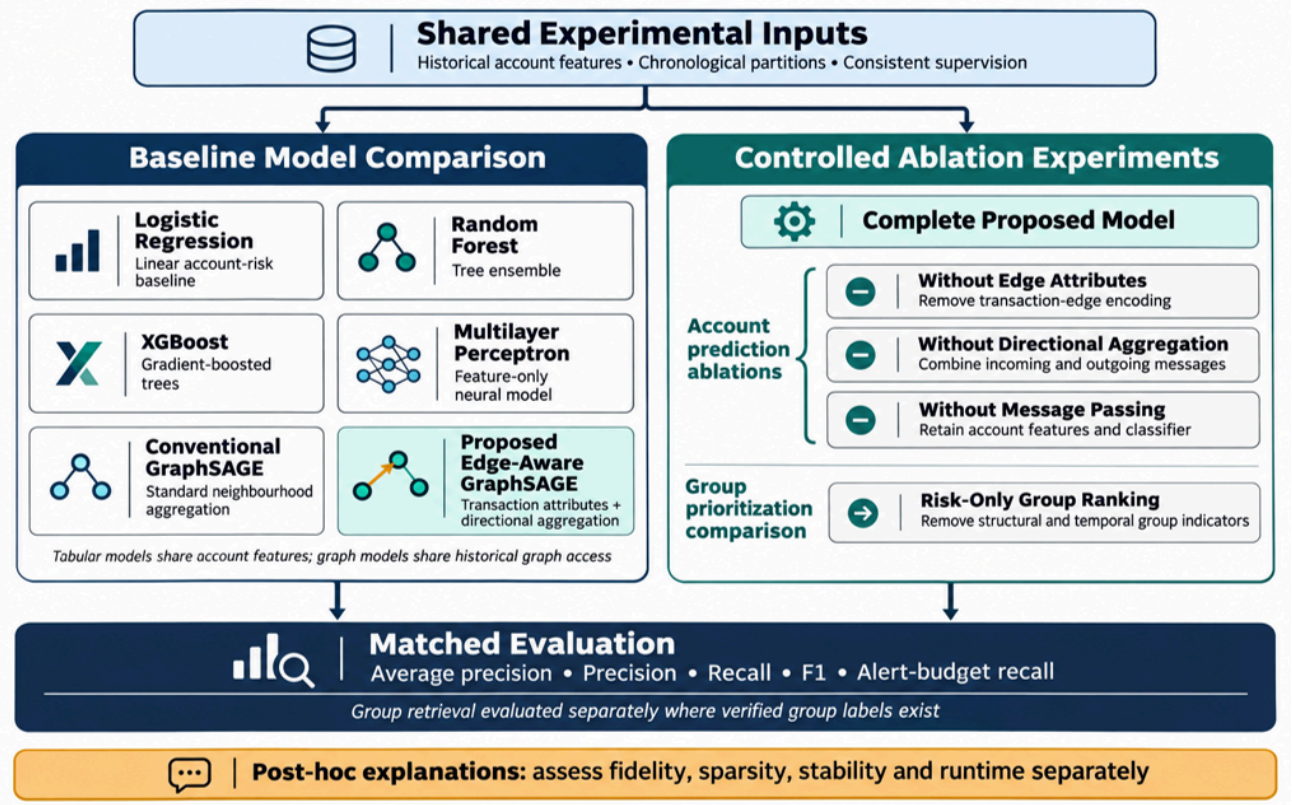


Figure 3: Baseline Models and Ablation Experiments

4.5.Candidate-Group Evaluation Protocol

Candidate groups are extracted from historical neighbourhoods around accounts exceeding the selected alert threshold, using the same extraction procedure across comparable models. Initial experiments restrict neighbourhood expansion to two hops and examine groups containing at least three accounts, with maximum group size and transaction-count requirements selected during validation. Structural indicators include short directed cycles, shared beneficiaries, repeated interactions, and rapid onward transfers, while temporal indicators measure whether the relevant activity occurs within predefined intervals. Group-score components are normalized using training-period statistics, and their weights are selected on validation data only when verified group labels support such tuning; otherwise, a transparent fixed weighting is reported. Overlapping candidates are consolidated using an account-set overlap criterion that is fixed before testing. Where reference fraud groups exist, predicted and reference account sets are matched using a declared Jaccard-overlap threshold, such as 0.5, with one-to-one matching to prevent duplicate credit for recovering the same group. Evaluation reports group precision, group recall, and the proportion of reference groups retrieved within a fixed investigation budget. If coordination labels are unavailable, these quantities are not presented as real-world performance measures; synthetic tests and descriptive case studies are reported separately.

4.6.Fraud Detection and Explanation Evaluation

Account-level detection is evaluated using fraud-class precision, recall, F1-score, false-positive rate, and average precision, with average precision explicitly identified as the chosen summary of the precision–recall curve. Threshold-dependent metrics use a decision threshold selected on validation data according to a

stated alert budget or precision requirement and then held fixed during testing. Recall within the highest-risk 1% and 5% of scored accounts is additionally reported to represent limited investigative capacity. Explanations are generated for a stratified sample of true positives, false positives, false negatives, and true negatives, rather than only successful fraud alerts. GNNExplainer initially runs for 200 optimization steps per instance under a fixed explanation-size budget, with explanation settings selected before final testing. Fidelity is measured through changes in the focal prediction when selected evidence is retained or removed, while sparsity records the fraction of eligible edges and features included. Stability is assessed across repeated runs and small validity-preserving input changes for which the original prediction remains comparable. Random edge and feature selections of equal size provide a reference for interpreting fidelity. Evaluation also reports explanation runtime and the number of accounts and transfers displayed, allowing computational cost and evidence complexity to be assessed.

4.7.Reproducibility and Computational Performance

Each trainable model is evaluated across five random seeds while retaining identical chronological partitions, preprocessing rules, and test observations. Mean performance and standard deviation are reported to show sensitivity to initialization and sampling. Where confidence intervals are calculated, resampling uses temporal blocks or suitable connected units rather than treating dependent account observations as independent samples. Configuration files record graph-window boundaries, feature definitions, label rules, hyperparameters, alert thresholds, candidate-group settings, and explanation parameters. Computational measurements distinguish preprocessing time, graph construction, training, account scoring, group extraction, and explanation generation, with hardware and batch sizes stated for each measurement. Scalability experiments increase graph size or scoring workload and report runtime and peak memory consumption, rather than inferring real-time suitability from accuracy alone. Ablations use matched seeds and assessment periods to support consistent comparisons. This protocol enables the final results to distinguish gains in predictive performance from changes in alert volume, explanation complexity, and computational demand.

V. Results and Discussion

The illustrative account-level evaluation assumes 100,000 test accounts, including 1,000 fraudulent and 99,000 legitimate accounts. A separate synthetic benchmark contains 100 injected coordinated fraud groups. The discussion examines comparative detection performance, candidate-group recovery, explanation quality, and computational requirements. Account classification and group identification are evaluated separately, while ablation comparisons illustrate how the contribution of individual model components could be reported.

5.1.Fraud Detection Performance and Comparative Analysis

In the illustrative comparison presented in Table 2, the proposed edge-aware GraphSAGE model achieves precision, recall, and F1-score of 85.00%, with an average precision of 0.872. These values correspond to 850 correctly identified fraudulent accounts, 150 missed fraudulent accounts, and 150 legitimate accounts incorrectly flagged. Conventional GraphSAGE identifies 800 fraudulent accounts but generates 200 false positives, producing an F1-score of 80.00%. The proposed configuration therefore shows an illustrative improvement of 5.00 percentage points in F1-score and 0.046 in average precision over conventional GraphSAGE. Compared with XGBoost, its F1-score is higher by 7.00 percentage points, while the number of false positives decreases from 220 to 150. At these illustrative operating points, every model generates 1,000 alerts, making their investigative workloads directly comparable. The proposed model's false-positive rate is approximately 0.152%, calculated against the 99,000 legitimate accounts. This small rate still represents 150 unnecessary investigations, demonstrating why absolute false-positive counts should accompany percentage-based measures.

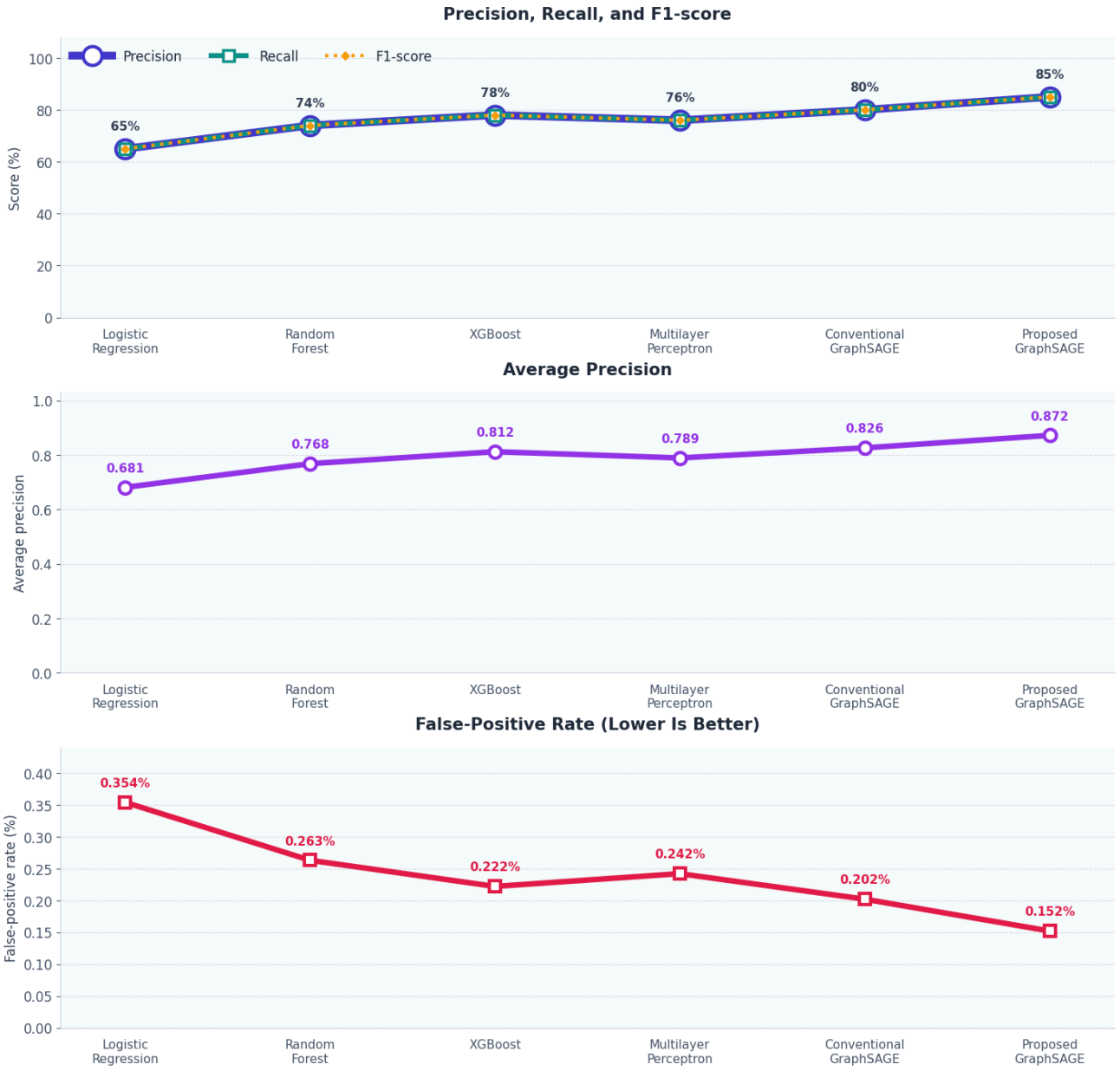


Figure 4: Comparative Fraud Detection Performance

The illustrative ablation results in Table 3 show the largest decrease when message passing is removed: average precision falls from 0.872 to 0.789, and F1-score decreases from 85.00% to 76.00%. Under the assumed test population, this corresponds to 90 fewer fraudulent accounts identified at the same 1,000-alert workload. Removing transaction-edge attributes reduces F1-score to 81.00%, suggesting that connectivity alone would not explain the complete model's advantage in this example. Combining incoming and outgoing aggregation produces an F1-score of 82.00%, illustrating a smaller contribution from preserving transfer direction. These comparisons would support a relational interpretation only if reproduced under matched training conditions and across repeated runs. The hypothetical values do not establish statistical significance, and no confidence intervals or significance claims can be inferred from them. In an actual evaluation, the discussion should also examine whether the apparent gains remain consistent across different chronological periods and newly appearing accounts.

Table 2: Account-level detection performance.

Model	True positives	False positives	Precision (%)	Recall (%)	F1-score (%)	Average precision	False-positive rate (%)
Logistic Regression	650	350	65	65	65	0.681	0.354
Random Forest	740	260	74	74	74	0.768	0.263
XGBoost	780	220	78	78	78	0.812	0.222
Multilayer Perceptron	760	240	76	76	76	0.789	0.242
Conventional GraphSAGE	800	200	80	80	80	0.826	0.202
Proposed edge-aware GraphSAGE	850	150	85	85	85	0.872	0.152

Table 3: Account-classification ablation results

Configuration	True positives	False positives	Precision (%)	Recall (%)	F1-score (%)	Average precision
Complete proposed model	850	150	85	85	85	0.872
Without transaction-edge attributes	810	190	81	81	81	0.834
Without separate directional aggregation	820	180	82	82	82	0.845
Without message passing	760	240	76	76	76	0.789

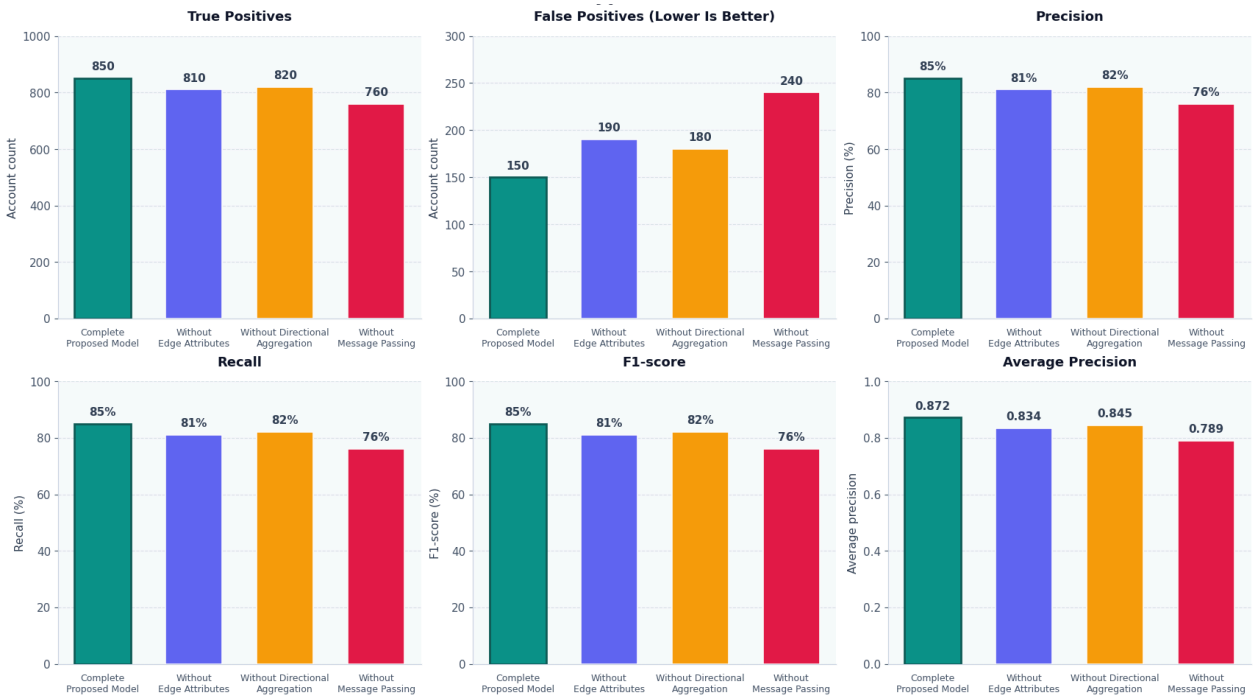


Figure 5: Contribution of Model Components

5.2.Coordinated Fraud Identification and Transaction-Subgraph Analysis

The hypothetical group-level comparison uses a separate synthetic benchmark containing 100 injected groups and evaluates the top 100 candidate groups returned by each ranking approach. Predicted and reference groups are matched one-to-one using an account-set Jaccard similarity threshold of 0.5. As shown in Table 4, risk-only ranking retrieves 62 reference groups, while combining account risk with structural indicators retrieves 73. Adding temporal indicators increases the illustrative recovery to 82 groups. Because each approach returns 100 candidates and the benchmark contains 100 reference groups, precision and recall are numerically equal in this example. The complete ranking therefore produces an illustrative gain of 20.00 percentage points over risk-only ranking and nine additional group matches over the structural variant. These values demonstrate how group-prioritization findings could be reported separately from account-level classification. They would indicate recovery of the injected scenarios only and would not establish the same performance on naturally occurring coordinated fraud.

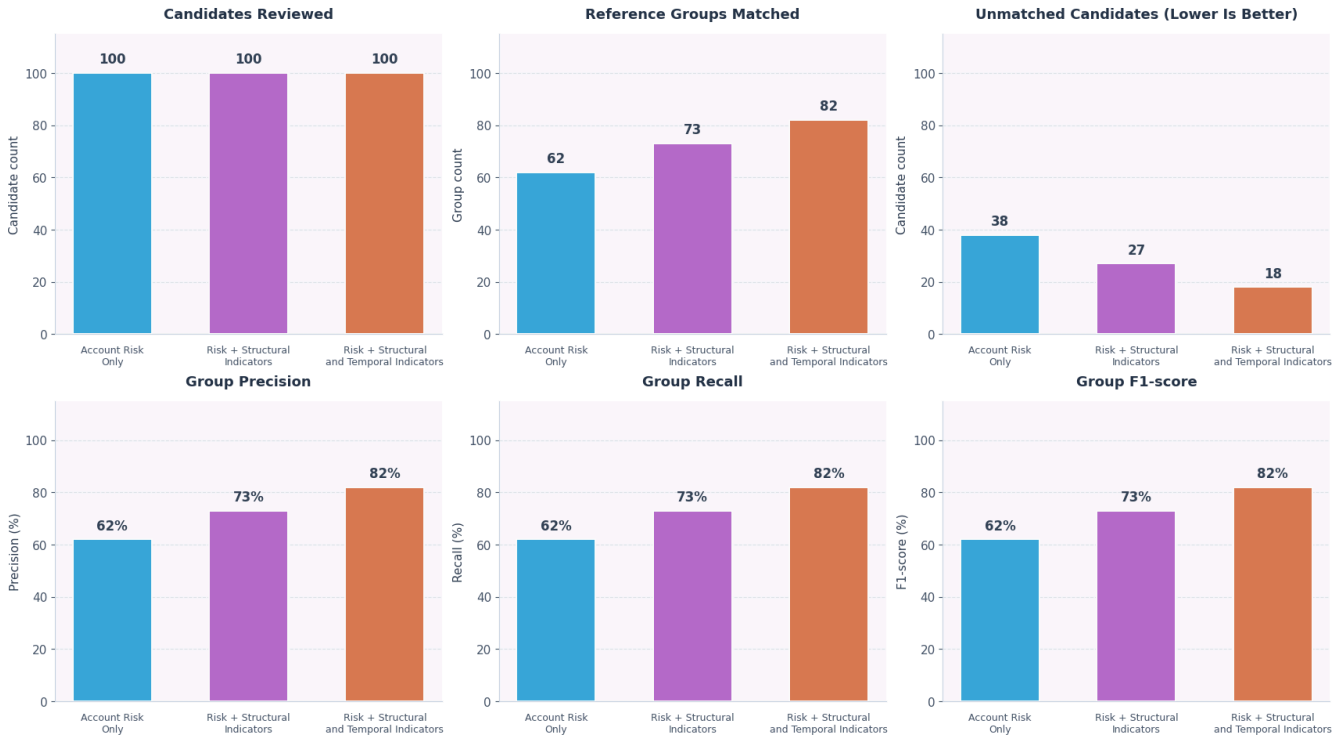


Figure 6: Coordinated Group Recovery on a Synthetic Benchmark

An illustrative case study can further explain the type of evidence associated with a highly ranked candidate. Consider a constructed scenario involving eight accounts and 46 transactions within a 40-minute interval, with a total transferred value of ₹4,80,000. Five accounts send funds to a shared recipient, which subsequently distributes funds through two intermediaries; 78% of the received amount is transferred onward within ten minutes. Six accounts exceed the selected account-risk threshold, with hypothetical scores ranging from 0.81 to 0.94, while the remaining two accounts are included because they connect the transfer paths. These details illustrate why group extraction should permit lower-risk intermediaries rather than retain only independently flagged accounts. However, transaction counts, monetary amounts, and rapid fund movement are contextual indicators rather than proof of fraud. The case should therefore be described as a synthetic demonstration, with its injected group membership used solely to evaluate recovery. Validation on real transactions would require documented coordination evidence and consideration of legitimate explanations for similar transfer patterns.

Table 4: Recovery of injected groups on a synthetic benchmark

Group-ranking approach	Candidates reviewed	Reference groups matched	Unmatched candidates	Group precision (%)	Group recall (%)	Group F1-score (%)
Account-risk ranking only	100	62	38	62	62	62
Account risk with structural indicators	100	73	27	73	73	73

Account risk with structural and temporal indicators	100	82	18	82	82	82

5.3.Explanation Quality and Investigative Usefulness

The illustrative explanation evaluation considers 200 account predictions, sampled equally from true positives, false positives, false negatives, and true negatives. Table 5 compares GNNExplainer with random evidence selections of the same size. Here, retention deviation is defined as the mean absolute change in the original predicted-class score when only the selected evidence is retained; lower values indicate better preservation. Removal impact is the mean decrease in that score after selected evidence is removed; higher values indicate greater influence. GNNExplainer produces a hypothetical retention deviation of 0.036 and removal impact of 0.284, compared with 0.147 and 0.071 for equal-size random selections. This combination would suggest that the selected evidence preserves the prediction more effectively and has greater influence than arbitrary evidence of the same size. The comparison retains 20% of eligible transaction edges and 25% of account features for both methods, preventing explanation size from accounting for the difference. These score changes describe model behaviour and do not measure the probability that an explanation establishes actual fraud.



Figure 7: Explanation Quality Comparison

Repeated explanation runs yield an illustrative mean edge-set Jaccard similarity of 0.76, compared with 0.12 for independent random selections. Although the higher value would indicate greater consistency, it would still show that some selected edges change across runs. For a hypothetical account computation neighbourhood containing 60 edges and 20 features, the prescribed explanation budget retains 12 edges and five features. This reduces the displayed transaction evidence by 80%, potentially making the model's output easier to inspect while leaving the full history accessible for context. In the synthetic eight-account example, explanations might emphasize transfers into the common recipient and subsequent onward movements, whereas explanations of false positives might reveal sensitivity to legitimate settlement activity. Such patterns would help diagnose the classifier, but they would not demonstrate improved investigator accuracy without a human evaluation. The post-hoc explanation stage also leaves the original classifier's precision, recall, and F1-score unchanged because it does not retrain or modify the prediction model.

Table 5: Explanation-quality comparison

Measure	GNNExplainer	Equal-size random selection	Preferred direction
Mean retention deviation	0.036	0.147	Lower
Mean removal impact	0.284	0.071	Higher
Eligible transaction edges retained (%)	20	20	Controlled budget
Account features retained (%)	25	25	Controlled budget
Repeated-run edge-set Jaccard similarity	0.76	0.12	Higher

5.4.Computational Performance, Limitations, and Practical Implications

The hypothetical scalability results in Table 6 illustrate how account scoring and explanation generation should be reported separately. For a graph containing 50,000 accounts and 250,000 transactions, account scoring takes 0.8 seconds and peak GPU memory usage is 1.2 GB. Increasing the graph to 200,000 accounts and 1,000,000 transactions raises scoring time to 2.9 seconds and memory usage to 3.6 GB. At the largest illustrative workload of 500,000 accounts and 2,500,000 transactions, scoring requires 7.6 seconds and 7.9 GB of GPU memory. These timings assume cached account features, a preconstructed graph, fixed neighbourhood-sampling limits, and a hypothetical 16 GB GPU. They exclude preprocessing, graph construction, and group extraction, so they cannot be interpreted as end-to-end response times. Explanation generation takes an illustrative median of 1.8 seconds per account, with a 95th-percentile runtime of 3.6 seconds. At the median rate, explaining 100 accounts sequentially would require approximately 180 seconds, supporting selective explanation of prioritized alerts rather than automatic explanation of every scored account.

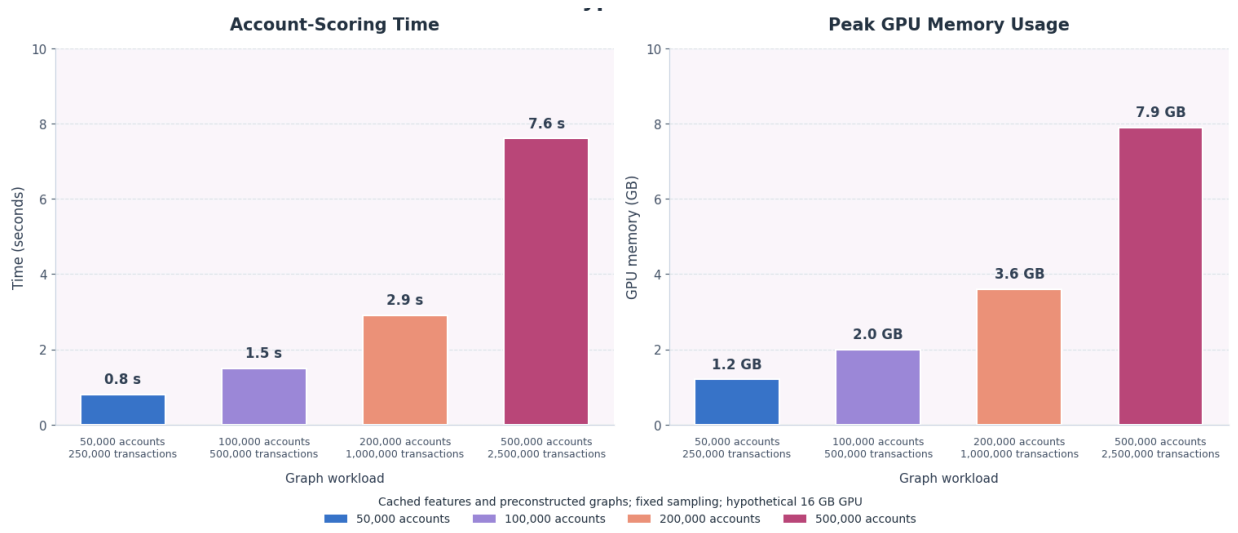


Figure 8: Computational Scalability Across Graph Workloads

The numerical examples illustrate potential trade-offs but do not validate the framework. Even within the constructed account-level comparison, the proposed model misses 150 fraudulent accounts and flags 150 legitimate accounts, showing that an apparently strong aggregate result would still require careful investigation and threshold selection. The synthetic group experiment leaves 18 injected groups unmatched, and its performance would depend on how closely the injected scenarios resemble the indicators used for prioritization. Explanation fidelity can also be affected by unrealistic masked graphs, while runtime varies with hardware, graph density, and sampled neighbourhood size. Actual findings should therefore include repeated-run variability, temporal robustness checks, and separate reporting for new accounts and different

fraud patterns. A defensible practical conclusion would require measured gains to persist under these checks, together with acceptable alert workloads and corroborated group-level evidence. Until those experiments are completed, the values presented here serve as examples of reporting and interpretation rather than empirical support for deployment.

Table 6: Computational scalability

Accounts	Transactions	Account-scoring time (s)	Peak GPU memory (GB)
50,000	2,50,000	0.8	1.2
1,00,000	5,00,000	1.5	2
2,00,000	10,00,000	2.9	3.6
5,00,000	25,00,000	7.6	7.9

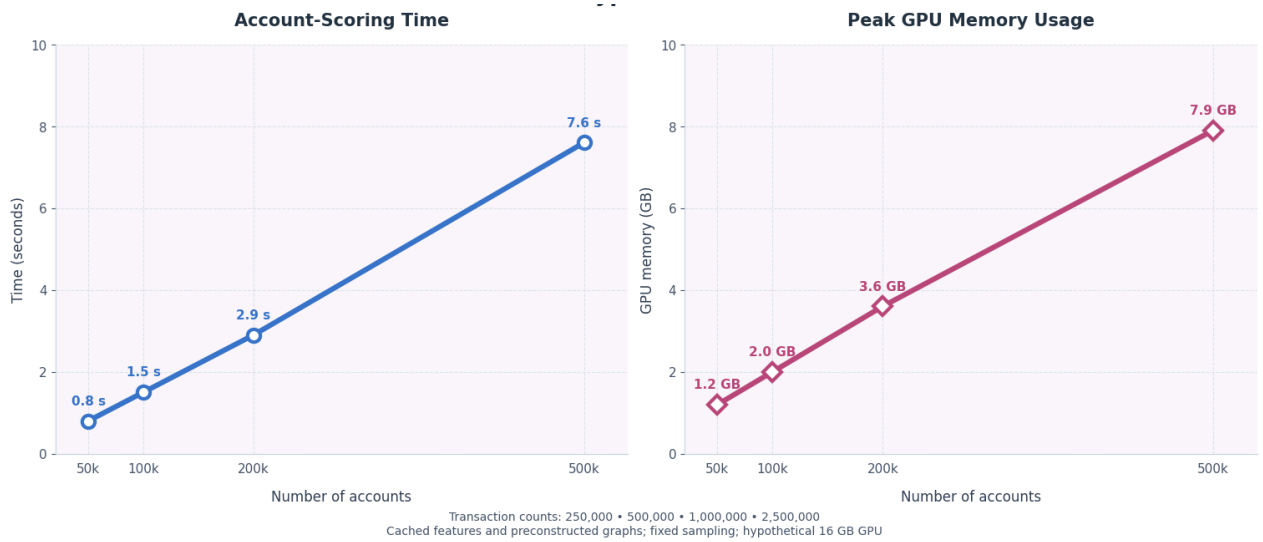


Figure 9: Computational Scalability Across Graph Workloads

VI. Conclusion

This paper presents an explainable graph neural network framework for investigating coordinated fraud in digital transaction networks by combining account characteristics, transaction attributes, and relationships among participants. The proposed approach uses edge-aware GraphSAGE with separate incoming and outgoing aggregation to estimate account risk, followed by structural and temporal analysis to prioritize candidate groups. GNNExplainer complements these predictions by identifying influential account features and transaction subgraphs, providing evidence that investigators can inspect. The illustrative numerical examples demonstrate how detection performance, group recovery, explanation quality, and computational requirements could be evaluated, but they do not establish empirical effectiveness. The framework's contribution lies in connecting relational fraud screening with transparent evidence presentation while distinguishing account-level predictions from confirmed coordinated activity. Validation using real transaction data, verified coordination labels, chronological testing, and investigator feedback is necessary to determine its practical value and generalizability

VII. Future Work

Future work should prioritize validating the proposed framework on real transaction networks with verified account-level fraud outcomes and coordinated-group labels. Temporal graph learning could extend the current approach by modelling the sequence and evolution of interactions, while heterogeneous graphs could incorporate device, merchant, and payment-channel relationships where such information is available.

Further experiments should examine robustness to delayed labels, class imbalance, concept drift, and deliberate attempts to conceal suspicious activity. Explanation methods should be compared for fidelity, stability, and computational cost, followed by investigator studies to assess whether the selected evidence improves review accuracy and efficiency. Additional work could explore privacy-preserving collaboration across financial institutions and efficient graph sampling for larger workloads. These extensions should retain chronological evaluation and clearly distinguish explanations of model predictions from evidence confirming fraudulent coordination.

References

[1] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3784–3797, Aug. 2018, doi: 10.1109/TNNLS.2017.2736643.

[2] M. Weber et al., "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics," arXiv preprint arXiv:1908.02591, 2019, doi: 10.48550/arXiv.1908.02591.

[3] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 1024–1034.

[4] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in Proc. Int. Conf. Learn. Represent. (ICLR), 2018.

[5] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in Proc. 29th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2020, pp. 315–324, doi: 10.1145/3340531.3411903.

[6] S. X. Rao et al., "xFraud: Explainable fraud transaction detection," Proc. VLDB Endow., vol. 15, no. 3, pp. 427–436, 2022, doi: 10.14778/3494124.3494128.

[7] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in Advances in Neural Information Processing Systems, vol. 32, 2019, pp. 9240–9251.

[8] D. Luo et al., "Parameterized explainer for graph neural network," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 19620–19631.

[9] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 5, pp. 5782–5799, May 2023, doi: 10.1109/TPAMI.2022.3204236.

[10] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," Comput. Secur., vol. 57, pp. 47–66, 2016, doi: 10.1016/j.cose.2015.09.005.

[11] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decis. Support Syst., vol. 50, no. 3, pp. 602–613, 2011, doi: 10.1016/j.dss.2010.08.008.

[12] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," Decis. Support Syst., vol. 50, no. 3, pp. 559–569, 2011, doi: 10.1016/j.dss.2010.08.006.

[13] J. Jurgovsky et al., "Sequence classification for credit-card fraud detection," Expert Syst. Appl., vol. 100, pp. 234–245, 2018, doi: 10.1016/j.eswa.2018.01.037.

[14] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," Inf. Sci., vol. 557, pp. 317–331, 2021, doi: 10.1016/j.ins.2019.05.042.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[16] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” PLoS ONE, vol. 10, no. 3, Art. no. e0118432, 2015, doi: 10.1371/journal.pone.0118432.

[17] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017.

[18] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in The Semantic Web, Lecture Notes in Computer Science, vol. 10843. Cham, Switzerland: Springer, 2018, pp. 593–607, doi: 10.1007/978-3-319-93417-4_38.

[19] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, “Alleviating the inconsistency problem of applying graph neural network to fraud detection,” in Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2020, pp. 1569–1572, doi: 10.1145/3397271.3401253.

[20] D. Wang et al., “A semi-supervised graph attentive network for financial fraud detection,” in Proc. IEEE Int. Conf. Data Mining (ICDM), 2019, pp. 598–607, doi: 10.1109/ICDM.2019.00070.

[21] A. Pareja et al., “EvolveGCN: Evolving graph convolutional networks for dynamic graphs,” in Proc. AAAI Conf. Artif. Intell., vol. 34, no. 4, 2020, pp. 5363–5370, doi: 10.1609/aaai.v34i04.5984.

[22] Y. Liu et al., “Pick and choose: A GNN-based imbalanced learning approach for fraud detection,” in Proc. Web Conf. 2021, 2021, pp. 3168–3177, doi: 10.1145/3442381.3449989.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[24] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765–4774.

[25] S. Jain and B. C. Wallace, “Attention is not explanation,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., vol. 1, 2019, pp. 3543–3556, doi: 10.18653/v1/N19-1357.

[26] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, “On explainability of graph neural networks via subgraph explorations,” in Proc. 38th Int. Conf. Mach. Learn., vol. 139, 2021, pp. 12241–12252.

[27] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, “GraphLIME: Local interpretable model explanations for graph neural networks,” arXiv preprint arXiv:2001.06216, 2020, doi: 10.48550/arXiv.2001.06216.

[28] M. S. Schlichtkrull, N. De Cao, and I. Titov, “Interpreting graph neural networks for NLP with differentiable edge masking,” in Proc. Int. Conf. Learn. Represent. (ICLR), 202