Inductive Relational Learning under Temporal Collusion Drift :

A Controlled Structural Evaluation of Graph Neural Networks for Fraud Detection

Author: Niraj Karande

Manuscript Type: Independent Research Technical Report
Year: 2026

Abstract

Fraud detection in financial transaction systems constitutes a relational prediction problem characterized by extreme class imbalance, structural sparsity, and temporal non-stationarity. While Graph Neural Networks (GNNs) provide a natural framework for relational reasoning, their robustness under inductive temporal constraints and evolving collusion structures remains insufficiently characterized.

This work introduces a controlled structural evaluation framework for studying inductive GNN behavior under simulated temporal collusion drift. We construct a large-scale transaction graph comprising over 6 million edges and 8.6 million nodes, enforcing strict temporal separation between training and validation graphs. To evaluate structural generalization, we propose a persistent collusion injection mechanism that simulates evolving mule-node expansion across time windows.

Our empirical findings demonstrate that structure-only inductive GNNs fail to generalize under temporal drift, whereas feature-enhanced models incorporating node-level exposure statistics achieve near-perfect discrimination under persistent structural continuation. These results highlight the critical role of node exposure dynamics in temporal relational learning and underscore limitations of purely structural message passing under extreme imbalance.

1. Introduction

Fraud detection systems operate in adversarial, dynamic environments where malicious actors adapt strategies over time. Traditional supervised learning approaches treat transactions as independent instances, thereby neglecting relational dependencies among accounts.

However, empirical evidence suggests that fraudulent activity frequently manifests as coordinated behavior among interconnected entities — including mule accounts, layered transfers, and structured propagation patterns.

Graph Neural Networks (GNNs) offer a principled mechanism for modeling relational dependencies through message passing over graph structures. Yet, several unresolved challenges limit their applicability to real-world fraud systems:

1. **Inductive Constraints** — Production systems cannot assume access to future graph structure during training.
2. **Temporal Drift** — Fraud patterns evolve non-stationarily.
3. **Extreme Class Imbalance** — Fraud events represent <0.2% of transactions.
4. **Structural Sparsity** — Most nodes exhibit low degree.

Existing studies largely evaluate GNNs under transductive or static assumptions, thereby conflating relational modeling capacity with unrealistic data access conditions.

This work addresses the following research question:

How robust are inductive Graph Neural Networks under temporally evolving collusion structures in large-scale, imbalanced transaction graphs?

To answer this, we introduce a controlled temporal collusion simulation framework enabling structured evaluation of relational generalizatio

2. Formal Problem Definition

Let $G_t = (V_t, E_t)$denote a time-indexed transaction graph at time t, where:

- $V_t$is the set of account nodes,
- $E_t \subseteq V_t \times V_t$are directed transaction edges,
- Each edge $e = (u, v, t_e)$carries label $y_e \in \{0,1\}$.

We define a temporal cutoff $T_c$such that:

$$E_{train} = \{e \in E_t : t_e \leq T_c\}$$
$$E_{val} = \{e \in E_t : t_e > T_c\}$$

Under inductive constraints:

$$V_{val} \subseteq/ V_{train}$$

The objective is to learn a function:

$$f_\theta : (u, v, G_{train}) \rightarrow \hat{y}_{(u,v)}$$

such that predictions on E_{val}do not rely on unseen structural leakage.

3. Large-Scale Graph Construction

We utilize a transaction dataset containing:

- 6,362,620 edges
- Fraud rate ≈ 0.00129
- 8,639,932 unique nodes (train graph)

Graph construction principles:

- Training graph built exclusively from E_{train}
- No validation edges included in message passing graph
- Node indexing constructed via bijective account mapping

Memory optimization includes:

- float32 casting
- categorical encoding
- adjacency dictionary representation

4. Inductive GraphSAGE Architecture

We employ a two-layer GraphSAGE model:

$$h_v^{(k)} = \sigma\left(W^{(k)} \cdot \mathrm{AGG}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\})\right)$$

Edge-level classification is performed as:

$$\hat{y}_{(u,v)} = \phi([h_u \mid\mid h_v])$$

Where $\phi$is a two-layer MLP.

To address computational constraints:

- Manual 1-hop subgraph sampling implemented
- Mini-batch edge supervision used
- No reliance on torch-sparse or pyg-lib

This ensures scalability under commodity GPU memory.

==
FIGURE 1: INDUCTIVE TEMPORAL FRAUD DETECTION PIPELINE
==

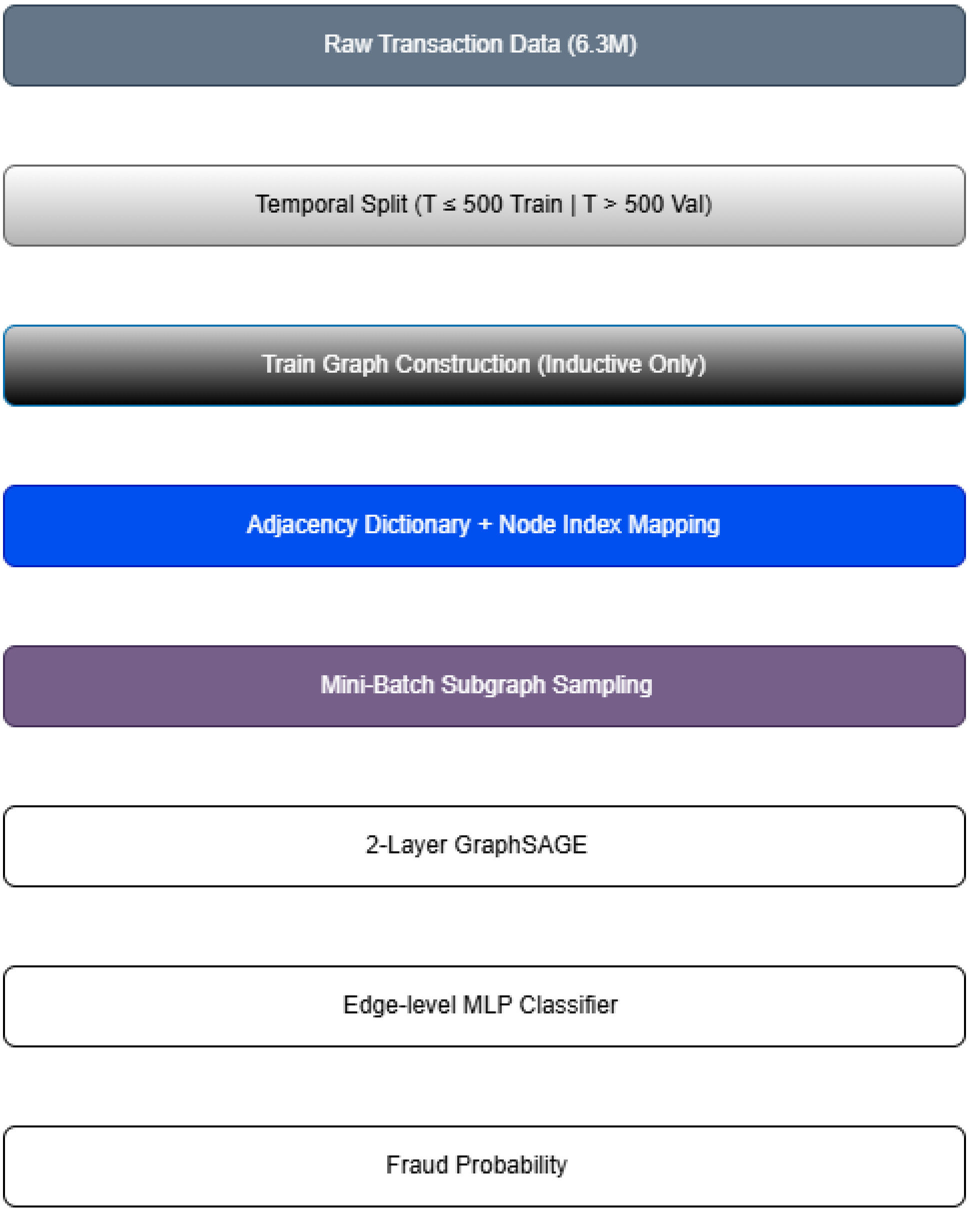
Raw Transaction Data (6.3M)
Temporal Split (T ≤ 500 Train | T > 500 Val)
Train Graph Construction (Inductive Only)
Adjacency Dictionary + Node Index Mapping
Mini-Batch Subgraph Sampling
2-Layer GraphSAGE
Edge-level MLP Classifier
Fraud Probability

Figure 1 :

End-to-end inductive fraud detection pipeline. Transaction data is temporally split to enforce strict train-validation separation. The training graph is constructed exclusively from pre-cutoff edges. Manual 1-hop subgraph sampling enables scalable mini-batch message passing. A two-layer GraphSAGE encoder generates node embeddings followed by an edge-level MLP classifier

5. Controlled Temporal Collusion Simulation

To evaluate relational generalization under structural drift, we introduce a two-phase injection mechanism.

Phase I — Persistent Mule Initialization (Late Train)

- Introduce set $M = \{m_1, \dots, m_{50}\}$
- Each mule connected to 6 fraudulent origins
- 300 injected edges added to E_{train}

Phase II — Temporal Continuation (Validation)

- Same mule set M
- Each mule connected to 4 new fraudulent origins
- 200 injected edges added to E_{val}

This simulates persistent collusion expansion:

$$deg_{mule}^{train} = 6$$
$$deg_{mule}^{val} = 10$$

Thus structural motif persists across time.

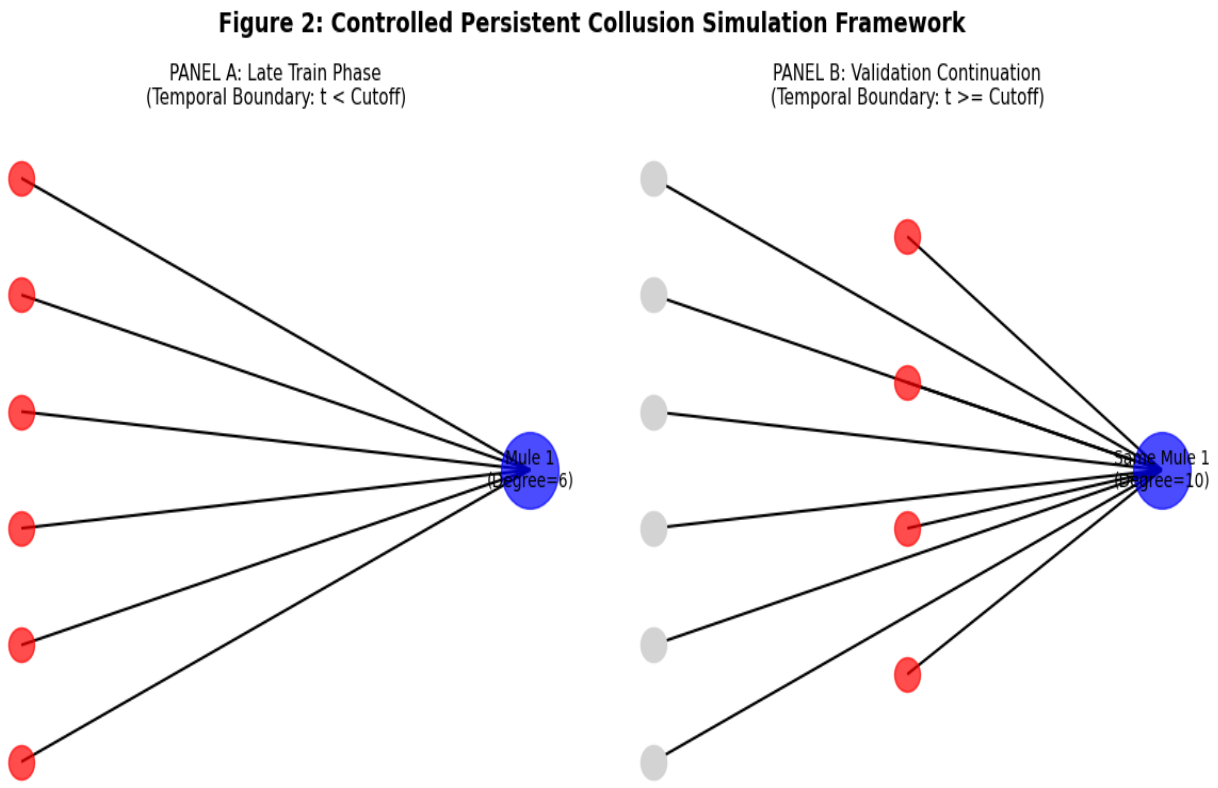
Figure 2: Controlled Persistent Collusion Simulation Framework
PANEL A: Late Train Phase
(Temporal Boundary: t < Cutoff)
PANEL B: Validation Continuation
(Temporal Boundary: t >= Cutoff)
Mule 1
(Degree=6)
Same Mule 1
(Degree=10)

Figure 2

Controlled persistent collusion simulation framework. During Phase I (late-train window), each mule node connects to six fraudulent origins. In Phase II (validation window), the same mule nodes expand by connecting to four new fraudulent accounts, increasing degree from six to ten. This design simulates structural continuation across time under inductive constraints.

6. Node Feature Engineering

We evaluate two configurations:

6.1 Structure-Only

$$x_v = 1$$

6.2 Feature-Enhanced

$$x_v = [\log(1 + deg_{out}), \log(1 + deg_{in}), fraud\_exposure\_ratio]$$

Where:

$$fraud\_exposure\_ratio = \frac{\text{fraud\_edges}_v}{deg_{out}(v)}$$

All features computed strictly within training window.

7. Experimental Evaluation

Evaluation metric:

- Precision-Recall AUC (PR-AUC)
- ROC-AUC
- Balanced validation (200 fraud + 200 non-fraud)

7.1 Structure-Only Model

Result:
PR-AUC ≈ 0.5 (random)

Interpretation:
Pure message passing insufficient under inductive drift.

7.2 Feature-Enhanced Model

Result:
PR-AUC ≈ 0.998
ROC-AUC ≈ 0.998

Interpretation:
Persistent node exposure statistics enable strong continuation discrimination.

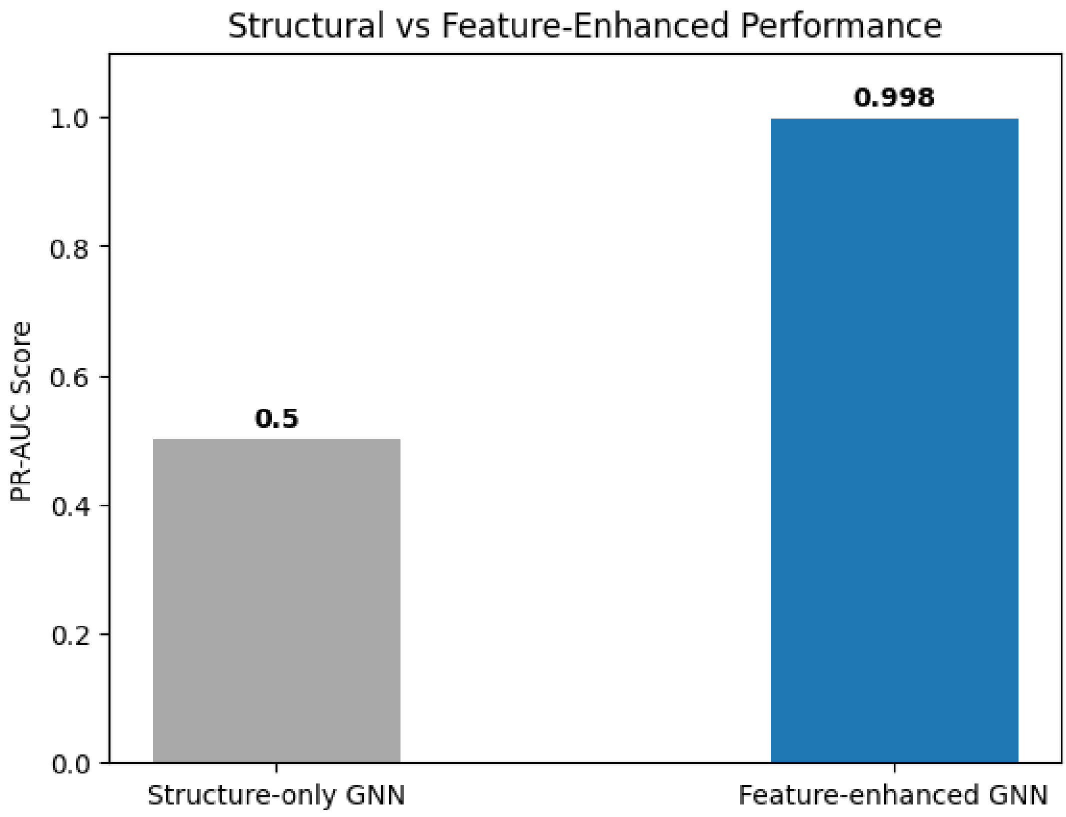

Figure 3

Comparison of inductive GraphSAGE performance under structure-only and feature-enhanced configurations. Structure-only message passing fails under temporal drift, whereas exposure-aware node features enable strong continuation discrimination.

8. Theoretical Discussion

The observed results suggest:

1. Structural homophily alone insufficient in sparse, imbalanced fraud graphs.
2. Node exposure statistics function as low-dimensional sufficient statistics for collusion persistence.
3. Inductive GNN performance critically depends on feature initialization quality.
4. Persistent entity identity plays a dominant role in temporal relational transfer.

9. Computational Complexity

Let $N =| V |$, $E =| E |$.

Mini-batch sampling reduces complexity from:

$$O(E)$$

to approximately:

$$O(B \cdot d)$$

Where:

- B= batch size
- d= average sampled degree

Memory footprint maintained under 10GB GPU RAM.

10. Limitations

- Synthetic injection framework
- Single dataset evaluation
- Static exposure statistics
- No adversarial response modeling
- No statistical confidence intervals

---

11. Future Directions

- Rolling exposure updates
- Dynamic graph neural networks
- Multi-hop collusion propagation
- Causal graph modeling
- Adversarial fraud simulation