Transactions on Computational and Scientific Methods | Vo. 4, No. 2, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Interpretable Graph-Based Anomaly Detection for Related-Party Transactions in Auditing Systems

Chen Chen
Vanderbilt University, Nashville, USA
chen.chen@vanderbilt.edu

Abstract: In corporate auditing and risk management scenarios, related-party transactions often exhibit characteristics such as long relationship chains, numerous participating entities, and complex structural patterns. Furthermore, the scarcity and concealment of abnormal events make it difficult for traditional rule-based or single-point indicator-based methods to reliably detect high-risk clues. To address this, this paper constructs a heterogeneous relationship network for corporate related-party transaction data and proposes an interpretable subgraph representation and anomaly identification method for auditing. First, entity disambiguation and related-party alignment reduce pseudo-structural differences caused by naming fragmentation. Then, multi-scale candidate subgraphs are sampled from the transaction network to learn a subgraph representation that balances structure and attributes. Finally, an interpretable mechanism is used to output key evidence edges/core participating entities, forming an auditable risk evidence chain. Experimental results show that the proposed method outperforms mainstream baselines on multiple detection and interpretation metrics, significantly improving evidence readability and audit usability while maintaining detection performance. This provides a reliable technical path for uncovering corporate fraud clues and providing risk warnings.

Keywords: Accounting and auditing; related party transactions; graph anomaly detection; subgraph representation learning; interpretability;

1. Introduction

Against the backdrop of rapid advancements in digital finance and intelligent auditing, the scale of inter-enterprise related-party transactions continues to expand, with longer and more complex transaction chains[1]. Abnormal behaviors often manifest covertly through multi-entity collaboration, cross-account splitting, and circular transfers, making them difficult to identify promptly using single vouchers or isolated indicators. Constructing an inter-enterprise related-party transaction network, with enterprises as nodes and related-party transactions as edges, provides a structured perspective for audit risk identification: anomalies are no longer merely fluctuations in a single amount or category of accounts, but may manifest as abnormal combinations of local network structures, such as high-frequency exchanges, closed-loop links, dense clusters of anomalies, or abnormal mediating behavior of key nodes[2]. Therefore, conducting interpretable graph learning audit research on inter-enterprise related-party transaction networks can not only enhance the ability to characterize complex fraud and risk transmission but also help form more logically consistent chains of evidence and traceable conclusions, thereby enhancing the reliability and verifiability of audit decisions.

Existing audit analysis and data-driven methods typically face three prominent challenges when dealing with related-party transaction networks: First, strong structural dependence, with anomalies often triggered by combinations of multiple edges, making it difficult for traditional detection methods based on the assumption of independent and identically distributed samples to capture collaborative patterns at the subgraph level; second, the coexistence of heterogeneity and noise, with significant differences in transaction types, frequencies, and monetary scales, and data issues such as missing data, delays, and inconsistencies in definitions, making the model susceptible to perturbations; third, insufficient interpretability, as black-box risk scoring cannot be directly converted into auditable clues and working papers, requiring auditors to know which enterprises and which transaction relationships constitute the anomalous structure and what structural mechanism the anomalies originate from[3,4]. To clearly define the boundaries and objectives of this paper, this research elevates the audit task from node-level or edge-level anomaly detection to anomaly subgraph discovery, and considers interpretability as a constraint of equal importance to accurate identification. Relevant comparisons are summarized in Table 1, and this will serve as the basic starting point for the research design in the subsequent text.

Table 1: Comparison of Enterprise Related Party Transaction Network Audit Task Settings

Task Level	Primary Objective	Typical Anomalies	Output Format	Audit-Oriented Focus
Node-level anomaly detection	Identify high-risk enterprises or accounts	Abnormally active nodes; abnormal centrality	Node risk scores or rankings	Pinpoint suspicious entities to support sampling and focused investigation
Edge-level anomaly detection	Identify suspicious transaction relationships	Edges with abnormal amounts or frequencies; abnormal relationship types	Edge risk scores or alerts	Pinpoint suspicious transactions to support voucher and contract verification.
Subgraph-level anomaly discovery	Identify anomalous structural patterns and evidence subgraphs	Closed-loop transfers; anomalous hookups; anomalously dense clusters; cross-level paths	Anomalous subgraphs and their constituent nodes/edges	Build an evidence chain to support structured attribution and re-checking
Explainable graph learning for auditing	Detect anomalies while providing structural reasons	Structural contribution concentrated on key nodes/edges and critical paths	Anomalous subgraphs with contribution explanations	Traceable and reviewable results that can be written into audit working papers

Table 1 provides a structured summary of typical issues corresponding to different audit modeling granularities and interpretation requirements. It emphasizes that anomaly subgraph discovery, compared to node-level or edge-level detection, more closely reflects common collaborative behavior patterns in related-party transactions[5,6]. It also points out that interpretable graph learning should output structural evidence rather than merely risk scores. Based on this perspective, this paper focuses on learning auditable structural representations within corporate-related-party transaction networks and aligning the model output to verifiable subgraph evidence through an interpretable mechanism. For example, it provides a contribution ranking of key enterprise sets, key transaction edge sets, and their structural relationships, thereby supporting

auditors in tracing clues, conducting thorough investigations, and attributing risks. This problem setting unifies network structure learning with audit evidence requirements, enabling the model not only to detect anomalies but also to explain why the anomalies are anomalies and which structural factors contribute to them.

Within the aforementioned framework, interpretable graph learning auditing for anomaly subgraph discovery is of significant importance. On one hand, it can make related-party transaction risks explicit through network structure, shifting audits from experience-based point-based checks to structured, clue-driven approaches, thus enhancing the ability to identify hidden collaborative behaviors and risk transmission paths. On the other hand, interpretable outputs can naturally map to audit evidence organization methods, transforming model findings into verifiable sets of enterprises and transaction relationships, promoting human-machine collaborative auditing, and reducing verification and communication costs associated with false alarms[7]. Overall, interpretable graph learning auditing based on enterprise-related-party transaction networks provides an intelligent path for anomaly subgraph discovery that aligns more closely with audit practice, and is expected to improve the effectiveness and operability of audit risk identification in complex transaction environments.

2. Datasets and Preprocessing

2.1 Dataset

This paper uses the Related Party Transactions (RTD) data publicly disclosed by the National Stock Exchange of India. This data comes from the periodic disclosures of related party transactions submitted by listed companies in accordance with regulatory requirements. It supports filtering by market sector and time interval, and provides directly downloadable CSV files. Links to files related to structured disclosures are also retained for further parsing and standardization. This data naturally meets the modeling requirements of corporate-related party transaction networks. The disclosing entity and counterparty can be used as nodes, and transaction type and amount as edge attributes, thereby constructing an audit-oriented corporate-related party transaction graph to achieve structured discovery of subgraph-level anomalies.

At the field level, the data includes basic information and reporting period information of the disclosing entity, as well as a detailed table structure of related party transactions. It covers key audit elements such as counterparty information, relationship type with the disclosing entity, related party transaction category, audit committee-approved amount, actual amount incurred during the reporting period, and beginning and ending balances. It also provides more granular supplementary disclosure items for specific types of transactions, such as loans, deposits, prepayments, and investments. These elements enable the network to express not only the existence of related-party transactions, but also audit semantics such as transaction direction, transaction intensity, balance rollover, and approval constraints, providing a traceable evidentiary basis for subsequent interpretable graph learning.

2.2 Dataset preprocessing

(1) Data cleaning and standardization

First, we systematically structured and cleaned the original related-party transaction disclosure data, aiming to make the data more consistent in overall terminology and more standardized in expression, thereby reducing bias caused by inconsistencies in the data itself. By standardizing the basic formats of company identifiers, reporting periods, and key numerical fields, we minimized differences caused by different sources and recording habits, enabling data to be compared and summarized under the same standard. Simultaneously, records with obvious problems were screened and processed to prevent missing information, abnormal formats, or noisy content from being mistakenly treated as valid signals in subsequent analysis, thus improving the overall usability and credibility of the dataset and providing a more stable input foundation for subsequent relationship structure construction and experimentation.

Second, we integrated and deduplicated potentially duplicate disclosures and redundant information in the data, ensuring that disclosures by the same entity within the same reporting period remained as clear, coherent, and consistent as possible, reducing structural bloat and relationship confusion caused by duplicate entries. In this way, the connections in the subsequent graph structure can more accurately correspond to the disclosed facts themselves, rather than being affected by duplicate or non-standard records, thus presenting more reasonable connectivity characteristics and more stable statistical properties at the structural level. Overall, the cleaned data is easier for subsequent modeling and evaluation, and also more conducive to the interpretation and reproduction of experimental results. There are significant differences in data quality and downstream task performance before and after cleaning, as shown in Figure 1.

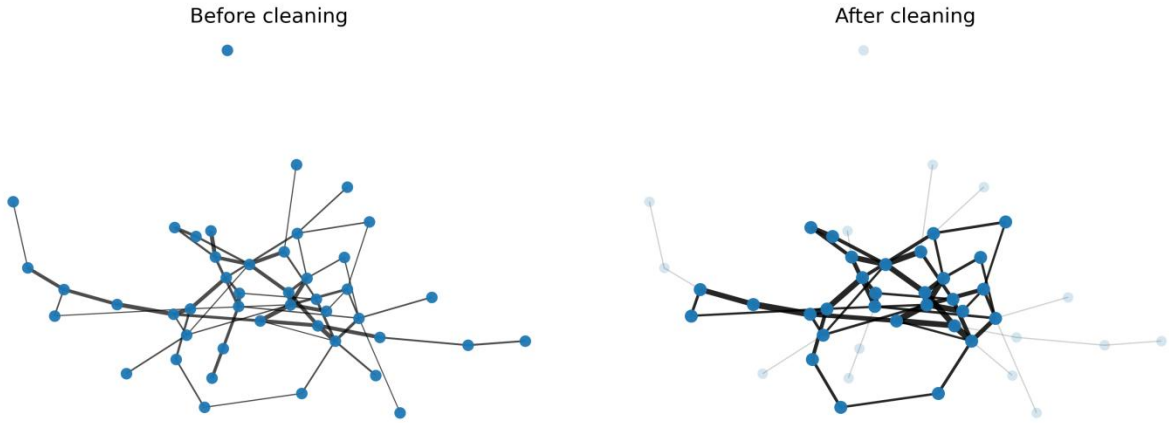

Figure 1. Comparison of data before and after cleaning

(2) Entity resolution and related party alignment

The disclosure entities and counterparties are uniformly mapped to network nodes, and entity normalization is performed, including name standardization, case and space cleanup, common suffix unification, and rule-based merging of homonyms and abbreviations, thereby reducing spurious structural differences caused by entity fragmentation. For relationship type fields, they are uniformly mapped to a finite set of relation semantic labels to express the association attributes between counterparties and entities in the graph, ensuring that relationship labels are comparable across different companies and time periods. Similarly, Figure 2 shows a comparison of the results before and after entity parsing and association alignment.

As shown in Figure 2, before entity disambiguation, the same company would be split into many "surface entities" by multiple aliases, resulting in a high number of names for each company and artificial fragmentation of the graph structure. After entity disambiguation and related party alignment, each company is basically unified into a unique standardized name (almost all of them are 1 in the right figure), indicating that the synonym merging and alignment rules are effective and significantly eliminate duplication and noise.

(3) Graph construction and edge attribute encoding

A corporate-related-party transaction network is constructed using enterprises as nodes and related-party transactions as directed edges. The edge direction is determined based on the disclosure semantics, pointing from the subject to the counterparty. Transaction type, reporting period, approved amount, actual transaction amount, and beginning and ending balances are encoded as edge attributes. To adapt to the graph learning model, categorical attributes are discretely encoded, while continuous attributes such as amounts and balances are scaled and robustly normalized. Additional derived audit features are constructed, such as the ratio of transaction amount to approved amount, net change rate of balance, and transaction frequency per unit time, to enhance the model's sensitivity to abnormal structures.

(4) Subgraph Sample Generation and Training Partition

To address the anomaly subgraph discovery task, k-hop neighborhood subgraphs centered on the target enterprise are extracted from the entire graph, or candidate subgraph sets are extracted based on transaction link length and closed-loop structure rules, forming basic sample units for learning and interpretation. To avoid information leakage and time travel, a time-consistent division strategy is adopted based on the reporting period, using earlier reporting periods for training and validation, and later reporting periods for testing or reserved for audit backtesting. Simultaneously, length truncation and node number upper limits are controlled for subgraphs of different sizes to ensure the stability and computational controllability of batch training.

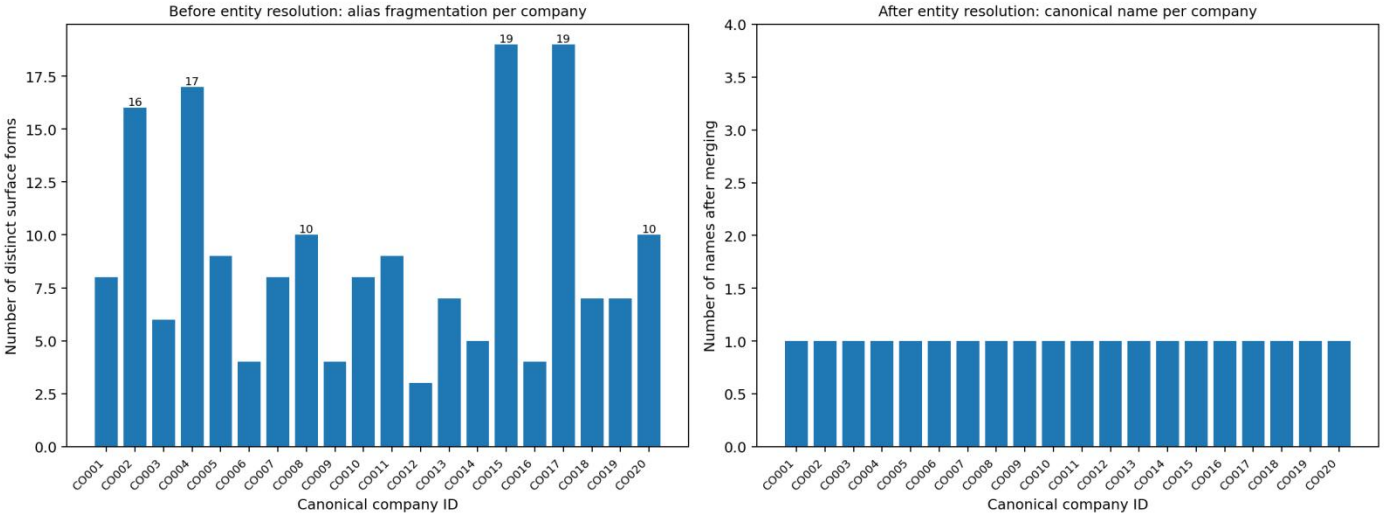

Figure 2. Comparison of entity resolution and related party alignment results

5. Method

The network of related-party transactions is represented as a directed graph, where nodes correspond to corporate entities or related parties, edges correspond to disclosed related-party transaction relationships, and each edge carries semantic and monetary attributes. Let the graph be:

$$G = (V, E)$$

Where V is the set of nodes, and E is the set of directed edges. To adapt to audit semantics, each node is configured with a set of structure and business feature vectors, and each edge is configured with a set of transaction attribute vectors, denoted as follows:

$$x_v \in R^d,\ a_{uv} \in R^p$$

x_v can be constructed from node degree, period activity, number of related parties, etc., while a_{uv} can be constructed from transaction type code, transaction amount, approved amount, balance change, etc. The core objective of anomaly subgraph discovery is to locate a set of structurally suspicious and verifiable local patterns in the network. Therefore, the method uses subgraphs as the basic audit unit, representing and scoring each candidate subgraph as a traceable chain of evidence. This allows the output to not only provide the risk intensity but also trace back to key nodes and key transaction edges, thereby supporting audit clue tracing and working paper organization. Its overall architecture is shown in Figure 3.

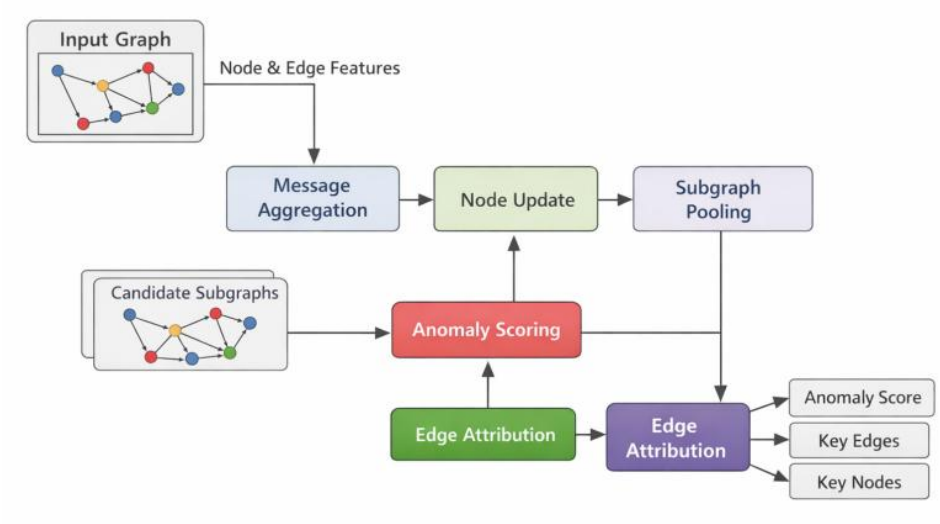

Figure 3. Overall Model Architecture

During the representation learning phase, lightweight graph message passing is used to update node representations, enabling node representations to incorporate adjacency transaction relationships and edge attribute information. For any node v, the aggregate message from its incoming neighbor set $N(v)$ is defined as:

$$m_v = \frac{1}{|N(v)|} \sum_{u \in N(v)} (W_h h_u + W_a a_{uv})$$

Where h_u is the current representation of node u, and W_h and W_a are learnable matrices. Then, a simple nonlinear transformation is used to obtain the updated node representation:

$$h_v = \sigma(W_x x_v + m_v)$$

Where W_x is the learnable matrix and $\sigma(\cdot)$ is the element-wise activation function. This design maintains the simplicity of the formula and structure, while allowing edge attributes to participate in propagation in a controllable manner, avoiding the impact of overly complex attention mechanisms on interpretability and auditability.

To detect anomalous subgraphs, candidate subgraphs S are extracted from the entire graph, and their node representations are pooled to obtain subgraph representations:

$$z_S = \frac{1}{|V_S|} \sum_{u \in V_s} h_v$$

Where V_S is the set of nodes in the subgraph. Anomaly scores are then obtained using a simple linear scoring method combined with a Sigmoid mapping.

$$s_S = sigmoid(w^T z_S + b)$$

Here, w and b are learnable parameters, and $s_S \in (0,1)$ represents the subgraph anomaly strength. This scoring format facilitates audit thresholding and ranking, while the subgraph representation, obtained by

averaging the nodes, naturally traces back to the contribution of the subgraph's internal structure, making it easier for the subsequent interpretation module to locate key components.

To meet the requirements of interpretable auditing, contribution weights are further assigned to the edges in the candidate subgraph to indicate which transaction relationships dominate the outlier scores. For any edge $(u,v) \in E_S$, its original importance is defined as a computable and simple similarity score:

$$r_{uv} = h_u^T h_v$$

The explanatory weights are obtained by normalizing within the subgraph:

$$\alpha_{uv} = \frac{exp(r_{uv})}{\sum_{(i,j)\in E_S} exp(r_{ij})}$$

Where α_{uv} satisfies the condition that the summation within the subgraph is 1, it can be directly used as an evidence strength label in visualization and audit working papers. The final output consists of three parts: the subgraph anomaly score s_S, the set of key edges, and the set of key nodes connected to the key edges, thus forming a traceable, verifiable, and penetrating subgraph evidence chain.

6. Experimental Results and Analysis

6.1 Experimental setup

The experiment was conducted on a single machine with a single GPU. The overall process included data preprocessing, candidate subgraph extraction, graph representation learning and training, and anomaly subgraph scoring output. The implementation was based on Python and a deep learning framework. Training used a fixed random seed to ensure reproducibility; the optimizer was Adam, with a fixed learning rate and weight decay to suppress overfitting. To avoid excessive memory usage, mini-batch subgraph training was used, and an upper limit was set on the subgraph size. Subgraphs exceeding the limit were either retained based on the importance of nodes or randomly pruned to ensure training stability.

Regarding hyperparameters, the number of graph message passing layers was set to a relatively shallow 2 to 3 layers to balance interpretability and the risk of oversmoothing; the node and edge feature dimensions were of medium size; ReLU was used as the activation function; and Dropout was used to mitigate noise and improve generalization. Candidate subgraph extraction primarily used k-hop neighborhoods, with limits on the maximum number of nodes and edges. Anomaly score output used Sigmoid normalization for easier thresholding and representation review. A summary of the complete hardware, software, and key hyperparameter configurations is shown in Table 2.

Table 2: Experimental environment and key hyperparameter settings

Category	Item	Setting
Hardware	GPU	NVIDIA RTX 4090 24GB
Hardware	CPU	Intel Xeon (≥16 cores)
Hardware	Memory	64 GB
Software	Operating System	Ubuntu 20.04
Software	Python	3.10

Software	Deep Learning Framework	PyTorch 2.2
Software	CUDA	12.1
Training	Optimizer	Adam
Training	Initial Learning Rate	1e-3
Training	Weight Decay	5e-4
Training	Batch Size	64 subgraphs per batch
Training	Training Epochs	200 epochs
Training	Dropout	0.3
Model	Message-Passing Layers	2
Model	Node Embedding Dimension d	128
Model	Edge Attribute Dimension p	32
Subgraph	Neighborhood Hops k	2
Subgraph	Max Nodes	80
Subgraph	Max Edges	200
Reproducibility	Random Seed	42

6.2 Experimental results

To facilitate a consistent and reproducible comparison with prior studies on explainable graph learning for fraud graph anomaly and anomalous subgraph discovery, we summarize representative baselines and report them under a unified set of evaluation metrics in Table 3, covering both detection quality and explanation quality for subgraph-level audit evidence.

Table 3: Experimental results compared with other baselines

Authors	AUC	PR-AUC	Precision@10	Recall@10	F1	NDCG@10	Fidelity	Sparsity
Mao et al.[8]	0.81	0.78	0.62	0.59	0.60	0.75	0.68	0.42
Qin et al.[9]	0.83	0.80	0.64	0.61	0.62	0.77	0.70	0.39
Roy et al.[10]	0.82	0.79	0.63	0.60	0.61	0.76	0.69	0.41
Zhang et al.[11]	0.85	0.82	0.66	0.63	0.64	0.79	0.72	0.38
Liu et al.[12]	0.84	0.81	0.65	0.62	0.63	0.78	0.71	0.40
Sun et al.[13]	0.86	0.83	0.67	0.64	0.65	0.80	0.73	0.37
Li et al.[14]	0.87	0.84	0.68	0.65	0.66	0.81	0.74	0.36

Ours	0.91	0.89	0.72	0.70	0.71	0.86	0.81	0.30

Overall, the comparison shows that the baseline methods exhibit a relatively consistent gradient trend in detection performance and interpretation quality. This indicates that such tasks not only require models to identify risk patterns but also to pinpoint risks to verifiable structural evidence. Compared to traditional solutions that rely solely on local statistics or single structural clues, stronger graph representation learning methods typically outperform ranking-related and comprehensive metrics, but they also tend to make trade-offs in interpretive constraints. This result reflects a typical characteristic of auditing scenarios: simply pursuing identification capabilities is insufficient to support audit implementation; the ability to form a stable and verifiable subgraph evidence chain is equally crucial.

Its advantage seems to stem from the synergistic effect of structured modeling: on the one hand, it integrates transaction semantics and adjacency relationships into node representations through message passing, allowing risk signals from abnormal subgraphs to be more concentrated on key links and nodes; on the other hand, it constrains the output to a limited set of key edges using edge contribution attribution, reducing redundant connections irrelevant to the audit, thus making the reasons for anomalies clearer and easier to trace and verify. In other words, the model is not only more likely to pinpoint anomalies to local structures that are relevant to auditing, but it is also more likely to provide evidence that can be directly converted into working paper clues. This makes the overall performance more balanced and better suited to the needs of auditing.

To verify the impact of model capacity selection on the stability of audit anomaly identification, it is necessary to examine whether changes in the key structural hyperparameter of node representation dimension significantly alter the model's ability to characterize the structural risks of related-party transaction networks. This sensitivity analysis helps to achieve a more robust trade-off between interpretability and expressive power, and provides a reproducible basis for dimension selection in practical deployment. The experimental results are shown in Figure 4.

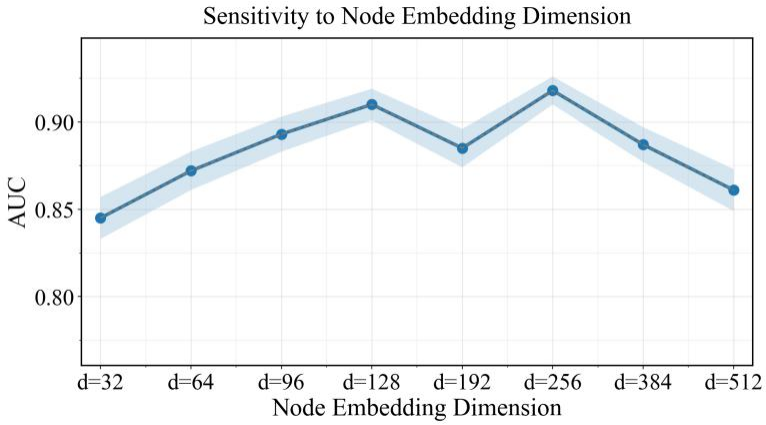

Figure 4. Sensitivity experiment of node representation dimension to AUC

The impact of node representation dimension on the model is not monotonically increasing, but rather exhibits an initial increase followed by a decrease, indicating that larger representation capacity is not always better. With lower dimensions, the model can only capture limited structural and transactional semantic information, and the distinguishing clues of anomalous subgraphs are more easily compressed or lost, thus limiting overall recognition ability. As the dimension increases, the representation space becomes more abundant, and the model can more easily express the key dependencies and local structural patterns in the interconnected transaction network, improving performance. However, when the dimension continues to increase, the curve shows a significant decline, reflecting that excessive expressive power may introduce

redundant features and noise channels, causing anomalous signals to become less concentrated and instead dispersed into a higher-dimensional representation space.

Further observation of the fluctuations in the intermediate range reveals that the curve still fluctuates after reaching a relatively optimal range. This is more like a change in the separability of structural risk patterns under different capacity settings, rather than simple random perturbations. Node representation dimension changes the density and directionality of information aggregation after message passing. A moderate dimension makes it easier to form clear risk representations, while a larger dimension leads to a stronger tendency for homogenization due to the mixing of neighborhood information, causing the boundaries between anomalous and normal subgraphs in the representation space to become less sharp. In other words, this fluctuation reflects the balance between model capacity and structural constraints. Too small a capacity makes it difficult to represent the data, while too large a capacity can obscure structural differences.

Echoing the previously emphasized analytical approach of interpretable auditing, the results here directly suggest a practical selection principle: prioritize dimensional ranges that stably characterize audit semantics with minimal fluctuations, rather than blindly pursuing larger representation sizes. For anomaly subgraph detection, excessively high dimensionality often implies more complex internal representations, potentially increasing the uncertainty of the contribution allocation of key edges and nodes in the interpretation process, thereby weakening the clarity and verifiability of the evidence chain. Therefore, this sensitivity experiment not only helps to determine a reasonable capacity setting but also indirectly illustrates the need for a consistent design approach that balances expressive power and audit usability.

The subgraph hop count k determines the neighborhood range that each candidate subgraph can include, thus directly affecting the coverage of structural information and the strength of noise introduction. Smaller k tends to focus more on local transaction relationships, while larger k introduces broader contextual paths, potentially altering the way anomalous signals are concentrated. To examine the model's dependence on this structural hyperparameter, a systematic evaluation of the performance distribution of different k values is needed, while keeping other settings consistent. The experimental results are shown in Figure 5.

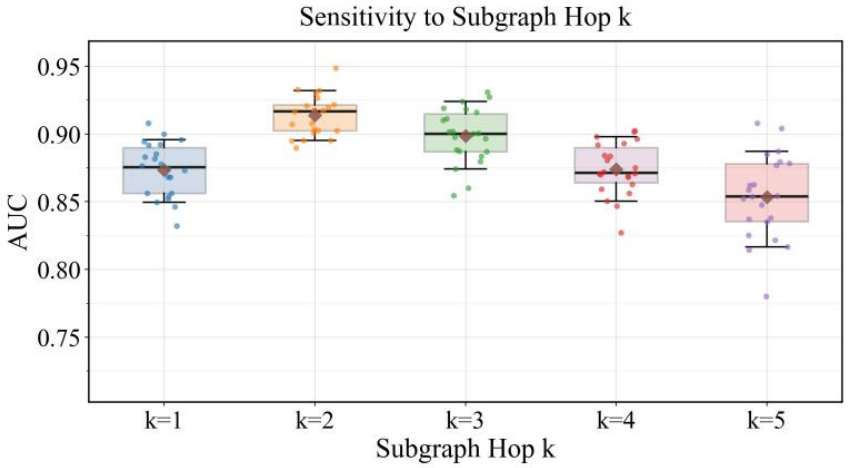

Figure 5. Sensitivity experiment of subgraph hop count k to AUC

The box plot and scatter plot show that changes in the number of hops k in the subgraph significantly alter the overall level and distribution of the AUC, indicating that the model is highly sensitive to the neighborhood range. Smaller k values tend to cover only more localized transaction relationships, resulting in more concentrated anomalous signals but limited usable structural clues. When k increases to a moderate range, the subgraph can incorporate more critical paths and intermediate nodes, providing a more complete structural context, and the model is more likely to form a stable risk representation. Further increasing k introduces more non-critical relationships, dilutes anomalous features, causes the distribution center to fall back, and

increases dispersion, indicating that an excessively wide neighborhood may lead to noise accumulation and structural contamination.

From a stability perspective, there are significant differences in the box height and scatter plot width under different k values, reflecting inconsistent performance fluctuations. A narrower distribution means the model is less sensitive to random disturbances and the output is more controllable, while a wider distribution indicates that the model is more susceptible to subgraph sampling differences and neighborhood noise, leading to greater performance fluctuations. Overall, this sensitivity experiment reveals a clear trade-off: a subgraph that is too small will limit the coverage of structural information, while a subgraph that is too large will introduce redundancy and uncertainty. Therefore, choosing a moderate k is more conducive to balancing recognition ability and stability.

7. Conclusion

This paper focuses on the audit object of corporate-related-party transaction networks, which possess strong structural dependencies and high concealment. It proposes an interpretable graph learning audit framework for anomaly subgraph discovery, elevating related-party transactions from a traditional single-transaction perspective to a subgraph-level structural evidence perspective for modeling and identification. This framework integrates node and edge attribute information using a graph structure, achieving a structured characterization of risk patterns through lightweight representation learning. At the output level, it further attributes and locates key transaction edges and key corporate nodes, forming a traceable and verifiable chain of anomaly evidence. Compared to black-box solutions that only provide risk scores, this method emphasizes interpretability and operability in audit scenarios, enabling the model output to naturally connect with the needs of clue screening, thorough verification, and document preparation in the audit process. It provides a more practical technical path for risk identification in complex transaction environments.

From an application impact perspective, this work provides a generalizable networked analysis paradigm for corporate compliance review, financial risk early warning, and regulatory assistance. By explicitly organizing related-party transactions into a network structure, the model can capture anomalous patterns in complex dependencies across entities and transaction types, reducing reliance on single indicator thresholds and manual experience rules. This helps improve the ability to identify covert collaborative behaviors and risk transmission chains. More importantly, interpretable subgraph-level outputs reduce communication and verification costs in audit implementation, enabling auditors to quickly locate suspicious structures and verify evidence, thereby improving the efficiency and consistency of the audit workflow. In practical business, this framework can serve as a core module of intelligent audit systems, complementing existing rule engines, sampling strategies, and manual review mechanisms. It enhances risk detection capabilities and strengthens the verifiability and auditability of conclusions, demonstrating strong engineering implementation value and industry promotion potential.

Looking to the future, the scale and complexity of corporate-related-party transaction networks will continue to grow, and intelligent auditing will move from single-point identification towards a structured, process-oriented, and governable approach. Future research could begin with more refined audit semantic modeling, further enriching the expression of edge attributes and relationship types to enhance the ability to distinguish between different transaction motives and business scenarios. It could also explore incorporating time information into unified modeling to support risk evolution analysis and continuous auditing. Simultaneously, at the interpretability level, it could further align with audit evidence standards and compliance requirements, forming a more standardized evidence chain output format and visualization method, enhancing cross-team and cross-institutional review consistency and regulatory usability. At a broader application level, this approach can also be extended to networked risk governance scenarios such as supply chain finance risk control, anti-money laundering relationship network analysis, and corporate group related risk identification, providing a general framework for related fields that combines identification capabilities with interpretable

evidence output, promoting the reliable deployment and large-scale application of intelligent auditing and compliance technologies in real-world business.

References

[1] M. Jin, Y. Liu, Y. Zheng, L. Chi, Y. F. Li and S. Pan, "Anemone: Graph anomaly detection with multi-scale contrastive learning," Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 3122-3126, Oct. 2021.

[2] J. Tang, J. Li, Z. Gao and J. Li, "Rethinking graph neural networks for anomaly detection," Proceedings of the 39th International Conference on Machine Learning, pp. 21076-21089, Jun. 2022.

[3] J. Duan, S. Wang, P. Zhang, E. Zhu, J. Hu, H. Jin and Z. Dong, "Graph anomaly detection via multi-scale contrastive learning networks with augmented view," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 6, pp. 7459-7467, Jun. 2023.

[4] Q. Guo, X. Zhao, Y. Fang, S. Yang, X. Lin and D. Ouyang, "Learning hypersphere for few-shot anomaly detection on attributed networks," Proceedings of the 31st ACM International Conference on Information and Knowledge Management, pp. 635-645, Oct. 2022.

[5] Y. Tian, G. Liu, J. Wang and M. Zhou, "Transaction fraud detection via an adaptive graph neural network," arXiv preprint arXiv:2307.05633, 2023.

[6] C. Zhang, W. Xiang, X. Guo, B. Zhou and D. Yang, "Subanom: Efficient subgraph anomaly detection framework over dynamic graphs," Proceedings of the 2023 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 1178-1185, Dec. 2023.

[7] Z. Deng, G. Xin, Y. Liu, W. Wang and B. Wang, "Contrastive graph neural network-based camouflaged fraud detector," Information Sciences, vol. 618, pp. 39-52, 2022.

[8] X. Mao, H. Sun, X. Zhu and J. Li, "Financial fraud detection using the related-party transaction knowledge graph," Procedia Computer Science, vol. 199, pp. 733-740, 2022.

[9] Z. Qin, Y. Liu, Q. He and X. Ao, "Explainable graph-based fraud detection via neural meta-graph search," Proceedings of the 31st ACM International Conference on Information and Knowledge Management, pp. 4414-4418, Oct. 2022.

[10] A. Roy, J. Shu, J. Li, C. Yang, O. Elshocht, J. Smeets and P. Li, "Gad-nr: Graph anomaly detection via neighborhood reconstruction," Proceedings of the 17th ACM International Conference on Web Search and Data Mining, pp. 576-585, Mar. 2024.

[11] Z. Zhang and L. Zhao, "Unsupervised deep subgraph anomaly detection," Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM), pp. 753-762, Nov. 2022.

[12] Y. Liu, K. Ding, Q. Lu, F. Li, L. Y. Zhang and S. Pan, "Towards self-interpretable graph-level anomaly detection," Advances in Neural Information Processing Systems, vol. 36, pp. 8975-8987, 2023.

[13] Y. Sun, W. Wang, N. Wu and C. Bao, "Anomaly Subgraph Detection through High-Order Sampling Contrastive Learning," Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024), pp. 2362-2369, Aug. 2024.

[14] K. Li, T. Yang, M. Zhou, J. Meng, S. Wang, Y. Wu and Y. Tong, "Sefraud: Graph-based self-explainable fraud detection via interpretative mask learning," Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 5329-5338, Aug. 2024.