Multi-Chain Graphs of Graphs: A New Approach to Analyzing Blockchain Datasets

Bingqiao Luo
National University of Singapore
luo.bingqiao@u.nus.edu

Zhen Zhang*
National University of Singapore
zhen@nus.edu.sg

Qian Wang
National University of Singapore
qiansoc@nus.edu.sg

Bingsheng He
National University of Singapore
hebs@comp.nus.edu.sg

Abstract

Machine learning applied to blockchain graphs offers significant opportunities for enhanced data analysis and applications. However, the potential of this field is constrained by the lack of a large-scale, cross-chain dataset that includes hierarchical graph-level data. To address this issue, we present novel datasets that provide detailed label information at the token level and integrate interactions between tokens across multiple blockchain platforms. We model transactions within each token as local graphs and the relationships between tokens as global graphs, collectively forming a "Graphs of Graphs" (GoG) approach. This innovative approach facilitates a deeper understanding of systemic structures and hierarchical interactions, which are essential for applications such as link prediction, anomaly detection, and token classification. We conduct a series of experiments demonstrating that this dataset delivers new insights and challenges for exploring GoG within the blockchain domain. Our work promotes advancements and opens new avenues for research in both the blockchain and graph communities. Source code and datasets are available at `https://github.com/Xtra-Computing/Cryptocurrency-Graphs-of-graphs`.

1 Introduction

Machine learning techniques applied to blockchain graphs present significant opportunities for in-depth data analysis and innovative applications [1, 2]. A comprehensive analysis of graph structures and patterns reveals valuable insights into transaction activities, including the investigation of transaction patterns, identification of key players, and deployment of tokenomics frameworks [3, 2, 4, 5]. Advanced graph learning algorithms have shown promise in enhancing the detection of various fraudulent activities [6, 7, 8, 9, 10] and predicting market trends [11, 12]. However, existing studies in this field face significant limitations due to the restricted availability of open and extensive datasets that include graph-level data. Most labeled blockchain datasets focus on node-level or edge-level data, lacking in-depth graph-level or advanced hierarchical graph-level studies [13, 14]. Furthermore, most existing datasets are confined to single-chain data, which restricts the ability to compare and understand the complex characteristics of various blockchain systems [15, 16, 17, 18, 3].

Concurrently, the study of Graphs of Graphs (GoG), which captures intricate relationships and structures across various domains, is gaining traction. This framework is particularly beneficial in scenarios involving multiple levels of interaction or dependency, such as in chemical, social media,

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

and document collection domains [19, 20, 21]. Despite these advancements, most GoG datasets remain small and static, focusing predominantly on chemical and molecular interactions task [19, 22]. It is not clear whether existing GoG machine learning models can achieve satisfied performance in large-scale, real-life transaction networks.

To bridge these two gaps, we introduce novel datasets and a new GoG approach tailored to the blockchain ecosystem. On the blockchain, a wide variety of digital tokens represent diverse assets, such as DeFi tokens related to decentralized finance products and MEME tokens inspired by internet memes. While these tokens are distinct, they are interconnected, as they are implemented on the same blockchain and can interact with the same user groups. This interconnectivity allows us to design an advanced hierarchical approach that includes local graphs representing individual crypto token transactions and global graphs depicting inter-token relationships within the blockchain ecosystem. Our dataset covers two crucial aspects: first, it includes detailed label information for each token graph, categorizing tokens by behavior, including fraud identification; second, it integrates interactions between tokens across multiple blockchain platforms. Specifically, our dataset covers 268,282,924 transactions conducted by 18,600,142 cryptocurrency addresses, covering the transaction history of 24,316 tokens on three main EVM chains: Ethereum, Polygon, and Binance Smart Chain (BSC).

We conduct an in-depth analysis of the constructed GoG, employing systematic graph analysis and extensive machine learning techniques. Our findings indicate that tokens belonging to the same class can exhibit distinct graph characteristics, such as varying graph size, reciprocity, and clustering coefficient, depending on the blockchain they are on. Furthermore, tokens with a high number of edges in local graphs tend to possess high centrality in global graphs. Through experiments on machine learning models, we observe that methods based on GoG can outperform traditional GNN methods in anomaly detection, multi-class classification, and link prediction on blockchain graphs under specific conditions. However, we also note that existing GoG models often underperform in minor class classification, highlighting the need for more advanced techniques.

In summary, this work makes several key contributions:

- We introduce large-scale, cross-chain graphs-of-graphs datasets, enriching blockchain research with unprecedented depth of analysis.
- Our analysis of graph structures within the hierarchical approach reveals intriguing characteristics, underscoring the diversity of token graph structures across different chains.
- We investigate traditional graph machine learning models and GoG-based models in the datasets. Experimental results demonstrate that our datasets present new avenues and challenges for the blockchain and graph community.

2 Literature Review

Blockchain Dataset. A number of datasets have been developed for machine learning tasks on blockchain platforms. For instance, [15, 16, 17, 18] proposed Ethereum datasets specifically for account detection and link prediction. [23] introduced datasets for tokens and liquidity pools, conducting statistical analyses of tokenomics for Ethereum and BSC. [24] utilized both on-chain and off-chain data for predicting crypto trading prediction. In the realm of blockchain graph datasets, one pioneering benchmark is Chartalist [14], which encompasses multiple tasks across Bitcoin, Ethereum, and Dashcoin. However, the inherent differences in blockchain types, such as unspent transaction output (UTXO) and account-based systems, pose challenges for comparing tasks across these varied architectures. Recent studies have also focused on Ethereum's NFT markets. For example, [3] introduced a live graph lab for temporal graphs, facilitating the study of open, dynamic, and real transaction graphs from Ethereum NFT transactions. Moreover, [13] highlighted the significant role of linking on-chain Ethereum accounts with off-chain X accounts, emphasizing the value of off-chain data in enhancing Ethereum analysis. Despite these advancements, many studies remain focused on single chains, predominantly Ethereum, and concentrate on node-level or edge-level tasks. This narrow focus may limit the generalizability of their findings.

Graph Representation Learning. Graph representation learning transforms high-dimensional, sparse graph data into compact, dense vectors [25]. The main objective is to produce representation vectors that effectively capture both structural and feature information of extensive graphs [26]. Among various tasks in this field, one key task is graph classification, which focuses on predicting

the properties of whole graphs [27]. This task is widely used in social community analysis [28, 25] and molecular property prediction [29, 30]. Numerous GNN-based algorithms have been proposed to address graph classification [31, 32, 33, 34]. Generally, these algorithms employ the message-passing paradigm to iteratively refine node representations, followed by a graph pooling function to generate graph-level representations [35, 26].

Graphs-of-Graphs. Graphs-of-graphs (GoG) extends traditional graph theory by structuring individual graphs as nodes within a larger, interconnected graph. This structure enables the analysis of complex relationships between distinct graph-structured data [36]. Initial studies applied GoG to rank nodes in domain-specific networks [37] and developed clustering methods using non-negative matrix factorization (NMF) for multi-view and multi-domain graph clustering [21]. Later research applied the GoG approach to GNNs to enhance graph classification tasks [20, 38]. Recent efforts have furthered GoG models to improve prediction capabilities in chemical and drug interactions [19, 22, 39]. Additionally, [40] explored the use of multi-layer network models within Ethereum and Ripple for anomalous event analysis, demonstrating the effectiveness of similarly structured concepts in blockchain analytics. However, the application of GoG in blockchain datasets remains underexplored, indicating a potential area for further research.

In summary, we provide a detailed comparison of related and public datasets with our dataset in Table 1. Specifically, our dataset includes these unique features: (1) as a graphs-of-graphs dataset, it comprises large-scale local graphs, dense global graph structures, and real-life temporal edges; (2) as a blockchain graph dataset, it stands as the first large-scale hierarchical graphs-of-graphs dataset, encompassing multi-chain transactions and graph-level labels.

Table 1: Comparisons among open-source graphs-of-graphs and blockchain graph datasets with ours. The symbol "-" indicates data that is not related or applicable.

Dataset	Field	Graph-level (token) labels	Multi-chain comparability	Density global graph	Avg. Num. local graph node	Avg. Num. local graph edge	Avg. global graph intra-inter edge	Dynamic
Graphs-of-graphs datasets								
CCI900 [39]	Chemical	-	-	4.4×10^{-4}	25.4	26.5	1707.3	✗
CCI950 [39]	Chemical	-	-	4.8×10^{-4}	26.2	27.4	511.4	✗
NetBasedDDI [39]	Drug	-	-	0.7×10^{-1}	24.8	26.7	442.2	✗
ZhangDDI [39]	Drug	-	-	0.3	25.2	27.0	1490.0	✗
ChChMiner [39]	Drug	-	-	0.5×10^{-1}	27.8	29.6	1418.9	✗
DeepDDI [39]	Drug	-	-	0.1	27.5	29.2	6153.8	✗
Arxiv [38]	Text	-	-	2.8×10^{-4}	30.9	200.1	23.31	✗
QQ [38]	Social	-	-	2.7×10^{-3}	291.2	2467.7	800.6	✗
Blockchain graph datasets								
Chartlist [14]	Blockchain	✗	✗	-	-	-	-	✓
LiveGraphLab [3]	Blockchain	✗	✗	-	-	-	-	✓
EX-Graph [13]	Blockchain	✗	✗	-	-	-	-	✓
Ours	Blockchain	✓	Ethereum	0.3	1493.7	2225.2	14273.2	✓
			Polygon	0.5	1184.2	2523.6	525.1	✓
			BSC	0.6	1650.5	3346.4	4660.0	✓

3 Dataset Details

3.1 Background

Blockchain and Cryptocurrency. Blockchain technology operates on a decentralized network secured by cryptographically linked blocks. This structure ensures data immutability and verifiability, supporting secure, irreversible, and transparent transactions. Cryptocurrencies, built on this technology, facilitate secure digital transactions without a central authority, enhancing user anonymity while complicating fraud detection.

EVM and ERC20. The Ethereum Virtual Machine (EVM) supports an account-based model that enables direct value transfers and complex features like smart contracts. This model has become a standard for blockchain networks and decentralized applications, utilized across prominent networks such as Ethereum, Polygon, and Binance Smart Chain (BSC). The ERC20 standard on Ethereum and the BEP20 standard on BSC provide a framework for fungible tokens, promoting interoperability and simplifying the trading process across platforms.

Accounts and Transactions. On EVM-compatible chains, two primary account types exist: externally owned accounts (EOAs) and smart contracts. EOAs resemble traditional bank accounts but are controlled by individual private keys, while smart contracts are programmable accounts that execute automatically under specified conditions. Each account has a unique address, maintains a balance, and is controlled by a private key. Transactions include details such as the sender's and receiver's addresses, timestamps, values, and the transaction hash, ensuring that each transaction is immutable and traceable.

3.2 Data Collection

This section summarizes the statistics of our datasets, focusing on ERC20 tokens on Ethereum and Polygon, and BEP20 tokens on BSC. These token standards are among the most popular in the blockchain industry [41]. Our datasets include three independent sets, each targeting a different blockchain, comprising a total of 268,282,924 transactions conducted by 18,600,142 addresses across 24,316 tokens. These transactions record the full history of these tokens from the inception to February 2024. Detailed statistics are presented in Table 2.

Table 2: Statistics of the datasets.

Chain	# Token	Start Month	End Month	# Transactions	# Addresses	# Categories
Ethereum	14,464	2016-02	2024-02	81,788,211	10,247,767	290
Polygon	2,353	2020-08	2024-02	64,882,233	1,801,976	112
BSC	7,499	2020-09	2024-02	121,612,480	6,550,399	149

Transaction Data. All blockchain transactions are transparent, traceable, and publicly available, achieved through the secure linkage of blocks using cryptographic techniques [42]. Prominent blockchain explorers provide tools to easily access blockchain transaction data. We utilize public APIs from Etherscan[2], Polygonscan[3], and Bscscan[4] to facilitate the collection of token transactions. Each transaction includes sender and recipient addresses, transfer value, timestamp, unique transaction hash, and other relevant details.

Tags. We collect category tags of the tokens from the three prominent blockchain explorers as labels. We reviewed the tags for all ERC20 and BEP20 tokens launched no later than February 2024 across these platforms. We filtered the tokens to include only those with more than 10,000 addresses or 1,000,000 transactions to ensure fair data distribution. The label details are as follows:

- For fraud cases, we labeled tokens flagged by explorers as suspicious phishing or hack tokens. These include various kinds of spammed tokens, such as those that have been spammed to many users or those that pretend to be famous tokens, like fake USDT. In total, 7,198 fraud tokens were identified, representing 29.6% of the dataset.
- For other classes, we labeled tokens using the category tags given by the explorers. The most popular classes in the dataset include DeFi tokens, which are related to decentralized finance products; MEME tokens, often inspired by internet memes and characters; and Gaming tokens, which are associated with electronic gaming.

Figure 1a represents the diversity of categories within our dataset as a label cloud. In total, 313 categories are found in these tokens. However, the distribution of categories is very skewed, as shown in Figure 1b. Specifically, the top 5 categories on each chain can cover more than half of all tokens in our dataset.

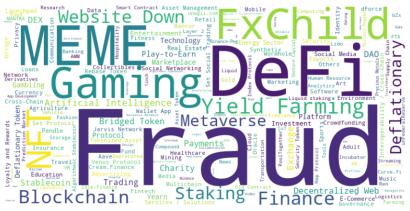
3.3 Graph Construction

In this section, we present how we construct our Graphs of Graphs (GoG) datasets. We build two types of graphs: local graphs that represent transactions of tokens, and global graphs that represent token-token relationships. Figure 2 illustrates a sample of our GoG structure.

[2] `https://etherscan.io/`

[3] `https://polygonscan.com/`

[4] `https://bscscan.com/`



(a) Tokens categories label cloud.

(b) Top 5 categories with highest total number.

Figure 1: Token categories analysis.

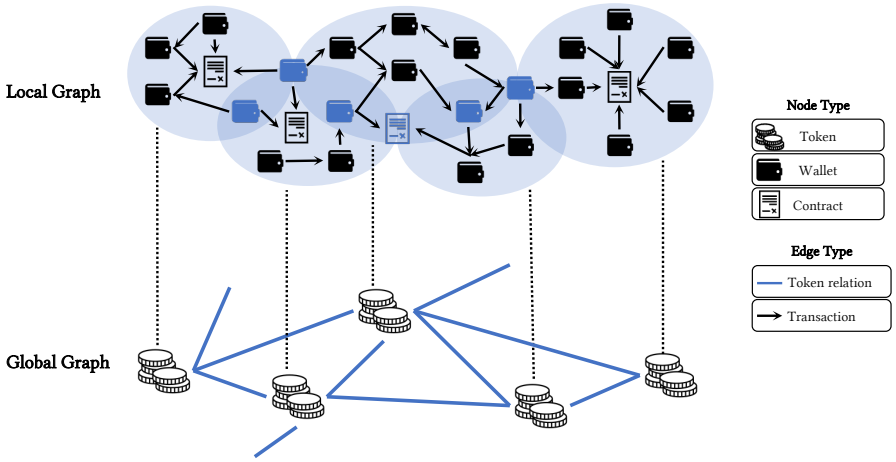


Figure 2: A sample GoG structure with 5 tokens. Blue wallets and contracts are involved in transactions of multiple tokens, while black wallets and contracts are involved in only one token's transactions.

3.3.1 Local Graph Construction

A local graph represents transactions of a single token. Defined as $G_{local} = (N_L, E_L)$, the graph consists of $|N_L|$ nodes representing accounts and $|E_L|$ edges representing transactions. Each edge $(e = (u, v, w, t))$ signifies a transaction from account u to v, involving a value w transferred at time t. The timestamp t indicates when the transaction occurs, with the first transaction timestamp marking when the token becomes active on-chain.

3.3.2 Global Graph Construction

A global graph aims to model the correlation of various tokens across blockchain platforms. Specifically, we model the transaction networks of individual tokens into nodes, forming a graph $G_{global} = (N_G, E_G)$. N_G denotes the set of all local graphs, and E_G represents the inter-token relationships. The edges in the global graph are weighted by the Jaccard Coefficient, defined as:

$$\text{Edge weight: } J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the node sets of the local transaction graphs for two distinct tokens. The Jaccard Coefficient quantifies the degree of overlap in user bases between different tokens, offering insights into the inter-connectedness and user-sharing across tokens. Each edge weight reflects this inter-connectedness, providing a measure of relational strength and activity overlap between tokens at a global scale. This approach of finding common addresses is inspired by previous studies on social media groups [20] and multi-layer blockchain analysis [40]. We remove the null address[5] when measuring to prevent all tokens involved in transactions with null addresses from being connected. As new transactions are executed on the blockchain platforms, global edges in our graph dynamically adapt to changes in local transaction information, as detailed in Appendix D.

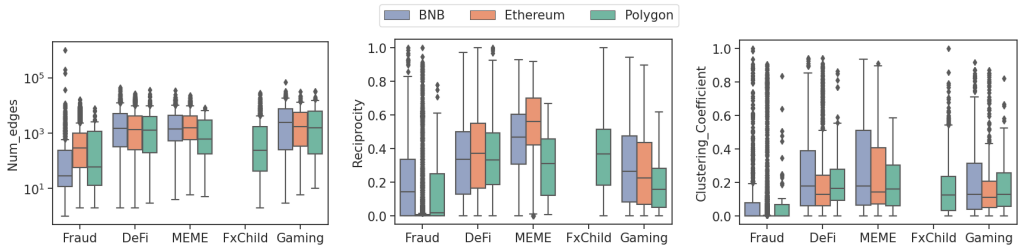
[5]Null address: 0x0000000000000000000000000000000000000000.

4 Observations and Analysis

4.1 Local Graph Analysis

We examine several graph properties within distinct classes of local graphs to deepen our understanding of token transfer networks. Figure 3 displays three crucial graph properties: **number of edges**, **reciprocity**, and **clustering coefficient**, segmented by the top five most prevalent classes across three blockchains. Notably, the "FxChild" class is exclusive to Polygon, while the other four classes are observed across all chains.

Finding (1): The distribution of token categories varies across different chains. Tokens within the same class can exhibit distinct network characteristics depending on the blockchain.



(a) Number of edges (b) Reciprocity (c) Clustering coefficient

Figure 3: Distribution of graph properties across three chains.

Figure 3a illustrates the distribution of the number of edges across token graphs on three blockchains. While DeFi, MEME, and Gaming tokens generally show a comparable number of edges, Fraud tokens consistently exhibit fewer edges across all platforms, with a notably higher node count on Polygon. This suggests less connectivity among participants in fraud-related activities.

Figure 3b shows the reciprocity of these token graphs, reflecting the proportion of mutual connections. Results indicate that MEME tokens display higher reciprocity, aligning with the interactive nature of these communities. In contrast, Fraud tokens show the lowest reciprocity, indicating that fraudulent transactions are less likely to be reciprocal, possibly due to their unilateral nature.

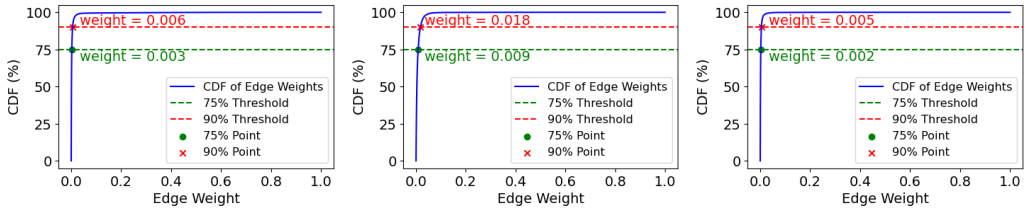
Figure 3c presents the clustering coefficient, which indicates how closely nodes in a graph cluster, reflecting the formation of tight-knit groups or collusive clusters. A higher average clustering coefficient suggests the presence of prevalent cliques or active trading communities. Fraud tokens demonstrate the lowest clustering coefficients, suggesting sparse connectivity, whereas MEME tokens exhibit the highest, indicative of tight-knit communities. Additionally, tokens on BSC display the widest range and highest clustering coefficients compared to those on Ethereum and Polygon, pointing to more clustered network structures on BSC.

4.2 Global Graph Analysis

To understand how the Graph of Graphs (GoG) approach enhances our comprehension of the intricate relationships and interactions within the ERC20 markets, we perform sophisticated network analyses, focusing on **edge weight analysis** and **node centrality** to identify influential tokens.

Finding (2): The predominance of low edge weights across the network suggests limited interaction between different tokens. High weights are predominantly observed among local graphs within the same class, especially those implicated in fraudulent activities.

Edge weights in the global graph, determined by the Jaccard coefficient of common nodes, illustrate the interconnectedness of tokens based on shared investors or smart contracts. We analyze the distribution of edge weights in the global networks, as demonstrated in Figure 4. This distribution is predominantly characterized by small values, indicating sparse connections across most token pairs, although there are exceptions with a few highly interconnected node pairs. Moreover, as shown in Table 3, the contract pairs with the highest weights on each blockchain consistently involve local graphs within the same class, notably marked by a prevalence of fraud-related contracts. This pattern suggests concentrated activity or potential collusive behaviors within these groups of tokens.



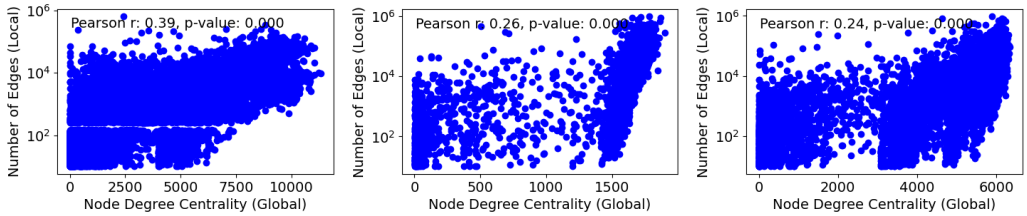
(a) Ethereum (b) Polygon (c) BSC

Figure 4: Cumulative distribution of edge weights in three global graphs.

Table 3: Top 3 connected contracts with highest weight across different chains.

Chain	Contract1	Contract2	Class1	Class2	Edge Weight
Ethereum	0x3d09...c61cce	0x6752...1e761a	Fraud	Fraud	1.0
	0xa034...145c1b	0x5fbf...762f24	Fraud	Fraud	1.0
	0xb3f6...7d8429	0x6249...29af88	Deprecated	Deprecated	1.0
Polygon	0x8db0...06f7ec	0x36f5...c72c5b	FxChild	FxChild	1.00
	0xbbcc...85e429	0x1a8a...6f5f68	FxChild	FxChild	0.98
	0xa7e8...9304f7	0xee35...52699c	Gaming	Gaming	0.97
BSC	0xaf71...020ac2	0x362d...881ffc	Play-to-Earn	Play-to-Earn	1.0
	0xaefe...55eab2	0x9775...4e74ec	Fraud	Fraud	1.0
	0x5fb0...42a2e4	0x9775...4e74ec	Fraud	Fraud	1.0

Finding (3): A higher number of edges in local graphs typically correlates with central roles in global graphs, highlighting a strong relationship between transaction activity and centrality within the blockchain ecosystem.



(a) Ethereum (b) Polygon (c) BSC

Figure 5: Global node degree centrality vs. local graph number of edge.

Node degree centrality is a critical metric for identifying hub nodes—those that are most interconnected within the market. We analyze node degree centrality within each global graph, aiming to identify tokens that act as central nodes in their respective markets. First, we find that tokens with high node degrees tend to belong to the most popular classes in the blockchain. More than 70% of the top 10 nodes with the largest degree centrality belong to the most popular classes, including Fraud, FxChild, and Gaming. Second, we explore the correlation between node degree centrality in global graphs and the number of edges in local graphs, as illustrated in Figure 5. Results reveal that token graphs with a higher number of edges are more likely to become high-degree nodes in global graphs. Interestingly, we found that very few tokens have between 150 and 250 transactions, which creates a noticeable gap in the Figure 5a.

5 Downstream Applications

The observations presented earlier provide a comprehensive overview of our GoG datasets. Our findings highlight how the GoG approach enhances our understanding of the complex transfer graphs across various token classes. These properties introduce new challenges for several downstream tasks. In this section, we investigate two studied tasks to address these questions:

- Q1: How can the GoG approach improve accurate classification of token graph categories?
- Q2: How can the GoG approach improve fraud detection performance on blockchain graphs?
- Q3: How do graph learning models perform across different blockchain datasets?

Additionally, we explore investment prediction as a future edge prediction task in Appendix F. These experiments demonstrate the capability of the GoG approach to handle various graph learning tasks in the blockchain domain, thereby underscoring its potential for practical applications.

5.1 Multi-class Graph Classification

Graph classification is a crucial aspect of graph learning research [43]. By predicting the attributes of each graph within a collection, graph classification facilitates exploration across various domains, including image classification [44], document analysis [45], and chemical discovery [46]. In this section, we focus on multi-class graph classification, aiming to categorize tokens into distinct classes. Given the scarcity of some minor classes, we concentrate on the top token categories within each chain for our classification task. Specifically, we classify tokens into two groups: (1) the top 3 categories, and (2) the top 5 categories.

Models. We compare two groups of models. Group 1 consists of GNN models applied to individual graphs, including: (1) GCN [31], (2) GAT [47], (3) GIN [48], (4) ResidualGCN [49], and (5) GraphSage [50]. Group 2 comprises GNN models tailored for handling collections of graphs, including: (1) SEAL [20], which applies a self-attentive graph embedding approach using GCN as the base model to embed individual graph instances into fixed-length vectors for classification; (2) GoGNN [19], which enhances GCN's capabilities by incorporating an attention-based pooling mechanism and GAT to effectively identify key substructures within local graphs; and (3) DVGGA [39], which combines a denoising autoencoder with a self-attentive GNN and readout function.

Settings. We filter out token graphs with fewer than five nodes or edges to maintain data integrity. After applying this criterion, less than 2% of tokens were removed from all three datasets, ensuring that our analysis still represents the majority of the data. Guided by insights from subsection 4.2, we establish a threshold for edge weight, including only weights exceeding 0.01 to identify closely connected tokens. When experimenting with both groups of models, we utilize incoming degree, outgoing degree, and total degree as node features for the local graphs. For a fair comparison, we conduct all experiments as supervised learning tasks. The dataset is divided into training and testing sets following an 80/20 ratio. Then, we employ Macro-F1 and Micro-F1 as evaluation metrics. Macro-F1 computes the F1-score separately for each class and then averages them, giving equal importance to all classes regardless of their frequency. In contrast, Micro-F1 assigns more weight to classes with higher frequencies, reflecting their prevalence in the dataset.Due to the imbalance in our dataset, we primarily use Macro-F1 for model comparisons. Each experiment is repeated three times with different seeds, and we report the average performance and standard deviation.

Results. The results of the 3-class and 5-class classification tasks are summarized in 3. Several notable observations emerge from these results. First, GoG models exhibit superior performance compared to individual GNN models across most tasks in both classification scenarios. Specifically, SEAL demonstrates the best F1-macro performance, showing up to 28% and 16% improvements respectively in the 3-class category on BSC, and up to 44% and 11% improvements respectively in the 5-class category on Ethereum, compared to the average performance of non-GoG models. Second, as the classification task becomes more complex by including additional minor classes, the performance of both model groups notably declines. The advantage of GoG models over individual GNN models diminishes in 5-class classification compared to 3-class classification, emphasizing the necessity for further development of advanced GoG models, especially for minor-class classification. Third, all model groups demonstrate less satisfactory performance on the Polygon dataset. This could be attributed to the dataset's smaller size and greater imbalance compared to others. Therefore, there is a clear need for devising robust graph learning models capable of effectively capturing the intricacies of the Polygon dataset. In Appendix G, we conducted experiments predicting the class label of younger tokens using the information about older tokens. Results show that for Ethereum and BNB, the performance shows slight differences from the results in Table 4. However, for Polygon, the performance deteriorates significantly.

Table 4: 3-class and 5-class classification performance by blockchain.

Model	Ethereum		Polygon		BSC	
	F1-macro	F1-micro	F1-macro	F1-micro	F1-macro	F1-micro
3-Class Classification						
GCN	62.48±6.31	85.05±1.38	28.82±1.86	74.24±0.83	51.43±5.93	57.02±3.37
GAT	60.22±7.04	84.62±1.23	29.90±2.60	73.94±1.79	54.48±6.15	59.96±3.19
GIN	39.79±11.02	78.58±3.07	28.82±1.53	74.26±0.83	43.29±2.93	55.90±2.86
ResidualGCN	62.85±6.07	84.18±1.50	28.50±0.35	74.37±0.18	50.73±4.59	56.78±2.29
GraphSage	64.17±8.53	85.51±2.05	31.71±2.56	74.48±0.68	56.70±6.12	61.36±2.78
SEAL	**67.31** ± 3.60	**86.65** ± 1.30	29.64±1.70	**74.51** ± 0.16	**63.77** ± 0.59	**65.59** ± 0.42
GoGNN	64.20±4.29	85.89±0.47	**36.11** ± 0.50	66.09±11.02	53.98±4.55	58.03±2.90
DVGGA	37.23±10.57	77.84±4.16	28.44±0.004	74.22±0.17	41.31±8.67	47.03±7.64
5-Class Classification						
GCN	36.88±4.90	78.37±1.81	16.40±0.65	68.79±0.66	27.49±3.27	43.09±2.12
GAT	36.46±4.44	78.46±1.42	**20.04** ± 3.54	68.85±1.08	29.85±3.12	44.97±2.77
GIN	19.24±4.62	71.31±2.18	16.41±0.65	68.82±0.63	22.01±2.32	41.33±3.19
ResidualGCN	33.72±5.56	76.69±1.91	16.47±0.86	68.6±12.30	23.82±5.09	40.33±3.90
GraphSage	38.91±5.31	79.73±1.97	18.01±1.64	68.88±0.65	30.22±3.34	46.15±2.20
SEAL	**45.09** ± 10.79	**81.59** ± 2.06	16.90±0.82	**69.02** ± 0.15	**31.83** ± 3.31	**46.50** ± 1.93
GoGNN	32.51±3.58	71.74±6.56	19.70±2.62	59.19±12.41	23.33±8.42	38.54±6.30
DVGGA	21.60±7.21	71.70±2.29	18.03±2.38	67.20±2.42	16.92±5.77	34.91±4.89

5.2 Graph Anomaly Detection

Anomaly detection is a significant task in machine learning with numerous applications, including anti-money laundering [51], social media analysis [52], and disease detection [53]. In this section, we focus on detecting anomalies in tokens, specifically identifying fraudulent tokens from non-fraudulent ones. Given the skewness of fraud and non-fraud tokens, we approach graph anomaly detection as an unsupervised learning task.

Models. We compare two groups of models. Group 1 includes anomaly detection methods for multivariate data, such as probabilistic and outlier ensemble methods. Specifically, we compare (1) COPOD [54], (2) IForest [55], (3) DIF [56], and (4) VAE [57]. Group 2 includes anomaly detection methods on graphs, primarily GNN+AE methods. Specifically, we compare (1) GAE [58], (2) DONE [59], (3) DOMINANT [60], (4) AnomalyDAE [61], and (5) CoLA [62]. Detailed introductions of these methods are presented in Appendix E.

Settings. We use the same dataset settings as in subsection 5.1 to filter out small token graphs and build global graphs. For multivariate data analysis, drawn from our observations in subsection 4.1, we measure various graph properties, including the number of nodes, edges, assortativity, density, and reciprocity. These features are normalized to ensure consistency across the dataset. Our experimental setup follows the frameworks provided by PyOD [63] and PyGOD [64]. The dataset is divided into train/validation/test sets following an 80/10/10 ratio. Evaluation metrics include the area under the curve (AUC) and Average Precision (AP). AUC measures the model's ability to rank anomalies higher than normal instances, while AP quantifies the precision-recall balance, providing insights into model performance regarding the anomaly detection task.

Results. The performance of graph anomaly detection methods across three blockchains is summarized in Table 7. Interestingly, on the BSC dataset, most graph outlier detection methods outperform those based on graph structural data. For example, AnomalyDAE shows up to 3.34% improvement in AUC and 54.20% improvement in AP, compared to the average performance of detection models based on graph structural data. However, on Ethereum and Polygon, methods based on graph structural data demonstrate superior performance. This variation may be attributed to differences in fraudulent token behaviors and network structures specific to each blockchain. Additionally, consistent with the findings in subsection 5.1, both groups of methods exhibit poorer performance on the Polygon dataset, highlighting the need for further research. In Appendix H, we explored an additional method to represent token graphs by employing the DeepWalk algorithms [65]. Results show that

Table 5: Graph anomaly detection performance by blockchain. We report the ratio of number of non-fraud:fraud case of each data at the top.

	Ethereum (8387: 6022)		Polygon (2257: 58)		BSC (6339: 1042)	
Model	AUC	AP	AUC	AP	AUC	AP
COPOD	83.27±1.09	27.25±0.4	60.52±13.27	**11.33** ± 6.49	52.87±2.09	14.18±0.69
IForest	84.10±0.55	26.93±0.56	64.33±11.43	10.79±5.67	58.36±2.83	11.58±1.57
DIF	**84.56** ± 1.31	32.69±0.95	68.04±10.11	7.99±2.06	51.57±0.49	17.52±2.05
VAE	67.25±1.61	31.46±0.49	**72.45** ± 10.41	10.56±5.09	59.03±0.20	18.70±1.13
GAE	70.85±2.58	31.21±0.68	62.16±0.09	3.85±0.01	56.33±1.25	17.11±0.35
DONE	74.93±2.91	29.03±0.92	62.21±0.30	1.95±0.07	65.86±3.70	10.64±1.10
DOMINANT	75.18±2.69	**43.14** ± 19.69	70.45±7.93	3.55±1.48	**78.87** ± 0.23	8.49±0.03
AnomalyDAE	65.82±8.47	39.24±10.09	60.94±3.06	3.72±0.42	62.49±9.23	**22.71** ± 6.98
CoLA	65.15±7.17	35.80±7.04	54.90±2.74	3.51±0.64	60.87±3.63	19.64±6.29

while GoG models benefit from the use of the DeepWalk algorithm, the performance of multivariate outlier detection methods decreases. In general, the adoption of the GoG approach presents new opportunities to enhance graph anomaly detection in the blockchain domain, as evidenced by the varied performance observed across different blockchains.

6 Conclusion

In this paper, we introduced a novel dataset based on the Graphs of Graphs (GoG) approach within the blockchain domain. Our dataset includes local graphs that detail individual token transactions and global graphs that model interactions between tokens across multiple blockchain platforms. This approach provides a comprehensive view of transaction activities within the blockchain ecosystem. We conducted systematic analyses and experiments using the GoG approach, revealing significant patterns and characteristics in the blockchain environment. Our findings suggest that GoG models have the potential improve various applications, such as link prediction, anomaly detection, and token classification, especially when compared to traditional GNN methods. We believe this work lays a foundation for future research in graph learning and encourages further exploration of the complex relationships within blockchain networks.

7 Border Impact and Limitation

Our datasets offer researchers fresh opportunities to explore and analyze blockchain graphs. Insights from this analysis could aid in developing stronger market structures and enhanced security protocols within crypto token platforms. Our classification and anomaly detection models aim to enhance predictive capabilities, particularly in situations where labels may be incomplete or unavailable. For instance, while only a small fraction of existing tokens has been labeled by blockchain explorers, there are over 900,000 ERC20 tokens on Ethereum, with new tokens being launched daily that may not yet be classified. Our models can predict these labels more quickly and provide detailed insights. By leveraging similarities within the data, the GoG approach reduces reliance on labeled data and improves the accuracy of predictions for unlabeled tokens.

However, our datasets have several limitations that must be acknowledged. First, the lack of restrictions on creating multiple accounts on the same blockchain allows a single entity to control multiple accounts. This can distort interaction patterns and connectivity metrics within our GoG datasets. Second, while our model is designed to enhance predictive capabilities in cases where labels are incomplete, it is limited by the fact that only a small fraction of existing tokens has been labeled by blockchain explorers. Although we rely on trusted blockchain explorers for labeling, there is always a risk of misclassification, especially with fraudulent tokens. Therefore, our model is intended to augment, not replace, human judgment; predictions should be viewed as suggestions that require further scrutiny. Third, the public nature of blockchain data raises privacy concerns. Our datasets link transactions to wallet addresses, potentially enabling the tracking of individual behaviors, which could lead to targeted advertising or surveillance.

Acknowledgments and Disclosure of Funding

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. The authors would like to thank reviewers for their helpful comments.

References

[1] Arijit Khan. Graph analysis of the ethereum blockchain data: A survey of datasets, methods, and future work. In *2022 IEEE International Conference on Blockchain (Blockchain)*, pages 250–257. IEEE, 2022.

[2] Qianlan Bai, Chao Zhang, Nianyi Liu, Xiaowei Chen, Yuedong Xu, and Xin Wang. Evolution of transaction pattern in ethereum: A temporal graph perspective. *IEEE Transactions on Computational Social Systems*, 9(3):851–866, 2021.

[3] Zhen Zhang, Bingqiao Luo, Shengliang Lu, and Bingsheng He. Live graph lab: Towards open, dynamic and real transaction graphs with nft. *Advances in Neural Information Processing Systems*, 36:18769–18793, 2023.

[4] Bartosz Kusmierz and Roman Overko. How centralized is decentralized? comparison of wealth distribution in coins and tokens. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6. IEEE, 2022.

[5] Weili Chen, Tuo Zhang, Zhiguang Chen, Zibin Zheng, and Yutong Lu. Traveling the token world: A graph analysis of ethereum erc20 token ecosystem. In *Proceedings of The Web Conference 2020*, pages 1411–1421, 2020.

[6] Sihao Hu, Zhen Zhang, Bingqiao Luo, Shengliang Lu, Bingsheng He, and Ling Liu. Bert4eth: a pre-trained transformer for ethereum fraud detection. In *Proceedings of the ACM Web Conference 2023*, pages 2189–2197, 2023.

[7] Sijia Li, Gaopeng Gou, Chang Liu, Chengshang Hou, Zhenzhen Li, and Gang Xiong. Ttagn: Temporal transaction aggregation graph network for ethereum phishing scams detection. In *Proceedings of the ACM Web Conference 2022*, pages 661–669, 2022.

[8] Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In *Proceedings of the 2018 world wide web conference*, pages 1409–1418, 2018.

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

[10] Bingqiao Luo, Zhen Zhang, Qian Wang, Anli Ke, Shengliang Lu, and Bingsheng He. Ai-powered fraud detection in decentralized finance: A project life cycle perspective. *arXiv preprint arXiv:2308.15992*, 2023.

[11] Wei Yin, Ziling Chen, Xinxin Luo, and Berna Kirkulak-Uludag. Forecasting cryptocurrencies' price with the financial stress index: a graph neural network prediction strategy. *Applied Economics Letters*, 31(7):630–639, 2024.

[12] Dan Lin, Jialan Chen, Jiajing Wu, and Zibin Zheng. Evolution of ethereum transaction relationships: Toward understanding global driving factors from microscopic patterns. *IEEE Transactions on Computational Social Systems*, 9(2):559–570, 2021.

[13] Qian Wang, Zhen Zhang, Zemin Liu, Shengliang Lu, Bingqiao Luo, and Bingsheng He. EXgraph: A pioneering dataset bridging ethereum and x. In *The Twelfth International Conference on Learning Representations*, 2024.

[14] Kiarash Shamsi, Friedhelm Victor, Murat Kantarcioglu, Yulia Gel, and Cuneyt G Akcora. Chartalist: Labeled graph datasets for utxo and account-based blockchains. *Advances in Neural Information Processing Systems*, 35:34926–34939, 2022.

[15] Dan Lin, Jiajing Wu, Qi Yuan, and Zibin Zheng. Modeling and understanding ethereum transaction records via a complex network approach. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(11):2737–2741, 2020.

[16] Dan Lin, Jiajing Wu, Qi Xuan, and K Tse Chi. Ethereum transaction tracking: Inferring evolution of transaction networks via link prediction. *Physica A: Statistical Mechanics and its Applications*, 600:127504, 2022.

[17] Jinhuan Wang, Pengtao Chen, Xinyao Xu, Jiajing Wu, Meng Shen, Qi Xuan, and Xiaoniu Yang. Tsgn: Transaction subgraph networks assisting phishing detection in ethereum. *arXiv preprint arXiv:2208.12938*, 2022.

[18] Tao Huang, Dan Lin, and Jiajing Wu. Ethereum account classification based on graph convolutional network. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(5):2528–2532, 2022.

[19] Hanchen Wang, Defu Lian, Ying Zhang, Lu Qin, and Xuemin Lin. Gognn: Graph of graphs neural network for predicting structured entity interactions. *arXiv preprint arXiv:2005.05537*, 2020.

[20] Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, and Junzhou Huang. Semi-supervised graph classification: A hierarchical graph perspective. In *The World Wide Web Conference*, pages 972–982, 2019.

[21] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. Flexible and robust multi-network clustering. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 835–844, 2015.

[22] Hanchen Wang, Defu Lian, Wanqi Liu, Dong Wen, Chen Chen, and Xiaoyang Wang. Powerful graph of graphs neural network for structured entity analysis. *World Wide Web*, 25(2):609–629, 2022.

[23] Federico Cernera, Massimo La Morgia, Alessandro Mei, and Francesco Sassi. Token spammers, rug pulls, and sniper bots: An analysis of the ecosystem of tokens in ethereum and in the binance smart chain (bnb). In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3349–3366, 2023.

[24] Yuan Li, Bingqiao Luo, Qian Wang, Nuo Chen, Xu Liu, and Bingsheng He. A reflective llm-based agent to guide zero-shot cryptocurrency trading. *arXiv preprint arXiv:2407.09546*, 2024.

[25] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, page 106207, 2024.

[26] Shima Khoshraftar and Aijun An. A survey on graph representation learning methods. *ACM Transactions on Intelligent Systems and Technology*, 15(1):1–55, 2024.

[27] Nan Yin, Li Shen, Mengzhu Wang, Long Lan, Zeyu Ma, Chong Chen, Xian-Sheng Hua, and Xiao Luo. Coco: A coupled contrastive framework for unsupervised domain adaptive graph classification. In *International Conference on Machine Learning*, pages 40040–40053. PMLR, 2023.

[28] Siyuan Liao, Shangsong Liang, Zaiqiao Meng, and Qiang Zhang. Learning dynamic embeddings for temporal knowledge graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 535–543, 2021.

[29] Kaveh Hassani. Cross-domain few-shot graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6856–6864, 2022.

[30] Yihang Yin, Qingzhong Wang, Siyu Huang, Haoyi Xiong, and Xiang Zhang. Autogcl: Automated graph contrastive learning via learnable view generators. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8892–8900, 2022.

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

[32] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yuguang Wang, Pietro Lio, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. *Advances in neural information processing systems*, 34:2625–2640, 2021.

[33] Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate learning of graph representations with graph multiset pooling. *arXiv preprint arXiv:2102.11533*, 2021.

[34] Zhengyang Mao, Wei Ju, Yifang Qin, Xiao Luo, and Ming Zhang. Rahnet: Retrieval augmented hybrid network for long-tailed graph classification. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3817–3826, 2023.

[35] Zemin Liu, Yuan Li, Nan Chen, Qian Wang, Bryan Hooi, and Bingsheng He. A survey of imbalanced learning on graphs: Problems, techniques, and future directions. *arXiv preprint arXiv:2308.13821*, 2023.

[36] Gregorio D'Agostino and Antonio Scala. *Networks of networks: the last frontier of complexity*, volume 340. Springer, 2014.

[37] Jingchao Ni, Hanghang Tong, Wei Fan, and Xiang Zhang. Inside the atoms: ranking on a network of networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1356–1365, 2014.

[38] Jia Li, Yongfeng Huang, Heng Chang, and Yu Rong. Semi-supervised hierarchical graph classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6265–6276, 2022.

[39] Han Chen, Hanchen Wang, Hongmei Chen, Ying Zhang, Wenjie Zhang, and Xuemin Lin. Denoising variational graph of graphs auto-encoder for predicting structured entity interactions. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[40] Dorcas Ofori-Boateng, I Segovia Dominguez, C Akcora, Murat Kantarcioglu, and Yulia R Gel. Topological anomaly detection in dynamic multilayer blockchain networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21*, pages 788–804. Springer, 2021.

[41] Dune. ERC and EIP Starter Kit. `https://dune.com/ilemi/erc-and-eip-starter-kit`, 2023. Accessed: June 4th, 2024.

[42] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[43] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] Yu Xie, Chuanyu Yao, Maoguo Gong, Cheng Chen, and A Kai Qin. Graph convolutional networks with multi-level coarsening for graph classification. *Knowledge-Based Systems*, 194:105578, 2020.

[45] Yinhua Piao, Sangseon Lee, Dohoon Lee, and Sun Kim. Sparse structure learning via graph neural networks for inductive document classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11165–11173, 2022.

[46] Junying Li, Deng Cai, and Xiaofei He. Learning graph-level representation for drug discovery. *arXiv preprint arXiv:1709.03741*, 2017.

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

[48] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[49] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

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

[51] Bogdan Dumitrescu, Andra Băltoiu, and Ştefania Budulan. Anomaly detection in graphs of bank transactions for anti money laundering applications. *IEEE Access*, 10:47699–47714, 2022.

[52] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12012–12038, 2021.

[53] Anjan Kr Dasgupta, Usha Sridhar, Panini Dasgupta, Amlan Chakrabarti, et al. Network approaches in anomaly detection for disease conditions. *Biomedical Signal Processing and Control*, 68:102659, 2021.

[54] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pages 1118–1123. IEEE, 2020.

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

[56] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[57] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[58] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[59] Sambaran Bandyopadhyay, Lokesh N, Saley Vishal Vivek, and M Narasimha Murty. Outlier resistant unsupervised deep architectures for attributed network embedding. In *Proceedings of the 13th international conference on web search and data mining*, pages 25–33, 2020.

[60] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 594–602. SIAM, 2019.

[61] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5685–5689. IEEE, 2020.

[62] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6):2378–2392, 2021.

[63] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019.

[64] Kay Liu, Yingtong Dou, Xueying Ding, Xiyang Hu, Ruitong Zhang, Hao Peng, Lichao Sun, and Philip S. Yu. PyGOD: A Python library for graph outlier detection. *Journal of Machine Learning Research*, 25(141):1–9, 2024.

[65] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[66] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[67] Vitalik Buterin et al. Ethereum white paper. *GitHub repository*, 1:22–23, 2013.

[68] Dune. EVM Blockchains analysis . `https://dune.com/KARTOD/blockchains-analysis`, 2024. Accessed: June 11th, 2024.

[69] Pin-Yu Chen, Chun-Chen Tu, Paishun Ting, Ya-Yun Lo, Danai Koutra, and Alfred O Hero. Identifying influential links for event propagation on twitter: a network of networks approach. *IEEE Transactions on Signal and Information Processing over Networks*, 5(1):139–151, 2018.

[70] Dongjie Wang, Zhengzhang Chen, Jingchao Ni, Liang Tong, Zheng Wang, Yanjie Fu, and Haifeng Chen. Interdependent causal networks for root cause localization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5051–5060, 2023.

[71] Mei Liu, Linyu Xu, and Pin-Chao Liao. Character-based hazard warning mechanics: a network of networks approach. *Advanced Engineering Informatics*, 47:101240, 2021.

[72] Chen Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. Fascinate: fast cross-layer dependency inference on multi-layered networks. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 765–774, 2016.

[73] Chen Chen, Jingrui He, Nadya Bliss, and Hanghang Tong. Towards optimal connectivity on multi-layered networks. *IEEE transactions on knowledge and data engineering*, 29(10):2332–2346, 2017.

[74] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Who to follow and why: link prediction with explanations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1266–1275, 2014.

[75] Peng Wang, BaoWen Xu, YuRong Wu, and XiaoYu Zhou. Link prediction in social networks: the state-of-the-art. *arXiv preprint arXiv:1411.5118*, 2014.

[76] Parrot Bamboo Crypto. Fastest evm blockchains: Bsc and polygon, 2024.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See section 1.
   (b) Did you describe the limitations of your work? [Yes] See section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See section 7.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines and ensured that our paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Not including in our work.
   (b) Did you include complete proofs of all theoretical results? [N/A] Not including in our work.
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See abstract.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 5. More details are in the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See section 5. Experiments have been run three times using different seed.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] Not including in our work.
   (b) Did you mention the license of the assets? [Yes] See supplemental material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See supplemental material and GitHub repo in abstract.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All cryptocurrency addresses are anonymous.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] All cryptocurrency addresses are anonymous.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] . All cryptocurrency addresses are anonymous.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] . Not including in our work.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] . Not including in our work.

A Background

In this section, we provide an overview of blockchain technology and cryptocurrency, laying the groundwork for understanding the subsequent discussions in this paper.

Blockchain and Cryptocurrency. Blockchain technology has gained growing attention recently for its strong security features and decentralized structure. It is characterized by a sequence of cryptographically secured blocks that operate on a network of nodes [42]. This design ensures data immutability and verifiability while allowing universal access, enabling participants to interact with the ledger from anywhere at any time. Once recorded on the ledger, transactions become irreversible and are executed securely and transparently, which helps safeguard the integrity of data exchanges.

With the support of blockchain technology, cryptocurrencies have surged in popularity as an innovative means of conducting secure digital transactions. Unlike traditional currencies, cryptocurrencies operate without a centralized authority and are managed through decentralized systems. This decentralization maintains participant anonymity, offering robust privacy protection; however, it complicates efforts to identify fraudulent activities within the market.

Blockchain Models and EVM Chains. Various operational models exist within blockchain technology. For instance, Bitcoin, the first cryptocurrency network, operates using the Unspent Transaction Output (UTXO) model [66]. In this model, each transaction utilizes unspent outputs from previous transactions as inputs, generating new unspent outputs for subsequent transactions. This method preserves transaction integrity by streamlining ownership verification and enhancing security measures related to transaction immutability.

In contrast, the Ethereum Virtual Machine (EVM) introduced an account-based model, akin to traditional banking systems, where balances are maintained in user accounts [67]. This model enables direct value transfer and supports advanced features such as smart contracts, which are self-executing agreements with terms embedded directly within the blockchain. Due to its versatility and strong developer support, the EVM has become the standard for building blockchain networks and decentralized applications. The three notable EVM-based networks discussed in this work are Ethereum, Polygon, and Binance Smart Chain [68]:

- *Ethereum*, the pioneering EVM chain, has developed a robust platform for decentralized applications. It supports a wide range of decentralized services, from financial transactions to games and autonomous organizations. Its native token, Ether, holds the second-largest market capitalization, second only to Bitcoin.
- *Polygon* enhances Ethereum's functionality as an EVM-compatible chain by offering faster transactions and reduced fees. Functioning as a sidechain to Ethereum, it addresses scalability issues with a multi-chain infrastructure, which is particularly advantageous for developers seeking efficient transaction throughput within the Ethereum ecosystem.
- *Binance Smart Chain* provides a similar EVM-compatible environment with a focus on scalability and user experience. It has carved out a niche by emphasizing rapid transactions and minimal fees, particularly attracting decentralized finance (DeFi) applications and NFTs.

ERC20 and BEP20 Standards. The ERC20 standard defines a framework for fungible tokens on the Ethereum blockchain. These fungible tokens are digital assets that are identical in type and value, making them interchangeable with one another. This standardization simplifies the process of trading and exchanging tokens and enhances their interoperability across various applications. Similarly, BEP20 is a standard used on the Binance Smart Chain (BSC), mirroring many of the functionalities of ERC20 while optimizing for faster transactions and lower fees.

Accounts and Transactions. EVM-compatible chains typically support two principal types of accounts: External Owned Accounts (EOAs) and smart contracts. EOAs function much like traditional bank accounts, as they are directly managed by users through a private key, granting them full autonomy over transactions. In contrast, smart contracts are autonomous programs that reside on the blockchain and execute automatically when predefined conditions are met. These programs are crucial for a variety of operations on EVM chains, from facilitating transactions in the token markets to managing decentralized finance (DeFi) protocols and automated governance mechanisms.

A transaction includes various details, such as the sender's and recipient's actions, signature, nonce, data, gas limit, maximum priority fee per gas, and maximum fee per gas. In the token market, these

transactions facilitate diverse blockchain events like token issuance and transfers. This architectural framework not only supports complex financial interactions but also enhances security across the blockchain ecosystem.

B Supplemental Related Work

Graphs-of-Graphs Analysis. The analysis of Graphs-of-Graphs (GoG) systems has become a crucial method for understanding complex relationships within and across different network layers in various domains. For instance, Chen et al. [69] examined the dynamics of event propagation on social platforms like Twitter. They analyzed follower link roles by grouping users based on their language settings, treating these groups as local graphs, with following or retweeting relationships represented as edges. Similarly, Wang et al. [70] modeled intra-level and inter-level causal relationships within interdependent networks, effectively tracing and identifying root causes in complex interconnected system structures. In more specialized applications, Liu et al. [71] employed GoG to enhance hazard identification at construction sites. They mapped interactions between characters and hazard networks, simplifying complex network structures to improve safety outcomes. Additionally, Chen et al. [72, 73] investigated the manipulation of connectivity in multi-layered networks, uncovering the structural dynamics that govern these complex systems. These studies underscore the powerful capability of GoG analysis in providing a deeper understanding of intricate graph systems.

C Basic Structure Properties

In this section, we explore several fundamental graph properties relevant to our analysis, as discussed in subsection 4.1 and subsection 5.2. We measure seven key graph properties: the number of nodes, the number of edges, density, assortativity, reciprocity, clustering coefficient, and effective diameter. These properties provide a comprehensive structural overview of the graph, which is essential for understanding its characteristics and implications in the context of token transfer networks.

First, we consider the number of nodes and edges, which quantitatively describe the scale and potential complexity of the graph. Density, assortativity, and reciprocity offer insights into the connectivity and interaction patterns among nodes, reflecting how edges are distributed and whether similar nodes preferentially connect to each other. Additionally, the clustering coefficient and effective diameter provide a view of the overall compactness and reachability within the graph.

Density. The density of a graph measures its compactness and connectivity. In this study, density is calculated as:

$$D = \frac{|E|}{|V|(|V|-1)}$$

where $|E|$ is the number of edges, and $|V|$ is the number of nodes. In token transfer graphs, a lower density suggests a fragmented or developing market, indicative of fewer interactions or participants. Conversely, a high density indicates a mature market with frequent transactions between participants. This distinction is crucial for understanding market dynamics.

Assortativity. The assortativity coefficient quantifies the tendency of nodes to connect with others that share similar attributes. Specifically, assortativity is calculated by:

$$r = \frac{\sum_{(i,j)\in E} (f(i) - f_1)(f(j) - f_2)}{\sqrt{\sum_{(i,j)\in E} (f(i) - f_1)^2 \sum_{(i,j)\in E} (f(j) - f_2)^2}}$$

This metric is particularly relevant in token transfer graphs, as it measures how frequently addresses transact with others of similar characteristics. A higher assortativity may indicate a market dominated by similar types of transactions or participants. However, it is important to note that this is a trend observed in our data rather than an absolute rule. Understanding this property aids in identifying market segmentation.

Reciprocity. Reciprocity measures the likelihood of directed connections being reciprocated. It is calculated by:

$$\rho = \frac{|\{(i,j) \in G : (j,i) \in G\}|}{|E(G)|}$$

This metric is crucial for understanding mutual interactions between addresses, such as reciprocal trading patterns. In token transfer graphs, a higher reciprocity suggests a strong bidirectional transactional relationship, indicating trust or partnership between nodes. This insight is vital for assessing the stability of relationships within the graph.

Clustering Coefficient. The clustering coefficient measures how closely nodes in a graph tend to cluster together. This metric is essential in token transfer graphs, as it indicates the extent to which nodes form tightly-knit groups, which may suggest collusive behavior or strong community structures. We primarily use the average clustering coefficient to assess overall network cohesion and the potential for collaborative behavior among participants. It is calculated as:

$$C_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} C_i$$

$$C_i = \frac{2T(i)}{k_i(k_i - 1)}$$

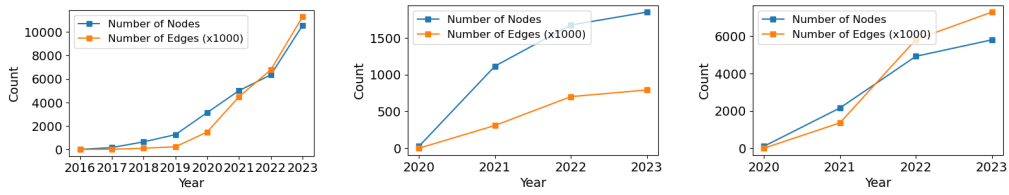
In the token transfer graph, a higher average clustering coefficient suggests a network characterized by prevalent cliques or groups that engage in frequent interactions, potentially indicating tight-knit trading communities.

Effective Diameter. The effective diameter provides insight into the average separation between node pairs across the graph. We measure the effective diameter by performing breadth-first search (BFS) from a sample of randomly selected nodes to provide a broad and representative overview of the graph's structure. The effective diameter is then defined as the 90th percentile of the shortest path lengths obtained from these BFS runs. This approach estimates how far apart nodes are on average, considering the most representative paths rather than extremes. The effective diameter reflects how easily a token can circulate within the network, a key factor in assessing liquidity and market efficiency. This metric is particularly important for understanding the graph's accessibility.

D Temporal Properties Analysis

To reveal the temporal changes in the GoG systems of the three blockchains, we analyze the yearly variation of some fundamental properties of the global graphs. Nodes represent tokens, and an edge between two nodes indicates that the tokens share common addresses during that year.

First, we examine the dynamics of the number of nodes and edges, as illustrated in Figure 6. Across the Ethereum, Polygon, and BSC ERC20 token networks, we observe a consistent trend of significant growth in both nodes and edges. This growth reflects increased adoption and diversification of blockchain platforms. Over the past three years, the average increase in the number of nodes in the global graphs is 42.49% for Ethereum, 33.08% for Polygon, and 65.18% for BSC. These figures indicate substantial changes in the dataset. Notably, Ethereum exhibits the most mature growth pattern, particularly with a significant acceleration since 2020. In contrast, Polygon shows robust growth; however, it has a slower increase in edges compared to nodes, suggesting a less interconnected network than Ethereum's GoG. Meanwhile, BSC experiences a rapid rise in both nodes and edges but begins to show signs of stabilization in 2023, indicating a maturing of its initial expansion phase. These patterns highlight that while all networks are expanding, the nature and rate of growth vary among the different blockchains.

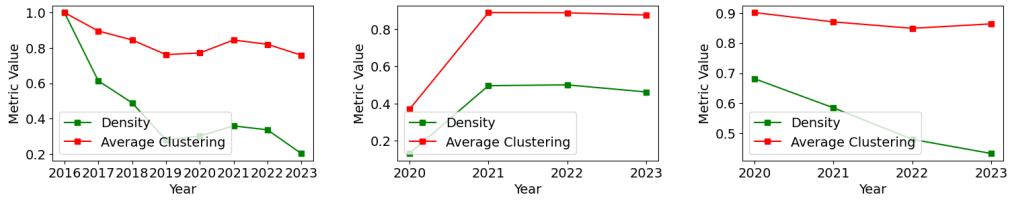


(a) Ethereum (b) Polygon (c) BSC

Figure 6: Yearly number of nodes and edges of three global graphs.

Second, we analyze the density and average clustering coefficient of the three global graphs, as shown in Figure 7. A common trend emerges across Ethereum and BSC: both density and clustering

coefficient tend to decrease as the network size increases. This trend indicates sparser connections as these networks expand, especially pronounced in the BSC network, which reflects significant diffusion from its originally dense structure. Conversely, Polygon exhibits a different pattern; both metrics initially increase and then stabilize. This indicates that the GoG not only grows but also effectively maintains or enhances its clustering. Such behavior suggests robust internal structuring that preserves community integrity even as the network scales. These observations highlight varied adaptive strategies within blockchain networks, with the Polygon GoG notably sustaining community cohesion amidst growth.



(a) Ethereum (b) Polygon (c) BSC

Figure 7: Yearly density and clustering coefficients of three global graphs.

E Model Implementation Details

In this section, we introduce the models and hyperparameters we used for the multi-class classification and anomaly detection tasks.

E.1 Multi-class classification

Models. We conduct experiments on two groups of models: (1) 5 GNN models on individual graphs, and (2) 3 GoG-based GNN models on graphs-of-graphs.

Group (1) includes GNN models for individual graphs:

- *Graph Convolutional Network (GCN)* [31] utilizes a layer-wise propagation rule based on spectral graph convolutions, enabling it to learn representations that capture graph structure and node features effectively.
- *Graph Attention Network (GAT)* [47] introduces an attention mechanism in the propagation step, allowing nodes to dynamically weigh the contributions of their neighbors.
- *Graph Isomorphism Network (GIN)* [48] is designed to capture the power of the Weisfeiler-Lehman graph isomorphism test. It approximates the Weisfeiler-Lehman graph isomorphism test by adjusting aggregators to better distinguish between different graph structures.
- *Residual Graph Convolutional Network (ResidualGCN)* [49] incorporates residual connections into the graph convolutional layers to improve gradient flow during training, which enhances the learning of deeper GNN architectures by mitigating the vanishing gradient problem.
- *GraphSAGE* [50] generates node embeddings by sampling and aggregating features from a node's local neighborhood. Its inductive learning framework supports embedding generation for unseen data, making it scalable and efficient for large graphs.

Group (2) includes models designed for graphs-of-graphs:

- *Semi-supervised Graph Classification via Cautious Iteration (SEAL)* [20] utilizes a self-attentive graph embedding method with GCN as a backbone to embed graph instances into fixed-length vectors, facilitating graph-based classification tasks. It enhances the encoding of local graph structures and their relationships within a larger graph context.
- *Graph of Graphs Neural Network (GoGNN)* [19] extends traditional GCN capabilities by integrating an attention-based pooling mechanism and GAT. It effectively identifies significant substructures within local graphs and interactions within the interaction graph, providing a powerful framework for analyzing complex graph relationships.

- *Denoising Variational Graph Autoencoder (DVGGA)* [39] employs a denoising variational autoencoder combined with a self-attentive graph neural network and a readout operation. This model is adept at handling noise in graph data, making it suitable for tasks requiring robust feature extraction and anomaly detection in noisy environments.

Model structures. For GNN models targeting individual graphs, we employ a configuration that includes two GNN layers followed by a fully connected layer for classification. This two-layer setup, consistent with the backbone design of SEAL [20], ensures fair comparisons. Each layer transforms node features to enhance feature extraction, using ReLU activation and dropout for regularization. Following the convolution layers, a global mean pooling layer aggregates node features into a cohesive graph-level representation. This representation is then processed through a fully connected layer, which outputs class probabilities using a logarithmic softmax function. For GoG models, we utilize publicly available code from the Github repositories of the original studies. For GoGNN and DVGGA, we adapt the original code from edge prediction to node classification tasks on the global graph.

Hyperparameters. For individual GNN models, we configure each layer with a dimension of 16, a dropout rate of 0, a learning rate of 0.01, and set the number of training epochs to 50. Cross-entropy serves as the loss function. For GoG-based models using a single GCN model as the backbone, we ensure that the dimensions and dropout rates are consistent with those of the individual GNN models. To fine-tune additional hyperparameters, we experiment with various settings listed in Table 6 to achieve optimal performance.

Table 6: GoG models parameter settings.

Model	Parameter	Values
SEAL	First dense neurons	16, 32, 64
	Second gcn dimensions	8, 16
	Number of epochs	50, 100, 150
	Weight	0, 0.001, 0.00001
GOGNN	Nhid	32, 64, 128
	Number of epochs	50, 100, 150
	Pooling rate	0.4, 0.5, 0.6
DVGGA	Vgae hidden dimensions	8, 16, 32
	Number of epochs	50, 100, 150

E.2 Anomaly Detection

Models. We test two groups of models: (1) 4 models for multivariate anomaly detection, and (2) 5 models for the graph anomaly detection.

Group (1) includes probabilistic-based and outlier ensembles methods designed for multivariate anomaly detection:

- *Copula-Based Outlier Detection (COPOD)* [54] is a probabilistic model that leverages the advantages of copulas for outlier detection. It does not assume a normal distribution of data, making it robust and effective in identifying outliers in various datasets with complex distributions.
- *Isolation Forest (IForest)* [55] utilizes a decision tree structure to isolate outliers by randomly selecting features and split values between the feature's maximum and minimum. Its efficiency and scalability make it well-suited for large datasets.
- *Deep Isolation Forest (DIF)* [56] extends the traditional isolation forest by incorporating deep learning techniques to enhance its capability to handle high-dimensional and complex structured data.
- *Variational Autoencoder (VAE)* [57] is a generative model that uses a neural network architecture to model data distributions and encode data into a latent space. It is widely used for anomaly detection by reconstructing inputs and measuring reconstruction errors to identify anomalies.

Group (2) includes anomaly detection methods on graphs, primarily utilizing GNN combined with Autoencoder techniques:

- *Graph Autoencoder (GAE)* [58] employs a graph convolutional network to encode the graph structure into a latent space, then reconstructs the graph to identify anomalies by measuring reconstruction loss.
- *Detection of Outliers in Network Data (DONE)* [59] integrates graph structural information with node feature information to detect anomalous nodes effectively within graph data.
- *Deep Anomaly Detection on Attributed Networks (DOMINANT)* [60] uses a deep autoencoder model adapted to graph data, enhancing the ability to capture non-linearities and complex patterns in the data, which helps in identifying both global and local anomalies in graphs.
- *Anomaly Detection with Autoencoder (AnomalyDAE)* [61] is an autoencoder-based model that particularly focuses on detecting anomalies in dynamic graphs by learning a representation that captures both the graph structure and changes over time.
- *Contrastive Learning for Anomaly Detection (CoLA)* [62] utilizes contrastive learning to differentiate between normal and abnormal nodes, leveraging the discriminative power of contrastive loss to enhance anomaly detection performance in graph settings.

Hyperparameters. We test on the following hyperparameters in Table 7 and select the best setting with superior performance.

Table 7: Models of anomaly detection parameter settings. n represents the number of features.

Model	**Parameter**	**Values**
COPOD	Contamination	0.01 to 0.1 (linear space)
Isolation Forest	Number of estimators	100, 200
	Maximum samples	256, 512
DIF	Contamination	0.01 to 0.05 (linear space)
VAE	Encoder neurons	$n/4$, $n/2$, min(20, n)
	Decoder neurons	$n/4$, $n/2$, min(20, n)
	Contamination	0.1 to 0.3 (linear space)
DOMINANT, DONE, GAE, AnomalyDAE, CoLA	Hidden dimensions	16, 32, 64
	Learning rate	0.01, 0.005, 0.1
	Number of epochs	50, 100, 150

F Global Link Prediction

Link prediction is an essential task in graph learning, widely applied in recommendation systems [74] and social media analysis [75]. In the context of blockchain analysis, predicting interactions between tokens is essential for forecasting future market behaviors. This section focuses on global edge prediction, specifically aiming to forecast interactions for newly launched tokens using information from existing tokens.

Models. We compare two groups of models based on the previous section subsection 5.1. The first group consists of traditional Graph Neural Network (GNN) models applied to global token graphs. The second group includes Graphs of Graphs (GoG) models, which leverage the hierarchical structure of token-to-token interactions. We provide a detailed comparison of performance metrics to substantiate our claims regarding the effectiveness of these models.

Settings. Our analysis focuses on the most recent tokens launched within the past year. We divided global token-token interactions into training and test sets, following an 80/20 ratio based on the tokens' launch times. Node degree serves as the primary feature for local graph embeddings, consistent with our approach in the classification task. We evaluate model performance using accuracy and AUC, supplemented by precision and recall to provide a comprehensive assessment.

Results. The performance of global edge prediction methods across three blockchains is summarized in Table 8. As shown, GoG models do not consistently outperform individual GNN models, particularly on the BSC dataset. One potential reason for these results is that the node degree, used as a node feature in this experiment, may not be as effective for predicting global edges as it is for

Table 8: Edge prediction performance by blockchain.

Model	Ethereum		Polygon		BSC	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
GCN	58.07±0.36	62.02±0.23	59.64±1.71	66.92±5.37	66.73±3.12	72.87±3.42
GAT	50.80±0.43	54.50±2.43	50.70±2.07	54.64±4.47	52.82±0.77	53.62±2.86
GIN	56.48±1.61	56.36±1.77	59.03±3.47	58.17±4.33	59.98±2.61	63.57±3.48
ResidualGCN	50.31±0.37	50.66±0.54	49.91±0.08	49.92±0.10	50.41±0.43	50.74±0.94
GraphSage	50.92±1.03	53.67±2.11	56.63±8.88	60.17±12.83	**71.02** ± 0.05	**78.07** ± 1.08
SEAL	57.09±1.64	64.74±4.83	56.98±4.93	64.62±10.34	56.52±4.62	58.05±6.04
GoGNN	**66.94** ± 2.08	**72.04** ± 2.41	57.10±5.21	56.72±4.75	58.99±2.77	66.25±1.84
DVGGA	50.40±1.79	62.93±1.73	**72.38** ± 1.36	**76.00** ± 0.32	63.63±4.94	69.11±3.95

classification tasks. This suggests that further exploration of edge feature engineering could enhance the predictive capabilities of GoG models for token-token interactions.

Additionally, the dynamic nature of blockchain networks presents opportunities to monitor and predict future token-token interactions, which could forecast significant market trends. However, most current GoG models are not designed with dynamic algorithms [19, 20], highlighting both challenges and potential areas for further research. We recommend future work to explore the integration of dynamic features and more sophisticated edge feature engineering to improve prediction accuracy. In summary, our findings indicate that while GoG models show promise, there is a need for further refinement and exploration of features to enhance their predictive performance in the context of blockchain networks.

G Multi-Class Graph Classification - Temporal Split

In this section, we present additional experiments that focus on predicting the class label of younger tokens using information derived from older tokens. To simulate a realistic scenario where future tokens are classified based on historical data, we implement a temporal split of the dataset. Specifically, we divide the tokens into training and test sets following an 80/20 ratio based on their first transaction timestamps. This approach enables evaluation of the model's performance within a time-sensitive context, which is crucial for applications in dynamic environments like blockchain.

The experimental settings align with those described in subsection 5.1. The results of these experiments are summarized in Table 9, which provides a comparative analysis of classification performance across different models and blockchain platforms.

Upon comparing these results with those presented in Table 4, we observe that the performance for Ethereum and BNB shows only slight differences regardless of the node-splitting method employed. However, for Polygon, we note a significant deterioration in performance. This discrepancy may be due to Polygon's status as the fastest of the major Ethereum-based chains [76], leading to varying transaction patterns across different time periods. These findings suggest that while our methods demonstrate competitive performance, further investigation is warranted to understand the underlying factors affecting classification accuracy across different blockchains.

H Graph Anomaly Detection with Deepwalk Embeddings

In this section, we present an effective method for representing token graphs in anomaly detection tasks by employing the DeepWalk algorithm [65]. DeepWalk is well-known for generating robust graph embeddings through the simulation of random walks. This approach captures the network topology and provides a nuanced representation of graph structures.

We configured DeepWalk with a walk length of 20 and performed 40 walks per node on each token transaction graph. This configuration strikes a balance between the depth and breadth of neighborhood exploration, ensuring that the embeddings accurately capture the structural and contextual nuances of the token graphs. We then aggregated these node embeddings into a unified graph-level representation by computing their mean, resulting in an embedding of 32 dimensions for each graph.

Table 9: 3-class and 5-class classification performance by blockchain (node split by time).

Model	Ethereum		Polygon		BSC	
	F1-macro	F1-micro	F1-macro	F1-micro	F1-macro	F1-micro
3-Class Classification						
GCN	60.16±5.60	87.70±0.83	22.37±0.57	48.22±0.67	50.01±5.27	57.39±3.78
GAT	57.50±6.25	87.33±1.16	26.00±2.67	48.91±1.02	51.15±6.52	59.48±5.58
GIN	60.38±5.76	87.68±0.94	21.74±1.21	48.03±0.63	42.56±2.73	56.59±3.65
ResidualGCN	40.62±8.06	83.83±1.41	22.86±1.02	48.24±0.55	48.09±5.30	60.13±2.95
GraphSage	61.71±6.27	88.25±0.97	24.91±1.87	48.72±0.55	53.86±6.99	62.16±4.28
SEAL	**67.42** ± 1.05	**88.72** ± 0.33	27.20±1.81	**49.37** ± 0.59	55.14±5.62	**64.03** ± 3.82
GoGNN	66.10±1.98	88.28±0.80	**30.85** ± 2.32	44.75±4.09	**61.80** ± 0.50	62.17±0.33
DVGGA	53.80±1.98	75.60±7.67	28.22±1.44	41.52±0.98	24.03±13.78	35.37±15.33
5-Class Classification						
GCN	38.75±5.44	85.18±0.93	12.11±0.53	41.16±0.95	26.76±3.74	47.21±4.33
GAT	37.02±5.64	85.24±1.07	**16.63** ± 3.04	42.10±2.27	28.43±4.08	49.37±5.87
GIN	22.69±1.43	80.65±0.52	12.15±0.77	41.06±0.76	22.02±2.97	43.48±5.83
ResidualGCN	41.19±5.45	85.00±1.13	12.03±0.58	41.15±0.70	24.38±4.34	47.78±6.34
GraphSage	40.51±5.82	86.31±1.10	14.97±1.69	41.98±0.69	27.89±5.48	49.06±6.83
SEAL	**48.85** ± 0.52	86.29±0.27	15.54±2.32	**42.41** ± 0.15	**30.41** ± 1.81	**52.65** ± 1.09
GoGNN	45.25±5.83	**86.36** ± 0.76	14.49±1.94	41.77±0.60	28.29±3.51	52.11±2.66
DVGGA	25.35±4.28	68.96±16.54	11.65±0.01	41.03±0.00	10.91±2.72	31.36±4.46

Table 10: Graph anomaly detection performance using DeepWalk. We report the ratio of number of non-fraud:fraud case of each data at the top.

Model	Ethereum (8387: 6022)		Polygon (2257: 58)		BNB (6339: 1042)	
	AUC	AP	AUC	AP	AUC	AP
COPOD	50.87±0.09	42.57±0.70	62.16±8.3	3.42±1.62	52.47±0.47	13.82±0.91
IForest	50.43±0.28	42.69±1.07	60.95±8.7	3.11±1.12	52.14±1.17	14.02±0.85
DIF	50.58±0.31	42.10±0.89	59.72±6.85	2.80±0.55	52.16±0.72	13.77±0.91
VAE	50.77±0.34	42.87±0.94	61.86±7.98	3.47±1.69	51.69±1.24	14.00±0.81
GAE	51.22±1.39	41.44±0.56	60.81±1.25	**5.40** ± 2.42	61.15±2.67	**24.02** ± 1.92
DONE	**68.86** ± 10.27	32.89±5.57	**71.29** ± 2.21	1.63±0.06	77.55±0.13	8.62±0.01
DOMINANT	60.92±4.57	38.12±2.63	67.15±3.41	2.43±1.00	**79.73** ± 0.07	8.42±0.01
AnomalyDAE	65.14±3.63	**46.12** ± 10.08	57.90±3.58	3.44±0.31	52.75±0.98	15.67±0.20
CoLA	50.51±0.42	41.90±0.44	59.61±3.94	2.67±0.77	54.87±0.03	14.88±0.56

The results of our anomaly detection analysis using the DeepWalk algorithm are presented in Table 10. Notably, the GoG models generally outperform multivariate outlier detection methods in our experiments, although this may vary depending on the specific characteristics of each dataset. When comparing the results in Table 7, the superiority of GoG models is evident across all three blockchains when using the DeepWalk algorithm, particularly in scenarios with high fraud rates.

It is important to note that the anomaly detection performance on Polygon remains the poorest among the chains, consistent with previous findings in subsection 5.2. While GoG models benefit from the use of the DeepWalk algorithm, the performance of multivariate outlier detection methods appears to decrease. This suggests that the DeepWalk algorithm significantly enhances the effectiveness of GoG models in identifying anomalies.

I Details of Compute Resources

We use two machine, one for experiements of inidividual GNN, one for experiements of GoG-based GNN. First, all experiments involving individual GNN models were conducted on machine outfitted

with eight NVIDIA GeForce GPUs, each with a maximum power capacity of 350W and 24,576 MiB of available memory. Second, all experiments utilizing GoG-based GNN models were carried out on the machine equipped with eight NVIDIA A100-SXM4-80GB GPUs. These GPUs, each with a maximum power capacity of 400W and a substantial 81,920 MiB of memory, are specifically chosen for their high performance and large memory capacity, which are ideal for the complex and memory-intensive computations required by GoG-based GNN models.

J License

The dataset is released under the Creative Commons Attribution-NonCommercial-ShareAlike (CC BY-NC-SA) license.

K Hosting Plan

We choose GitHub as our hosting platform for both code and data due to its ease of use, cost-effectiveness, and scalability. Ensuring easy access to our data is crucial. To facilitate straightforward and reliable data retrieval, we will maintain a curated interface. We are committed to keeping our platform stable and functional, with regular updates and maintenance to ensure our repository remains up-to-date, bug-free, and efficient.

Our project is driven by a commitment to open access. By regularly updating our GitHub repository, we ensure that users have timely access to the latest data. We believe that GitHub's user-friendly environment will provide a dependable and efficient solution for sharing our data with the global community.