Hierarchical Network With Local-Global Awareness for Ethereum Account De-anonymization

Jiahui Huang, Teng Huang, Changyu Dong, Sisi Duan, and Yan Pang

***Abstract*—The expansion of blockchain applications, particularly on platforms like Ethereum, brings escalating security challenges as account anonymity provides breeding grounds for criminals to commit crimes and cause significant economic losses. As the mainstream architecture of de-anonymization technology, graph neural networks (GNNs) provide empirical tools for law enforcement agencies to investigate illegal activities. However, the limited expressiveness of current GNNs leads to performance degradation for Ethereum account de-anonymization. To address this challenge, we propose an innovative Local-Global Awareness (LGA) framework, which consists of a Local Structure-Aware (LSA) module and a Global Information-Aware (GIA) module. LSA integrates subgraph-level encoding strategies with local attention to enhance the capture of microscopic interactions. As a complementary measure, GIA introduces global attention to facilitate the understanding of macroscopic information. The LGA framework meticulously captures subgraph-level account behavior patterns at a granular level while simultaneously incorporating global contextual insights, demonstrating higher-level expressive power and receptive fields over conventional GNN. The efficacy of the LGA framework is corroborated by experimental evaluations conducted on the lw-AIG dataset. Our framework achieves exceptional performance, significantly outstripping state-of-the-art GNN-based methods in terms of the micro F1 score metric, with relative improvements ranging from 0.14% to 6.63%. Through its detailed and comprehensive analysis of account interactions, the LGA framework aims to provide a potent solution to the complex security challenges faced in the expanding blockchain landscape. The code for LGA is available at https://github.com/deepang-ai/LGA.**

***Index Terms*—Attention mechanism, blockchain, de-anonymization, Ethereum, subgraph.**

I. Introduction

As ONE of the essential supporting technologies in the digital economy era, blockchain technology provides an efficient, stable, and secure system architecture for the development of the digital economy. It is expected to be utilized in digital finance [1], supply chain management [2], identity verification [3], and other fields [4], [5]. In recent years, the scale of investment in the industry has shown a continuous growth trend. According to data from Precedence Research,[1] a leading international strategic market insight organization, the global blockchain technology market is projected to reach USD 9.31 billion in 2024 and is expected to grow at a compound annual growth rate of 4.63% from 2024 to 2034. The continuously growing market highlights the key role of blockchain as a "trusted network" and "trusted machine" to support the development of emerging digital ecosystems.

Escalating risks accompany the disorderly expansion of the blockchain market. In particular, Ethereum network attackers are becoming more sophisticated, deftly manipulating various malicious incidents. Typical incidents, such as phishing scam [6], money laundering [7], and cybercrime [8] are rampant on the Ethereum network, causing massive economic damage. According to data provided by Scam Sniffer,[2] in the first half of 2024, 266 713 victims suffered losses of $314 million on Ethereum. The driving factor behind this deterioration is the pseudonymity of Ethereum accounts, which provides a natural shelter for criminals. Ethereum accounts adopt the last 20 bytes of the hash of the public key as a pseudonymous address, a virtual identity that has no explicit association with a real identity [9]. Pseudonymous addresses are designed to protect user privacy, but the unbridled use of pseudonymous identities facilitates evasion of regulation.

In order to crack down on cybercrime that exploits anonymous identities, various initiatives have been proposed to deanonymize Ethereum accounts to date. These initiatives record behavior patterns of anonymous accounts as empirical evidence and label them into meaningful account categories, such as initial coin offering (ICO) wallets, exchanges, miners, phishers, hackers, etc. Existing de-anonymization methods generally fall into two categories. One includes conventional machine learning (ML)-based methods [10], [11], [12], [13], [14], [15], [16], implemented by training a model that learns the relationship between features and account categories from training data and maps input account features to output account categories in the

Received 11 March 2025; revised 20 April 2025; accepted 14 May 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB2704300; in part by the Guangdong Natural Science Foundation under Grant 2025A1515010276; and in part by the Science and Technology Projects in Guangzhou 2024 Guangzhou School (Institute) Enterprise Joint Funding Project under Grant 2024A03J0166. This article was recommended by Associate Editor D. O. Olson. *(Corresponding authors: Teng Huang; Yan Pang.)*

Jiahui Huang, Teng Huang, and Changyu Dong are with the School of Artificial Intelligence, Guangzhou University, Guangzhou 511370, China (e-mail: jiahuihuang@e.gzhu.edu.cn; huangteng1220@gzhu.edu.cn; changyu.dong@gzhu.edu.cn; yanpang@gzhu.edu.cn).

Sisi Duan is with the Institute for Advanced Study, Tsinghua University, Beijing 100084, China (e-mail: duansisi@tsinghua.edu.cn).

Yan Pang is with the Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China (e-mail: yanpang@siat.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2025.3571795.

Digital Object Identifier 10.1109/TSMC.2025.3571795

[1] https://www.precedenceresearch.com/rigid-bulk-packaging-market

[2] https://x.com/realScamSniffer/status/1809089851859611908

inference phase. However, ML-based methods exhibit two key limitations: 1) they necessitate laborious manual feature design contingent upon specific domain expertise, and 2) they ignore to incorporate the intricate relationships between entities and the rules that govern their interactions within the network.

Addressing the limitations inherent in ML-based approaches, graph neural network (GNN)-based methods present a promising alternative for de-anonymization tasks, leveraging their inherent sophistication, generalizability, and flexibility [17], [18]. These approaches [19], [20], [21], [22], [23], [24] leverage the topological features between entities, enabling models to achieve combinatorial generalization and facilitate flexible relational reasoning. By computing structured representations, GNN-based approaches excel in capturing a richer tapestry of information compared to conventional ML-based approaches. This inherent advantage translates to superior performance in the challenging task of de-anonymizing Ethereum accounts. However, a critical examination of current GNN-based implementations reveals a potential limitation: their constrained expressiveness and receptive fields may not adequately capture the intricate behavioral patterns and feature distributions inherent to Ethereum network account de-anonymization tasks.

1) *GNNs With Limited Expressiveness:* GNNs based on the Weisfeiler–Lehman (1-WL) [25] algorithm expose limitations in the Ethereum account de-anonymization task. The 1-WL algorithm iteratively aggregates the features of adjacent nodes to the central node and encodes the root subtree around the central node as its final representation. The inherent limitations of the tree structure make it fail to represent arbitrary subgraphs, especially complex subgraphs containing cycles. This defect allows malicious transaction participants to take advantage of special transaction channels to evade detection, as shown in Fig. 1(a). In the Ethereum peer-to-peer network, transactions are allowed to be initiated between any participants, which self-evolves into a large number of cyclic structures. Unfortunately, the limited expressiveness of GNNs leaves these potential transactions unregulated.
2) *GNNs With Bounded Receptive Fields:* The design paradigm of GNNs follows 1-hop message passing, where node representations are updated by the node's direct neighbors, as shown in Fig. 1(a). This design implicitly assumes homophily in the graph, where neighboring nodes belong to the same class as the target node. However, as a ubiquitous property in the Ethereum network, heterophily [26], [27] significantly limits the performance of tailor-made GNNs with an implicit homophily assumption. Specifically, the "opposites attract" phenomenon is observed in heterophilic networks [28], [29], [30], [31], where nodes of the same class are distributed over long-term distances beyond 1-hop. GNNs with bounded receptive fields pose significant challenges to learning and identifying nodes with semantic similarity on heterophilic graphs.

In response to these limitations of existing GNNs in Ethereum account de-anonymization, we propose the Local-Global Awareness (LGA) framework. This end-to-end network

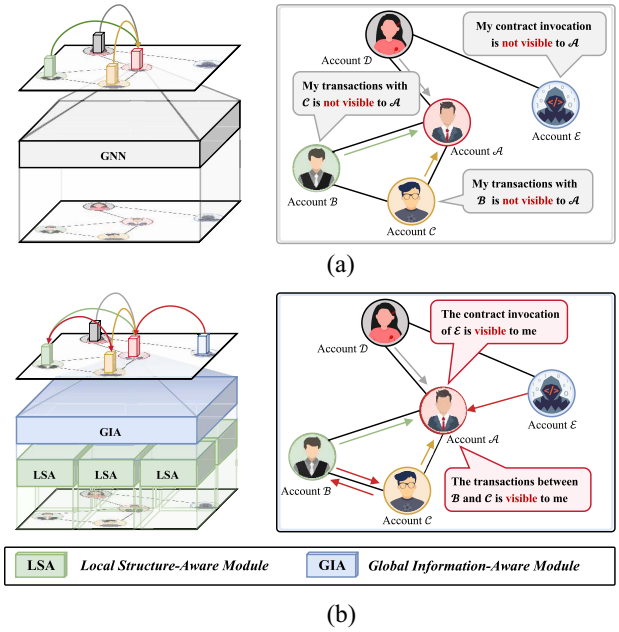

Fig. 1. Comparison of our proposed LGA framework with existing GNN frameworks for Ethereum account de-anonymization. (a) Existing GNN methods encode only a 1-hop subtree around the target node, which limits their expressiveness and fails to capture transactions between neighboring nodes or information from nodes multiple hops away. (b) In contrast, LGA effectively identifies potential transactions between neighboring nodes and expands the receptive field to consider the entire graph, enabling the aggregation of information from all nodes to the target node.

architecture offers a comprehensive solution by characterizing and analyzing account behavior patterns at both micro and macro levels, as depicted in Fig. 1(b). The core components of LGA include the Local Structure-Aware (LSA) module and the Global Information-Aware (GIA) module. The LSA module adopts a local self-attention mechanism to perform message passing under the constraints of structured subgraphs, accurately capturing various complex transaction patterns. The GIA module uses a global self-attention mechanism to learn long-range pairwise dependencies between nodes, effectively widening the receptive field to exchange equivalent information. This dual-module design works synergistically, with LSA providing differentiated structural representations for diverse structural patterns from a local perspective, combined with GIA to discriminate the similarity of structures and attributes from a global perspective. Compared with the popular GNNs design paradigm, the subgraph encoding strategy integrated by LSA supports encoding of arbitrarily complex transaction patterns, and the global attention mechanism of GIA allows ignoring the assumption of strong heterophily. Extensive evaluation on lightweight account interaction graph (lw-AIG) and Elliptic datasets confirms that LGA achieves state-of-the-art performance in account de-anonymization. The advanced achievements of LGA reflect its capabilities in the task of account de-anonymization, particularly in capturing complex transaction patterns to prevent criminals from evading detection through specific transaction linkages. Coupled with global information awareness for comprehensive evidence collection, this all-encompassing perception capability provides

TABLE I
COMPARISON OF GRAPH-BASED LEARNING MODELS FOR ACCOUNT DE-ANONYMIZATION

Type	Method	Key approach	Structure awareness	Feature awareness	Attention operation	Subgraph awareness	Heterophilic graph modeling
RW	DeepWalk [32]	Unbiased random walk strategy.	✓	-	-	-	-
	Struc2Vec [33]	Hierarchical measurement of structural similarity.	✓	-	-	-	-
	Trans2Vec [34]	Importance-based walking strategy	✓	✓	-	-	✓
	Node2Vec [35]	Parameterized walking strategy.	✓	-	-	-	-
	Role2Vec [36]	Attributed random walking strategy.	✓	-	-	-	-
GNN	GIN [37]	Aggregation function over the multiset.	✓	✓	-	-	-
	GCN [38]	Feature aggregation based on convolution.	✓	✓	-	-	-
	GAT [39]	Local self-attention mechanism.	✓	✓	✓	-	-
	TTAGN [22]	Temporal transaction relationship modeling.	✓	✓	✓	-	-
	I^2BGNN [20]	Local sampling mechanism.	✓	✓	-	-	-
	Ethident [21]	Hierarchical graph attention encoder.	✓	✓	✓	-	-
	AEtransGAT [40]	Local transaction importance calculation.	✓	✓	✓	-	-
	ETGNN [41]	Attention-based edge feature modeling.	✓	✓	✓	-	-
	LGA	Subgraph encoding and long-range modeling.	✓	✓	✓	✓	✓

valuable insights for enhancing the security governance of the blockchain transaction ecosystem.

In summary, our work makes the following contributions.

1) This article conducts the first in-depth analysis and study of complex transaction patterns and long-range dependencies for the Ethereum account interaction graph characterized by heterophily. A novel LGA framework LGA is designed to capture local transaction patterns and aggregate global information.
2) The LGA framework comprises two core modules: the Local Structure Aware module, which encodes subgraph-level structural information for precise transaction pattern representation, and the Global Information Aware module, which models long-range dependencies to capture account similarities and facilitate information exchange on heterophilic graphs.
3) Extensive experiments conducted on the lw-AIG and Elliptic datasets confirm the effectiveness of our proposed framework. Our findings indicate that LGA significantly outperforms state-of-the-art methods, achieving a relative performance improvement in micro F1 score ranging from 0.14% to 6.63%.

II. RELATED WORK

In this section, we briefly review the research on account de-anonymization based on graph learning, which can be categorized into two categories: 1) random walk (RW)-based graph analytics and 2) GNN-based graph analytics, as shown in Table I.

A. RW-Based Graph Analytics

DeepWalk [32] is a graph mining algorithm that employs RWs and Word2Vec [42] to generate low-dimensional vector representations of network nodes. These embeddings serve as input for subsequent tasks, such as account de-anonymization. Building upon the foundations of DeepWalk and Node2Vec [35], Trans2Vec [34] proposes a refined network embedding technique. By incorporating transaction timestamps and amounts into its RW process, Trans2Vec captures the temporal and monetary attributes of account relationships, thereby enriching the resulting node embeddings. Combined with a support vector machine (SVM), it distinguishes between normal nodes and phishing nodes. Inspired by Trans2Vec, other studies such as [19] and [43] aim to capture more comprehensive attributes of dynamic transaction networks. They propose modeling the Ethereum transaction network as a time-weighted, directed multigraph and developing various time-walking strategies to learn richer node representations for this large-scale network. Béres et al. [44] conducted quantitative comparisons of the latest DeepWalk-based algorithms on real datasets from sources like Etherscan and Tornado. Role2Vec [36] is statistically identified as the most effective method. DeepWalk-based methods demonstrate strong expressive capabilities and robust performance in node de-anonymization tasks. However, RW-based methods have limited generalizability to unseen nodes and do not easily incorporate node or edge features. To address these issues, GNN-based methods offer generalizability to unseen nodes, consider node or edge attributes, and provide task-specific training and training based on node similarity.

B. GNN-Based Graph Analytics

The massive amount of transaction data in Ethereum can be modeled as an account interaction graph, where nodes represent accounts and edges represent transactions. GNNs are used to explore the relational inductive bias in the encoding of graph connectivity, allowing this rich information to be used to discover the true identity of anonymous accounts. In typical GNN-based de-anonymization studies [19], [20], [21], [22], [23], [24], [40], [41], end-to-end graph convolutional network (GCN) models [38] map transaction patterns to account readl identity. Addressing the lack of temporal information in transactions, TTAGN [22] models the temporal relationships of historical transactions between nodes, and GCN is used to integrate the features of edges around nodes into the node representations. To address the computational overhead of training GCNs on the entire transaction graph, I^2BGNN [20] introduces a sampling mechanism to constrain the size of the graph, and GCN is combined to learn representations of the sampled graph to infer blockchain account identities. However, methods based on GCN models are limited in learning the correlations between interacting accounts. To address this issue, Ethident [21], AEtransGAT [40], and

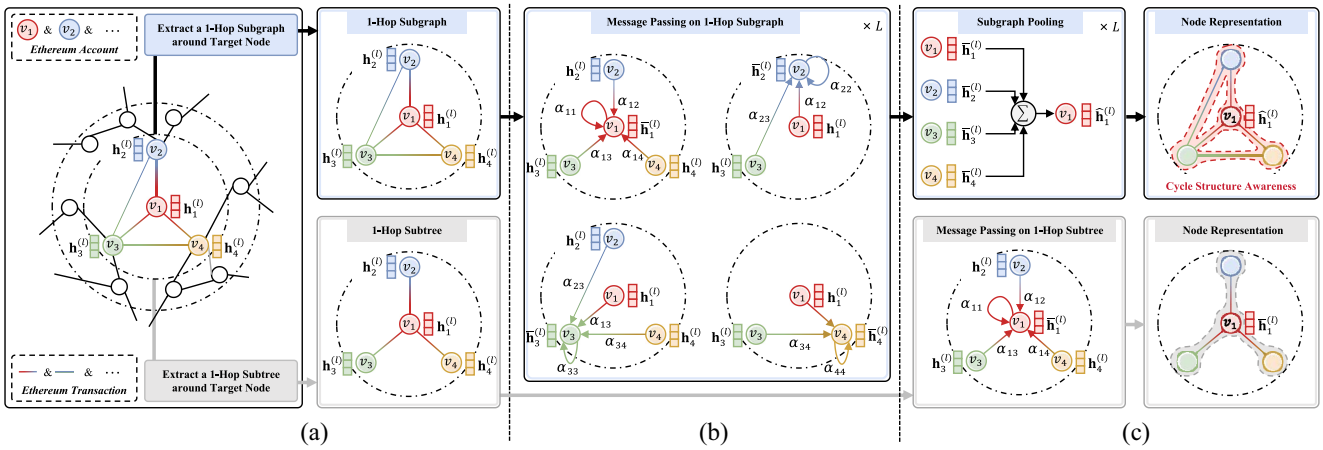

Fig. 2. Message-passing flow of the LSA. It consists of three key phases: (1) Extracting 1-hop subgraphs around the target node from the original graph; (2) Applying the local self-attention mechanism independently to each node on the subgraph; (3) Aggregating the node representations into a target node representation using subgraph pooling. The subgraph-level representation highlights the cycle structure awareness, which is absent in the subtree-level representation. Specifically, the cycle structures formed by nodes (v_1, v_2, v_3), and (v_1, v_3, v_4) are effectively captured by the subgraph-level representation. (a) Local structure extraction. (b) Subgraph-level message passing. (c) Local representation aggregation.

ETGNN [41] employ graph attention encoders based on the graph attention network (GAT) [39], effectively representing node-level account feature correlations. However, these GAT-based approaches still struggle with capturing complex transaction patterns and modeling long-range dependencies in heterophilic Ethereum networks. In response, the LGA framework is proposed to perceive local complex structures and aggregate global equivalent information, highlighting its importance in identifying Ethereum account categories accurately.

III. Method

The LGA framework is specifically designed to enhance Ethereum account de-anonymization by leveraging the strengths of an attention-based hierarchical structure. As illustrated in Fig. 1(b), LGA is a hierarchical framework that incorporates a sophisticated attention mechanism, consisting of two integral modules: 1) the LSA module, which focuses on encoding local subgraphs with a self-attention mechanism to enhance transaction pattern recognition, and 2) the Global Information-Aware (GIA) module, which employs a global self-attention mechanism to capture long-range pairwise dependencies between nodes, thus broadening the analytical scope and preventing data over-smoothing. This dual-module approach allows LGA to provide a comprehensive analysis tool for Ethereum transactions, combining detailed local insights with broad global perspectives.

A. Local Structure-Aware Module

As a critical component of the LGA framework, the LSA module is specifically designed to enhance the detection and analysis of transaction patterns. It introduces a robust approach that surpasses the limitations of traditional GNNs, which primarily focus on encoding rooted subtrees around central nodes [21], [45], [46]. Addressing this challenge, the LSA module extracts subgraphs instead of subtrees, enabling a deeper and more accurate representation of transaction dynamics. As shown in Fig. 2, a comparison between subgraph encoding and subtree encoding is provided, with the former message-passing flow indicated in blue and the latter in gray. Subgraph-level message passing captures the direct surroundings of the target node and perceives transactions between neighboring nodes, thereby enhancing the detection and analysis of complex transaction patterns. The message-passing flow of the LSA involves three main stages, as shown in Fig. 2.

1) *Local Structure Extraction:* Extracts 1-hop subgraphs around the target node from the original graph.
2) *Subgraph-Level Message Passing:* Independently applies a local self-attention mechanism to each node within the subgraph.
3) *Local Representation Aggregation:* Aggregates the node representations into a target node representation through subgraph pooling.

1) Local Structure Extraction: In the context of graph classification, let $G = (V, E)$ denote a graph, where $V = \{v_1, .., v_n\}$ represents the set of nodes and $E \subseteq V \times V$ represents the set of edges. Each node $v_i \in V$ is associated with a feature vector $\mathbf{x}_i \in \mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the feature matrix. The neighborhood of a node v_i is defined as $\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in E\}$. For a given target node v_u, its neighborhood is defined as $N = \mathcal{N}(v_u) \cup \{v_u\}$. The LSA module's structure extractor operates on this neighborhood N to derive the induced subgraph S. This subgraph S comprises all nodes in N and the edges between them, thus forming a subgraph of the original graph G, denoted by $S \subseteq G$.

2) Subgraph-Level Message Passing: Within the context of these extracted subgraphs, the LSA module employs a sophisticated local self-attention mechanism to compute attention coefficients between a node v_i and its neighbors v_j in

$$e_{ij}^{(l)} = \phi_1\left(\mathbf{W}_e^{(l)} \cdot \left[\mathbf{h}_i^{(l)} \| \mathbf{h}_j^{(l)}\right]\right) \tag{1}$$

where $\mathbf{h}_i^{(l)}$ is the hidden state of node v_i at the lth layer, $\mathbf{W}_e^{(l)}$ is a layer-specific linear transformation matrix and ϕ_1 is a nonlinear activation function, such as LeakyReLU. This operation highlights the importance of the hidden features

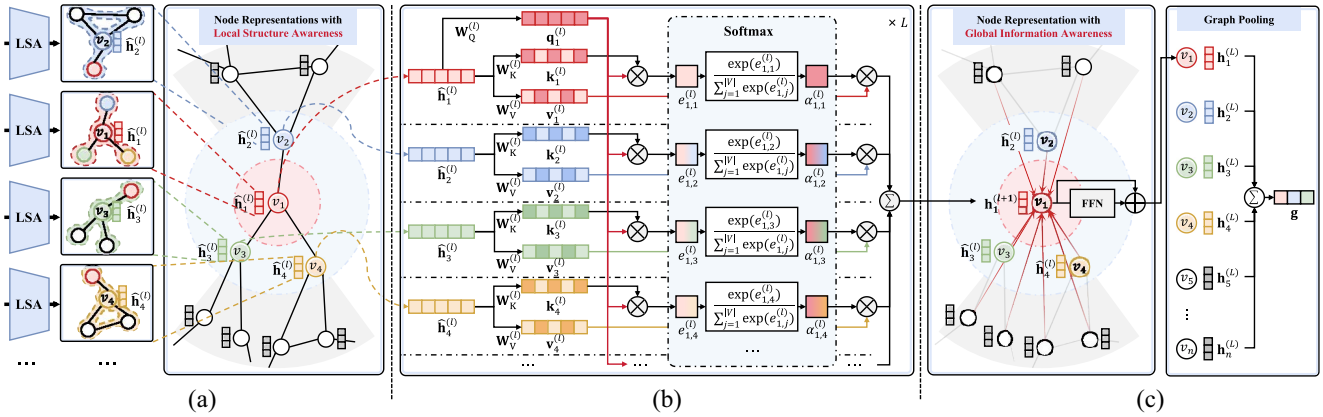

Fig. 3. Message-passing flow of GIA. It consists of three key phases: (1) Extracting node representations encoding local subgraphs using LSA; (2) Generating node representations encoding global information using the global self-attention mechanism; (3) Aggregating the final node representations into a graph representation using graph pooling. (a) LSA-based representations extraction. (b) Global information aggregation. (c) Graph representation learning.

of account v_j concerning account v_i at each layer l of the model, with the concatenation $[\mathbf{h}_i^{(l)} \| \mathbf{h}_j^{(l)}]$ combining the features of both nodes to feed into the linear transformation. The initial hidden states are set as $\mathbf{h}_i^{(0)} = \mathbf{x}_i$, where $\mathbf{x}_i$ are the initial feature inputs of the nodes. The softmax function is employed to compute the attention coefficients of neighboring nodes in (2), ensuring that the computed attention coefficients remain contextually relevant across different nodes within the subgraph

$$\alpha_{ij}^{(l)} = \phi_2\left(e_{ij}^{(l)}\right) = \frac{\exp\left(e_{ij}^{(l)}\right)}{\sum_{v_x \in \mathcal{N}(v_i|S)\cup\{v_i\}} \exp\left(e_{ix}^{(l)}\right)} \tag{2}$$

where $\mathcal{N}(v_i|S)$ is the set of v_i's neighbors on the subgraph S and ϕ_2 denotes the Softmax function. Once the attention coefficients are obtained, the neighborhood context is aggregated to update the features of the target node

$$\bar{\mathbf{h}}_i^{(l)} = \phi_3\left(\alpha_{ii}^{(l)} \cdot \mathbf{W}_\alpha^{(l)} \cdot \mathbf{h}_i^{(l)} + \sum_{v_j \in \mathcal{N}(v_i|S)} \alpha_{ij}^{(l)} \cdot \mathbf{W}_\alpha^{(l)} \cdot \mathbf{h}_j^{(l)}\right) \tag{3}$$

where a linear transformation parameterized by $\mathbf{W}_\alpha^{(l)}$ and a ReLU activation function ϕ_3 are applied to produce the hidden representation $\mathbf{h}_i^{(l)}$. It's worth noting that the local node representations are aggregated in a position-sensitive manner from the representations of neighboring nodes $v_j \in \mathcal{N}(v_i)$, ensuring effective capture of the local structural relationships.

3) Local Representation Aggregation: In order to integrate local structure into node representations, we aggregate the representations of all nodes within the subgraph using pooling functions, such as summation to obtain a subgraph representation. Subsequently, the original features of the target node are enhanced by concatenating them with the subgraph representation, resulting in the updated representation of the target node v_u

$$\hat{\mathbf{h}}_u^{(l)} = \mathbf{h}_u^{(l)} + \sum_{v_i \in \mathcal{N}(v_u)\cup\{v_u\}} \bar{\mathbf{h}}_i^{(l)} \tag{4}$$

where $\hat{\mathbf{h}}_u^{(l)}$ is a structure-aware node representations of target node v_u and $\mathbf{h}_u^{(l)}$ represents the hidden feature of target node v_u before feeding into the LSA module.

B. Global Information-Aware Module

Building on the capabilities of the LSA module, the GIA module is designed to capture long-range dependencies across large-scale graph data, expanding the receptive field for better performance and enhancing the flexibility and pattern recognition of the LGA framework. While traditional graph transformer-based methods [47], [48], [49] only encode positional relationships, the GIA module addresses this limitation by explicitly encoding structural information and correlations between all node representations on top of the LSA module. The synergistic integration of LSA and GIA simultaneously captures both structural and attribute similarities among nodes, ensuring a detailed and comprehensive analysis. The message-passing flow of the GIA module also includes three stages, as illustrated in Fig. 3.

1) *LSA-Based Representations Extraction:* Utilizes LSA to extract node representations that encode local subgraphs.
2) *Global Information Aggregation (GIA):* Generates node representations incorporating global information through the global self-attention mechanism.
3) *Graph Representation Learning:* Aggregates these node representations into a comprehensive graph representation via graph pooling.

1) LSA-Based Representations Extraction: This process provides a comprehensive analysis of transaction dynamics across all accounts by modeling the structural interactions of all nodes V in the graph G. Specifically, the LSA module is used to encode the local subgraph around each node in the entire graph to obtain structure-aware node representations. These node representations effectively capture the transaction patterns and attribute information of each node in G to provide analysis from a global perspective. The global attention mechanism is then introduced to learn diverse transaction patterns and attribute information under a global window, enabling the aggregation of various equivalent complex transaction patterns and attribute information.

2) Global Information Aggregation: The GIA module aims to preserve the relevance of all accounts in the Ethereum interaction graph by learning node representations that correlate the target node representation, $\hat{\mathbf{h}}_u^{(l)}$, with all other node representations within the graph G. Using a global attention mechanism, the GIA module ignores the original graph structure and models interactions between all node pairs. It efficiently infers node similarities through the aggregation function in

$$\mathbf{h}_u^{(l+1)} = \sum_{v_v \in V} \frac{\kappa_{\exp}\left(\hat{\mathbf{h}}_u^{(l)}, \hat{\mathbf{h}}_v^{(l)}\right)}{\sum_{v_w \in V} \kappa_{\exp}\left(\hat{\mathbf{h}}_u^{(l)}, \hat{\mathbf{h}}_w^{(l)}\right)} f\left(\hat{\mathbf{h}}_v^{(l)}\right) \tag{5}$$

where $f(\mathbf{h}) = \mathbf{W_V h}$ represents a linear function, and $\kappa_{\exp}$ denotes an exponential kernel parameterized by $\mathbf{W_Q}$ and $\mathbf{W_K}$. The exponential kernel is defined as in

$$\kappa_{\exp}(\mathbf{h}, \mathbf{h}') := \exp\left(\langle \mathbf{W_Q h}, \mathbf{W_K h}' \rangle / \sqrt{d_{\text{out}}}\right) \tag{6}$$

where the dot product $\langle \cdot, \cdot \rangle$ captures the similarity between node features. The LSA module's node features, which integrate local structures, allow the exponential kernel to effectively identify both feature and structural similarities.

3) Graph Representation Learning: This process retains complex patterns within graph structures. Node representations are initially processed through a feed-forward network (FFN) that incorporates residual connections in

$$\begin{aligned} \mathbf{h}_u^{(L)\prime} &= \mathbf{h}_u^{(L-1)} + \mathbf{h}_u^{(L)} \\ \mathbf{h}_u^{(L)\prime\prime} &= \text{FFN}\left(\mathbf{h}_u^{(L)\prime}\right) := \phi_3\left(\mathbf{h}_u^{(L)\prime} \mathbf{W_1}\right) \mathbf{W_2}. \end{aligned} \tag{7}$$

After this step, a graph pooling layer aggregates the refined node representations into a final graph representation in

$$\mathbf{g} = \sum_{v_u \in V} \mathbf{h}_u^{(L)\prime\prime}. \tag{8}$$

Combining the local structural learning of LSA with the global reasoning abilities of GIA provides a comprehensive understanding of transaction networks. The LSA module excels at learning localized subgraph patterns, capturing intricate neighborhood structures, while the GIA module adds a layer of global reasoning that recognizes long-range dependencies between nodes. This complementary relationship enables accurate mapping of structural and attribute similarities across the entire graph, significantly improving the ability to identify complex patterns. The result is an enhanced model capable of handling heterophilic networks like Ethereum, providing robust performance in tasks, such as account de-anonymization.

C. Loss Function

In our approach to the Ethereum account de-anonymization task, we employ a dual-component loss function that integrates both classification and self-supervised learning mechanisms. The classification task is formulated using a cross-entropy loss, where the mapping function $f(\cdot)$ transforms graph representations into labels that reflect account identities. This mapping is quantified through the loss function $\mathcal{L}_p$ in

$$\mathcal{L}_p = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot \log(f(\mathbf{g}_i)) \tag{9}$$

where y_i are the true labels for the account identities.

In order to alleviate the scarcity of labeled data, the model incorporates a self-supervised loss function that enhances its discriminative ability. This function utilizes a contrastive mechanism, maximizing the cosine similarity between two augmented views of the same graph. By doing so, the model focuses on aligning related graph representations in the contrastive space, as defined in (10), which effectively strengthens its ability to discern subtle patterns and relationships within the dataset

$$\mathcal{L}_i = -\log \frac{e^{\cos(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau}}{\sum_{j=1, j\neq i}^{N} e^{\cos\left(\mathbf{z}_i^1, \mathbf{z}_j^2\right)/\tau}} \tag{10}$$

where $\mathbf{z}_i^1$ and $\mathbf{z}i^2$ are two different augmented representations of the same graph, τ is the temperature parameter, and cos denotes the cosine similarity function. The overall self-supervised loss, $\mathcal{L}s$, is then calculated by averaging $\mathcal{L}_i$ over all pairs in

$$\mathcal{L}_s = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i. \tag{11}$$

The complete loss function combines these two aspects with a balancing hyperparameter λ, integrating the self-supervised and classification losses in

$$\mathcal{L} = \mathcal{L}_p + \lambda \cdot \mathcal{L}_s \tag{12}$$

where λ adjusts the emphasis between the two loss functions, allowing for tailored learning dynamics based on the available dataset characteristics.

IV. Experiments

A. Datasets

1) lw-AIG: The lw-AIG dataset is sourced from the blockchain data of the Xblock website, encompassing transactions, smart contracts, and publicly labeled account identities [50]. The dataset is constructed from Ethereum user relationships, using interaction merging and feature construction operations to form a lw-AIG [21]. The lw-AIG dataset contains a total of 5732 graphs for binary classification. In lw-AIG, nodes represent accounts, node features represent the contract call information of the account, and edges indicate transaction data and contract callbacks.

The lw-AIG dataset is divided into four subsets based on account identity labels obtained from the Ethereum blockchain explorer's Label Word Cloud: 1) ICO wallets; 2) mining accounts; 3) exchange accounts; and 4) phishing&hack accounts, as shown in Table II. Each subset exhibits distinct interaction patterns. ICO wallets are characterized by a high number of outbound edges, which distribute investment returns

TABLE II
STATISTICS OF GRAPH DATASETS SAMPLED FROM THE ACCOUNT INTERACTION GRAPH OF ETHEREUM. $|G|$ IS THE NUMBER OF GRAPHS, AVG. $|N|$ AND AVG. $|E|$ ARE THE AVERAGE NUMBER OF NODES AND EDGES IN GRAPHS, RESPECTIVELY, $|x|$ AND $|e|$ ARE THE NUMBER OF NODE AND EDGE FEATURES IN GRAPHS

Dataset	Strategies	$\|G\|$	Avg.$\|N\|$	Avg.$\|E\|$	$\|x\|$	$\|e\|$
Eth-ICO	Amount	146	42.5	141.3	14885	2
	Times		52.2	152.6		
	avgAmount		42.4	140.7		
Eth-Mining	Amount	130	23.7	72.9	14885	2
	Times		24.7	67.2		
	avgAmount		29.0	91.9		
Eth-Exchange	Amount	386	33.6	123.7	14885	2
	Times		38.0	113.4		
	avgAmount		38.6	148.6		
Eth-Phish&Hack	Amount	5070	37.3	110.8	14885	2
	Times		37.8	101.6		
	avgAmount		37.8	111.3		

TABLE III
HYPERPARAMETERS FOR LGA MODELS TRAINED ON LW-AIG DATASET AND ELLIPTIC DATASET. THE ELLIPTIC DATASET FOLLOWS THE NUMBER OF SAMPLES PER TIME STEP AS THE BATCH SIZE

Hyperparameter	lw-AIG	Elliptic
Number of layers	2	2
Hidden dimensions	64	64
FFN hidden dimensions	128	128
Attention heads	8	8
Dropout rate	0.1	0.1
Graph pooling	mean	none
Learning rate	0.0003	0.001
Patience	20	20
Number of epochs	1000	1000
Batch size	32	-
Weight decay	0.0001	0.0001
Class weight	0.7:0.3	1:1

from a central ICO account to surrounding supporters. Mining accounts display numerous outbound edges from a central pool to peripheral mining nodes and accumulated rewards. Due to frequent client interactions, exchange accounts function as highly central hub nodes with significant in-degree and out-degree. Phishing&hack accounts are identified by a few bidirectional edges between the central node and surrounding nodes, with high in-degree and low out-degree for the central node. Long-distance nodes are common in these graphs, and transactions between nodes form a variety of complex cycle structures. These diverse categories and complex transaction patterns provide reliable empirical scenarios for exploring the effectiveness of LGA.

Each graph in lw-AIG is sampled by TopK according to different edge information (transaction amount, transaction times, or average transaction amount), generating three types of graph data sets represented by "Amount," "Times," and "avgAmount," respectively, as shown in Table II. For each type of identity label (ICO wallet, mining, exchange, and phishing&hacking), lw-AIG samples the target account graphs with the specified identity label as positive samples, and randomly samples the same number of account graphs with other labels as negative samples to ensure data balance. Each graph in lw-AIG corresponds to a target account, forming a dataset $D = \{(g_i, y_i) \mid \forall(v_i, y_i) \in Y\}$ for binary classification. The labels of target accounts are assigned to the graphs, facilitating the learning of a function that maps the graph patterns to the identity labels of the accounts.

2) Elliptic: We also curate the Bitcoin transaction network dataset, Elliptic, for experiments to demonstrate the scalability of the proposed LGA framework. The Elliptic dataset originates from a labeled Bitcoin transaction dataset released by Elliptic. This dataset associates Bitcoin transactions with real-world entities, classified into licit categories (exchanges, wallet providers, miners, and licit services) and illicit categories (fraud, malware, terrorist organizations, ransomware, and Ponzi schemes). It encompasses Bitcoin transactions across 49 time intervals, each approximately two weeks in duration. The network comprises 203 769 nodes and 234 355 edges, where nodes represent transactions and edges indicate that the output of one transaction serves as the input for the subsequent transaction. Furthermore, there are 166 features associated with each node. About 33% of transactions are labeled, of which 4545 are illicit and 42 019 are licit.

B. Comparison Methods

For the lw-AIG dataset, we systematically evaluate the effectiveness of LGA by comparing two principal methodological categories for account identification: 1) graph embedding techniques and 2) GNN-based approaches. In the graph embedding category, we utilize established techniques, such as DeepWalk [32], Node2Vec [35], Struc2Vec [33], and Trans2Vec [34] to transform the account data into vector spaces. These embeddings are then analyzed using ML classifiers, specifically logistic regression (LR) and random forest (RF), to perform the identification tasks. Concurrently, we explore the capabilities of GNN-based methods employing models, such as GIN [37], GAT [39], and GCN [38], which are enhanced for subgraph classification through the integration of pooling layers and prediction heads. Our approaches are benchmarked against advanced models, such as Ethident [21], AEtransGAT [40], and ETGNN [41], which leverages node-level attention for high precision, and the I^2BGNN [20] method, offering two variants, I^2BGNN-A and I^2BGNN-T, to address account identification challenges using varied edge information. The comprehensive performance comparison of these methods is detailed in Table IV. For the Elliptic dataset, Trans2Vec, I^2BGNN, AEtransGAT, and ETGNN are not included in the comparison methods because they require edge attributes that are not provided by the Elliptic dataset. The comprehensive performance comparison of illicit transaction classification is shown in Table V.

C. Implementation Details

The LGA is implemented using the PyTorch framework and Torch Geometric library, and it operates on an NVIDIA Tesla

TABLE IV

MICRO F1 SCORE (%) COMPARISONS WITH STATE-OF-THE-ART METHODS FOR ETHEREUM ACCOUNT DE-ANONYMIZATION ON THE LW-AIG DATASETS. THE PERFORMANCE METRICS OF OUR METHOD ARE HIGHLIGHTED IN BOLD. COMPARING THE PERFORMANCE OF DIFFERENT METHODS WITH LGA, THE GAPS OVER ARE RECORDED IN RED. ∗ INDICATES A METHOD DESIGNED SPECIFICALLY FOR THE ACCOUNT DEANONYMIZATION TASK

Method	Dataset (with different sampling strategies)											
	Eth-ICO			Eth-Mining			Eth-Exchange			Eth-Phish&Hack		
	Amount	Times	avgAmount	Amount	Times	avgAmount	Amount	Times	avgAmount	Amount	Times	avgAmount
DeepWalk + LR	56.69+40.26	58.96+37.12	59.64+36.34	56.94+32.73	60.26+29.01	62.07+27.79	59.17+34.34	63.82+29.87	62.53+31.05	58.73+39.45	67.99+30.03	64.29+33.95
DeepWalk + RF	73.24+23.71	68.93+27.15	67.12+28.86	65.13+25.24	71.58+17.69	65.40+24.46	76.31+17.20	80.53+13.16	80.19+13.39	91.14+7.04	89.77+8.25	92.71+5.53
Struc2Vec + LR	61.00+35.95	58.96+37.12	55.10+40.88	51.82+37.85	61.56+27.71	59.80+30.06	63.48+30.03	57.02+36.67	59.09+34.39	57.02+41.16	59.47+38.55	54.89+43.35
Struc2Vec + RF	61.45+35.5	60.32+35.76	60.09+35.89	63.61+26.06	69.44+19.83	60.81+29.05	74.07+19.44	71.66+22.03	69.60+23.98	66.33+31.85	68.93+29.09	65.02+33.22
Trans2Vec + LR*	73.77+23.18	61.05+35.03	59.64+36.34	71.41+18.26	53.78+35.49	75.87+13.99	57.52+35.99	76.81+16.88	79.75+13.83	68.67+29.51	63.05+34.97	58.21+40.03
Trans2Vec + RF*	86.50+10.45	71.02+25.06	73.16+22.82	73.34+16.33	61.85+27.42	77.87+11.99	72.91+20.60	82.65+11.04	87.31+6.27	89.94+8.24	89.75+8.27	89.98+8.26
Node2Vec + LR	80.95+16	62.36+33.72	79.37+16.61	64.36+25.31	72.85+16.42	79.24+10.62	66.06+26.91	82.77+10.92	81.05+12.53	66.53+31.65	82.58+15.44	79.73+18.51
Node2Vec + RF	88.21+8.74	78.91+17.17	89.34+6.64	74.37+15.30	78.48+10.79	78.72+11.14	83.12+10.39	88.98+4.71	86.56+7.02	92.04+6.14	94.06+3.96	92.17+6.07
GIN	75.20+21.75	78.66+17.42	75.17+20.81	59.07+30.60	57.48+31.79	63.99+25.87	81.56+11.95	84.35+9.34	85.42+8.16	95.89+2.29	95.79+2.23	95.88+2.36
GCN	87.57+9.38	86.73+9.35	86.89+9.09	75.25+14.42	82.91+6.36	83.52+6.34	90.23+3.28	89.85+3.84	89.79+3.79	96.49+1.69	96.15+1.87	96.49+1.75
GAT	88.44+8.51	90.02+6.06	86.11+9.87	80.28+9.39	78.63+10.64	82.71+7.15	91.01+2.50	91.42+2.27	90.70+2.88	95.87+2.31	95.68+2.34	95.91+2.33
I^2BGNN-A*	92.13+4.82	91.10+4.98	92.41+3.57	82.24+7.43	79.52+9.75	80.20+9.66	89.64+3.87	91.63+2.06	88.76+4.82	95.94+2.24	95.93+2.09	96.07+2.17
I^2BGNN-T*	91.02+5.93	91.72+4.36	90.13+5.85	82.64+7.03	82.13+7.14	83.84+6.02	89.28+4.23	91.87+1.82	89.87+3.71	95.99+2.19	95.98+2.04	96.11+2.13
Ethident*	94.05+2.90	92.76+3.32	94.05+1.93	86.38+3.29	87.00+2.27	83.50+6.63	93.16+0.35	93.55+0.14	93.34+0.24	97.93+0.25	97.58+0.44	97.98+0.26
AEtransGAT*	93.11+3.84	91.09+4.99	91.75+4.23	86.13+3.54	86.90+2.37	86.89 +2.97	92.25+1.26	89.15+4.54	88.37+5.21	93.80+4.38	97.55+0.47	97.67+0.57
ETGNN*	94.50+2.45	90.42+5.66	93.14+2.84	86.89+2.78	87.68+1.59	84.58 +5.28	92.74+0.77	93.26+0.43	92.48+1.10	95.18+3.00	97.53+0.49	96.92+1.32
LGA*	**96.95**	**96.08**	**95.98**	**89.67**	**89.27**	**89.86**	**93.51**	**93.69**	**93.58**	**98.18**	**98.02**	**98.24**

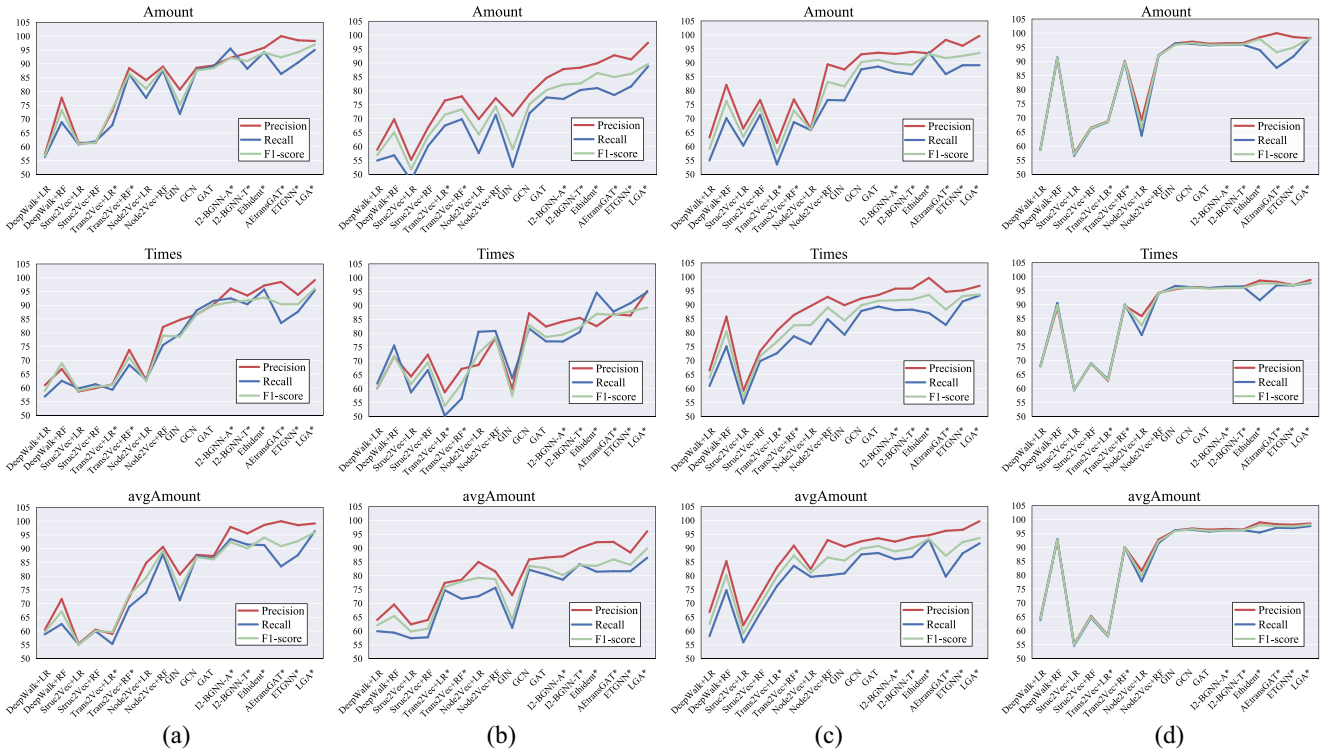

Fig. 4. Comparison with state-of-the-art methods for Ethereum account de-anonymization on the lw-AIG dataset, reporting precision, recall, and F1 score. (a) Eth-ICO. (b) Eth-Mining. (c) Eth-Exchange. (d) Eth-Phish&Hack.

V100S-PCIE-32GB GPU for computational support. Table III summarizes the hyperparameters of training LGA models on different datasets. The training settings include the initial learning rates of 0.0003 for the lw-AIG dataset and 0.001 for the Elliptic dataset, respectively. The training regimen features a 10-epoch linear warm-up, followed by a cosine annealing learning rate scheduler to optimize performance. The AdamW optimizer, with a weight decay of 0.0001, manages the optimization process. We conduct the training over 1000 epochs, integrating early stopping with a patience

TABLE V
ILLICIT TRANSACTION CLASSIFICATION RESULTS ON THE ELLIPTIC DATASET, REPORTING PRECISION (%), RECALL (%), F1 SCORE (%), AND MICRO F1 SCORE (%). THE GAPS ARE RECORDED IN RED

Method	Precision	Recall	F1	Micro-F1
DeepWalk + LR	0.00+86.06	0.00+65.00	0.00+74.07	93.50+3.54
DeepWalk + RF	10.14+75.92	00.65+64.35	01.22+72.85	93.17+3.87
Struc2Vec + LR	0.00+86.06	0.00+65.00	0.00+74.07	93.50+3.54
Struc2Vec + RF	6.06+80.00	0.18+64.82	0.36+73.71	93.33+3.71
Node2Vec + LR	0.00+86.06	0.00+65.00	0.00+74.07	93.50+3.54
Node2Vec + RF	11.49+74.57	00.92+64.08	1.71+72.36	93.10+3.94
GIN	57.61+28.45	54.85+10.15	56.20+17.87	94.45+2.59
GCN	81.38+4.68	54.48+10.52	65.27+8.80	96.23+0.81
GAT	84.05+2.01	62.79+2.21	71.88+2.19	96.81+0.23
Ethident*	80.61+5.45	51.06+13.94	62.52+11.55	96.02+1.02
LGA*	**86.06**	**65.00**	**74.07**	**97.04**

threshold of 20 epochs to mitigate overfitting. To obtain statistically significant experimental results, a threefold cross-validation is performed 10 times, and the average result and its standard deviation are reported. We divide the lw-AIG dataset into training, validation, and testing sets according to the proportion specified by Ethident [21]. For the Elliptic dataset, the task is a binary classification of nodes, where the two classes of licit and illicit transactions are unbalanced, and the identification of a small number of illicit transactions is more important. To this end, we train the LGA model using a weighted cross-entropy loss with a ratio of 0.7:0.3 for the illicit and licit classes to give higher importance to illicit samples. Since the Elliptic dataset is used for node-level classification, the Graph pooling layer is abolished. The Elliptic dataset is divided into training and testing sets according to the 0.7:0.3 temporal split. The first 34 time steps are used to train the model, and the last 15 time steps are used for testing.

D. Evaluation Metrics

The task of classifying accounts can be transformed into a binary classification problem. We consider precision, recall, F1 score, and micro F1 score as evaluation metrics. A higher precision indicates that the model is less likely to misclassify negative samples as positive samples. A higher recall indicates that the model is more capable of capturing samples that are truly positive. F1 score can be considered the harmonic mean of the model's precision and recall. Micro F1 provides a single score that reflects the model's performance across all classes.

E. Classification Performance

We conducted a comprehensive comparison of our proposed LGA framework with other baseline methods on the lw-AIG dataset, as shown in Table IV and Fig. 4, where our model demonstrated state-of-the-art results. The micro F1 score achieved by LGA on four subdatasets significantly exceeded all the methods using graph embeddings, with improvements ranging from 3.96% to 40.88%, as shown in Table IV. This enhancement in performance can be attributed to our LGA's superior capability in capturing the behavior patterns of different accounts when acquiring subgraph features. In comparison with the state-of-the-art GNN-based methods, such as Ethident [21], ETGNN [41], and AEtransGAT [40], the micro F1 score achieved by LGA improved by 1.93%–6.63% on the Eth-ICO and Eth-Mining subdatasets. Improvements of approximately 0.1%–0.5% were also observed in the Eth-Exchange and Eth-Phish&Hack datasets compared to the best performance. We further visualize the comparison results of LGA and the comparison methods in terms of precision, recall, and F1 score, as shown in Fig. 4.

The extended experiments are conducted on the Elliptic dataset derived from the Bitcoin transaction network to verify the transferability of LGA on the illicit transactions detection task. The detailed experimental results are shown in Table V. From these results, it is evident that the proposed LGA outperforms all other methods, achieving significantly better outcomes. Specifically, the precision, recall, F1 score, and micro F1 score for LGA are 86.06%, 65.00%, 74.07%, and 97.04%, respectively, surpassing the second-best method, GAT, which recorded scores of 84.05%, 62.79%, 71.88%, and 96.81%. Notably, we observe a significant drop in performance for graph embedding methods. In the Elliptic dataset, each node is associated with transaction-related features that provide insights into various categories of entities. Therefore, we conclude that graph embedding methods that do not incorporate node or edge features are insufficient to address the challenges of illicit transactions detection.

From the experimental results, it was observed that the performance of graph embedding methods was significantly lower than that of GNN-based methods, and they ranked relatively lower. Graph embedding methods are unable to learn end-to-end task-relevant features and are highly dependent on the chosen classifiers to achieve relatively high performance. In contrast, GNN methods can learn transaction patterns from the graph's topological structure, thus obtaining more expressive information about account behaviors, giving them a relative performance advantage. However, GNN-based methods still have shortcomings in learning critical information at the edges, and the generated subtree structures are overly simplistic and lack rich semantic expression. In contrast to these methods, LGA extracts node subgraphs for information aggregation to obtain subgraph-level representations of nodes. Additionally, the GIA module is employed to capture long-range dependencies between global nodes and implement structural and attribute similarity detection between nodes, thus achieving state-of-the-art performance on the lw-AIG dataset.

F. Sensitivity Analysis

To evaluate the performance of LGA in multiclass classification tasks, we further extracted an equal number of 65 positive samples (limited by the number of positive samples in the mining subdataset) from four subdatasets to form a dataset for multiclass classification. Table VI reports key metrics, such as Precision, Recall, and F1-score. According to different sampling strategies, LGA shows variations in its sensitivity to the multiclass nature of the dataset. For example, under the

TABLE VI
MULTICLASS CLASSIFICATION RESULTS ON THE LW-AIG DATASET. THE EXPERIMENTAL RESULTS UNDER THREE SAMPLING STRATEGIES: AMOUNT, TIMES, AND AVGAMOUNT ARE REPORTED

Type	Dataset (with four categories for multi-classification tasks)											
	Eth-ICO			Eth-Mining			Eth-Exchange			Eth-Phish&Hack		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Amount	77.95	46.32	57.77	77.96	73.95	74.81	67.68	64.50	65.97	60.71	87.81	71.19
Times	57.89	49.27	53.23	86.76	81.52	83.72	75.00	55.19	62.53	56.02	78.35	64.95
avgAmount	67.56	43.15	52.27	81.83	72.58	75.98	72.82	70.71	70.78	57.92	84.63	68.36

TABLE VII
COMPARATIVE ANALYSIS OF EFFICIENCY ON THE ETH-ICO "VOLUME" SUBDATASET. REPEAT THE THREEFOLD CROSS VALIDATION 10 TIMES AND REPORT THE AVERAGE INFERENCE DURATION (S), AVERAGE INFERENCE SPEED (GRAPH/S) AND STANDARD DEVIATION

Method	Duration ↓	Speed ↑
GIN	$0.144_{\pm 0.009}$	$341.19_{\pm 22.20}$
GCN	$0.082_{\pm 0.003}$	$596.52_{\pm 21.71}$
GAT	$0.087_{\pm 0.007}$	$562.87_{\pm 37.15}$
I^{2}BGNN-A*	$0.085_{\pm 0.004}$	$573.48_{\pm 26.44}$
I^{2}BGNN-T*	$0.087_{\pm 0.007}$	$565.38_{\pm 42.81}$
Ethident*	$0.114_{\pm 0.009}$	$431.68_{\pm 33.93}$
AEtransGAT*	$0.111_{\pm 0.006}$	$441.85_{\pm 25.21}$
ETGNN*	$0.140_{\pm 0.009}$	$350.32_{\pm 23.34}$
LGA*	$\mathbf{0.108}_{\pm \mathbf{0.006}}$	$\mathbf{462.15}_{\pm \mathbf{25.94}}$

"Amount" sampling strategy, the recall for Eth-Phish&Hack is 87.81%, significantly higher than Eth-Mining's 73.95%; whereas in the "Times" sampling strategy, Eth-Mining leads with a recall of 81.52%; under the "avgAmount" sampling strategy, Eth-Phish&Hack leads again with a recall of 84.63%, demonstrating high sensitivity in identifying malicious account behavior. These results emphasize the effectiveness and flexibility of LGA as a malicious behavior detection tool.

G. Efficiency Performance

Assume a graph has n nodes with a maximum degree of d, and the maximum number of nodes in a rooted subgraph is m. In the LSA module, message passing needs to be performed for the rooted subgraphs of all n nodes, with a time complexity of $\mathcal{O}(n \cdot md)$. To improve the scalability of LSA, a small k-hop setting can be used to achieve a smaller m. GIA interacts with the features of all nodes in the global perspective graph, with a time complexity of $\mathcal{O}(n^2)$. Focusing on the item with the highest growth rate, the total time complexity of LGA is $\mathcal{O}(n^2)$. In order to comprehensively evaluate the practical value of LGA, a statistical analysis of the inference execution time of different methods under the "Amount" sampling strategy of the Eth-ICO subdataset was performed. As shown in Table VII, compared with methods specifically designed for account deanonymization, such as I^{2}BGNN-A, I^{2}BGNN-T, Ethident, AEtransGAT, and ETGNN, LGA ranks third in efficiency performance. Compared with Ethident, AEtransGAT, and ETGNN, the inference duration is reduced by 0.006, 0.003, and 0.032 s, respectively, and the inference speed is increased by 30.47graph/s, 20.30graph/s, and 111.83graph/s, respectively. Compared with I^{2}BGNN, although LGA introduces additional computation time, it significantly improves the classification performance, as shown in Table IV.

H. Ablation Studies

1) Evaluating the Effectiveness of Local Attention Mechanism: Table VIII provides an ablation study on how the local attention mechanism affects account classification performance, showing the micro F1 scores obtained by LSA and LSA without local attention LSA(w/o LA). From the experimental results, it is evident that LSA does not achieve significant positive gains compared to LSA (w/o LA). Furthermore, we investigate how the local attention mechanism affects the performance of the LGA framework. Table VIII demonstrates the performance differences between LGA(w/o LA) and LGA. Specifically, the integration of the local attention mechanism has precisely captured the relevance of interacting accounts, resulting in an increase of the micro F1 score by 0.49%–2.73%. These results indicate that focusing on local correlation is insufficient to achieve good performance, and the introduction of global correlation promotes better performance.

2) Evaluating the Effectiveness of Structural Subgraph: Table VIII provides a detailed analysis of the impact of different encoding strategies on account classification performance. This study first explores the impact of two encoding strategies: 1) subtree encoding and 2) subgraph encoding, on the performance of the LSA module. Experimental results show that the LSA module based on subgraph coding does not show obvious performance advantages over the one based on subtree coding. In contrast, LGA with subgraph encoding achieves an overall performance improvement, ranging from 0.41% to 3.20%. This finding suggests that the subgraph encoding strategy can better capture behavioral patterns, effectively enhancing account identification performance. Another insight from these results is that the synergistic effect of the GIA module with structured subgraphs enables the LGA framework to precisely delineate the subtle differences between all transaction patterns.

3) Evaluating the Effectiveness of LSA Module: To investigate the impact of the LSA module on account classification performance, we conduct ablation experiments on the LSA module. The micro F1 score is reported in Table VIII. Clearly,

TABLE VIII
PERFORMANCE COMPARATIVE ANALYSIS FOR ETHEREUM ACCOUNT DE-ANONYMIZATION ON THE LW-AIG DATASET WITH DIFFERENT CONFIGURATIONS OF LGA, REPORTING MICRO F1 SCORE (%) IN DIFFERENT SETTINGS. THE GAPS ARE RECORDED IN RED

Method		Dataset (with different sampling strategies)											
		Eth-ICO			Eth-Mining			Eth-Exchange			Eth-Phish&Hack		
		Amount	Times	avgAmount	Amount	Times	avgAmount	Amount	Times	avgAmount	Amount	Times	avgAmount
GIA		92.80+3.85	92.19+3.89	92.19+3.79	84.93+4.74	85.47+3.80	85.77+4.19	92.46+1.05	91.95+1.74	91.17+2.41	96.68+1.50	96.72+1.30	97.02+1.22
LSA (w/o LA)	subtree	93.42+3.53	92.11+3.97	93.08+2.90	85.02+4.65	85.78+3.49	86.32+3.54	91.45+2.06	91.63+2.03	91.09+2.49	97.03+1.15	96.74+1.28	97.02+1.22
	subgraph	93.35+3.60	91.99+4.09	92.45+3.53	85.40+4.27	86.54+2.73	85.78+4.08	91.87+1.64	91.87+1.64	91.24+2.34	97.29+0.89	96.82+1.20	97.35+0.89
LSA	subtree	92.80+4.15	91.70+4.38	92.46+3.52	84.86+4.81	85.09+4.18	86.62+3.24	91.92+1.59	91.82+1.87	91.19+2.39	96.87+1.31	96.57+1.45	97.14+1.10
	subgraph	92.46+4.49	91.57+4.51	93.21+2.77	85.85+3.82	85.01+4.26	86.63+3.23	91.50+2.01	91.97+1.72	91.58+2.00	97.27+0.91	97.02+1.00	97.27+0.97
LGA (w/o LA)	subtree	94.73+2.22	94.18+1.90	94.93+1.05	86.63+3.04	86.24+3.03	87.39+2.47	92.15+1.36	92.44+1.25	91.17+2.41	97.45+0.73	97.02+1.00	97.34+0.90
	subgraph	94.93+2.02	93.97+2.11	95.13+0.85	87.39+2.28	86.54+2.73	87.86+2.00	91.89+1.62	91.97+1.72	91.55+2.03	97.46+0.72	96.83+1.19	97.46+0.78
LGA	subtree	94.52+2.43	93.83+2.25	94.52+1.46	86.93+2.74	86.07+3.20	88.23+1.63	92.18+1.33	92.15+1.54	91.56+2.02	97.54+0.64	96.99+1.03	97.32+0.92
	subgraph	**96.95**	**96.08**	**95.98**	**89.67**	**89.27**	**89.86**	**93.51**	**93.69**	**93.58**	**98.18**	**98.02**	**98.24**

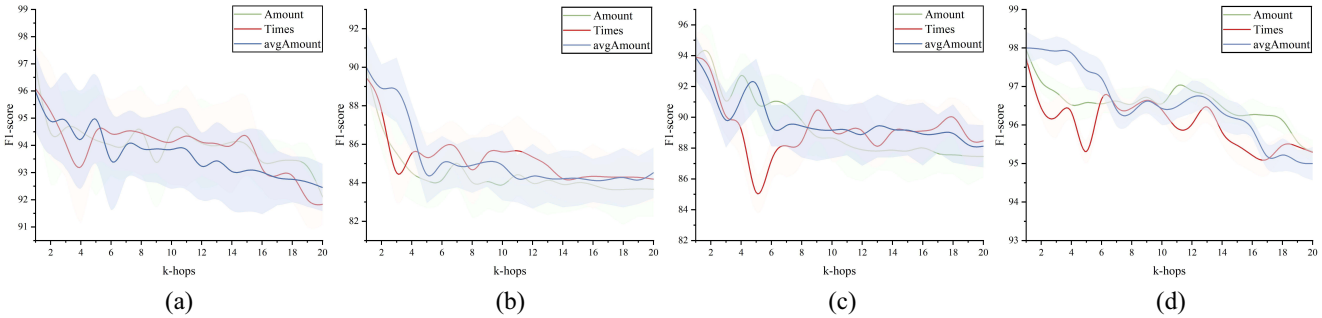

Fig. 5. Performance of LGA with different hyperparameters is analyzed on the lw-AIG dataset, showing how changing the k-hops of the subgraph affects the performance. Perform threefold cross-validation ten times and report the average micro F1 score. The filled areas represent the standard deviation. (a) Eth-ICO. (b) Eth-Mining. (c) Eth-Exchange. (d) Eth-Phish&Hack.

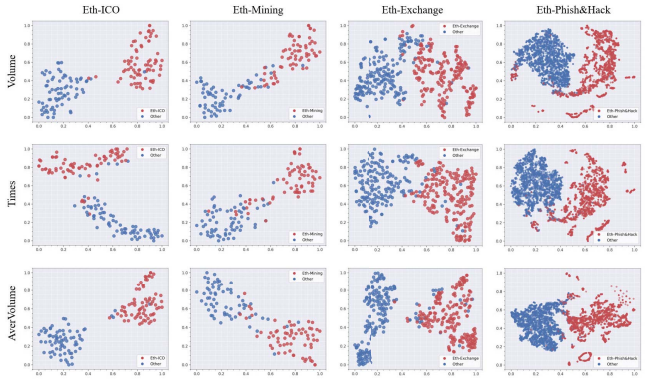

Fig. 6. T-SNE visualization of graph embeddings learned by LGA, where the account graph embeddings for the specified identity label are shown in red, and the sampled account graph embeddings with other labels are shown in blue.

the performance of the LGA framework significantly outperforms that of the GIA module, highlighting the differences between the complete model and the variant that lacks the LSA module. Specifically, the integration of the LSA module allows the LGA to capture various complex transaction patterns, improving the LGA's micro F1 score from 0.98% to 4.74%. Different types of accounts are associated with their unique transaction patterns, which are crucial for accurately categorizing account types. The standalone GIA module fails to capture the distinctions between different patterns, resulting in a decrease in classification performance.

4) Evaluating the Effectiveness of GIA Module: To explore the contribution of the GIA module in enhancing account classification performance, we conduct ablation experiments on the GIA module and verify the micro F1 scores on the lw-AIG dataset, as shown in Table VIII. The results show that integrating the GIA module on top of the LSA module leads to significant performance improvements, with increases ranging from 0.68% to 4.51%. These findings indicate that the introduction of a global attention mechanism plays a crucial role in heterophilic Ethereum transaction networks. This effect stems from establishing long-range dependencies between nodes in large-scale graph datasets and adaptively filtering appropriate accounts for information aggregation through attention weights. It is noteworthy that the excellent performance results from the synergistic action between the LSA and GIA modules. The underlying idea is that the LSA module captures the transaction patterns of all accounts, while the flexible integration of the LSA and GIA modules simultaneously captures the similarities of transaction patterns and features among accounts.

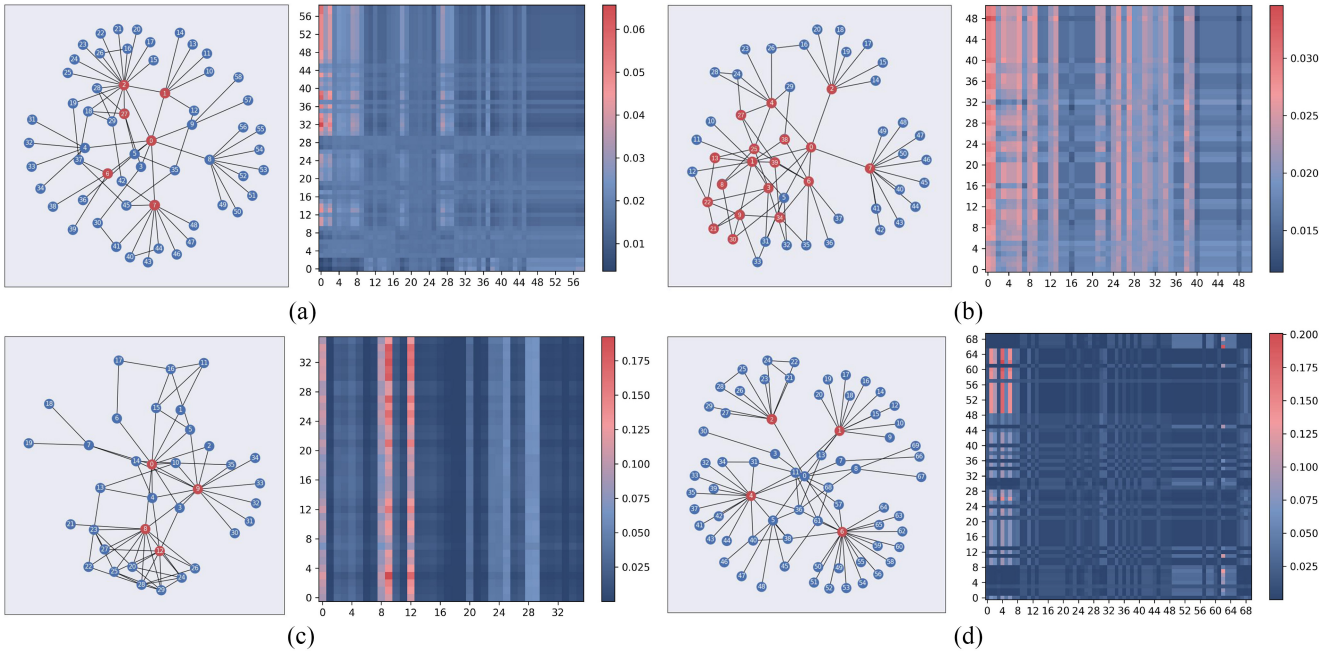

Fig. 7. Example graph and attention map retrieved from LGA. The vertical axis corresponds to the attention weights that the target node receives from other nodes, and the horizontal axis corresponds to the attention weights that a specific node allocates to other nodes. Attention weights sum to 1 on the horizontal axis. Nodes that receive more attention from other nodes are displayed in red, while nodes that receive less attention are displayed in blue. (a) Eth-ICO. (b) Eth-Mining. (c) Eth-Exchange. (d) Eth-Phish&Hack.

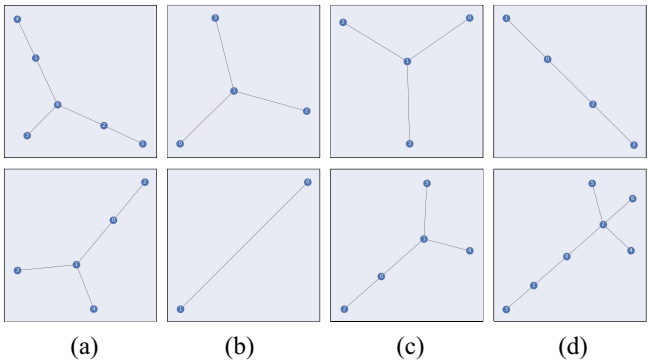

Fig. 8. Specific example analysis of binary classification errors. (a) Eth-ICO. (b) Eth-Mining. (c) Eth-Exchange. (d) Eth-Phish&Hack.

I. Hyperparameter Studies

LSA explicitly incorporates local structural information into the local attention mechanism. It is worth noting that multihop LSA exerts a wide receptive field and forms a functional localization that overlaps with GIA. In order to explore how LSA and GIA work together to achieve their respective advantages, we use error bar with filled areas to show how the choice of k-hops of the subgraph affects the results, where the filled area represents the standard deviation. Fig. 5 shows the impact of changing the k-hops hyperparameters of LGA on the test micro F1 score on the lw-AIG dataset. It is observed that smaller k-hops can significantly improve performance. As k-hops increase, the performance of LGA gradually decreases. Stacking GNN layers aggregate information from outside the local neighborhood, effectively expanding the GNN receptive field. For example, node A must pay attention to remote node B that is k hops away, which can be achieved by stacking k layers of GNN. However, as the depth increases [51], the performance of GNN drops drastically because node A receives signals from other nodes, thus diluting the signal from node B. GNNs with multiple layers cause node representations to become equivalent throughout the graph, a phenomenon called over-smoothing [52]. Node representations are not equivalent across the entire graph when k-hop is small, and the GIA module uses a global attention mechanism to perform global reasoning and magnify the global interactions of equivalent nodes. As a powerful global reasoning module, GIA's global reasoning capability effectively routes equivalent information on the highly heterophilic Ethereum network by modeling long-range dependencies.

J. Qualitative Visualization

1) t-SNE Visualization of Embeddings Learned by LGA: Assume that the LGA model has learned relevant information about account categories based on the features of the accounts. In this case, we should expect to observe account clusters in the embedding space, where accounts with the same label belong to the same cluster. We use t-Distributed Stochastic Neighbour Embedding (t-SNE) [53] to simplify these embedding vectors into 2-D vectors, so that we can intuitively observe the distribution of the embedding vectors on a standard 2-D scatter plot. Fig. 6 shows the t-SNE visualization of the graph embeddings generated by the four subdatasets.

2) Example Graph and Attention Map Retrieved From LGA: We carefully chose several example graphs from our dataset to illustrate the distinct transaction patterns inherent to various types of accounts. Further, we retrieve the attention map from the LSA to reveal the fundamental insights driving the outstanding performance of the LGA. An ICO wallet is an account designated for storing the proceeds from token

sales, which serve to raise funds for blockchain projects by issuing tokens, as shown on the left side of Fig. 7(a). The exchange account provides asset trading services to users, which is manifested in frequent interactions with customers. It is a hub node with extremely high centrality in the graph, as shown in Fig. 7(b). The mining team finds new blocks through the sharing of computational power resources, with the associated transaction pattern distinctly represented as a cycle structure on the left side of Fig. 7(c). Phishers and hackers are malicious accounts that spread illegal transactions, and in some cases cooperate to trick recipients into making direct remittances in order to obtain Ether, as shown in Fig. 7(d). The LSA module is specifically designed to harness a variety of complex transaction patterns, using graph connectivity encoding to facilitate selective message aggregation and cycle structure awareness. In collaboration with LSA, the GIA module allocates special attention weights to these essential transaction patterns, as shown in the attention map on the right side of Fig. 7.

3) Analysis of Error Classification Examples: In order to explore the limitations of LGA, we collect specific examples for a more detailed analysis of binary classification errors. As shown in Fig. 8, atypical interaction patterns are observed in these error examples. These atypical patterns lack complex structures or multihop neighbors, which limit LGA from exerting its unique advantages. In future work, dynamically adjusting the LGA framework according to the characteristics of the input graph is expected to break the existing limitations of LGA and further optimize the performance.

V. Conclusion and Future Work

We introduced the LGA Framework, which integrates the novel LSA and GIA modules for effective feature extraction and global pattern analysis, respectively. Our empirical tests on real-world Ethereum transaction data clearly demonstrate LGA's superiority over existing baseline methods, thereby confirming its potential to enhance security governance on the Ethereum platform. Importantly, this study serves as a foundational exploration into the de-anonymization space, urging further research into sophisticated evasion tactics by malicious entities, the scalability of large-scale graph analyses, and the real-time application constraints on Ethereum. Furthermore, we posit that the LGA framework holds promise for broader application across various blockchain platforms, contingent upon verification with multiplatform data. Moving forward, we aim to refine our methodologies to develop more efficient and broadly applicable de-anonymization techniques for improving blockchain security.

References

[1] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.

[2] J. Leng et al., "ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 182–192, Jan. 2019.

[3] Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K.-K. R. Choo, "Blockchain-based identity management systems: A review," *J. Netw. Comput. Appl.*, vol. 166, p. 102731, Sep. 2020.

[4] Y. Pang et al., "Slim UNETR: Scale hybrid transformers to efficient 3D medical image segmentation under limited computational resources," *IEEE Trans. Med. Imag.*, vol. 43, no. 3, pp. 994–1005, Mar. 2024.

[5] T. Huang et al., "AdaptFormer: An adaptive hierarchical semantic approach for change detection on remote sensing images," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, Apr. 2024.

[6] B. He et al., "TxPhishScope: Towards detecting and understanding transaction-based phishing on Ethereum," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Copenhagen, Denmark, Nov. 2023, pp. 120–134.

[7] D. Lin, J. Wu, Y. Yu, Q. Fu, Z. Zheng, and C. Yang, "DenseFlow: Spotting cryptocurrency money laundering in Ethereum transaction graphs," in *Proc. World Wide Web Conf. (WWW)*, Singapore, May 2024, pp. 4429–4438.

[8] L. Su et al., "Evil under the sun: Understanding and discovering attacks on Ethereum decentralized applications," in *Proc. 30th USENIX Conf. Secur. Symp.*, Aug. 2021, pp. 1307–1324.

[9] H. Du, Z. Che, M. Shen, L. Zhu, and J. Hu, "Breaking the anonymity of Ethereum mixing services using graph feature learning," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 616–631, 2024.

[10] T.-H. Chang and D. Svetinovic, "Improving bitcoin ownership identification using transaction patterns analysis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 9–20, Sep. 2018.

[11] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the Ethereum blockchain," *Exp. Syst. Appl.*, vol. 150, Jul. 2020, Art. no. 113318.

[12] Y. Pang et al., "Efficient breast lesion segmentation from ultrasound videos across multiple source-limited platforms," *IEEE J. Biomed. Health Inform.*, early access, Feb. 19, 2025, doi: 10.1109/JBHI.2025.3543435.

[13] F. Victor, "Address clustering heuristics for Ethereum," in *Proc. Financ. Cryptogr. Data Secur.*, Kinabalu, Malaysia, Feb. 2020, pp. 617–633.

[14] S. Linoy, N. Stakhanova, and S. Ray, "De-anonymizing Ethereum blockchain smart contracts through code attribution," *Int. J. Netw. Manag.*, vol. 31, no. 1, Aug. 2021, Art. no. e2130.

[15] R. Agarwal, T. Thapliyal, and S. K. Shukla, "Detecting malicious accounts showing adversarial behavior in permissionless blockchains," 2021, *arXiv:2101.11915*.

[16] C. Wang et al., "Demystifying Ethereum account diversity: Observations, models and analysis," *Front. Comput. Sci.*, vol. 16, pp. 1–12, Dec. 2022.

[17] Y. Pang et al., "Sparse-Dyn: Sparse dynamic graph multirepresentation learning via event-based sparse temporal attention network," *Int. J. Intell. Syst.*, vol. 37, no. 11, pp. 8770–8789, Jul. 2022.

[18] Y. Pang et al., "Graph decipher: A transparent dual-attention graph neural network to understand the message-passing mechanism for the node classification," *Int. J. Intell. Syst.*, vol. 37, no. 11, pp. 8747–8769, Jul. 2022.

[19] D. Lin, J. Wu, Q. Yuan, and Z. Zheng, "T-edge: Temporal weighted multidigraph embedding for Ethereum transaction network analysis," *Front. Phys.*, vol. 8, p. 204, Jun. 2020.

[20] J. Shen, J. Zhou, Y. Xie, S. Yu, and Q. Xuan, "Identity inference on blockchain using graph neural network," in *Proc. Int. Conf. Blockchain Trustworthy Syst.*, Aug. 2021, pp. 3–17.

[21] J. Zhou, C. Hu, J. Chi, J. Wu, M. Shen, and Q. Xuan, "Behavior-aware account de-anonymization on Ethereum interaction graph," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3433–3448, 2022.

[22] S. Li, G. Gou, C. Liu, C. Hou, Z. Li, and G. Xiong, "TTAGN: Temporal transaction aggregation graph network for Ethereum phishing scams detection," in *Proc. World Wide Web Conf. (WWW)*, New York, NY, USA, Apr. 2022, pp. 661–669.

[23] S. Hu, Z. Zhang, B. Luo, S. Lu, B. He, and L. Liu, "BERT4ETH: A pre-trained transformer for Ethereum fraud detection," in *Proc. World Wide Web Conf. (WWW)*, Apr. 2023, pp. 2189–2197.

[24] H. Zheng, M. Ma, H. Ma, J. Chen, H. Xiong, and Z. Yang, "TEGDetector: A phishing detector that knows evolving transaction behaviors," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 3, pp. 3988–4000, Jun. 2024.

[25] B. J. Weisfeiler and A. A. Lehman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsiya*, vol. 2, no. 9, pp. 12–16, 1968.

[26] X. Liu, Z. Tang, P. Li, S. Guo, X. Fan, and J. Zhang, "A graph learning based approach for identity inference in DApp platform blockchain," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 438–449, Sep. 2020.

[27] T. Huang, D. Lin, and J. Wu, "Ethereum account classification based on graph convolutional network," *IEEE Trans. Circuits Syst. Exp. Briefs*, vol. 69, no. 5, pp. 2528–2532, May 2022.

[28] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 5453–5462.

[29] Z. Wu, P. Jain, M. Wright, A. Mirhoseini, J. E. Gonzalez, and I. Stoica, "Representing long-range context for graph neural networks with global attention," in *Proc. 35th Conf. Neural. Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 13266–13279.

[30] Y. Pang et al., "Online self-distillation and self-modeling for 3D brain Tumor segmentation," *IEEE J. Biomed. Health Inform.*, early access, Jan. 16, 2025, doi: 10.1109/JBHI.2025.3530715.

[31] Q. Wu et al., "SGFormer: Simplifying and empowering transformers for large-graph representations," in *Proc. 37th Conf. Neural. Inf. Process. Syst.*, vol. 36, Dec. 2024, pp. 64753–64773.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, New York, NY, USA, Aug. 2014, pp. 701–710.

[33] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Aug. 2017, pp. 385–394.

[34] J. Wu et al., "Who are the phishers? Phishing scam detection on Ethereum via network embedding," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 1156–1166, Sep. 2020.

[35] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Aug. 2016, pp. 855–864.

[36] N. K. Ahmed et al., "Learning role-based graph embeddings," 2018, *arXiv:1802.02896*.

[37] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2018, *arXiv:1810.00826*.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[40] L. Yu, F. Zhang, J. Ma, L. Yang, Y. Yang, and W. Jia, "Who are the money launderers? Money laundering detection on blockchain via mutual learning-based graph neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2023, pp. 1–8.

[41] S. Fan, H. Xu, S. Fu, Y. Luo, and M. Xu, "Edge-feature modeling-based topological graph neural networks for phishing scams detection on Ethereum," in *Proc. IEEE/ACM 32nd Int. Symp. Qual. Service (IWQoS)*, 2024, pp. 1–10.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[43] D. Lin, J. Wu, Q. Yuan, and Z. Zheng, "Modeling and understanding Ethereum transaction records via a complex network approach," *IEEE Trans. Circuits Syst. Exp. Briefs*, vol. 67, no. 11, pp. 2737–2741, Jan. 2020.

[44] F. Béres, I. A. Seres, A. A. Benczúr, and M. Quintyne-Collins, "Blockchain is watching you: Profiling and deanonymizing Ethereum users," in *Proc. IEEE Int. Conf. Decent. Appl. Infrastruct. (DAPPS)*, Aug. 2021, pp. 69–78.

[45] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, and Z. Zheng, "Phishing scams detection in Ethereum transaction network," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–16, Dec. 2020.

[46] D. Lin, J. Wu, T. Huang, K. Lin, and Z. Zheng, "Who is who on Ethereum? Account labeling using heterophilic graph convolutional network," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 3, pp. 1541–1553, Mar. 2024.

[47] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 7134–7143.

[48] J. Zhang, H. Zhang, C. Xia, and L. Sun, "Graph-bert: Only attention is needed for learning graph representations," 2020, *arXiv:2001.05140*.

[49] P. Li, Y. Wang, H. Wang, and J. Leskovec, "Distance encoding: Design provably more powerful neural networks for graph representation learning," in *Proc. 34th Conf. Neural. Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 4465–4478.

[50] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "XBlock-ETH: Extracting and exploring blockchain data from Ethereum," *Open J. Comput. Soc.*, vol. 1, pp. 95–106, 2020.

[51] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 3538–3545.

[52] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, Feb. 2020, pp. 3438–3445.

[53] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, Nov. 2008.