Association-Focused Path Aggregation for Graph Fraud Detection

Tian Qiu [1,2] **Wenda Li** [1,2] **Zunlei Feng** [1,2] **Jie Lei** [3] **Tao Wang** [4*]
Yi Gao [1] **Mingli Song** [1,2] **Yang Gao** [1,2*]

[1] State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2] Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[3] College of Computer Science, Zhejiang University of Technology
[4] Department of Planning, Ministry of Emergency Management Big Data Center

{tqiu,lwdup,zunleifeng,gaoyi,songml,roygao}@zju.edu.cn
jasonlei@zjut.edu.cn
hbliu@qdu.edu.cn

Abstract

Fraudulent activities have caused substantial negative social impacts and are exhibiting emerging characteristics such as intelligence and industrialization, posing challenges of high-order interactions, intricate dependencies, and the sparse yet concealed nature of fraudulent entities. Existing graph fraud detectors are limited by their narrow "receptive fields", as they focus only on the relations between an entity and its neighbors while neglecting longer-range structural associations hidden between entities. To address this issue, we propose a novel fraud detector based on Graph Path Aggregation (GPA). It operates through variable-length path sampling, semantic-associated path encoding, path interaction and aggregation, and aggregation-enhanced fraud detection. To further facilitate interpretable association analysis, we synthesize G-Internet, the first benchmark dataset in the field of internet fraud detection. Extensive experiments across datasets in multiple fraud scenarios demonstrate that the proposed GPA outperforms mainstream fraud detectors by up to +15% in Average Precision (AP). Additionally, GPA exhibits enhanced robustness to noisy labels and provides excellent interpretability by uncovering implicit fraudulent patterns across broader contexts. Code is available at https://github.com/horrible-dong/GPA.

1 Introduction

With the evolution of information technology, there is a pronounced spillover of risks from cyberspace into human society and the physical world. The sharp rise in fraudulent activities has made fraud detection [1–4] an increasingly prominent research area. Fraud encompasses various domains such as the internet, finance, social networks, and online reviews, bringing severe negative social impacts including economic losses, trust damage, disruption of fair competition, and infringement of consumer rights. Therefore, combating fraud and safeguarding public interests are of utmost importance.

As fraudulent activities become increasingly intelligent and industrialized, the main challenges in fraud detection include: (1) high-order interactions often exist between fraudulent entities, (2) dependencies among fraudulent entities are intricate, and (3) fraudulent entities are sparse and highly concealed. Given the complex dependencies and topological structures among fraud entities, researchers have employed Graph Neural Networks (GNNs) [5–7] for fraud detection. However,

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

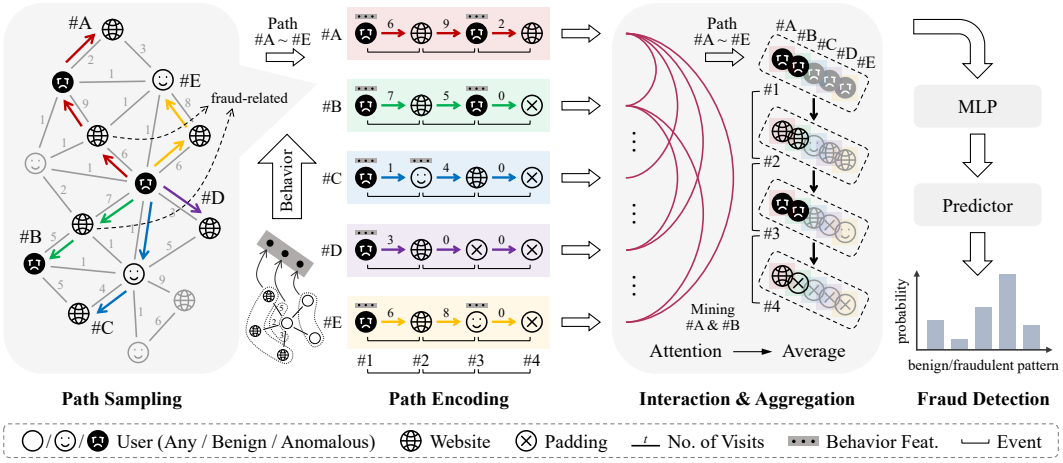


Figure 1: An illustration of the proposed fraud detector based on Graph Path Aggregation (GPA) under the internet fraud scenario, which operates through variable-length path sampling, semantic-associated path encoding, path interaction and aggregation, and aggregation-enhanced fraud detection. Note that the proposed method is generally applicable to other fraud detection scenarios.

standard GNNs such as GCN [8] and GAT [9] are local-aware, unable to process graph structures with long-range dependencies as they primarily focus on knowledge from one-hop neighbors with limited utilization of global contexts. Graph Transformers [10–12] can capture global contexts. However, they include all nodes for self-attention, facing computational challenges in large-scale fraud detection graphs while overlooking the inherent structural associations among fraudulent entities.

In recent years, significant advancements have been made in graph-based fraud detection techniques, which can be broadly categorized into spectral methods and spatial methods. Spectral methods emphasize leveraging different frequency components of graph signals to enhance the ability to capture and process fraudulent clues on graphs through strategies such as spectral energy distribution analysis [13], label-aware edge indicators [14], and multi-frequency signal combinations [15, 16]. Spatial methods focus on spatial relations among nodes, employing techniques like context and relation fusion [17], class balance and sampling optimization [18], reinforced relation-aware neighbor selection [19], homogeneous and heterogeneous connection management [20], anomalous feature recognition and constraint [21], implicit fairing-inspired layer-wise propagation rule [22], and low-variance relation generating [23] to improve GNNs' ability to detect and identify fraudulent activities. Although these methods demonstrate excellent performance, their "receptive fields" remain limited, as they primarily rely on information from low-hop neighbors. For more concealed and complex fraud that involves long-range structural associations, they often struggle to respond effectively.

To address the aforementioned challenges in graph fraud detection, it is imperative to develop a global-aware technique that can effectively capture long-range structural dependencies in graphs to unveil hidden fraudulent patterns within massive data. Toward this goal, we propose a novel fraud detector based on Graph Path Aggregation (GPA), as illustrated in Figure 1. GPA integrates variable-length path sampling, semantic-associated path encoding, path interaction and aggregation, and aggregation-enhanced fraud detection to effectively deal with the high-order interactions, intricate dependencies, and the sparse yet concealed nature of fraudulent entities. This approach overcomes the limitation of mainstream fraud detectors in handling graph structures with long-range associations.

Due to privacy restrictions, existing fraud datasets have been anonymized, making it impossible to discern the specific meanings of entities, thereby hindering the analysis of model interpretability. To further facilitate this line of research, we focus on the currently prevalent yet underexplored field of internet fraud detection. Based on established fraud rules, we synthesize the first internet fraud dataset, G-Internet, to support research on interpretable association analysis. Extensive experiments across datasets in multiple fraud scenarios encompassing the internet, finance, social networks, and online reviews demonstrate that the proposed GPA outperforms various mainstream fraud detectors in Area Under the Curve (AUC) and Average Precision (AP). Meanwhile, GPA exhibits stronger robustness to noisy labels. It also provides excellent interpretability that can uncover common patterns

within fraud-related paths through global pattern interaction and similarity computation, showcasing a more comprehensive view of the associations among diverse fraudulent entities.

The main contribution of this paper is the proposal of a novel fraud detector based on Graph Path Aggregation (GPA), which goes beyond the local view of neighbors and explores hidden and complex fraud associations at the path level. To enrich research on interpretable associations, we synthesize the first internet fraud dataset named G-Internet. Experimental results demonstrate that GPA exhibits superior performance compared to mainstream fraud detectors across various fraud scenarios. Furthermore, GPA presents enhanced robustness to noisy labels and excellent interpretability, providing new insights for unveiling global association patterns in fraudulent activities.

2 Related Work

2.1 Graph Neural Networks

Given the topological structures and intricate dependencies among entities, researchers introduced Graph Neural Networks (GNNs) [5–7] as a powerful tool for processing unstructured data. Built upon message-passing, GNNs have been widely employed for various downstream tasks [24–26].

GCN [8] is among the earliest classical GNNs, which updates node representations by aggregating information from neighbor nodes through graph convolution. GAT [9] introduces attention mechanisms [27] to learn interaction weights between neighbor nodes. Their common drawback is that they only utilize knowledge from one-hop neighbors, making them inadequate for handling graph structures with long-range dependencies. Graph Transformers [10–12] are designed to capture global context. However, they perform self-attention across all nodes, which is not only computationally infeasible for large-scale graphs in fraud detection but also neglects the structural associations inherent among fraudulent entities. While some general approaches like PathNet [28] and RAW-GNN [29] claim to incorporate path-level structures, their core design is in fact to address heterophily issues. Not specifically resolving long-range dependency challenges, they employ short path lengths (2-7 hops) that essentially maintain a local-aware framework. Additionally, these methods suffer from slow encoding, lack path interaction, and are unable to uncover implicit fraud patterns.

2.2 Graph Fraud Detectors

Traditional graph fraud detectors [30–32] primarily rely on structural information to assess the node's contribution, commonly utilizing metrics such as centrality, PageRank [33], and HITS [34]. These methods are effective in identifying behaviors similar to known fraudulent patterns, but exhibit limitations in detecting unknown types of fraud. To address this issue, graph representation learning like DeepWalk [35], LINE [36], and node2vec [37] has emerged as a promising solution.

In recent years, advanced graph fraud detectors have been mainly categorized into spectral methods and spatial methods. Regarding spectral methods, AMNet [15] adaptively combines multiple frequency signals to capture anomalies across different bands. BWGNN [13] employs Beta graph wavelets for band-pass filtering in spectral and spatial domains. GHRN [14] addresses over-smoothing [38] by calculating similarities with a label-aware edge indicator and pruning heterogeneous edges based on this. AHFAN [16] builds a semantic fusion module based on Chebyshev polynomial filtering to capture high- and low-frequency components of graph signals. CARE-GNN [39] leverages context embeddings, neighborhood metrics, and relation attention to mitigate fraud inconsistencies. PC-GNN [18] employs a label-balanced sampler and a neighborhood sampler to resolve feature dilution from class imbalance. RioGNN [19] combines reinforced relation-aware neighbor selection with label-aware similarity measures for more effective representation learning. H^2-FDetector [20] tackles fraudsters hiding in graphs by incorporating both homophilic and heterophilic connections, along with a novel aggregation strategy guided by prototype priors to propagate similar and distinct signals. GDN [21] identifies key anomalous features to mitigate the influence of heterogeneous neighbors and uses prototype vectors to manage the distribution of anomalous features. GFCN [22] introduces a layer-wise propagation rule motivated by the concept of implicit fairing in geometry processing. HedGe [23] emphasizes reducing the high homophily variance between categories by generating new low-variance relations rather than modifying the original ones. Due to reliance on low-hop neighbors, these methods are also local-aware. Therefore, there remains an absence of methods that are both global-aware and capable of effectively uncovering long-range structural fraudulent patterns.

3 Methodology

To address the challenges posed by high-order interactions, intricate dependencies, and the sparse yet concealed nature of fraudulent entities, along with the limitations of GNN-based anti-fraud methods in dealing with long-range dependencies in graph structures, we propose a novel fraud detector based on Graph Path Aggregation (GPA), as illustrated in Figure 1. GPA works through variable-length path sampling, semantic-associated path encoding, path interaction and aggregation, and finally the aggregation-enhanced fraud detection. To facilitate an intuitive understanding, the methodology is presented in the context of internet fraud detection. Note that the proposed method is generally applicable to other fraud detection scenarios.

3.1 Problem Definition

Let $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ be a graph in the context of an internet fraud scenario. Let $\mathbb{V} = \{v_1, v_2, ..., v_N\}$, $\mathbb{E} = \{e_1, e_2, ..., e_M\}$, $\mathbb{X} = \{\mathbf{x}_{v_1}, \mathbf{x}_{v_2}, ..., \mathbf{x}_{v_N}\}$, and $\mathbb{T} = \{t_{e_1}, t_{e_2}, ..., t_{e_M}\}$ denote the set of nodes, edges, node features, and edge weights. The nodes include two types: users and websites. $v_n \in \mathbb{N}$ is the node index, among which $\mathbb{V}^{\text{user}} = \{v_1, v_2, ..., v_U\}$ is the set of user nodes and $\mathbb{V}^{\text{web}} = \{v_{U+1}, v_{U+2}, ..., v_N\}$ is the set of website nodes. $\mathbf{x}_{v_n} \in \mathbb{R}^D$ is the feature of node v_n, where D is the dimension. $e_m \in \mathbb{N}$ is the edge index, which includes two types of bidirectional relations: "user visits website" and "user to user". $t_{e_m} \in \mathbb{N}^+$ represents the number of visits within the edge e_m. $\mathbb{Y}^{\text{user}} = \{y_{v_1}, y_{v_2}, ..., y_{v_U}\}$ is the set of labels for the user nodes. The essence of graph-based internet fraud detection is a binary classification task for user nodes, $y_{v_n \in \mathbb{V}^{\text{user}}} \in \{0, 1\}$, signifying the user node v_n is benign or anomalous. Additionally, in alignment with realistic settings, the types and labels of website nodes are treated as unknown, leaving the model to mine them.

3.2 Variable-Length Path Sampling

For a given user node, starting from itself, sample I paths $\mathbf{R} \in \mathbb{N}^{I \times J}$ through random walk. Each path $\mathbf{R}_i \in \mathbb{N}^J$ starts from this user node and contains J nodes (of any type) connected in sequence. To further enrich the diversity of path lengths, the sampled paths are randomly masked. Define a binary matrix $\mathbf{M} \in \mathbb{1}^{J \times J}$, $\mathbf{M} = \mathbf{U} - \mathbf{I}$, where $\mathbf{U} \in \mathbb{1}^{J \times J}$ is an upper triangular matrix filled with ones, and $\mathbf{I} \in \mathbb{1}^{J \times J}$ is an identity matrix. Randomly and repeatably sample I rows from $\mathbf{M}$ and get $\tilde{\mathbf{M}} \in \mathbb{1}^{I \times J}$. Then, obtain variable-length paths $\tilde{\mathbf{R}} \in \mathbb{R}^{I \times J}$ as follows:

$$\tilde{\mathbf{R}}_{i,j} = \begin{cases} \mathbf{R}_{i,j}\,, & if\ \ \tilde{\mathbf{M}}_{i,j} = 0 \\ -1\,, & if\ \ \tilde{\mathbf{M}}_{i,j} = 1 \end{cases}. \tag{1}$$

To enhance the quality of the sampled paths, an additional selection is sometimes necessary. The importance of each path is evaluated by summing the degree centrality of its nodes. From the randomly sampled I paths, the top-K most valuable paths $\tilde{\mathbf{R}} \in \mathbb{R}^{K \times J}$ are selected as follows:

$$\tilde{\mathbf{R}} \in \mathbb{R}^{K \times J} \leftarrow \text{Top-}K^{\downarrow \text{degree}}(\tilde{\mathbf{R}} \in \mathbb{R}^{I \times J}). \tag{2}$$

3.3 Semantic-Associated Path Encoding

In order to facilitate subsequent path interaction, the sampled variable-length paths are uniformly encoded into fixed-length embeddings based on atomic events. Additionally, a vector of behavior feature is designed based on the global behavior of the node and integrated into the path encoding to enrich the feature representation.

For the k-th path $\tilde{\mathbf{R}}_k$, the set of node features it contains is $\{\mathbf{x}_{\tilde{\mathbf{R}}_{k,j}}\}_{j=1}^{J}$. The variable-length paths sampled in §3.2 are padded at the tail with nodes indexed as -1, and the feature $\mathbf{x}$ corresponding to the "-1" node is filled with zeros. Each link in the dataset can be viewed as an event. Taking the link "user visits website" as an example, this event consists of three elements: user (node), website (node), and the number of visits (edge weight). To produce the representation for this event, operate by adding the user feature with the product of the website feature and the number of visits. For all events in the path, the same operation is applied successively, and then get the path feature $\breve{\mathbf{h}}_k \in \mathbb{R}^{DJ}$:

$$\breve{\mathbf{h}}_k = \bigcup \{\mathbf{x}_{\tilde{\mathbf{R}}_{k,j}} + t_{\tilde{\mathbf{R}}_{k,j} \to \tilde{\mathbf{R}}_{k,j+1}} \cdot \mathbf{x}_{\tilde{\mathbf{R}}_{k,j+1}}\}_{j=1}^{J}, \tag{3}$$

where $\bigcup$ denotes the multi-element "concatenate" operation, $\tilde{\mathbf{R}}_{k,j} \rightarrow \tilde{\mathbf{R}}_{k,j+1}$ is the edge between nodes $\tilde{\mathbf{R}}_{k,j}$ and $\tilde{\mathbf{R}}_{k,j+1}$, and $\mathbf{x}_{\tilde{\mathbf{R}}_{k,J+1}}$ is a padding feature filled with zeros. The length of the path feature after concatenating is $D \cdot J$. For other datasets with the meaning of relations/events unknown, just simply concatenate the node features along the path and get $\breve{\mathbf{h}}_k = \bigcup\{\mathbf{x}_{\tilde{\mathbf{R}}_{k,j}}\}_{j=1}^{J}$. After that, a linear projection is applied to the path feature to obtain the path embedding $\mathbf{h}_k \in \mathbb{R}^C$:

$$\mathbf{h}_k = \breve{\mathbf{h}}_k \cdot \mathbf{W} + \mathbf{b}, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{DJ \times C}$ and $\mathbf{b} \in \mathbb{R}^C$ are learnable weight and bias respectively.

For a further enhanced path feature, the behavior feature $\mathbf{x}^{\text{behav}}$ for each node is introduced. For user nodes, the behavior feature includes the number of visited websites, the total number of website visits, and the number of connected users. For website nodes, no behavior feature is set, and instead, a zero vector with the same length as the user behavior feature is used. The variable-length paths sampled in §3.2 are padded at the tail with nodes indexed as -1, and the behavior feature $\mathbf{x}^{\text{behav}}$ corresponding to the "-1" node is filled with zeros. The behavior features of all nodes are denoted as $\{\mathbf{x}^{\text{behav}}_{\tilde{\mathbf{R}}_{k,j}}\}_{j=1}^{J}$, where $\mathbf{x}^{\text{behav}}_{\tilde{\mathbf{R}}_{k,j}} \in \mathbb{R}^S$ and S is the length of the feature. Then, normalize the behavior features. Next, string together the behavior features for all nodes within a path as follows and get $\breve{\mathbf{h}}^{\text{behav}}_k \in \mathbb{R}^{2SJ}$:

$$\breve{\mathbf{h}}^{\text{behav}}_k = \bigcup\{\mathbf{x}^{\text{behav}}_{\tilde{\mathbf{R}}_{k,j}} \cup \mathbf{x}^{\text{behav}}_{\tilde{\mathbf{R}}_{k,j+1}}\}_{j=1}^{J}, \tag{5}$$

where $\cup$ and $\bigcup$ denote two-element and multi-element "concatenate" operations respectively, and $\mathbf{x}^{\text{behav}}_{\tilde{\mathbf{R}}_{k,J+1}}$ is a padding behavior feature filled with zeros. The length of the behavior feature after concatenating is $2S \cdot J$. For different scenarios, unique behavior feature can be designed on a case-by-case basis. After that, a linear projection is applied to the concatenated behavior feature to obtain the behavior embedding $\mathbf{h}^{\text{behav}}_k \in \mathbb{R}^C$:

$$\mathbf{h}^{\text{behav}}_k = \breve{\mathbf{h}}^{\text{behav}}_k \cdot \mathbf{W}^{\text{behav}} + \mathbf{b}^{\text{behav}}, \tag{6}$$

where $\mathbf{W}^{\text{behav}} \in \mathbb{R}^{2SJ \times C}$ and $\mathbf{b}^{\text{behav}} \in \mathbb{R}^C$ are learnable weight and bias respectively.

Finally, integrate the behavior embedding into the original path embedding and get $\mathbf{z}_k \in \mathbb{R}^C$:

$$\mathbf{z}_k = \mathbf{h}_k + \mathbf{h}^{\text{behav}}_k. \tag{7}$$

3.4 Path Interaction and Aggregation

Interaction will be conducted between the paths of a user node to uncover common fraudulent patterns. For a given user node, stack all the path embeddings starting from this node and get $\mathbf{Z} \in \mathbb{R}^{K \times C}$:

$$\mathbf{Z} = [\mathbf{z}_1;\ \mathbf{z}_2;\ ...;\ \mathbf{z}_K]. \tag{8}$$

Then, self-attention between paths is employed to produce the updated embeddings $\hat{\mathbf{Z}} \in \mathbb{R}^{K \times C}$:

$$\mathbf{Q} = \mathbf{Z} \cdot \mathbf{W}^{\text{query}}, \mathbf{K} = \mathbf{Z} \cdot \mathbf{W}^{\text{key}}, \mathbf{V} = \mathbf{Z} \cdot \mathbf{W}^{\text{value}}, \tag{9}$$

$$\hat{\mathbf{Z}} = \text{softmax}(\mathbf{Q} \cdot \mathbf{K}^{\text{T}}) \cdot \mathbf{V}, \tag{10}$$

where $\mathbf{W}^{\{\text{query, key, value}\}} \in \mathbb{R}^{C \times C}$ are learnable weights. $\text{softmax}(\mathbf{Q} \cdot \mathbf{K}^{\text{T}})$ can yield the attention scores of each path to itself and to other paths. With increasing training data, the model can identify fraudulent and benign path patterns more clearly. Additionally, compared to local-aware models such as GCN and GAT, the "receptive field" of path interaction is much larger, making it more suitable for detecting fraudulent associations that are concealed through all sorts of means.

After path interaction, path aggregation is performed. Unlike traditional GNNs that aggregate features from neighbor nodes, the proposed GPA aggregates features from high-order paths. Specifically, aggregate the embeddings $\hat{\mathbf{Z}} \in \mathbb{R}^{K \times C}$ of all paths as follows to produce $\hat{\mathbf{z}} \in \mathbb{R}^C$:

$$\hat{\mathbf{z}} = \bigcap\{\hat{\mathbf{Z}}_k\}_{k=1}^{K}, \tag{11}$$

where $\bigcap$ denotes the multi-element "aggregate" operation. In this paper, the average aggregation is adopted, and the aggregated embedding $\hat{\mathbf{z}}$ serves as the updated embedding of the central user node.

3.5 Aggregation-Enhanced Fraud Detection

The updated user embedding $\hat{\mathbf{z}}$ from path aggregation is fed into a Multi-Layer Perceptron (MLP) and subsequently a sigmoid predictor to yield a prediction $p \in [0, 1]$:

$$p = \text{sigmoid}(\text{MLP}(\hat{\mathbf{z}})), \tag{12}$$

representing the probability that the central user node is predicted to be anomalous. Then, together with the label y computes the binary cross-entropy loss $\mathcal{L}$:

$$\mathcal{L} = y \cdot \log p + (1 - y) \cdot \log(1 - p). \tag{13}$$

Finally, optimize the model through standard backward propagation and gradient descent.

4 Dataset Construction

Existing fraud datasets are typically anonymized, which obscures the specific meanings of entities and consequently hinders interpretability analysis. Moreover, emerging internet fraud is rapidly evolving and widespread with concealed tactics, and research on internet fraud detection remains limited. Considering these gaps, we construct G-Internet, the first benchmark dataset for internet fraud detection, featuring a transparent structure. Due to privacy restrictions, real-world data is not available. Therefore, we synthesize the dataset through simulation. The anomaly rate in G-Internet is only 2.96%, the lowest among existing fraud datasets. This section provides an overview of the construction process for G-Internet, with more detailed descriptions available in Appendix §A.1.

Fraud rules. Types of internet fraud are varied, and the user behaviors regarding website visits exhibit specific patterns within each type. To more accurately simulate anomalous activities, we have designed 12 common rules as shown in Table 5 & 6, which are based on real cases and account for the diversity and complexity of user behaviors.

User nodes. A total of 9 attributes have been generated for users. Based on existing references, we determine the attribute distribution for each user category and sample the attribute accordingly. Since our primary focus is to detect fraud based on the behavior of users visiting websites rather than on the users' inherent characteristics, we intentionally minimize the distinction between benign and anomalous users when constructing user attributes.

Website nodes. In total, we have included 11 types of fraud-related websites and 27 types of normal websites in the dataset. We employ large language models to generate keywords for each website, then derive the corresponding website description from these keywords, and finally encode the description into the website feature.

"User visits website" relation. For an anomalous user, randomly select one of the designed fraud rules and generate connections between the user and websites that meet the rule. Anomalous users may also visit a large number of normal websites while visiting the websites involved in the fraud rules. These connections are also taken into account and randomly generated. For benign users, they visit more normal websites, but they may also visit fraud-related websites as long as they do not meet any of the fraud rules.

"User to user" relation. There may be relations between users, such as phone calls or other interactions. Since our primary focus is to detect fraud based on the behavior of users visiting websites rather than on the connections between users, we set both benign users and anomalous users to maintain a closer connection with benign users, so as to minimize the difference in communication behaviors between these two types of users.

In summary, the constructed G-Internet aligns with practical situations and has a clear and transparent structure. This interpretable dataset facilitates the demonstration of the interpretable GPA method (§3), which in turn enables the interpretable association analysis of fraud-related entities (§5.4).

5 Experimental Study

Datasets. We adopt our released G-Internet dataset for internet fraud detection. Additionally, we adopt the well-known Elliptic [40] and T-Finance [13] datasets for finance fraud detection, T-Social [13] dataset for social networks fraud detection, YelpChi [41] and Amazon [42] datasets for online

reviews fraud detection. Table 1 provides a brief overview of the datasets' statistics. For datasets except Elliptic, the split ratio for training, validation, and testing is 4:2:4. Detailed dataset descriptions are provided in Appendix §A.2.

Table 1: A brief overview of dataset statistics.

Scenario	Dataset	#Nodes	#Edges	#Feats.	Anomaly
Internet	G-Internet	160655	1972292	106	2.96%
Finance	Elliptic	46564	73248	93	9.76%
	T-Finance	39357	42445086	10	4.58%
Social	T-Social	5781065	146211016	10	3.01%
Reviews	YelpChi	45954	7693958	32	14.53%
	Amazon	11944	8796784	25	6.87%

Compared methods. The proposed GPA is compared with a range of methods including *Non-GNNs*: MLP [43] and KNN [44], *Standard GNNs*: GCN [8], GraphSAGE [45], and GAT [9], *Spectral GNNs*: BernNet [46], AMNet [15], BWGNN [13], GHRN [14], and AHFAN [16], and *Spatial GNNs*: GAS [47], DCI [48], CARE-GNN [39], PC-GNN [18], RioGNN [19], H^2-FDetector [20], GDN [21], GFCN [22], and HedGe [23].

Experimental settings. The experiments for the proposed GPA are conducted using the AdamW optimizer with an initial learning rate of 1e-4 or 1e-3 and a weight decay of 5e-4. It adopts minibatch sampling of user nodes per iteration, training for a maximum of 200 epochs from scratch. The baseline models use the officially recommended settings. All numerical results are the averages across 10 different random seeds. Detailed GPA model settings can be found in Appendix §A.3.

Evaluation metrics. Due to the significant positive-negative imbalance in fraud datasets, evaluation metrics should take both Precision and Recall into account. In this paper, Area Under the Curve (AUC, %) and Average Precision (AP, %) are adopted to assess the model performance. For more details on the metrics, please refer to Appendix §A.4.

5.1 Performance Comparison

The proposed GPA is comprehensively compared with non-, standard, spectral, and spatial GNNs in the internet, finance, social networks, and online reviews fraud detection scenarios. As shown in Table 2, the proposed GPA performs excellently and is the most stable across all datasets, achieving significant AP improvements of +15.7%, +3.4%, and +9.2% on the challenging G-Internet, Elliptic, and T-Social datasets. Other methods exhibit high instability and often perform poorly on several of these datasets. Standard GNNs, while able to capture some graph structural features, struggle when handling higher-order interactions. Spectral and spatial methods, designed specifically for fraud detection, perform better but still only consider low-hop neighbors, resulting in a limited "receptive field". Moreover, these methods generally perform poorly on G-Internet. The possible reason is that the inherent features of benign and anomalous users in G-Internet are not distinguishable. Without deeper mining of associations within user behaviors, the detection effect will be severely weakened. In comparison, GPA has broader perceptions and excels in identifying more complex and hidden fraudulent patterns, thus outperforming existing techniques.

5.2 Robustness to Noisy Labels

Mislabeled data has a detrimental impact on model performance [49, 50]. In real-world scenarios, due to the hidden nature of anomalies, it is often challenging to label anomalous nodes accurately, resulting in the widespread presence of noisy labels, particularly anomalous labels that are incorrectly marked as benign labels. Table 3 presents the impact of noisy labels on model performance. GAT, which also employs an attention mechanism, struggles to resist label noise. The explanation is that GAT focuses only on neighbor nodes and therefore lacks a global view of fraudulent patterns, making it more prone to noise. In contrast, under both asymmetric and symmetric label noise, the proposed GPA shows stronger robustness compared to other methods. This may be attributed to the implicit associations in paths, which capture long-range interactions among nodes, allowing the extraction of rich benign and fraudulent patterns even from a limited set of accurately labeled data. This global view enables the model to more comprehensively understand the underlying logic behind node behaviors, thereby mitigating the interference caused by noisy labels to some extent.

5.3 Ablation Study

This section conducts ablation study of the proposed GPA. As presented in Table 4, each component of GPA can contribute to enhancing the model, and any combination of the components involved can further enhance performance, with the optimal achieved when all components are used.

Table 2: Model performance comparison in various fraud scenarios. "/" denotes "out of memory".

Method	Scenario	Internet		Finance				Social		Reviews			
	Dataset	G-Internet		Elliptic		T-Finance		T-Social		YelpChi		Amazon	
	Metric	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
Non-GNN	MLP	61.0	4.7	88.3	43.8	92.2	74.2	73.1	9.7	81.6	47.7	96.9	87.3
	KNN	51.7	3.0	88.0	61.0	92.7	75.0	77.6	36.3	84.6	54.4	94.9	84.4
Standard	GCN	98.5	81.9	81.7	25.4	94.6	78.2	96.6	76.4	58.6	20.9	85.2	45.7
	GraphSAGE	98.6	78.9	87.6	57.8	95.6	84.7	95.7	75.3	82.9	46.6	90.9	82.5
	GAT	94.2	57.9	86.3	27.5	95.8	82.7	90.3	32.1	79.1	43.6	97.1	87.9
Spectral	BernNet	91.5	57.6	87.5	38.3	96.7	89.2	93.7	44.3	83.5	51.9	95.8	84.9
	AMNet	90.8	56.0	88.9	69.5	96.4	88.9	92.5	37.7	81.9	46.9	96.9	88.4
	BWGNN	95.8	72.7	89.6	48.4	96.9	89.4	96.9	78.9	87.1	61.5	98.3	91.5
	GHRN	94.3	66.1	90.0	55.2	96.5	87.6	97.1	86.8	84.6	55.4	**98.3**	89.5
	AHFAN	87.6	72.8	87.5	72.7	86.8	73.3	87.1	73.5	86.8	72.7	98.2	88.2
Spatial	GAS	96.7	72.9	86.7	29.8	96.5	86.0	95.0	62.4	76.4	35.1	92.7	81.4
	DCI	82.4	23.6	85.7	27.4	87.9	63.7	84.0	13.0	78.4	39.9	95.4	85.2
	CARE-GNN	70.2	15.5	87.8	37.2	90.0	61.8	78.3	41.2	84.0	53.0	97.0	85.6
	PC-GNN	79.6	22.4	86.5	42.7	94.0	83.3	96.9	80.3	80.8	44.5	98.0	89.3
	RioGNN	72.8	15.1	86.4	29.1	91.3	62.6	81.7	17.6	85.7	56.5	96.9	87.6
	H^2-FDetector	83.7	27.9	63.2	10.5	/	/	/	/	89.9	57.5	96.1	84.9
	GDN	90.7	45.7	88.7	65.2	95.8	85.7	88.4	52.3	90.4	67.4	97.3	86.8
	GFCN	85.9	43.4	85.8	45.5	92.6	82.1	86.1	47.4	87.6	68.4	95.5	83.8
	HedGe	91.2	49.7	88.9	64.5	96.5	89.0	96.8	83.7	91.3	70.7	**98.3**	92.3
Path	GPA	**99.8**	**97.6**	**91.3**	**76.1**	**97.3**	**89.6**	**99.6**	**96.0**	**91.8**	**73.7**	98.1	**92.5**

Table 3: Impact of noisy labels on model performance. The dataset used for evaluation is G-Internet. "a→b" denotes the proportion of anomalous labels that are incorrectly marked as benign labels, and "b→a" represents the proportion of benign labels that are incorrectly marked as anomalous labels.

a→b b→a		0% 0%	10% 0%	20% 0%	40% 0%	80% 0%	0% 10%	0% 20%	0% 40%	0% 80%	10% 10%	20% 20%	40% 40%	80% 80%
GAT	AUC	94.2	94.0	92.5	89.9	86.1	84.3	81.0	73.0	63.6	83.1	76.5	56.1	51.3
	AP	57.9	56.6	54.4	44.4	23.7	33.9	27.4	14.4	5.5	32.8	19.5	3.6	3.1
BWGNN	AUC	95.8	95.0	94.7	93.3	88.0	91.8	86.3	82.8	73.6	90.2	86.9	70.1	42.3
	AP	72.7	71.8	67.2	64.1	42.9	58.7	42.9	24.7	10.3	53.3	31.0	8.4	2.5
GPA	AUC	99.8	99.7	99.6	99.3	98.3	98.4	97.4	96.8	79.4	98.7	97.3	78.0	52.6
	AP	97.6	96.5	95.2	92.9	83.8	93.8	89.3	82.2	12.6	93.3	82.6	10.8	3.2

Table 4: Ablation study of the proposed GPA on G-Internet.

Variable-Length Paths	✓	×	✓	✓	✓	✓	×	×	×	×
Path Selection	✓	✓	×	✓	✓	×	✓	×	×	×
Behavior Encoding	✓	✓	✓	×	✓	×	×	✓	×	×
Path Interaction	✓	✓	✓	✓	×	✓	✓	✓	✓	×
AUC	**99.83**	99.79	99.75	99.26	86.63	99.04	99.17	99.75	98.66	86.48
AP	**97.62**	97.17	97.06	90.41	34.15	88.81	89.84	96.94	88.17	34.59

5.4 Interpretability

Firstly, we use the constructed G-Internet dataset to explore the associations between users falling victim to internet fraud and their website-visiting behaviors, as well as their internet environments. The paths starting from a user interact, and the attention scores between paths are computed through Equation 10. The attention maps are shown in Figure 2.

(1) Whether the path length is set short or long, the proposed GPA maintains clear interpretability. As shown in Figure 2(a)(c), when the central user exhibits anomalous status, some paths consistently attract attention during interactions. These paths typically pass through fraud-related websites visited by the user, whereas paths through normal websites receive little to no attention. This indicates that the attention mechanism can highlight the contribution of visited websites to the fraud incident. Besides, in Figure 2(b)(d), when the central user is benign, paths through normal websites receive more attention, which sharply contrasts with the results in Figure 2(a)(c) and aligns with the expected interpretability. Moreover, since the construction of G-Internet does not impose strong regularity on user connections (see §4), the proposed GPA, as expected, pays more attention to users' website-visiting behaviors rather than the connections between users, which is also interpretable.

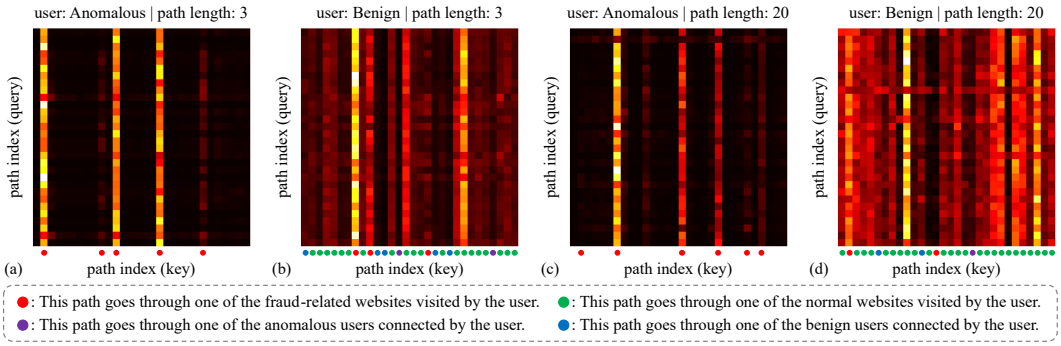


Figure 2: Attention map between all paths starting from a user in G-Internet during path interaction. The brighter the area, the higher the attention score.

(2) In Figure 2(c), some paths through fraud-related websites visited by the user also receive less attention. This is because, in longer paths, the model evaluates the contribution of user-visited websites to the fraud incident within a broader context, allowing it to filter out less relevant fraud-related websites and uncover other implicit patterns at higher levels. Consequently, Figure 2(c)(d) exhibit improved detection performance compared to Figure 2(a)(b).

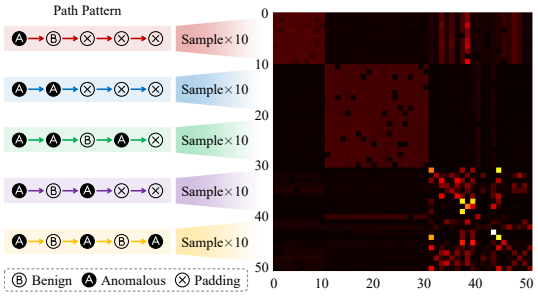


Figure 3: Attention map between paths from five patterns in T-Social during path interaction. The brighter the area, the higher the attention score.

Next, as shown in Figure 3, to investigate the potential associations between path patterns, we use the T-Social dataset and select five types of path patterns. Ten paths are sampled from each pattern, totaling 50 paths. After encoding, these paths interact, and the attention scores between paths are computed through Equation 10. It can be observed from the attention map that there is a close association between all paths in the same path pattern, and there may be common associations between similar path patterns. In the following discussions, "B" denotes "Benign node" and "A" denotes "Anomalous node".

(1) "A→A" and "A→A→B→A" are highly correlated, probably because both involve continuous anomalous nodes, indicating that some frauds tend to be clustered or continuous.

(2) "A→B→A" and "A→B→A→B→A" are highly correlated, probably because both show the strategy of fraud spreading or hiding through benign nodes. "A→B→A" indicates that anomalous nodes may spread fraud with the help of benign nodes to increase concealment. "A→B→A→B→A" reveals that more covert fraudulent patterns may be produced by bridging multiple benign nodes.

(3) "A→B" is highly correlated with "A→B→A" and "A→B→A→B→A", probably because "A→B" represents the initial of fraud and is the basis of more complex patterns. When analyzing more complex patterns, attention will be paid to whether they are developed from "A→B".

(4) "A→B→A" and "A→B→A→B→A" are almost irrelevant to "A→B", probably because they contain context that has gone beyond the scope of "A→B", focusing more on their own complexity.

6 Conclusion

This paper presents a novel fraud detector based on Graph Path Aggregation (GPA). By incorporating path sampling, encoding, interaction, and aggregation, GPA tackles the challenges of high-order interactions, intricate dependencies, and the sparse yet concealed nature of fraudulent entities, overcoming limitations of existing methods in handling graph structures with long-range associations. Furthermore, we construct G-Internet, the first benchmark dataset for internet fraud detection, to support the research. Experiments demonstrate GPA's superior performance over mainstream methods, robustness to label noise, and interpretability by unveiling implicit fraudulent patterns.

Acknowledgments and Disclosure of Funding

This work is funded by the National Key Research and Development Project (No. 2022YFB2703100), the Hangzhou Joint Fund of the Zhejiang Provincial Natural Science Foundation of China (No. LHZSD24F020001), the Fundamental Research Funds for the Central Universities (No. 226-2025-00067), and ZJU-China Unicom Digital Security Joint Laboratory.

References

[1] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: A survey. *Data mining and knowledge discovery*, 29:626–688, 2015.

[2] 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.

[3] Jing Ren, Feng Xia, Ivan Lee, Azadeh Noori Hoshyar, and Charu Aggarwal. Graph learning for anomaly analytics: Algorithms, applications, and challenges. *ACM transactions on intelligent systems and technology*, 14(2):1–29, 2023.

[4] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in neural information processing systems*, 36:29628–29653, 2023.

[5] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[6] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[7] Yu Zhou, Haixia Zheng, Xin Huang, Shufeng Hao, Dengao Li, and Jumin Zhao. Graph neural networks: Taxonomy, advances, and trends. *ACM transactions on intelligent systems and technology*, 13(1):1–54, 2022.

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

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

[10] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

[11] Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35:27387–27401, 2022.

[12] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023.

[13] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pages 21076–21089. PMLR, 2022.

[14] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference*, pages 1528–1538, 2023.

[15] Ziwei Chai, Siqi You, Yang Yang, Shiliang Pu, Jiarong Xu, Haoyang Cai, and Weihao Jiang. Can abnormality be detected by graph neural networks? In *International joint conference on artificial intelligence*, pages 1945–1951, 2022.

[16] Xiang Wang, Hao Dou, Dibo Dong, and Zhengyu Meng. Graph anomaly detection based on hybrid node representation learning. *Neural networks*, page 107169, 2025.

[17] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the international ACM SIGIR conference on research and development in information retrieval*, pages 1569–1572, 2020.

[18] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference*, pages 3168–3177, 2021.

[19] Hao Peng, Ruitong Zhang, Yingtong Dou, Renyu Yang, Jingyi Zhang, and Philip S Yu. Reinforced neighborhood selection guided multi-relational graph neural networks. *ACM transactions on information systems*, 40(4):1–46, 2021.

[20] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM web conference*, pages 1486–1494, 2022.

[21] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the ACM international conference on web search and data mining*, pages 357–365, 2023.

[22] Mahsa Mesgaran and A Ben Hamza. Graph fairing convolutional networks for anomaly detection. *Pattern recognition*, 145:109960, 2024.

[23] Rui Zhang, Dawei Cheng, Xin Liu, Jie Yang, Yi Ouyang, Xian Wu, and Yefeng Zheng. Generation is better than modification: Combating high class homophily variance in graph anomaly detection. *arXiv preprint arXiv:2403.10339*, 2024.

[24] Shunxin Xiao, Shiping Wang, Yuanfei Dai, and Wenzhong Guo. Graph neural networks in node classification: Survey and evaluation. *Machine vision and applications*, 33(1):4, 2022.

[25] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical mechanics and its applications*, 553:124289, 2020.

[26] Zhaohui Wang, Huawei Shen, Qi Cao, and Xueqi Cheng. Survey on graph classification. *Journal of software*, 33(1):171, 01 2022.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Yifei Sun, Haoran Deng, Yang Yang, Chunping Wang, Jiarong Xu, Renhong Huang, Linfeng Cao, Yang Wang, and Lei Chen. Beyond homophily: Structure-aware path aggregation graph neural network. In *IJCAI*, pages 2233–2240, 2022.

[29] Di Jin, Rui Wang, Meng Ge, Dongxiao He, Xiang Li, Wei Lin, and Weixiong Zhang. Raw-gnn: Random walk aggregation based graph neural network. *arXiv preprint arXiv:2206.13953*, 2022.

[30] Rafał Dreżewski, Jan Sepielak, and Wojciech Filipkowski. The application of social network analysis algorithms in a system supporting money laundering detection. *Information sciences*, 295:18–32, 2015.

[31] Kai Wang and Danwei Chen. Graph structure based anomaly behavior detection. In *International conference on computer engineering, information science & application technology*, pages 531–538. Atlantis press, 2016.

[32] Yuan Wang, Liming Wang, and Jing Yang. Egonet based anomaly detection in e-bank transaction networks. In *IOP conference series: Materials science and engineering*, volume 715, page 012038. IOP publishing, 2020.

[33] Lawrence Page. The pagerank citation ranking: Bringing order to the web. Technical report, 1999.

[34] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5):604–632, 1999.

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

[36] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the international conference on world wide web*, pages 1067–1077, 2015.

[37] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, pages 855–864, 2016.

[38] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

[39] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the ACM international conference on information & knowledge management*, pages 315–324, 2020.

[40] 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.

[41] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, pages 985–994, 2015.

[42] Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the international conference on world wide web*, pages 897–908, 2013.

[43] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[44] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

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

[46] Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in neural information processing systems*, 34: 14239–14251, 2021.

[47] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. Spam review detection with graph convolutional networks. In *Proceedings of the ACM international conference on information and knowledge management*, pages 2703–2711, 2019.

[48] Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the international ACM SIGIR conference on research and development in information retrieval*, pages 1239–1248, 2021.

[49] Enyan Dai, Charu Aggarwal, and Suhang Wang. Nrgnn: Learning a label noise resistant graph neural network on sparsely and noisily labeled graphs. In *Proceedings of the ACM SIGKDD conference on knowledge discovery & data mining*, pages 227–236, 2021.

[50] Jingyang Yuan, Xiao Luo, Yifang Qin, Yusheng Zhao, Wei Ju, and Ming Zhang. Learning on graphs under label noise. In *IEEE international conference on acoustics, speech and signal processing*, pages 1–5, 2023.

A Appendix

A.1 Detailed Dataset Construction

Existing fraud datasets are typically anonymized, which obscures the specific meanings of entities and consequently hinders interpretability analysis. Moreover, emerging internet fraud is rapidly evolving and widespread with concealed tactics, and research on internet fraud detection remains limited. Considering these gaps, we construct G-Internet, the first benchmark dataset for internet fraud detection, featuring a transparent structure. Due to privacy restrictions, real-world data is not available. Therefore, we synthesize the dataset through simulation. We meticulously consider a range of key factors, including fraud rules, the physical meanings, construction, and diversity of user and website node features, the behavioral differences between anomalous and benign users in terms of website visits, and the interconnections among users. In an overview, G-Internet contains 51268 user nodes, 109387 website nodes, 1572836 edges of the "user visits website" (or equivalently, "website visited by user") relation, and 399456 edges of the "user to user" relation.

Fraud rules. Types of internet fraud are varied, and the user behaviors regarding website visits exhibit specific patterns within each type. To more accurately simulate anomalous activities, we have designed 12 common rules as shown in Table 5 & 6, which are based on real cases and account for the diversity and complexity of user behaviors. Using the "pig-butchering scam" as an example, we establish the following rule: within a specified time frame, a user who downloads APKs more than twice, accesses blacklisted IPs and domains more than once each, and visits banking websites more than three times can be identified as having fallen into the trap of "pig-butchering scam".

Table 5: Types of fraud and their rules.

Type of fraud	Rule
Pig-butchering scam	$\delta > 2$ && $\zeta > 1$ && $\eta > 1$ && $\gamma > 3$
Brushing scam	$(\alpha > 2$ || $\beta > 2)$ && $\gamma > 2$
Phishing scam	$(\zeta > 1$ || $\eta > 1$ || $\epsilon > 5)$ && $\gamma > 2$
Credit repair scam	$\lambda > 1$ && $(\alpha > 1$ || $\kappa > 1)$ && $\gamma > 2$
Loan scam	$\theta > 5$ && $(\delta > 1$ || $\eta > 1)$ && $\gamma > 2$
Investment scam	$(\zeta > 5$ || $\eta > 5)$ && $(\beta > 3$ || $\kappa > 3)$ && $\gamma > 2$
Game trading scam	$\iota > 3$ && $\delta > 1$ && $(\alpha > 2$ || $\beta > 2)$
Hookup scam	$(\epsilon > 2$ || $\zeta > 5$ || $\eta > 5)$ && $\beta > 3$ && $\gamma > 2$
Honeytrap scam	$(\zeta > 5$ || $\eta > 5)$ && $\beta > 3$ && $\gamma > 2$
Online gambling scam	$\epsilon > 1$ && $\zeta > 5$ && $\eta > 5$ && $\gamma > 2$
Escort scam	$\epsilon > 3$ && $\zeta > 6$ && $\eta > 7$ && $\gamma > 1$
Nude chat scam	$\epsilon > 1$ && $\kappa > 1$ && $\gamma > 5$

Table 6: Meanings of each code in fraud rules.

Code	Meaning
α	Number of times a user visits zoom-like (online meeting) websites in a fixed time period
β	Number of times a user visits meiqia-like (customer service) websites in a fixed time period
γ	Number of times a user visits banking websites in a fixed time period
δ	Number of times a user downloads APKs in a fixed time period
ϵ	Number of times a user visits gambling websites in a fixed time period
ζ	Number of times a user visits blacklisted IPs in a fixed time period
η	Number of times a user visits blacklisted domains in a fixed time period
θ	Number of times a user visits loan websites in a fixed time period
ι	Number of times a user visits gaming websites in a fixed time period
κ	Number of times a user visits niche chat websites in a fixed time period
λ	Number of times a user visits credit repair websites in a fixed time period

User nodes. A total of 9 attributes have been generated for users, including age, gender, city, level, number of login times within a fixed time period, average login time, device, operating system, and active time periods. For instance, the age distribution often differs between benign and anomalous users. Based on existing references, we determine the age distribution for each user category and sample the age accordingly when constructing the attribute. Similar procedures are applied to the other attributes. Since our primary focus is to detect fraud based on the behavior of users visiting websites rather than on the users' inherent characteristics, we intentionally minimize the distinction between benign and anomalous users when constructing user attributes. Then, the user feature is

obtained through binary (0/1) encoding applied to each attribute. The dimension after binary encoding is 106. Additionally, we randomly mask some user attributes to increase the challenge.

Website nodes. The types of websites are diverse. In total, we have included 11 types of fraud-related websites and 27 types of normal websites in the dataset. The 11 types of fraud-related websites include zoom-like (online meeting), meiqia-like (customer service), banking, APKs, gambling, blacklisted IPs, blacklisted domains, loan, gaming, niche chat, and credit repair. The 27 types of normal websites include news, information, wikipedia, e-commerce, social media, blogs & personal, enterprise, education, forums & community, entertainment, government & non-profit, travel & food, image sharing, video, content sharing, art & design, technology & product review, travel guide, charity, community service platform, university official website, job search, real estate agency, real estate information platform, sports & fitness, flight & hotel booking, and financial management. Collecting and annotating a vast number of websites requires long-term investment. Due to the simulation nature of the dataset, we simplify the process by employing large language models (LLMs) to generate keywords for each website (except for websites on sensitive topics like gambling, pornography, etc., for which we collect keywords manually), then generate the website description based on the selected keywords, and finally use the BERT model to encode the description as the website feature, whose dimension is 768. To make the dimension match those of the user feature and to increase the difficulty, we only take the first 106 elements of the original website feature. Additionally, we randomly mask some elements in the website feature to increase the challenge.

"User visits website" relation. For an anomalous user, randomly select one of the designed fraud rules and generate connections between the user and websites that meet the rule. Anomalous users may also visit a large number of normal websites while visiting the websites involved in the fraud rules. These connections are also taken into account and randomly generated. For benign users, they visit more normal websites, but they may also visit fraud-related websites as long as they do not meet any of the fraud rules.

"User to user" relation. There may be relations between users, such as phone calls or other interactions. Since our primary focus is to detect fraud based on the behavior of users visiting websites rather than on the connections between users, we set both benign users and anomalous users to maintain a closer connection with benign users, so as to minimize the difference in communication behaviors between these two types of users.

In summary, the constructed G-Internet aligns with practical situations and has a clear and transparent structure. This interpretable dataset facilitates the demonstration of the interpretable GPA method (§3), which in turn enables the interpretable association analysis of fraud-related entities (§5.4).

A.2 Detailed Dataset Descriptions

In addition to the constructed internet fraud dataset, G-Internet, this paper adopts another five commonly used datasets from three typical fraud scenarios—finance, social networks, and online reviews—to evaluate the proposed method. These datasets include Elliptic [40], T-Finance [13], T-Social [13], YelpChi [41], and Amazon [42].

G-Internet has been thoroughly detailed in §A.1. The dataset is partitioned into training, validation, and testing sets with proportions of 40.00%, 20.00%, and 40.00%, respectively.

Elliptic [40] is derived from a real-world Bitcoin transaction network. It is a graph-structured dataset where nodes represent Bitcoin transactions and edges denote transaction flows. Transactions are categorized as legal, illegal, or unknown, encompassing entities such as exchanges, miners, and activities. Each node is associated with 93 features. Elliptic is particularly valuable for blockchain security and financial fraud detection, enabling the identification of illegal activities like money laundering. The dataset is split into training, validation, and testing sets with proportions of 45.86%, 18.34%, and 35.80%, respectively, based on transaction timestamps as per official recommendations.

T-Finance [13] focuses on detecting anomalous accounts in financial transaction networks. Nodes represent anonymous accounts characterized by 10-dimensional features related to registration duration, logging activities, and interaction frequencies. In the graph, edges signify account transactions, while anomalous nodes (e.g., money laundering, online gambling, etc.) are labeled by experts. The dataset is partitioned into training, validation, and testing sets with proportions of 40.00%, 20.00%, and 40.00%, respectively.

T-Social [13] is tailored for identifying anomalous accounts in social networks. It shares the same feature annotations as T-Finance. Two nodes are connected if they maintain a friendship for over three months. T-Social is characterized by its impressive scale, comprising 5.78 million nodes and 1.5 billion edges. This sheer volume of data poses a significant computational challenge. The dataset is partitioned into training, validation, and testing sets with proportions of 40.00%, 20.00%, and 40.00%, respectively.

YelpChi [41] is a behavioral graph dataset built from Yelp (a major U.S. review platform, which boasts a substantial customer base and significant social influence). It incorporates individual and graph-structured features stored in sparse matrix form, encompassing businesses, reviews, and user information. YelpChi is widely used in financial risk management and anti-money laundering research. The dataset is partitioned into training, validation, and testing sets with proportions of 40.00%, 20.00%, and 40.00%, respectively.

Amazon [42] is a large-scale e-commerce dataset provided by Amazon. Much like YelpChi, it encompasses a vast array of product metadata along with user-generated reviews. The dataset includes product reviews under the Musical Instruments category, with users and reviews as nodes. Amazon is frequently used for fraud detection validation. The dataset is partitioned into training, validation, and testing sets with proportions of 40.00%, 20.00%, and 40.00%, respectively.

A.3 Model Settings

The model settings for each dataset are shown in Table 7. n_path (I) denotes the number of initially sampled paths, k_path (K) denotes the number of selected top-K paths, l_path (J) denotes the maximum length of the path, d_path (C) denotes the dimension of path encoding, and n_attn and n_head denote the number of attention layers and attention heads, respectively.

Table 7: Model settings for each dataset.

Dataset	n_path (I)	k_path (K)	l_path (J)	d_path (C)	n_attn	n_head
G-Internet	200	150	20	128	2	8
Elliptic	100	100	5	256	2	1
T-Finance	200	150	5	128	2	1
T-Social	200	100	30	64	2	8
YelpChi	200	200	20	256	2	8
Amazon	60	30	20	64	2	1

A.4 Evaluation Metrics

Considering the significant class imbalance between positive and negative samples in fraud datasets, it is essential to use evaluation metrics that account for both Precision and Recall. This study employs Area Under the Curve (AUC) and Average Precision (AP) as key performance metrics.

AUC (%) generally refers to AUC-ROC (Area Under the Receiver Operating Characteristic Curve). The ROC curve illustrates the trade-off between the true positive rate and the false positive rate across different classification thresholds.

AP (%) is derived from the Precision-Recall (P-R) curve, which plots Precision against Recall as the classification threshold varies. AP reflects the area under the P-R curve, representing the model's average precision over all thresholds.

Fraud datasets are often highly imbalanced, with anomalous (positive) samples being significantly fewer than benign (negative) samples. As a result, models tend to achieve high recall for benign samples, potentially leading to an overly optimistic AUC score. The P-R curve, on the other hand, better reflects the model's ability to identify anomalous samples, as false negatives significantly impact the AP score. Therefore, AP offers a more reliable metric for assessing model performance in identifying rare but critical anomalous samples within imbalanced datasets.

Additionally, we provide two extra metrics: **Recall@K** (%) and **F1-Score** (%). Recall@K [4] is determined by the recall of true positives (anomalous samples) within the top-K highest-confidence predictions from the model, where the value of K is set as the number of anomalous samples in the test dataset. F1-Score represents the harmonic mean of Precision and Recall, offering a balanced assessment. Model performance comparison on these two metrics is shown in Table 8.

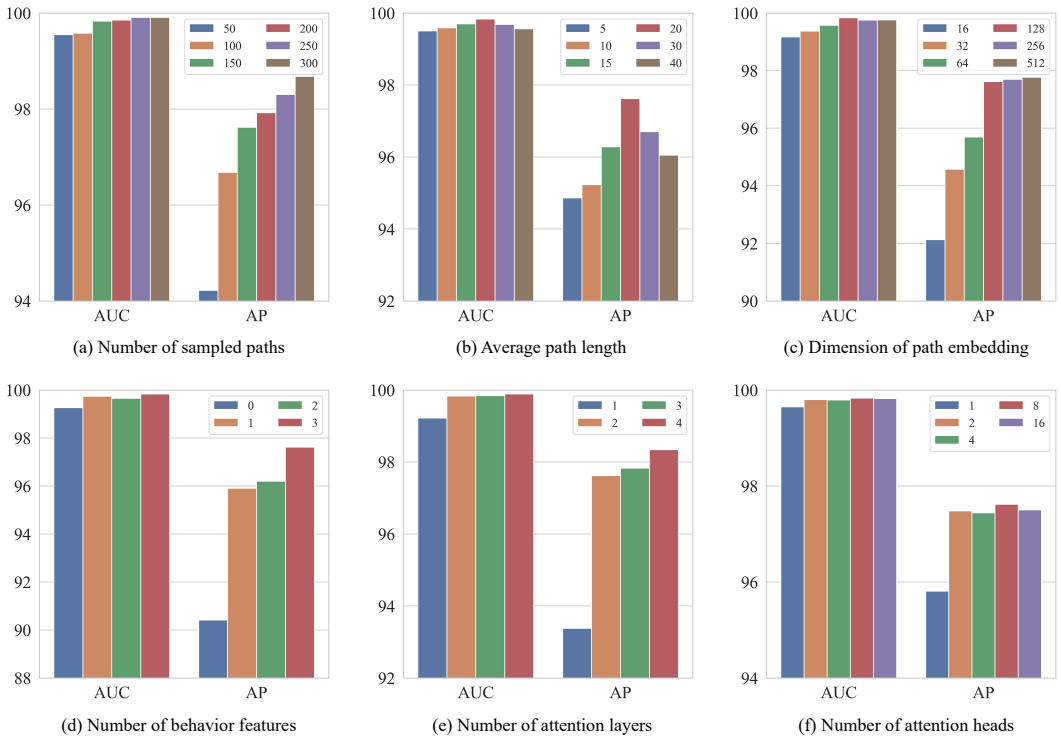
Table 8: Model performance comparison on Recall@K (%) and F1-Score (%) metrics.

Dataset	G-Internet		Elliptic		T-Finance		T-Social		YelpChi		Amazon	
Metric	R@K	F1	R@K	F1	R@K	F1	R@K	F1	R@K	F1	R@K	F1
GAT	58.6	77.6	37.9	57.3	79.8	64.9	42.1	63.4	44.2	57.1	82.6	70.5
BWGNN	64.1	77.7	42.5	59.2	84.2	89.1	75.8	85.3	56.7	76.6	85.9	91.5
PC-GNN	26.3	56.4	43.8	62.9	79.1	58.2	73.5	48.6	43.8	63.7	85.3	89.9
GPA	**93.4**	**96.5**	**69.9**	**81.7**	**84.5**	**92.2**	**90.3**	**95.4**	**68.2**	**79.0**	**86.1**	**92.3**

A.5 Hyperparameter Impact Analysis

This section explores the impact of various hyperparameters on the proposed GPA model, including the number of sampled paths and average path length in variable-length path sampling, the dimension of path embedding and number of behavior features in semantic-associated path encoding, and the number of attention layers and attention heads in path interaction and aggregation. Figure 4 illustrates the impact of these hyperparameters on G-Internet.

Experimental results indicate that, in variable-length path sampling, increasing the number of sampled paths allows the model to interact with more paths, helping it detect a broader range of fraudulent patterns, thus improving common pattern mining. However, this also raises computational costs. As the average path length increases, the model performance rises first and then drops. This may be because shorter paths can only achieve low-order pattern interactions and fail to capture more complex associations, while too-long paths may lead to information redundancy, and the longer the path, the more severe the information loss after path encoding. In semantic-associated path encoding, increasing both the dimension of path embedding and the number of behavior features enhances the model's knowledge base, leading to improved performance. Regarding path interaction and aggregation, the model performance improves as the number of attention layers increases. Unlike traditional GNNs such as GCN and GAT, increasing the number of attention layers in the proposed GPA will not cause "over-smoothing" that has negative effects. Additionally, as the number of attention heads increases, the model performance improves and gradually reaches saturation, and an excessive number of attention heads will impose computational burden.



(a) Number of sampled paths (b) Average path length (c) Dimension of path embedding

(d) Number of behavior features (e) Number of attention layers (f) Number of attention heads

Figure 4: Impact of the hyperparameters of the proposed GPA on G-Internet.

A.6 Necessity of Mining Long-Range Associations

Fraudulent activities often form implicit patterns through long-range associations. For example:

User A visits website X, but determining whether X is anomalous requires further analysis of whether other users visiting X (such as user B) exhibit anomalous behavior, and user B's anomalous behavior may be reflected in the website Y he visits, thus forming a multi-hop association chain. Although the "user visits website" relation in the dataset appears as a one-to-one relationship, the long-range association "A→X→B→Y" can better help accurately determine whether user A is anomalous.

From the graph in Figure 1, we can observe the following rules through long-range analysis:

(1) Fraud-related websites are mainly visited by anomalous users.
(2) Websites frequently visited by anomalous users are not necessarily related to fraud, as they are also frequently visited by benign users.

Therefore, mining long-range associations is of great necessity for interpretable fraud detection.

A.7 Computational Costs

We measure the model's running time. For training, we record the average time taken by the model to process a batch of data (including forward pass, backward pass, and gradient descent). For inference, we record the average time taken by the model to process a batch of data (forward pass only). The dataset used is G-Internet. The model settings used for measuring running time are kept the same as those in the main paper. All experiments use a batch size of 1024 and are conducted on a single NVIDIA GeForce RTX 3090 GPU. The results are presented in Table 9 & 10.

The proposed method is faster in both training and inference than many mainstream fraud detectors. Due to pytorch's low-level optimizations, the time required for path interaction does not increase significantly even if the number of paths grows a lot. The main computational overhead of the proposed method actually stems from path encoding, followed by path sampling and path interaction.

Table 9: Computational costs comparison.

Method	CARE-GNN	GDN	GPA
Training time / batch	821.5ms	340.3ms	95.4ms
Inference time / batch	688.3ms	119.3ms	25.4ms

Table 10: Computational costs for each component of the proposed GPA.

GPA	Path Sampling	Path Encoding	Path Interaction	Path Aggregation	Fraud Detection	Total
Inference time / batch	8.731ms	14.025ms	2.502ms	0.034ms	0.110ms	25.402ms

A.8 Limitation and Discussion

The limitation of our work lies in the simulated nature of the constructed G-Internet dataset. This is in fact constrained by data privacy policies. Initially, to establish this benchmark dataset, we consulted our collaborating institutions. Due to privacy restrictions, they were unable to provide real-world data directly. However, they did share several empirical fraud rules (as presented in Table 5 & 6), based on which we constructed the "user visits website" relation. Additionally, "user to user" connections (such as phone calls) are also actively utilized as evidence in rule-based fraud detection by our collaborating authorities. Therefore, we integrated these two relation types into the fraud detection system, with their distributions being freely adjustable to accommodate various real-world fraud scenarios. Anyway, based on existing fraud rules, we have made every effort to construct a dataset that is both close to reality and adjustable to different fraud scenarios, fully leveraging its advantages. Moving forward, we remain committed to continuous improvement and refinement.

A.9 Societal Impacts

This study strengthens the prevention and governance of various fraudulent activities across scenarios including the internet, finance, social networks, online reviews, etc., helping safeguard public welfare, promote social harmony and stability, and foster the healthy development of the digital economy.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: "Abstract" & §1 - "Introduction"

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: §A.8 - "Limitation and Discussion"

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: §3 - "Methodology" & §5 - "Experimental Study" & §A.3 - "Model Settings" & The source code provided & The constructed dataset provided

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code has been provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: §5 - "Experimental Study - Experimental settings" & §5 - "Experimental Study - Datasets" & §A.2 - "Detailed Dataset Descriptions" & §A.3 - "Model Settings" & §A.5 - "Hyperparameter Impact Analysis" & §A.7 - "Computational Costs"

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The numerical results are the average under 10 different random seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: §A.7 - "Computational Costs"

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: §A.9 - "Societal Impacts"

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The source code provided & The citations in the paper

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The source code provided

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: §A.1 - "Detailed Dataset Construction - Website nodes"

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.