Global Attribute-Association Pattern Aggregation for Graph Fraud Detection

Mingjiang Duan[1,2], **Da He**[1,4], **Tongya Zheng**[3,1],
Lingxiang Jia[1,4], **Mingli Song**[1,4], **Xinyu Wang**[1,2], **Zunlei Feng**[1,4*]
[1]State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2]Bangsheng Technology Co,Ltd.
[3]Big Graph Center, Hangzhou City University
[4]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
{duanmj, zunleifeng}@zju.edu.cn

Abstract

Fraud is increasingly prevalent, and its patterns are frequently changing, posing challenges for fraud detection methods such as random forests and Graph Neural Networks (GNNs), which rely on bin-based and mixture features separately. The former may lose crucial graph-associated features, while the latter face incorrect feature fusion. To overcome these limitations, we propose an approach based on attribute-association pattern that leverages the distinct attribute and association patterns differentiating fraudulent from benign behaviors, to enhance fraud detection capabilities. Attribute features are adaptively split into separate bins to eliminate incorrect attribute fusion and combine association patterns through graph neighbor message passing, thereby deriving attribute-association pattern features. Using the learned attribute-association patterns, the fraud patterns between a single pattern and the patterns across the entire graph are globally aggregated. Extensive experiments comparing our approach with 24 methods on 7 datasets demonstrate that the proposed method achieves SOTA performance.

Code — https://github.com/AtwoodDuan/GAAP

Introduction

The widespread adoption of the Internet and electronic payments has led to a rise in fraudulent activities, resulting in significant economic losses for society and businesses. In fraud detection, data are often represented as graphs, with nodes containing complex features such as age, gender, and transaction history. The relationships among these attributes are intricate and dynamic, which presents numerous challenges. Evolving patterns of fraudulent behavior make it difficult for traditional detection methods to address these dynamic scenarios effectively.

Extensive research has been conducted to address these challenges in various domains such as social networks (Li et al. 2023), mobile payments (Zheng et al. 2023), insurance fraud (Zhang et al. 2024b), and e-commerce (Wang et al. 2023). Fraud detection methods can be broadly classified into two categories: traditional machine learning approaches and Graph Neural Networks (GNNs).

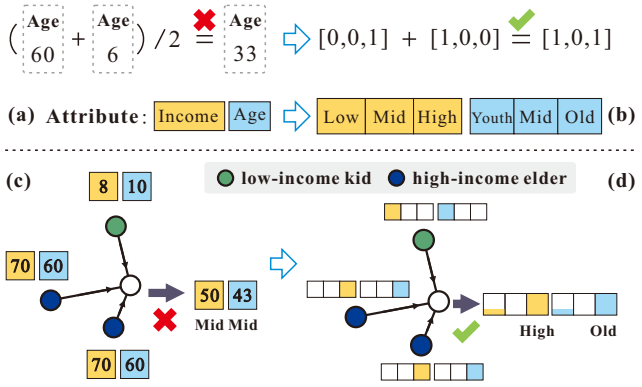

Figure 1: Attribute-pattern (top) and association-pattern (down). **(a)** Adding normal attribute features leads to erroneous feature values. **(b)** In contrast, utilizing the attribute-pattern (single attribute features are represented in a fragment-wise vector) preserves the correct meaning. For two easily misled groups ("low-income kids" and "wealthy elders"), **(c)** feature aggregation using normal attribute features results in inaccuracies ("middle-income middle-aged people"). Conversely, **(d)** when employing attribute-pattern aggregation, accurate association patterns emerge (indicating that fraudsters tend to target "high-income elders").

Traditional machine learning methods, such as Random Forest (Breiman 2001), effectively capture attribute feature patterns and identify feature intervals. Despite their effectiveness, they overlook the associations between nodes in graph-structured data, leading to suboptimal performance.

In contrast, Graph Neural Networks (GNNs) (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018) aggregate information from neighboring nodes to capture associative patterns within graph structures. However, when processing attribute features, GNNs often conflate similar and distinct types of attributes.

For example, averaging two ages (e.g. 60: older and 6: child) leads to erroneous feature values (e.g. 33: middle-aged). Furthermore, merging disparate attributes, such as age and income, can result in nonsensical outcomes. This mixing complicates the model's ability to identify attribute

*Corresponding author.

patterns accurately. As a result, it compromises both the accuracy and interpretability of fraud detection.

In fraud scenarios, fraudulent behaviors can be distinguished from legitimate ones by category. For instance, financial institutions have noted that fraudsters often target high-income elderly individuals. Similarly, e-commerce platforms report that low-income young users fall victim to fake rebate schemes, resulting in lost prepaid funds. Thus, accurately differentiating fraudulent behavior patterns from legitimate ones in complex graph-structured data is a critical challenge in current fraud detection research. Existing studies have focused largely on optimizing information aggregation within graph structures while often neglecting various fraud patterns.

To address these limitations, we propose a global attribute-association pattern aggregation framework for graph fraud detection, composed of three components. First, each attribute value is adaptively split into bins to eliminate incorrect attribute fusion. For example, as shown in Fig. 1 (a), income and age attributes are divided into bins (low, middle, high) and (youth, mid, old), respectively. Using attribute-patterns (single attribute features are represented as fragment-wise vectors), the fusion of attribute values retains their correct meanings, as depicted in Fig. 1 (b).

Next, the GNN aggregates the attribute-patterns of each node to derive association patterns between central and neighboring nodes. In Fig. 1 (d), feature-pattern aggregation accurately identifies fraudulent patterns (e.g., fraudsters often target "high-income elderly"), while aggregation using conventional features can lead to inaccuracies (e.g., "middle-income middle-aged individuals"). Finally, the attribute-association patterns from each node are globally aggregated across the entire graph, addressing the sparsity of fraudulent behaviors.

Extensive experiments demonstrate that global attribute-association pattern aggregation framework significantly enhances fraud detection performance and reveals interpretable fraud patterns, thus improving the practical applicability of the proposed method in real-world scenarios.

Overall, our contribution is the first to introduce a **G**lobal **A**ttribute-**A**ssociation **P**attern aggregation framework (GAAP) for fraud detection. In addition, a differentiable binning strategy is devised to construct attribute-pattern features, thereby improving the distinguishability of non-additive and distinct attribute combinations. Furthermore, we propose a global aggregation strategy that leverages global cross-attention to reduce computational complexity. Comprehensive experiments demonstrate that the proposed method achieves state-of-the-art performance.

Related Work

Traditional Machine Learning based Method. Traditional machine learning methods have played a significant role in fraud detection due to their broad applicability and interpretability. In industrial practice, ensemble tree learning algorithms such as Random Forest (Breiman 2001), LightGBM (Ke et al. 2017), and XGBoost (Chen and Guestrin 2016) continue to be widely used. These methods effectively capture attribute feature patterns, for instance, by segmenting age into different groups through information gain and combining them with income levels. However, they heavily rely on manual feature engineering and are unable to extract patterns from complex graph associations.

GNN-based Method. The GNN-based method has been a research hotspot in recent years due to its powerful potential for learning graph structures. Mainstream studies aim to improve the mechanism of message propagation on graphs from either the spatial or spectral perspective. From the spatial perspective, methods such as CARE-GNN (Dou et al. 2020), Rio-GNN (Peng et al. 2021), PC-GNN (Liu et al. 2021), DiG-In-GNN (Zhang et al. 2024a), and AO-GNN (Huang et al. 2022a) select neighbors based on similarity or customized loss functions. H2FDetector (Shi et al. 2022), PMP (Zhuo et al. 2024), and DGA-GNN (Duan et al. 2024) adapt different information weights for nodes with different properties from the domain. From the spectral perspective, AMnet (Chai et al. 2022), BWGNN (Tang et al. 2022), and GHRN (Gao et al. 2023) use low-pass and high-pass filters for targeted fraud adaptation designs. However, these methods rarely focus on the targeted mining of feature patterns of node attributes on fraud detection graphs.

Combined Method. To combine the advantages of the above two methods, some researchers have explored new approaches. DGA-GNN (Duan et al. 2024) uses preprocessing techniques to one-hot encode the original features. XGBGraph and RFGraph perform GNN-style feature engineering using non-parametric methods before applying ensemble tree models for classification predictions (Tang et al. 2024). Although these approaches have shown some success, they are not end-to-end solutions and tend to yield suboptimal performance.

Feature Binning. Feature binning is a traditional technique but remains active in the industrial practice of fraud detection, such as in credit scoring (Hand and Henley 1997). The earliest systematic discussion is represented by the work of Dougherty, Kohavi, and Sahami, which provides an overview of some classic approaches to binning. PLE (Gorishniy, Rubachev, and Babenko 2022) improves upon one-hot encoding by introducing lossless piecewise linear representations of the original scalar values. However, issues with two-stage optimization persist.

Globe Message Aggregation. Some studies have highlighted that global message aggregation can mitigate issues with the neighborhood message passing mechanism of GNNs, such as over-squashing and long-range dependencies. However, global message aggregation can lead to scalability problems as the number of nodes increases. Nodeformer (Wu et al. 2022) and SGFormer (Ren et al. 2023) address these scalability issues by randomly partitioning large graphs and approximating the entire graph using each partition, followed by linear transformations within each partition. Nevertheless, this approach inevitably results in the obstruction of some global information paths.

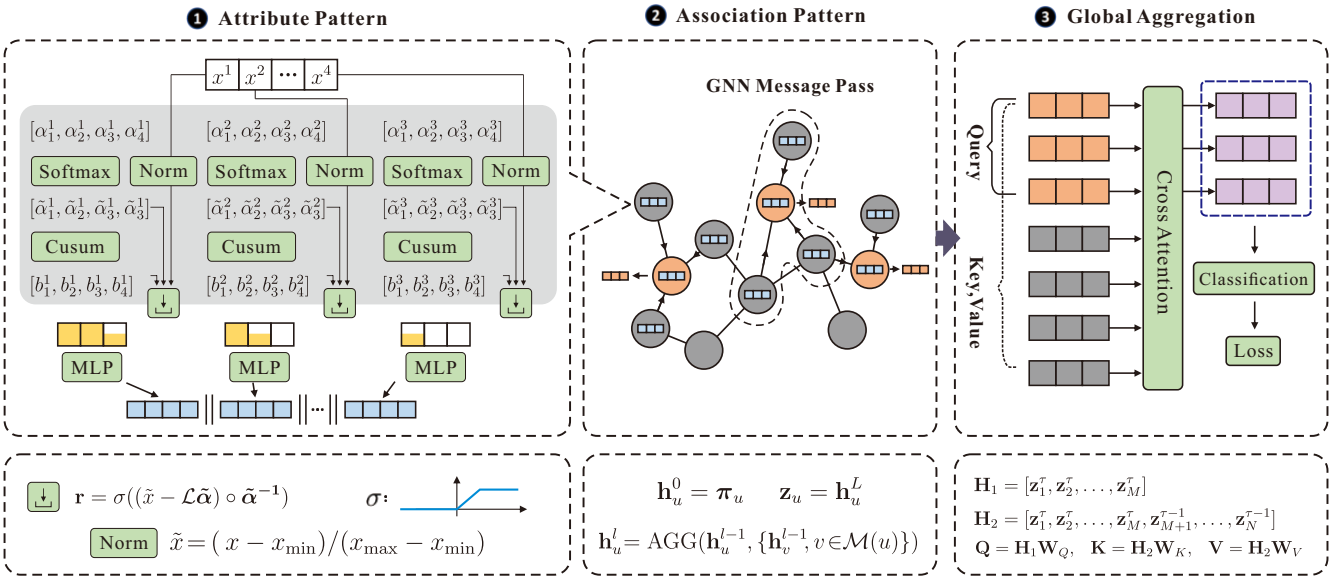

Figure 2: The global attribute-association pattern aggregation framework, composed of the extraction of attribute patterns, extraction of association patterns, and global aggregation of the attribute-association patterns. A dynamic binning embedding technique is employed to construct the feature patterns. Subsequently, a GNN aggregates the association patterns between central nodes and their neighbors. Finally, the attribute-association patterns for each node are globally aggregated with those of the entire graph, effectively leveraging global cross-attention while minimizing computational complexity.

Methodology

In this paper, we propose a global attribute-association pattern aggregation framework for graph fraud detection, which is composed of the attribute pattern extraction module, association pattern aggregation module, and global aggregation of the attribute-association patterns. The entire framework is illustrated in Fig. 2.

Notation. Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ denote a graph with a node set $\mathcal{N}$, comprising $N = |\mathcal{N}|$ nodes, and an edge set $\mathcal{E}$, consisting of $E = |\mathcal{E}|$ edges. The node attribute feature matrix is represented by $\mathbf{X} \in \mathbb{R}^{N \times d}$. Each node $u \in \mathcal{N}$ is assigned node features $\mathbf{x}_u \in \mathbb{R}^{1 \times d}$ and a label $y_u \in \{0, 1\}$. The vector corresponding to the i-th attribute feature column is represented by $\mathbf{x}^i \in \mathbb{R}^{N \times 1}$. The i-th attribute feature of node u, which is a scalar, is denoted by x_u^i.

Dynamic Binning Embedding

To capture feature patterns and address the limitations of overly smooth decision boundaries (Tang et al. 2024) and non-additive features (Duan et al. 2024) in graph fraud detection, we propose Dynamic Binning Embedding (DyBEM), which automatically learns distinctive embeddings for fraud features in an end-to-end manner. First, the feature value domain is dynamically split using a set of learnable vectors, a process known as *dynamic splitting*. Then, based on the binning results, specific feature values undergo *binary encoding*. Finally, *bin embedding* is applied to enhance the stability of the learning process.

Formally, for the i-th numerical feature, we split its value range into the disjoint T interval set $\{B_1^i, B_2^i, \ldots, B_T^i\}$, each of which we call *bins*: $B_t^i = [b_{t-1}^i, b_t^i)$. In previous research, bins were primarily determined using preprocessing, which often resulted in suboptimal bin optimization. It is necessary to design a learnable binning strategy. From now on, we omit the feature index i for simplicity.

Dynamic Splitting. Given a attribute feature x of a node, without loss of generality, Min-Max normalization is performed to scale it to the range $[0, 1]$ as follows:

$$\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{1}$$

where the scaled result is defined as $\tilde{x}$, $x_{\min}$ and $x_{\max}$ denote the minimum and maximum values of the attribute feature i in the training set, respectively.

In order to bin $\tilde{x}$ properly, a set of valid partition coordinates $\{b_0, b_1, \ldots, b_t, \ldots, b_T\}$ must be defined. These coordinates must satisfy the conditions $b_0 = 0$, $b_T = 1$, and each subsequent partition coordinate b_{t+1} must be strictly greater than b_t. To achieve this, the following construction is carried out. First, a vector $\boldsymbol{\alpha} \in \mathbb{R}^T$ is randomly initialized:

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_T), \tag{2}$$

where T represents the number of bins.

Then, the softmax transformation is applied to α to obtain $\tilde{\alpha}$, ensuring that the sum of all elements equals to 1 as follows:

$$\tilde{\alpha}_t = \frac{\exp(\alpha_t)}{\sum_{j=1}^{T} \exp(\alpha_j)}, \quad t \in \{1, 2, \ldots, T\}, \tag{3}$$

where $\exp(\cdot)$ is the exponential function with the natural base e. Due to the nature of the softmax transformation, the

sum of all elements in the resulting vector α is 1. At this point, a set of partition coordinates satisfying the requirements can be obtained:

$$b_t = \begin{cases} 0, & t = 0 \\ \sum_{i=1}^{t} \alpha_i. & t > 0 \end{cases} \tag{4}$$

Binary Encoding. Once the bins are determined, we define the dynamic encoding (DyBEN) scheme as follows:

$$\text{DyBEN}(x) = [r_1, \ \ldots, \ r_T], r_t = \sigma\left(\frac{\tilde{x} - b_{t-1}}{b_t - b_{t-1}}\right). \tag{5}$$

The activation function σ represents different encoding schemes. When $\sigma = \sigma_1$, the overall behavior corresponds to one-hot encoding (Dougherty, Kohavi, and Sahami 1995), and when $\sigma = \sigma_2$, it corresponds to PLE encoding. In most cases, PLE can achieve better performance in downstream tasks because it retains more precision of the original values and the encoding scheme carries ordinal information (Gorishniy, Rubachev, and Babenko 2022). These are formally expressed as follows:

$$\sigma_1(x) = \begin{cases} 0, & x < 0 \ \texttt{OR} \ x \geq 1 \\ 1, & x \geq 0 \ \texttt{AND} \ x < 1 \end{cases} \tag{6}$$

$$\sigma_2(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 1 \\ x. & x \geq 0 \ \texttt{AND} \ x < 1 \end{cases} \tag{7}$$

Moreover, to facilitate accelerated training and automatic gradient backpropagation in modern deep learning frameworks, we present the matrix row form as follows:

$$\mathbf{r} = \sigma((\tilde{x} - \mathcal{L}\tilde{\boldsymbol{\alpha}}) \circ \tilde{\boldsymbol{\alpha}}^{-\mathbf{1}}), \tilde{\boldsymbol{\alpha}} = \text{softmax}(\boldsymbol{\alpha}), \tag{8}$$

where $\circ$ denotes the hadamard product, and $\mathbf{r} \in \mathbb{R}^T$ represents a binary encoded vector transformed from a scalar x. $\mathcal{L}$ is a lower triangular matrix with ones below the diagonal:

$$\mathcal{L} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix}.$$

Because the dynamic encoding scheme may vary during training, employing dynamic PLE encoding proves beneficial for enhancing the stability of the training procedure. Up to this point, we have developed a fully differentiable auto-encoding module.

Binning Embedding. To increase the model's information capacity, we assign a trainable embedding $\mathbf{v}_t \in \mathbb{R}^d$ to each bin B_t. The final feature embedding is then derived by aggregating the bin embeddings weighted by e_t and subsequently adding a bias term $\mathbf{v}_0$, as follows:

$$\text{DyBEM}(x) = \text{Lin}(\text{DyBEN}(x)) = \mathbf{v}_0 + \sum_{t=1}^{T} r_t \cdot \mathbf{v}_t, \tag{9}$$

where $\text{Lin}(\cdot)$ represents a single-layer MLP. When using dynamic one-hot encoding, the binning embedding is similar in form to word embedding, where only one embedding vector is selected for optimization at each training step. However, when using dynamic PLE, all embedding vectors at positions less than or equal to t are selected at each training step.

To enhance the stability of binning learning, inspiration has been drawn from traditional Random Forests. It is beneficial to extend the mechanism by employing multiple initializations of $\boldsymbol{\alpha}$ for multi-head binning. Specifically, multiple independent sets of parameters are utilized to execute the transformation as Eqn.(9). The encoded results of binning embeddings with different attributes and different initializations are then concatenated to obtain the output representation $\boldsymbol{\pi}$, which is subsequently used as input features for downstream GNN models.

Message-Passing GNN

To capture association patterns, message-passing Graph Neural Networks (GNNs) are introduced. The process can be formalized as follows:

$$\mathbf{h}_u^l = \text{AGG}(\mathbf{h}_u^{l-1}, \{\mathbf{h}_v^{l-1} \mid v \in \mathcal{M}(u)\}),$$

where the hidden state of node u at layer l, denoted by $\mathbf{h}_u^l$, is initialized as $\mathbf{h}_u^0 = \boldsymbol{\pi}_u$. The aggregation function $\text{AGG}(\cdot)$ is responsible for combining the hidden state of node u from the previous layer, $\mathbf{h}_u^{l-1}$, with those of its neighboring nodes, represented by $\mathbf{h}_v^{l-1}$ for each v in the neighbor set $\mathcal{M}(u)$.

We employ GraphSAGE as the GNN module. It has been observed that GraphSAGE models with simple max pooling functions are effective. The framework is compatible with GNN models that conform to the message-passing paradigm, such as GAT and PMP. The embedding from the final layer $\mathbf{h}_u^L$ is then utilized as the input $\boldsymbol{z}_u$ to the global aggregation module.

Pattern Global Aggregation

To improve the extraction of attribute-association patterns by integrating global information, and to address the issues of over-smoothing and long-range dependencies (Wu et al. 2022; Ren et al. 2023) commonly introduced by GNN aggregation, we introduce a novel all-pair attention mechanism.

The primary computational and storage bottleneck of traditional self-attention lies in its $O(N^2)$ complexity. We observe that in message-passing-based spatial GNNs, the final layer of each step after information aggregation typically contains only a small number of nodes. During each epoch, we save the GNN output vectors from the previous iteration as historical embeddings. By computing the cross-attention between these step choice nodes and the entire node embeddings from previous epochs, global attention can be captured. This approach reduces the $O(N^2)$ complexity to $O(N \cdot M)$, where M represents the batch size.

The self-attention mechanism can be formally expressed as follows:

$$\mathbf{H}_1 = [\mathbf{z}_1^\tau, \mathbf{z}_2^\tau, \ldots, \mathbf{z}_M^\tau] \in \mathbb{R}^{M \times d_1}, \tag{10}$$

$$\mathbf{H}_2 = [\mathbf{z}_1^\tau, \mathbf{z}_2^\tau, \ldots, \mathbf{z}_M^\tau, \mathbf{z}_{M+1}^{\tau-1}, \ldots, \mathbf{z}_N^{\tau-1}] \in \mathbb{R}^{N \times d_1}, \tag{11}$$

$$\mathbf{Q} = \mathbf{H}_1\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{H}_2\mathbf{W}_K, \quad \mathbf{V} = \mathbf{H}_2\mathbf{W}_V, \tag{12}$$

Rec@K	Methods	Amazon	Yelp.	T-Fin.	Ellip.	Tolo.	DGraph.	T-Social	**Ave.**
Traditional Methods	MLP (Hinton 1990)	83.15	46.62	68.93	57.43	39.88	4.04	16.86	45.27
	KNN (Cover and Hart 1967)	79.35	51.92	70.04	56.60	37.85	1.98	44.08	48.83
	RF (Breiman 2001)	86.41	70.23	75.59	<u>72.76</u>	39.72	4.21	45.48	56.34
	XGBoost (Chen et al. 2016)	86.41	75.08	76.01	72.58	41.90	4.30	21.86	54.02
	NA (Yang et al. 2016)	<u>87.50</u>	73.38	75.31	71.74	42.21	4.04	21.70	53.70
General GNNs	GCN (Kipf and Welling 2017)	44.02	23.85	74.90	33.52	39.41	7.05	73.23	42.28
	GraphSAGE (Ham. et al. 2017)	78.26	47.15	78.09	56.69	48.75	6.84	73.74	55.65
	GAT (Veličković et al. 2018)	82.61	44.23	79.75	37.86	44.24	7.14	42.07	48.27
	GIN (Xu et al. 2018)	80.98	36.15	73.37	32.13	39.41	6.32	64.47	47.55
	SGC(Wu et al. 2019)	46.20	22.46	67.41	22.99	39.72	3.87	24.93	32.51
	GT (Shi et al. 2020)	78.80	44.62	81.55	30.75	44.39	6.92	43.58	47.23
	PNA (Corso et al. 2020)	**90.78**	30.00	72.54	36.47	42.68	5.16	26.95	43.51
	BGNN (vanov et al. 2021)	64.67	30.77	77.25	59.56	45.33	<u>7.70</u>	<u>96.89</u>	54.60
Specialized GNNs	GAS (Li et al. 2019)	80.43	38.00	79.75	37.49	47.04	6.02	64.58	50.47
	DCI (Wang et al. 2021)	80.98	40.46	71.98	35.27	37.85	5.85	18.27	41.52
	PCGNN (Liu et al. 2021)	85.33	43.77	79.06	43.77	43.15	6.66	73.53	53.61
	BernNet (He et al. 2021)	82.61	49.69	83.63	49.77	44.70	5.55	48.23	52.03
	AMNet (Chai et al. 2022)	83.15	45.38	84.05	30.56	42.52	4.21	43.21	47.58
	BWGNN (Tang et al. 2022)	85.87	56.69	84.19	42.47	50.31	7.57	75.78	57.55
	GHRN (Gao et al. 2023)	85.33	51.85	81.97	50.51	46.57	6.96	82.33	57.93
	PMP (Zhuo et al. 2024)	83.15	61.69	**85.99**	46.64	44.39	3.31	81.11	58.04
Combined Methods	RFGraph (Tang et al. 2024)	83.15	75.31	84.05	72.58	52.18	3.22	93.58	66.30
	XGBGraph (Tang et al. 2024)	85.87	83.15	85.02	71.93	53.43	6.96	93.53	68.56
	DGA-GNN (Duan et al. 2024)	85.87	<u>84.23</u>	84.33	72.76	<u>55.14</u>	7.52	95.97	69.40
Ours	GAAP	<u>87.50</u>	**88.54**	<u>85.71</u>	**73.32**	**56.08**	**7.73**	**97.25**	**70.88**

Table 1: The comparison results with 25 methods on 7 datasets. The recall score within top-K (Rec@K) is calculated, where K is set as the number of fraudulent samples in the test set for each dataset. '**Ave.**' signifies the average score across all datasets. The best results are highlighted in **bold**, and the second-best results are <u>underlined</u>. All scores are presented in %.

where $\mathbf{H}_1$ is an input sequence of vectors for the target nodes in the current step, $\mathbf{H}_2$ is a sequence of vectors for all nodes saved from the previous epoch, and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d_1 \times d_2}$ are learned weight matrices. The output of the self-attention layer is given by:

$$\mathbf{Z} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{sim}\left(\mathbf{Q}\mathbf{K}^\top\right)\mathbf{V}, \tag{13}$$

where $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{M \times N}$ is the dot product of the query and key matrices. Generally, the computation of $\text{sim}(\cdot)$ includes a softmax operation, necessitating the calculation of $\mathbf{Q}\mathbf{K}^\top\mathbf{V}$ from left to right. The first step has a complexity of $O(Md_2N)$, and the second step has a complexity of $O(MNd_2)$. The overall complexity is $O(2MNd_2)$. When M and d_2 are much smaller than N, the overall complexity can be approximated as $O(N)$.

Furthermore, when the scale of nodes increases and the magnitude of Md_2 becomes unacceptable, linear attention mechanisms, such as random feature approximation or simple global attention (Wu et al. 2022; Ren et al. 2023), can be considered. In this case, the calculation of $\mathbf{Q}\mathbf{K}^\top\mathbf{V}$ from right to left reduces the complexity to $O((M+N)d_2^2)$.

It is noteworthy that, since our input vectors are the output of a GNN and already contain structural similarity information, additional positional encoding operations are unnecessary. This allows for the simultaneous consideration of both structural and global information on the graph during cross-attention. We use a standard binary classification loss function in the loss component. To utilize the embeddings from previous epochs during training and inference, we cache the historical GNN embeddings at the end of each epoch.

Experiments

Datasets. Experiments were conducted on seven real-world fraud detection datasets, which are given as follows: *YelpChi*, a dataset aimed at identifying abnormal reviews that unfairly promote or demote products or businesses on Yelp.com (Rayana and Akoglu 2015). *Amazon*, a dataset containing users who write fake reviews in the musical instruments category on Amazon.com (McAuley and Leskovec 2013). *T-Finance*, a financial transaction fraud dataset (Tang et al. 2022). *T-Social*, a dataset for detecting abnormal accounts in social networks (Tang et al. 2022). *Elliptic*, designed for illicit Bitcoin transaction detection (Weber et al. 2019). *Tolokers*, a dataset for detecting fraudulent users on the Toloka crowd-sourcing platform (Platonov et al. 2023). *DGraph-Fin*, a credit default detection dataset provided by the Finvolution Group, constructed using guarantor contact information (Huang et al. 2022b). For more com-

Dataset	#Nodes	#Edges	#Dim.	Relation Concept	Attribute Feature Description
Amazon	11,944	4,398,392	25	Review Correlation	Hand-crafted user features and statistics
YelpChi	45,954	3,846,979	32	Reviewer Interaction	Hand-crafted review features and statistics
T-Finance	39,357	21,222,543	10	Transaction Record	User profile details such as registration days
Elliptic	203,769	234,355	166	Payment Flow	Timestamps and transaction information
Tolokers	11,758	519,000	10	Work Collaboration	User profile with task performance statistics
DGraph-Fin	3,700,550	4,300,999	17	Loan Guarantor	Timestamps and user profiles details
T-Social	5,781,065	73,105,508	10	Social Friendship	User profile details such as logging activities

Table 2: Dataset descriptions about node attribute feature types, feature dimensions, and additional details.

parative experiments on the GADBench (Tang et al. 2024) datasets, please refer to the appendix. The statistical information of the datasets is shown in Tab. 2.

Baselines. We employ four distinct groups of baseline methodologies for comparison with the proposed method:

- The first group is composed of traditional methods, encompassing Multilayer Perceptron (MLP) (Hinton 1990), k-Nearest Neighbors (KNN) (Cover and Hart 1967), Random Forest (RF) (Breiman 2001), XGBoost (Chen and Guestrin 2016), and Neighborhood Averaging (NA) (Yang, Rahardja, and Franti 2023). This group serves as a foundational reference to observe outcomes when graph information is absent.
- The second group comprises general GNNs, including Graph Convolutional Network (GCN) (Kipf and Welling 2017), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GIN (Xu et al. 2018), SGC (Wu et al. 2019), Graph Transformer (GT) (Shi et al. 2020), and BGNN (Ivanov and Prokhorenkova 2021).
- The third group consists of algorithms specifically designed for graph fraud detection. In the spatial domain, this includes GAS (Li et al. 2019), DCI (Wang et al. 2021), PCGNN (Liu et al. 2021), and PMP (Zhuo et al. 2024); in the spectral domain, it includes BernNet (He et al. 2021), AMNet (Chai et al. 2022), BWGNN (Tang et al. 2022), and GHRN (Gao et al. 2023).
- The fourth group of baselines combines the advantages of traditional machine learning and GNNs, including RFGraph and XGBGraph, proposed in GADBench (Tang et al. 2024), as well as DGA-GNN. These GNN-related algorithms are mostly implemented using the DGL framework provided by GADBench (Tang et al. 2024). The official implementations are used for DGA-GNN and PMP.

Metrics. In accordance with existing anomaly detection literature, we employed the recall score within top-k predictions (Rec@K) as a performance metric. The value of K was set to the number of fraudulent samples in the test set. In industrial practice, Rec@K has significant business relevance, whereas AUROC and AUPRC can yield misleadingly high scores in the presence of imbalanced samples. The Rec@K results are shown in Tab. 1. Complete comparison results, including AUROC and AUPRC, are provided in the appendix.

Parameters Setting. To ensure fairness in hyperparameter tuning, we followed the recommendations of GADBench, using random search to optimize the hyperparameters and reporting test set results corresponding to the highest validation scores. In each trial on every dataset, a set of hyperparameters was randomly selected from a predefined search space for each model. Early stopping was performed on the validation set, and the scores on the test set were reported. All experiments were run on an Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz with multiple NVIDIA A6000 GPUs. For our proposed method, the number of bins T was selected from a range of 4 to 40, the number of GNN layers L ranged from 1 to 4, and the mini-batch size varied from 32 to 5000 depending on the dataset. For more information on default hyperparameters, search space, and other implementation details, please refer to the appendix.

Comparison with SOTA

Overall, Table 1 demonstrates the superiority of our proposed method compared to 24 baseline methods across 7 datasets. On the YelpChi dataset, an improvement of 4.3% was achieved. Additionally, state-of-the-art (SOTA) performance was reached on 5 datasets, while the second-best performance was achieved on the Amazon dataset, possibly due to noise in the graph structure information of Amazon, which may have caused instability during optimization.

From the perspective of method groups, General GNNs performed the worst, likely due to their lack of task-specific design for fraud detection. Specialized GNNs and traditional methods were evenly matched, as they focus on mining attribute patterns and association patterns, respectively. Combined methods performed the best, as they leverage the strengths of both attribute and association pattern mining.

The methods in the Combined Methods category demonstrated superiority in the overall comparison across the four groups. This is because fraud detection tasks require the extraction of information from both attribute patterns and association patterns. DGA-GNN outperformed XGBGraph likely because the feature extraction for the association pattern in XGBGraph occurs before the learning process, which does not receive optimal optimization. Ours outperformed DGA-GNN due to the double optimization problem in the attribute pattern of DGA-GNN, whereas our proposed end-to-end architecture effectively overcomes this issue. For more information on the experimental analysis and visualization analysis, please refer to the appendix.

Model	YelpChi Rec@K	YelpChi AUPRC	YelpChi AUROC	T-Finance Rec@K	T-Finance AUPRC	T-Finance AUROC	T-Social Rec@K	T-Social AUPRC	T-Social AUROC
w/o DyBEM	47.62	46.80	83.19	79.07	85.34	96.33	75.82	79.92	97.78
w OHT	75.33	83.90	94.99	84.49	88.81	**97.24**	95.52	98.69	99.87
w/o GNN	61.62	64.88	88.62	72.63	78.95	93.75	23.32	24.72	77.56
w/o GA	86.29	92.49	97.26	84.12	89.58	96.14	96.85	97.01	99.51
w GP	68.19	75.11	92.04	66.41	72.25	90.79	75.62	79.59	93.43
Ours	**87.51**	**93.89**	**98.47**	**85.22**	**90.32**	97.13	**97.05**	**99.49**	**99.96**

Table 3: The results of the ablation study on different modules and strategies. DyBEM refers to dynamic binning embedding, OHT denotes static one-hot encoding, GA represents our proposed cross-attention-based global aggregation module, and GP refers to the graph partition-based aggregation method. For each dataset, the reported results(%) are the average of 10 runs.

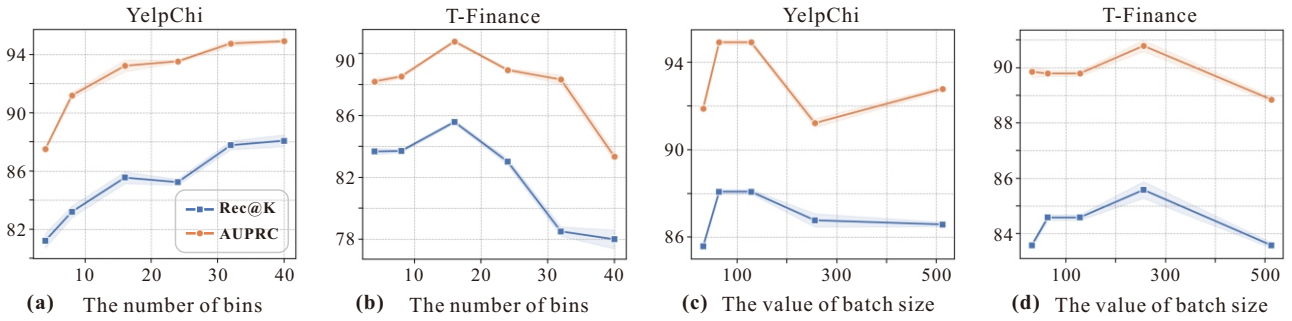

Figure 3: The ablation study on the number of bins and batch size. The Recall score within top-k predictions (Rec@K) and receiver operating characteristic curve (AUROC) are reported. All scores are in %.

Ablation Study

As shown in Tab. 3 and in Fig. 3, we selected three datasets that benefit significantly from the graph structure for ablation experiments. Based on the hyperparameter search conducted in the main experiment, we ran each dataset 10 times and averaged the results. It can be observed that all three components of proposed method produced clear benefits.

- For the attribute pattern part, removing the DyBEM module had the greatest impact on the YelpChi dataset, as it includes the largest number of node attributes with high information density. Replacing dynamic binning with static one-hot binning resulted in a performance decline, which demonstrates that our proposed dynamic binning better adapts to the overall optimization process and meets the demands of downstream tasks. As shown in Fig. 3(a) and 3(b), we analyzed the impact of the number of bins. For the YelpChi dataset, an increase in the number of bins resulted in an approximately monotonically increasing benefit, albeit with diminishing marginal returns. On the T-Finance dataset, the best performance was achieved with around 8 bins. This difference is attributed to the varying complexity of attribute information in YelpChi and T-Finance.
- For association pattern part, removing the GNN module had the greatest impact on the T-Social dataset, as the extraction of graph structure information is a crucial factor for its fraud detection performance.
- For the pattern global aggregation part, first, removing the global aggregation (GA) module results in some performance degradation. Secondly, we replaced the proposed cross-attention aggregation method with a global graph partition(GP) followed by all-pair attention within each partition. It was observed that this aggregation method led to negative results due to the loss of randomness in node connections. These demonstrate the effectiveness of our module design. As shown in Fig. 3(c) and 3(d) analysis is on the batch size, which corresponds to the size of the query part in cross-attention. The experiments showed that the batch size had no significant impact on the overall performance of the model. This is because our attention mechanism ensures the completeness of potential overall information transmission.

Conclusion

The proposed attribute-association pattern-based approach effectively overcomes the limitations of traditional fraud detection by adaptively splitting attribute features and aggregating neighbor features. This method enhances interpretability and enables the global aggregation of fraud patterns, leading to improved detection across various scenarios. Extensive experiments confirm our method's SOTA performance. The dynamic binning embedding shows promise in tabular data, CV, and NLP tasks with statistical features. Targeted improvements to the framework's three components could be a valuable avenue for future research.

Acknowledgments

This work is funded by National Key Research and Development Project (Grant No: 2022YFB2703100), Zhejiang Province "JianBingLingYan+X" Research and Development Plan (No. 2024C01114), and Ningbo Natural Science Foundation (2023J281).

References

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Chai, Z.; You, S.; Yang, Y.; Pu, S.; Xu, J.; Cai, H.; and Jiang, W. 2022. Can Abnormality be Detected by Graph Neural Networks? In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 1945–1951.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Corso, G.; Cavalleri, L.; Beaini, D.; Liò, P.; and Veličković, P. 2020. Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33: 13260–13271.

Cover, T.; and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1): 21–27.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3844–3852.

Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; and Yu, P. S. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 315–324.

Dougherty, J.; Kohavi, R.; and Sahami, M. 1995. Supervised and unsupervised discretization of continuous features. In *Machine learning proceedings 1995*, 194–202. Elsevier.

Duan, M.; Zheng, T.; Gao, Y.; Wang, G.; Feng, Z.; and Wang, X. 2024. DGA-GNN: Dynamic Grouping Aggregation GNN for Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11820–11828.

Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference 2023*, 1528–1538.

Gorishniy, Y.; Rubachev, I.; and Babenko, A. 2022. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35: 24991–25004.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

Hand, D. J.; and Henley, W. E. 1997. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160: 523–541.

He, M.; Wei, Z.; Xu, H.; et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34: 14239–14251.

Hinton, G. E. 1990. Connectionist learning procedures. In *Machine learning*, 555–610. Elsevier.

Huang, M.; Liu, Y.; Ao, X.; Li, K.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022a. Auc-oriented graph neural network for fraud detection. In *Proceedings of the ACM web conference 2022*, 1311–1321.

Huang, X.; Yang, Y.; Wang, Y.; Wang, C.; Zhang, Z.; Xu, J.; and Chen, L. 2022b. DGraph: A Large-Scale Financial Dataset for Graph Anomaly Detection. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Ivanov, S.; and Prokhorenkova, L. 2021. Boost then convolve: Gradient boosting meets graph neural networks. *arXiv preprint arXiv:2101.08543*.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, A.; Qin, Z.; Liu, R.; Yang, Y.; and Li, D. 2019. Spam review detection with graph convolutional networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2703–2711.

Li, Y.; Zhu, J.; Zhang, C.; Yang, Y.; Zhang, J.; Qiao, Y.; and Wang, H. 2023. THGNN: An Embedding-based Model for Anomaly Detection in Dynamic Heterogeneous Social Networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1368–1378.

Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021. Pick and Choose: A GNN-Based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference 2021*, 3168–3177.

McAuley, J. J.; and Leskovec, J. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, 897–908.

Peng, H.; Zhang, R.; Dou, Y.; Yang, R.; Zhang, J.; and Yu, P. S. 2021. Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. *ACM Transactions on Information Systems*, 40(4): 69:1–69:46.

Platonov, O.; Kuznedelev, D.; Diskin, M.; Babenko, A.; and Prokhorenkova, L. 2023. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*.

Rayana, S.; and Akoglu, L. 2015. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 985–994.

Ren, S.; Yang, X.; Liu, S.; and Wang, X. 2023. Sg-former: Self-guided transformer with evolving token reallocation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6003–6014.

Shi, F.; Cao, Y.; Shang, Y.; Zhou, Y.; Zhou, C.; and Wu, J. 2022. H2-FDetector: A GNN-Based Fraud Detector with Homophilic and Heterophilic Connections. In *Proceedings of the ACM Web Conference 2022*, 1486–1494.

Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; and Sun, Y. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.

Tang, J.; Hua, F.; Gao, Z.; Zhao, P.; and Li, J. 2024. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Advances in Neural Information Processing Systems*, 36.

Tang, J.; Li, J.; Gao, Z.; and Li, J. 2022. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning*, 21076–21089.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, Y.; Zhang, J.; Guo, S.; Yin, H.; Li, C.; and Chen, H. 2021. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 1239–1248.

Wang, Z.; Wu, Q.; Zheng, B.; Wang, J.; Huang, K.; and Shi, Y. 2023. Sequence as genes: An user Behavior modeling framework for fraud transaction detection in E-commerce. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5194–5203.

Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D. K. I.; Bellei, C.; Robinson, T.; and Leiserson, C. E. 2019. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. arXiv:1908.02591.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871.

Wu, Q.; Zhao, W.; Li, Z.; Wipf, D. P.; and Yan, J. 2022. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 27387–27401.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Yang, J.; Rahardja, S.; and Franti, P. 2023. Neighborhood averaging for improving outlier detectors. *arXiv preprint arXiv:2303.09972*.

Zhang, J.; Xu, Z.; Lv, D.; Shi, Z.; Shen, D.; Jin, J.; and Dong, F. 2024a. DiG-In-GNN: Discriminative Feature Guided GNN-Based Fraud Detector against Inconsistencies in Multi-Relation Fraud Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9323–9331.

Zhang, R.; Cheng, D.; Yang, J.; Ouyang, Y.; Wu, X.; Zheng, Y.; and Jiang, C. 2024b. Pre-trained Online Contrastive Learning for Insurance Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 22511–22519.

Zheng, W.; Xu, B.; Lu, E.; Li, Y.; Cao, Q.; Zong, X.; and Shen, H. 2023. MIDLG: Mutual Information based Dual Level GNN for Transaction Fraud Complaint Verification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 5685–5694.

Zhuo, W.; Liu, Z.; Hooi, B.; He, B.; Tan, G.; Fathony, R.; and Chen, J. 2024. Partitioning message passing for graph fraud detection. In *The Twelfth International Conference on Learning Representations*.