Context-aware Graph Neural Network for Graph-based Fraud Detection with Extremely Limited Labels

Pengbo Li [*], Hang Yu [*], Xiangfeng Luo [†]

The School of Computer Engineering and Science, Shanghai University, Shanghai, China
{lpb, yuhang, luoxf}@shu.edu.cn

Abstract

Graph-based fraud detection is crucial in identifying illegal activities in social networks, finance, and other sectors. Despite recent progress in this area, most of current researches typically require a large amount of annotated data to demonstrate its benefits. In practice, obtaining sufficient high-quality annotated data is challenging, limiting the effectiveness of model training. Therefore, leveraging extremely limited label information is crucial to enhance model performance. We propose a context-aware graph neural network (CGNN) to address this. CGNN performs category semantic decomposition on the contextual neighbor features of the center node to enrich the category semantics. In the neighbor message aggregation stage, the denoising attention mechanism enables the center node to adaptively aggregate heterophilic and homophilic information from neighbors. Particularly for unlabeled data, feature augmentation within the category subspace and consistency regularization driven by entropy minimization ensure that such data can further enhance model performance under explicit semantic guidance. We demonstrate on four real-world datasets that CGNN significantly outperforms other baseline methods with extremely limited labels.

1 Introduction

Fraud detection has been widely studied as a risk control method, but initial research primarily focused on structured data, such as tabular data (Zhou et al. 2020; Kurshan, Shen, and Yu 2020). However, tabular data is insufficient in reflecting the interactions between entities. For example, financial fraud often involves interactive transactions between bank cards or between bank cards and merchants. Therefore, it is necessary to use unstructured data, such as graph-structured data, to improve fraud detection.

With the emergence of graph neural networks, there has been a growing interest among researchers in graph-based fraud detection. GADBench (Tang et al. 2024) introduces a benchmark for graph anomaly detection tasks and provides a comprehensive analysis of current graph anomaly detection algorithms. Notable works such as CARE-GNN (Dou et al. 2020), BWGNN (Tang et al. 2022), and AMNet (Chai et al. 2022) have demonstrated promising results in graph-based fraud detection tasks. However, these methods typically require a substantial amount of labeled data to achieve optimal model performance. In practice, the availability of labeled data is often extremely limited, and most of the samples are unknown to their labels, called the gray samples. Hence, recent research has increasingly focused on improving fraud detection performance with few labeled data. For instance, GTAN (Xiang et al. 2023) and ConsisGAD (Chen et al. 2024) leverage label propagation and learnable data augmentation techniques to enhance model performance with limited labeled data.

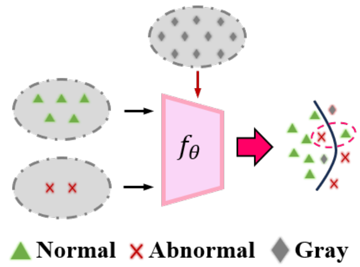


Figure 1: Improper use of unlabeled gray samples can adversely affect the model. For example, assigning pseudo labels to gray samples or assuming that gray samples are normal can introduce errors.

Motivation. However, the aforementioned methods have shown success primarily stems from the efficient use of the label data, and overlooking the information from gray samples. Even the latest work ConsisGAD (Chen et al. 2024) proposes a learnable data augmentation module to utilize gray samples explicitly, this method lacks explicit semantic information guidance. In fact, as shown in Figure 1, the unreasonable use of gray samples will have a huge impact on the accuracy of the detection results. Hence, effectively leveraging a small number of labeled samples alongside a large number of gray samples for fraud detection is our research motivation.

Challenges. This is nontrivial due to the following challenges: (1) Effective utilization of gray samples. Existing self-supervised or semi-supervised learning methods for un-

[*]These authors contributed equally.

[†]Corresponding author

labeled samples rely on multi-view data augmentation (Liu et al. 2022). However, data augmentation can disrupt original fraud patterns, resulting in suboptimal model performance. (2) Joint optimization of models with labeled and gray samples. Gray samples lack clear semantic meaning, making it challenging to optimize them alongside labeled samples. Existing pseudo-labeling methods can lead to error accumulation, rendering them less than ideal (Kim et al. 2022).

Solution. To address the challenges of graph-based fraud detection with few labels, we propose a Context-aware Graph Neural Network (CGNN). It performs category semantic decomposition on the contextual neighbor features of the center node to enrich the category semantics. In the neighbor message aggregation stage, the denoising attention mechanism enables the center node to adaptively aggregate heterophilic and homophilic information from neighbors. In particular, for unlabeled data, we conduct targeted perturbations on the feature subspace formed by category semantic decomposition to form different feature augmentation forms. Finally, the idea of consistency regularization is used to optimize the model. Highlights of our contributions are as follows:

- The category semantic decomposition of contextual neighbors significantly enhances GNNs' ability to extract semantic information from gray samples, alleviating the issue of semantic deficiency in scenarios with limited labeled data.
- Feature augmentation within the category subspace and consistency constraints based on minimizing entropy enable the utilization of gray samples under explicit semantic guidance.
- Extensive experiments are conducted on four datasets. It is fully verified that our proposed CGNN can effectively utilize gray sample information to improve model performance compared with the baseline methods.

2 Related Work

2.1 Graph-based Fraud Detection

Graph-based fraud detection primarily focuses on graph neural networks (GNNs) (Li et al. 2023; Jin et al. 2021; Liu et al. 2023; Xiang et al. 2023; Xu et al. 2024), which can be categorized into spatial and frequency domain approaches. In the spatial domain, the focus is on propagating and aggregating neighborhood information. CARE-GNN (Dou et al. 2020) employs reinforcement learning to adaptively sample neighbors, effectively filtering out noise. PC-GNN (Liu et al. 2021) enhances this approach by introducing conditional node sampling to combat fraud camouflage. FRAUDRE (Zhang et al. 2021) integrates outputs from different layers to manage varying complexities. In the frequency domain, the emphasis is on designing filters to capture target signals. AMNet (Chai et al. 2022) aligns anomalous node signals with high-frequency information for anomaly detection. BWGNN (Tang et al. 2022) uses band-pass filters to manage spectral energy shifts. GHRN (Gao et al. 2023) employs a label-aware edge indicator to reduce noise. PMP-GNN (Zhuo et al. 2024) bridges the spatial and spectral domains by using adaptive node-level spectrogram filters to address heterogeneity.

2.2 Graph-based Semi-supervised Learning

Semi-supervised learning is a very effective learning paradigm when you have a small amount of labeled and a large amount of unlabeled data (Song et al. 2022; Kim et al. 2022; Yang et al. 2021). It is mainly implemented through consistency regularization or pseudo-label learning. Consistency regularization is implemented based on the assumption that the model should have the same predictions for different inputs after reasonable perturbations (Laine and Aila 2016; Li, Xiong, and Hoi 2021; Yang et al. 2024). Pseudo-label learning is based on the low-density assumption of the decision boundary, and uses high-confidence unlabeled samples as training targets, making such samples far away from the decision boundary (Lee et al. 2013; Berthelot et al. 2019). There are also some methods that combine the two to further improve the effect of the model, such as FixMatch (Sohn et al. 2020), Grafn (Lee et al. 2022), etc. Usually these methods require different dimensions of perturbation of the data, such as random cropping or flipping of images in the field of computer vision, and random masking of node features or cropping of edges in the field of graph learning (Yang, Cohen, and Salakhudinov 2016). It is difficult to ensure the rationality of this random augmentation method. This paper uses the learned category semantic features to achieve feature augmentation in category subspace, ensuring the rationality and controllability of the augmentation.

3 Problem Definition

We denote the graph by $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{X}, \mathcal{C})$, where $\mathcal{V} = \{v_1, ..., v_N, v_{N+1}, ..., v_{N+M}\}$ represents the set of nodes in the graph. For ease of description, we use $V = \{v_i\}_1^N$ to represent the set of labeled nodes, and $\mu V = \{v_i\}_{N+1}^{N+M}$ to represent the set of unlabeled nodes. $\mathcal{A}$ represents the adjacency matrix. $\mathcal{X} = \{x_1, x_2, ..., x_N\} \in \mathbb{R}^d$ represents the node features. $\mathcal{C}$ represents the label of the node.

Unlike previous work, our focus is on alleviating the problem of mediocre model performance in the presence of only a small number of labels. Hence, the number of instances with real labels is denoted as $|\mathcal{C}| = |\mathcal{C}_0| + |\mathcal{C}_1| \ll N$, where $|\mathcal{C}_0| = \eta\,|\mathcal{C}_1|$ and $|\mathcal{C}_1|$ represents the counts of labeled normal and fraudulent nodes, respectively. The parameter η signifies the ratio of positive to negative samples. Following (Wan et al. 2021), we set $|\mathcal{C}_1|$ to 1 and 5, while η maintains the original ratio of the dataset, to study fraud detection under extremely limited labels.

4 Methodology

The overall framework of our proposed CGNN is shown in Figure 2. The key core of CGNN is based on the assumption of interactivity between nodes. That is, *for any node in the graph, it has the characteristics of maintaining its own category and interacting with other categories.* In other words, node features can be decomposed according to node categories (Ma et al. 2019).

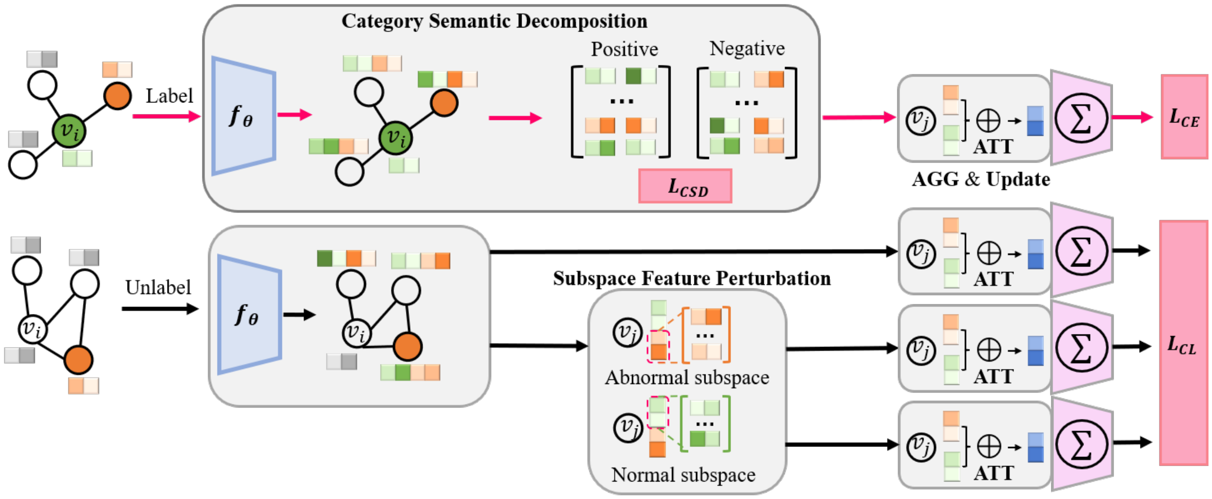


Figure 2: The overall framework of CGNN. CGNN improves the ability of the center node to perceive the category semantics of its neighbors, especially the unlabeled nodes, through category semantic decomposition. At the same time, it performs consistency regularization training on the category subspace feature perturbations (SFP) of the unlabeled nodes, further improving the performance of the model.

4.1 Context-aware Graph Neural Network

Category Semantic Decomposition In reality, the lack of annotated data makes it difficult to directly train an effective model. Therefore, CGNN fully mines the contextual information of the center node through category semantic decomposition (CSD), thereby alleviating the problem of less annotated data. Specifically, for the center node and its contextual neighbors, we first use a linear layer to map all node features in the graph to the same latent space.

$$x = \sigma(\mathbf{FC}(x)) \tag{1}$$

where **FC** represents the fully connected layer, and σ represents the Leaky ReLU nonlinear activation function.

Based on the aforementioned interaction hypothesis between nodes, we believe that each node has different characteristic components to determine the link tendency with other specific categories of nodes. Since graph fraud detection is a binary classification problem, we can decompose the node features into two subfeature components: normal and abnormal.

Specifically, the features of the neighbor v_j of the center node v_i are equally divided into two parts, and then two different linear layers project each part to different category subspaces as follows:

$$\begin{aligned} x_j^0 &= W_{nor}(x_j[:m,:]) + b_{nor} \\ x_j^1 &= W_{abnor}(x_j[m:,:]) + b_{abnor} \end{aligned} \tag{2}$$

where $m = d//2$ represents the feature dimension after decomposition. $\cdot//\cdot$ represents integer division operation. W_{nor}, W_{abnor}, b_{nor} and b_{abnor} are all learnable parameters. If v_j is a normal node, then x_j^0 is the component that maintains its own category characteristics, and x_j^1 is the tendency component to interact with other category nodes. The reverse is also true for fraudulent nodes.

Therefore, we adopt feature consistency contrastive learning to constrain the semantics of the two categories of normal and abnormal. We keep the semantics of the feature subspaces of the same category close, while the semantics of the subspaces of different categories are far away (Khosla et al. 2020). So the positive sample pair $\langle x_j^k, x_j^p \rangle \,|_{k=p}$ comes from the feature components of the same category subspace, while the negative sample pair $\langle x_j^k, x_j^a \rangle \,|_{k \neq a}$ is the node feature component of different category subspaces. Specifically, we implement the contrast constraints between positive and negative sample pairs as follows:

$$L_{CSD} = \sum_{k \in N} -\frac{1}{|P(k)|} \log \left\{ \sum_{p \in P(k)} \frac{exp(\frac{\mathcal{S}\langle x_j^k, x_j^p \rangle}{\tau})}{\sum_{a \in A(k)} exp(\frac{\mathcal{S}\langle x_j^k, x_j^a \rangle}{\tau})} \right\} \tag{3}$$

where $P(k)$ is the set of positive sample pairs, that is, other samples with the same label as sample c. $A(k)$ is a set of negative sample pairs, that is, other samples with different labels from sample k. $\mathcal{S}\langle \cdot, \cdot \rangle$ denotes the cosine similarity and τ is the temperature coefficient. Category semantic decomposition gives explicit semantic information to the contextual neighbors of the center node, especially the unlabeled nodes. This enables the center node to obtain more explicit information from its neighbors.

Denoising Attention Further clarify and refine the information that need to be propagated to the center node. We employ an attention mechanism to determine the messages content from each neighbor node that should be propagated to the center node.

$$\begin{aligned} \alpha_j &= softmax(W_{Att} Tanh(W(x_j^0 + x_j^1) + b)) \\ x_j &= \alpha_j x_j^0 + (1 - \alpha_j) x_j^1 \end{aligned} \tag{4}$$

where α_j represents the attention weight. The above operations enable the center node to adaptively extract neighbor information in its area.

Aggregation and Update We use the messages propagated by context neighbors to update the feature of the center node as follows:

$$x_i = \mathbf{FC}(x_i + \frac{1}{\sqrt{d_i}} \sum x_j) \tag{5}$$

where **FC** represents the fully connected layer, and d_i represents the degree of the center node.

Prediction Take x_i as the input of the classifier based on multi-layer perceptron to obtain the final classification probability $\hat{y}_i$. The classification process can be optimized by cross entropy.

$$L_{CE} = -\sum_i \eta^c y_i^c log(\hat{y}_i^c) \tag{6}$$

where L_{CE} calculates the classification loss with cross-entropy. η^c is the imbalance weight corresponding to class c. y_i^c represents the label of node i on class c, and $\hat{y}_i^c$ is the corresponding predicted probability.

Theoretical Insights In the spectral domain, CGNN's contextual neighbor perception functions as a node-level adaptive spectral graph filter.

Theorem 1. Consider an attributed graph $\mathcal{G}$, Let $L = U\Lambda U^T$ represent the eigendecomposition of the symmetric normalized Lpalacian $L = I - D^{-1/2}AD^{-1/2}$, where U is the matrix of eigenvectors and $\Lambda = diag([\lambda_i]_{i=1...N})$ is the diagonal matrix of eigenvalues and $0 = \lambda_1 \leq ... \leq \lambda_n \leq 2$. The different feature components of node v_i can be regarded as two sets of different graph signals, namely XW_{nor} and XW_{abnor}. W_{nor} and W_{abnor} are similar to the parameters of Eq.(2) that map node features to normal and abnormal subspaces. CGNN can be represented as a node-level adaptive graph filter. The specific spectral convolution of node v_i on XW_{nor} and XW_{abnor} can be defined as:

$$\begin{aligned} H(v_i) &= g_{nor}^i(L)XW_{nor} + g_{abnor}^i(L)XW_{abnor} \\ &= Ug_{nor}^i(\Lambda)U^T XW_{nor} + Ug_{abnor}^i(\Lambda)U^T XW_{abnor} \end{aligned} \tag{7}$$

where the spectral convolution filters are diagonal matrices defined as:

$$\begin{aligned} g_{nor}^i(\Lambda)[j,j] &= \gamma_i(1-\lambda_j), \qquad && v_j \in \mathcal{N}(v_i) \\ g_{abnor}^i(\Lambda)[j,j] &= (1-\gamma_i)(1-\lambda_j), && v_j \in \mathcal{N}(v_i) \end{aligned} \tag{8}$$

where $H(v_i)[i,:]$ represents the feature representation of node v_i after convolution. $\gamma_i = \frac{1}{d}\sum \alpha_j$ is the filter adaptation coefficient. d is the degree of the center node. The proof process is shown in Appendix A.1.

As we all know, different center nodes are located in different areas of the graph, which causes the center nodes to show different homogeneity and heterogeneity. Therefore, it is not optimal for existing works (Chai et al. 2022; Dong et al. 2021; Yang et al. 2023) to use the same graph filtering method for all nodes. Under the condition of very few labels, the labels of the neighbors of the center node are almost unknown. As a node-level adaptive graph filter, CGNN realizes the perception of the homogeneity and heterogeneity of the center node through category semantic decomposition. In other words, CGNN can effectively combine contextual neighbor information to obtain regional unique information.

4.2 Consistency Training

For unlabeled gray samples, consistency regularization within the feature subspace are created by category semantic decomposition to enhance the model's generalization ability.

Specifically, for the unlabeled node v_i, its contextual neighbor v_j can be decomposed into two components x_j^0 and x_j^1 by the Eq.(2). Then similarly combining x_j^0 and x_j^1 according to the Eq.(4) can obtain the feature x_j of the unlabeled node.

Subspace Feature Perturbation After category semantic decomposition, the normal and abnormal feature sets $\{x_j^0\}$ and $\{x_j^1\}$ of neighbor nodes are equivalent within their respective feature subspaces. Hence, we employ random subspace feature perturbation (SFP) to accomplish feature augmentation from two distinct perspectives: normal and abnormal. Combining Eq.(4), we can get the neighbor message after disturbance as follows:

$$\begin{aligned} \tilde{x}_j^0 &= \alpha_j x_{FA}^0 + (1-\alpha_j)x_j^1, \ x_{FA}^0 \in SFP(\{x_j^0\}) \\ \tilde{x}_j^1 &= \alpha_j x_j^0 + (1-\alpha_j)x_{FA}^1, \ x_{FA}^1 \in SFP(\{x_j^1\}) \end{aligned} \tag{9}$$

where $\tilde{x}_j^0$ and $\tilde{x}_j^1$ are the messages of the context neighbor nodes after feature augmentation in the normal and abnormal feature subspaces, respectively. $SFP(*)$ is a random permutation operation of features within a subspace. $\{x_j^0\}$ and $\{x_j^1\}$ represent the feature sets of neighbor nodes in the normal and abnormal subspaces, respectively. x_{FA}^0 and x_{FA}^1 represent the features after feature augmentation in the normal and abnormal subspaces, respectively.

Finally, for x_j, $\tilde{x}_j^0$ and $\tilde{x}_j^1$, the corresponding classification probabilities $\hat{y}$, $\hat{y}^0$ and $\hat{y}^1$ can be obtained through Eq.(5) and the classifier. According to consistency regularization (Li, Xiong, and Hoi 2021), the prediction $\hat{y}$ of unlabeled nodes should be consistent with $\hat{y}^0$ and $\hat{y}^1$ respectively. Therefore, we constrain the consistency relationship between the three through minimum entropy as follows:

$$L_{CL} = -\frac{1}{N}\sum \mathcal{S}\left\langle \hat{y}, \hat{y}^0 \right\rangle log(\mathcal{S}\left\langle \hat{y}, \hat{y}^1 \right\rangle) \tag{10}$$

where $\mathcal{S}\langle\cdot,\cdot\rangle$ denotes the cosine similarity.

4.3 Objective Function

To sum up, the objective function of CGNN in the training process consists of three parts: supervised classification loss function L_{CE}, category semantic decomposition loss L_{CSD}, and consistency training loss L_{CL}. Therefore, the overall objective function can be expressed as follows:

$$L = L_{CE} + \lambda L_{CSD} + \beta L_{CL} \tag{11}$$

where λ and β are constant coefficients respectively used to balance the objective function. It is worth noting that the feature space is not perturbed during the testing phase, so the prediction results can be obtained directly through CGNN.

For ease of understanding, we list the entire training process of the model in Algorithm 1.

Algorithm 1: CGNN

1: **Input**: Attribute graph $\mathcal{G}$; Few labeled nodes V and corresponding labels $\mathcal{C}$. Unlabeled nodes μV are used for semi-supervised learning. Number of training iterations $iters$.
2: **Output**: Predict the result $\hat{y}$.
3: Initialize the corresponding model parameters.
4: **while** $iter \longleftarrow iters$ **do**
5: Input the attribute graph $\mathcal{G}$, and map all nodes of V and μV to the same feature space through Eq.(1).
6: Eq.(2) and Eq.(3) respectively realize the category semantic decomposition and the corresponding semantic constraints.
7: Eq.(4) selectively adjusts the neighbor messages of the center node V. Eq.(5) realizes the aggregation and update of the center node message.
8: The prediction $\hat{y}$ for the labeled node V is obtained through the classifier and further optimized using Eq.(6).
9: Eq.(9) perform subspace feature perturbation and denoising operations on the context neighbor nodes of the unlabeled center node.
10: Eq.(10) constrains the consistency of predictions for unlabeled nodes.
11: Calculate the total optimization loss according to Eq.(11).
12: Update model parameters.
13: **end while**
14: **return** $\hat{y}$.

5 Experiments

In this section, we conduct experiments to validate the performance of CGNN on four real-world graph fraud detection datasets while also answering the following questions:

- **RQ1:** Does CGNN outperform state-of-the-art methods for graph-based fraud detection with extremely few labels?
- **RQ2:** What impact does the core module of CGNN have on model performance?
- **RQ3:** How important parameters in CGNN affect model performance?

5.1 Experimental Setup

Datasets We selected the review-oriented datasets **Amazon** (Liu et al. 2021) and **YelpChi** (Dou et al. 2020), and the financial datasets **Elliptic** (Chai et al. 2022) and **T-Finance** (Tang et al. 2022) to evaluate the performance of CGNN. Table 1 summarizes the basic information of these four datasets.

Baselines For evaluating the performance of CGNN, we establish a baseline using state-of-the-art GNN-based methods. This baseline comprises well-known graph neural networks like GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018), and GraphSAGE (Hamilton, Ying, and Leskovec 2017). Additionally, it includes specialized graph models explicitly crafted for the task of graph fraud detection, namely CARE-GNN (Dou et al. 2020), PC-GNN (Liu et al. 2021), FRAUDRE (Zhang et al. 2021), AMNet (Chai et al. 2022), BWGNN (Tang et al. 2022), GHRN (Gao et al. 2023), Semi-GNN (Wang et al. 2019), GTAN (Xiang et al. 2023), BSL (Yu, Liu, and Luo 2024), ConsisGAD (Chen et al. 2024) and PMP-GNN (Zhuo et al. 2024). It is worth noting that Semi-GNN, GTAN, BSL, ConsisGAD and PMP-GNN all emphasize their model capabilities with few labels. Additional details provided in the Appendix A.6

Datasets	#Nodes	#Edges	#Features	Abnormal(%)
Elliptic	46,564	73,248	93	9.76
Amazon	11,944	4,398,392	25	6.87
T-Finance	39,357	21,222,543	10	4.58
YelpChi	45,954	3,846,979	32	14.53

Table 1: The statistics of datasets

Evaluation Metrics We cast the graph fraud detection task as a binary classification problem under imbalanced conditions. Consequently, for model evaluation, we adopt standard metrics including Macro-F1, Macro-Recall, and Macro-Precision, in line with established methodologies.

Implementation Details In all experiments, we use Adam (Kingma and Ba 2014) to optimize the model, where the learning rate is 0.01 and the weight decay is 1e-4. On all four datasets, we set the embedding size to 64, the batch size to 16, and the balance coefficients λ and β of the loss function to 1. For the temperature coefficient τ in L_{CSD} it is set to 0.8 in T-Finance and 0.01 in other datasets.

For data split, we use $D = \{D_{train} \cup D_{valid} \cup D_{test}\}$ to represent all data, where D_{train}, D_{valid} and D_{test} represent training, validation and test sets respectively. These sets account for 10%, 30% and 60% of the total data respectively. Specifically, in order to reflect the few-label and gray samples, we set the training set $D_{train} = \{D_{label} \cup D_{gray}\}$, where D_{label} and D_{gray} represent labeled and unlabeled samples, respectively. For D_{label}, we first randomly sample 1 or 5 abnormal nodes without replacement, and then randomly sample normal nodes without replacement according to the positive-to-negative sample ratio of the original data. The nodes not sampled are used as D_{gray}. In addition, to reflect the stability of the model in experiments, we report the mean and standard deviation of five experiments. See the Appendix A.2 for more implementation information.

5.2 Performance Comparison of Baseline (RQ1)

Table 2 presents the evaluation results of CGNN and baseline methods with 1 or 5 black samples during training. The results show that basic GNN models like GCN, GAT, and GraphSAGE are ineffective due to fraud camouflage. However, specialized methods like PC-GNN, BWGNN, and GHRN fail to leverage gray samples, limiting their performance under extremely limited labels.

CGNN is compared to recent graph-based fraud detection methods using gray samples, including GTAN, BSL and ConsisGAD. Although they use gray samples, the lack

Method	Amazon				Elliptic			
	1		5		1		5	
	F1	Recall	F1	Recall	F1	Recall	F1	Recall
GCN	58.40±8.87	60.91±9.35	57.60±7.44	60.13±9.24	50.55±2.04	60.55±5.17	49.86±1.85	50.47±2.57
GAT	45.94±16.85	46.31±16.46	55.23±7.94	56.46±8.37	26.09±1.77	26.98±2.63	25.93±2.15	24.02±1.85
GraphSAGE	71.95±10.75	69.36±10.40	68.11±15.06	75.27±13.58	47.68±0.76	58.89±2.54	47.34±0.72	47.87±1.08
CARE-GNN	75.51±14.74	72.26±13.44	88.87±0.73	**87.40±1.76**	50.95±5.46	57.33±10.43	60.86±5.81	65.21±8.48
FRAUDRE	65.47±11.63	61.42±9.44	73.80±11.98	80.15±4.31	49.49±5.25	54.93±8.90	58.46±5.69	68.30±9.17
PC-GNN	73.76±14.12	69.33±11.28	82.17±5.36	76.08±5.53	44.45±14.61	58.33±9.18	66.00±8.24	66.47±8.53
AMNet	61.21±11.31	58.91±7.93	80.25±6.86	74.37±7.23	53.99±13.11	54.98±9.97	60.31±16.06	61.69±14.45
BWGNN	72.89±15.14	70.55±14.04	86.72±3.26	83.63±3.06	62.28±6.46	64.05±6.94	73.90±8.61	72.29±8.35
GHRN	76.53±12.21	76.07±10.22	89.99±1.15	86.36±1.68	61.81±6.17	64.40±7.05	74.28±8.59	72.26±8.26
GTAN	79.31±6.52	77.16±9.62	83.04±10.95	79.93±9.67	57.46±5.56	68.95±8.54	70.94±4.73	77.40±4.65
Semi-GNN	49.03±21.54	58.73±11.97	73.29±8.79	77.81±4.20	56.95±8.42	62.28±11.89	70.65±7.93	70.86±7.81
ConsisGAD	80.08±13.70	75.89±12.29	88.27±1.91	84.09±3.10	59.71±15.37	60.15±14.30	75.64±11.1	73.95±10.91
BSL	78.40±15.41	77.22±14.42	88.66±1.66	84.69±3.06	65.13±9.63	70.25±11.74	73.28±9.23	73.57±9.44
PMP-GNN	79.26±3.42	75.80±3.12	87.39±3.00	83.70±3.35	54.03±13.13	55.49±11.02	68.65±17.34	67.00±13.91
CGNN (ours)	**85.56±8.49**	**82.26±8.50**	**90.08±0.72**	86.51±1.21	**76.26±7.07**	**76.29±4.46**	**82.79±4.48**	**80.81±0.40**

Method	T-Finance				YelpChi			
	1		5		1		5	
	F1	Recall	F1	Recall	F1	Recall	F1	Recall
GCN	55.77±7.28	54.84±5.07	55.92±8.52	56.00±7.75	46.08±0.00	50.00±0.00	46.08±0.00	50.00±0.00
GAT	32.81±9.28	36.91±10.74	31.37±9.83	34.99±9.84	46.08±0.00	50.00±0.00	46.08±0.00	50.00±0.00
GraphSAGE	64.49±7.93	62.55±6.38	69.08±11.27	68.36±9.41	46.73±0.83	50.13±0.13	46.40±0.34	50.08±0.10
CARE-GNN	66.24±12.84	74.21±15.18	66.62±9.40	80.93±2.57	52.36±0.93	52.69±0.59	53.09±3.03	**56.63±2.40**
FRAUDRE	76.03±9.18	73.63±9.01	75.60±14.33	83.13±4.17	48.35±2.43	49.88±3.07	41.93±8.57	53.73±2.01
PC-GNN	74.32±13.60	76.42±13.23	84.95±1.56	82.61±0.79	46.34±0.51	50.11±0.23	54.05±3.06	54.37±2.39
AMNet	70.19±5.86	68.62±4.73	80.37±2.53	75.99±5.01	48.70±2.29	50.86±0.73	50.43±3.80	52.10±2.05
BWGNN	73.21±5.11	74.05±4.95	85.75±3.07	82.42±3.44	49.29±1.42	50.56±0.74	54.59±1.25	54.71±0.98
GHRN	69.46±10.39	70.97±8.38	85.98±1.56	83.53±2.87	49.14±1.87	50.63±0.56	55.04±1.12	55.10±1.11
GTAN	74.79±14.43	70.52±12.60	85.44±2.82	84.44±1.84	47.47±4.11	51.30±0.85	51.20±3.46	52.73±1.98
Semi-GNN	56.05±9.64	71.19±9.62	52.15±1.89	72.60±5.27	32.72±16.36	50.00±0.00	31.96±15.89	50.18±0.28
ConsisGAD	80.10±6.63	79.37±8.61	88.37±2.33	84.06±4.44	51.90±3.05	52.69±2.13	55.05±1.96	56.25±2.78
BSL	66.94±3.47	67.68±8.97	64.76±14.45	75.10±6.21	49.06±0.88	51.79±2.20	52.12±3.28	52.84±1.83
PMP-GNN	78.97±7.44	76.31±7.69	84.83±1.72	81.09±0.90	33.42±15.62	50.02±0.06	46.13±0.03	50.02±0.02
CGNN (ours)	**86.36±3.59**	**82.94±4.36**	**90.10±1.62**	**87.17±1.81**	**52.71±1.13**	**52.73±1.17**	**56.08±1.86**	56.49±1.63

Table 2: Performance(%) of fraud detection on four datasets. The 1st/2nd best results are marked in bold/underline.

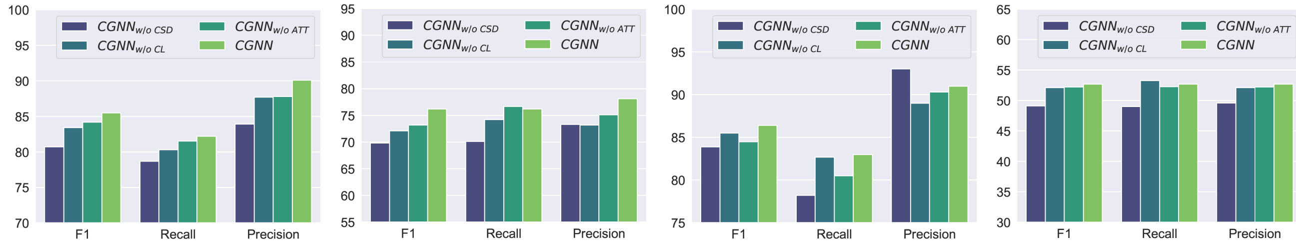
of clear category semantics limits their effectiveness. PMP-GNN only focuses on the gray neighbor information corresponding to the central node. CGNN performs well with 1 and 5 black samples, owing to node-level adaptive spectral filtering and consistent training with gray samples. Even on the challenging YelpChi dataset, CGNN demonstrates comparable performance. Notably, CGNN achieves the best Macro F1 scores across all four datasets.

In addition, the standard deviation reveals that CARE-GNN, PC-GNN, and similar models are more sensitive to labeled data distribution with few labels, resulting in larger standard deviations. In contrast, CGNN's spectral adaptation mechanism leads to a smaller standard deviation, indicating stronger robustness to changes in data distribution.

5.3 Ablation Analysis (RQ2)

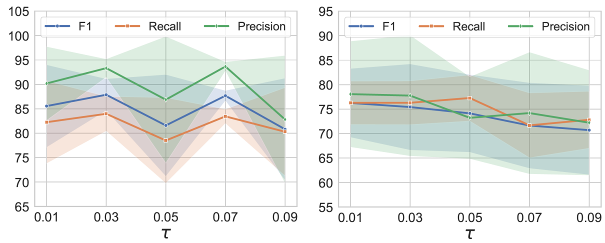
To evaluate the effectiveness of the key designs of CGNN, we conducted a series of ablation experiments by excluding each module. As shown in Figure 3, $CGNN_{w/o\ CSD}$ excludes the category semantic decomposition module, and $CGNN_{w/o\ CL}$ excludes the consistency training process. $CGNN_{w/o\ ATT}$ means that the attention mechanism in Eq.(4) is not used, but the normal and abnormal components are directly added together.

It can be observed that the category semantic decomposition module in CGNN improves the utilization of labels and lays a good foundation for the consistency training part by assigning category semantics to the context neighbor information of the center node under extremely limited labeling



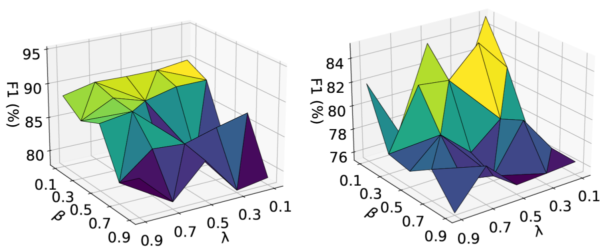
(a) Amazon (b) Elliptic (c) T-Finance (d) YelpChi

Figure 3: Ablation analysis for four datasets.



(a) Amazon (b) Elliptic

Figure 4: Temperature coefficient sensitivity analysis.



(a) Amazon (b) Elliptic

Figure 5: Weight factor analysis of the loss function.

conditions. Consistency training further improves the model effect through consistent regularization of different perspectives of unlabeled data. It is worth noting that the theoretical analysis in 4.1 shows that the attention weight coefficient will adjust the spectrum energy, making CGNN an adaptive filter. From the ablation result $CGNN_{w/o\ ATT}$, when the attention mechanism is removed, the model effect decreases. This demonstrates the effectiveness of CGNN as a node-level adaptive spectral filter.

5.4 Parameter Sensitivity Analysis (RQ3)

Temperature coefficient analysis. In the category semantic decomposition module, Eq.(3) plays an important role. The smaller the temperature coefficient, the sharper the similarity distribution, which will make the model pay more attention to details but prone to overfitting. Conversely, the similarity distribution will be flatter. We therefore evaluate the sensitivity of the temperature coefficient to model performance at different settings. Specifically, in the Amazon, Elliptic and YelpChi datasets we set $\tau \in \{0.01, 0.03, 0.05, 0.07, 0.09\}$. For T-Finance, we set $\tau \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We conducted five experiments under corresponding settings. Figure 4 shows the specific effects in Amazon and Elliptic datasets. It can be seen that the temperature coefficient τ has a slight impact on the effect of the model. For example, as τ increases, the performance of the model shows a slightly decreasing trend in the Elliptic dataset. But overall, better performance can be easily achieved through appropriate and reasonable adjustments.

Weight factor analysis of the loss function. As shown in Eq.(11), the two factors λ and β are critical in balancing the supervised classification loss, category semantic decomposition loss, and consistency regularization loss. Hence, we evaluate our model's sensitivity to these terms under different settings. Specifically, we run CGNN on the four datasets for $\lambda, \beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ based on grid search and report the results of Macro-F1 in Figure 5. It can be observed that the loss weight parameters λ and β have a slight impact on the performance of CGNN. But it is worth noting that no matter how λ and β change, CGNN can show competitive results. In addition, we found that reasonable settings of the two parameters λ and β can achieve better results than those reported in Table 2. For example, for the elliptic dataset, CGNN shows better results when λ and β are set to 0.1 and 0.2 respectively. The parameter sensitivity analysis results of T-Finance and YelpChi are shown in Appendix A.3.

6 Conclusion

In this study, we propose a novel context-aware graph neural network (CGNN) with extremely limited labels. CGNN performs category semantic decomposition on the contextual neighbor features of the center node to enrich the category semantics. In the neighbor message aggregation stage, the denoising attention mechanism enables the center node to adaptively aggregate heterophilic and homophilic information from neighbors. Particularly for unlabeled data, feature augmentation within the category subspace and consistency regularization driven by entropy minimization ensure that such data can further enhance model performance under explicit semantic guidance. We verify that CGNN exhibits meaningful improvements over other baseline methods under extremely limited labels on four real-world datasets.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China No.2021YFC3300602, the National Natural Science Foundation of China under Grant No.62302287, the Outstanding Academic Leader Project of Shanghai under the grant No.20XD1401700.

References

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2019. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In *International Conference on Learning Representations*.

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 Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI), Vienna, Austria*, 23–29.

Chen, N.; Liu, Z.; Hooi, B.; He, B.; Fathony, R.; Hu, J.; and Chen, J. 2024. Consistency Training with Learnable Data Augmentation for Graph Anomaly Detection with Limited Supervision. In *The Twelfth International Conference on Learning Representations*.

Dong, Y.; Ding, K.; Jalaian, B.; Ji, S.; and Li, J. 2021. Adagnn: Graph neural networks with adaptive frequency response filter. In *Proceedings of the 30th ACM international conference on information & knowledge management*, 392–401.

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.

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.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Jin, M.; Liu, Y.; Zheng, Y.; Chi, L.; Li, Y.-F.; and Pan, S. 2021. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3122–3126.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673.

Kim, J.; Min, Y.; Kim, D.; Lee, G.; Seo, J.; Ryoo, K.; and Kim, S. 2022. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision*, 674–690. Springer.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Kurshan, E.; Shen, H.; and Yu, H. 2020. Financial crime & fraud detection using graph computing: Application considerations & outlook. In *2020 Second International Conference on Transdisciplinary AI (TransAI)*, 125–130. IEEE.

Laine, S.; and Aila, T. 2016. Temporal Ensembling for Semi-Supervised Learning. In *International Conference on Learning Representations*.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896. Atlanta.

Lee, J.; Oh, Y.; In, Y.; Lee, N.; Hyun, D.; and Park, C. 2022. Grafn: Semi-supervised node classification on graph with few labels via non-parametric distribution assignment. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2243–2248.

Li, J.; Xiong, C.; and Hoi, S. C. 2021. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9475–9484.

Li, P.; Yu, H.; Luo, X.; and Wu, J. 2023. LGM-GNN: A local and global aware memory-based graph neural network for fraud detection. *IEEE Transactions on Big Data*.

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.

Liu, Y.; Ding, K.; Lu, Q.; Li, F.; Zhang, L. Y.; and Pan, S. 2023. Towards Self-Interpretable Graph-Level Anomaly Detection. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Philip, S. Y. 2022. Graph self-supervised learning: A survey. *IEEE transactions on knowledge and data engineering*, 35(6): 5879–5900.

Ma, J.; Cui, P.; Kuang, K.; Wang, X.; and Zhu, W. 2019. Disentangled graph convolutional networks. In *International conference on machine learning*, 4212–4221. PMLR.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33: 596–608.

Song, Z.; Yang, X.; Xu, Z.; and King, I. 2022. Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*.

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. PMLR.

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

Wan, S.; Zhan, Y.; Liu, L.; Yu, B.; Pan, S.; and Gong, C. 2021. Contrastive graph poisson networks: Semi-supervised learning with extremely limited labels. *Advances in Neural Information Processing Systems*, 34: 6316–6327.

Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; and Qi, Y. 2019. A semi-supervised graph attentive network for financial fraud detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, 598–607. IEEE.

Xiang, S.; Zhu, M.; Cheng, D.; Li, E.; Zhao, R.; Ouyang, Y.; Chen, L.; and Zheng, Y. 2023. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 14557–14565.

Xu, F.; Wang, N.; Wu, H.; Wen, X.; Zhao, X.; and Wan, H. 2024. Revisiting Graph-based Fraud Detection in Sight of Heterophily and Spectrum. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 9214–9222.

Yang, Y.; Jiang, N.; Xu, Y.; and Zhan, D. 2024. Robust Semi-Supervised Learning by Wisely Leveraging Open-Set Data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(12): 8334–8347.

Yang, Y.; Yang, J.; Bao, R.; Zhan, D.; Zhu, H.; Gao, X.; Xiong, H.; and Yang, J. 2023. Corporate Relative Valuation Using Heterogeneous Multi-Modal Graph Neural Network. *IEEE Trans. Knowl. Data Eng.*, 35(1): 211–224.

Yang, Y.; Zhan, D.; Wu, Y.; Liu, Z.; Xiong, H.; and Jiang, Y. 2021. Semi-Supervised Multi-Modal Clustering and Classification with Incomplete Modalities. *IEEE Trans. Knowl. Data Eng.*, 33(2): 682–695.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 40–48. PMLR.

Yu, H.; Liu, Z.; and Luo, X. 2024. Barely Supervised Learning for Graph-Based Fraud Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16548–16557.

Zhang, G.; Wu, J.; Yang, J.; Beheshti, A.; Xue, S.; Zhou, C.; and Sheng, Q. Z. 2021. FRAUDRE: Fraud Detection Dual-Resistant to Graph Inconsistency and Imbalance. In *2021 IEEE International Conference on Data Mining (ICDM)*, 867–876. IEEE.

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81.

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*.