Barely Supervised Learning for Graph-Based Fraud Detection

Hang Yu*, Zhengyang Liu*, Xiangfeng Luo†

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

Abstract

In recent years, graph-based fraud detection methods have garnered increasing attention for their superior ability to tackle the issue of camouflage in fraudulent scenarios. However, these methods often rely on a substantial proportion of samples as the training set, disregarding the reality of scarce annotated samples in real-life scenarios. As a theoretical framework within semi-supervised learning, the principle of consistency regularization posits that unlabeled samples should be classified into the same category as their own perturbations. Inspired by this principle, this study incorporates unlabeled samples as an auxiliary during model training, designing a novel barely supervised learning method to address the challenge of limited annotated samples in fraud detection. Specifically, to tackle the issue of camouflage in fraudulent scenarios, we employ disentangled representation learning based on edge information for a small subset of annotated nodes. This approach partitions node features into three distinct components representing different connected edges, providing a foundation for the subsequent augmentation of unlabeled samples. For the unlabeled nodes used in auxiliary training, we apply both strong and weak augmentation and design regularization losses to enhance the detection performance of the model in the context of extremely limited labeled samples. Across five publicly available datasets, the proposed model showcases its superior detection capability over baseline models.

Introduction

Fraud detection is a pivotal task that serves to mitigate societal losses. Early fraud detection (Phua et al. 2010; Maes et al. 2002; Hooi et al. 2016) depended on shallow machine learning methods such as decision trees and support vector machines, but most of them are fit to process multidimensional tabular data, and the relation between users is ignored. Actually, scenarios such as fraudulent product reviews, illicit Bitcoin transactions, and account laundering can be modeled as graph-structured data, where the entities involved are represented as nodes and their interactions as edges. Consequently, fraud detection can be transformed into unbalanced node classification, whereby the objective is to discern whether a given node is fraudulent or legitimate.

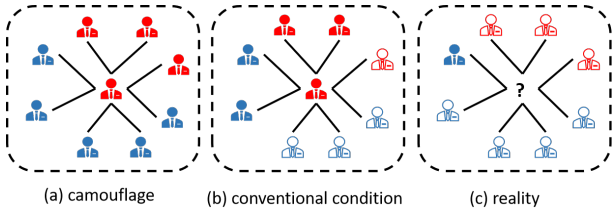


Figure 1: An example to explain the gap in identifying camouflage. Conventional methods rely on certain proportions of labeled neighbors that provide an opportunity to approximately exhibit the typical camouflage, as shown in the middle subplot. On the contrary, it is difficult to identify camouflage with few labeled neighbors in the reality, as shown in the right subplot. The black question mark means it is difficult to indicate the concerned node's class with little information on neighbors' labels, while blue, red, and white nodes indicate normal users, fraudsters, and unlabeled accounts, respectively.

Given the effective expressive capacity of graph neural networks (GNN) (Kipf and Welling 2016; Ren et al. 2023) for graph-structured data, there has been an increasing amount of research on employing GNN for fraud detection (Ma et al. 2018; Liu et al. 2018, 2019; Liang et al. 2019; Liu et al. 2020; Li et al. 2023), which has shown certain levels of success. These studies typically focus on addressing the issue of camouflage in fraud detection. That is to say, the fraudsters disguise themselves by engaging in interactions with normal users, making their characteristics similar to those of normal users. As a result, detection models tend to lean towards predicting the normal class. The camouflage problem challenges the consistency assumption of GNN (Zhu et al. 2020), which posits that neighboring nodes have similar feature representations and belong to the same class as the central node. To tackle this problem, several efforts, such as utilizing attention mechanisms to aggregate neighbors of different types (Wang et al. 2019a) and designing a strategy of sampling neighbors for aggregation (Liu et al. 2021), have been made from different perspectives.

However, annotating samples in the graph-structured data

*These authors contributed equally.
†Corresponding Author is Xiangfeng Luo.

of fraudulent scenarios is a costly endeavor. The aforementioned fraud detection methods often overlook the issue that there are very few labeled samples in real-world scenarios. As shown in Figure 1, insufficient labeled neighbors may prevent these methods from identifying camouflage under such extreme conditions. These methods often assume a significant proportion of samples as training and validation sets, but reality may not always meet such conditions. We need to find ways to achieve strong detection capabilities with limited labeled samples.

To address the issue of camouflage in fraud detection with scarce annotated samples, we propose a barely supervised detection approach based on disentangled representation learning, where barely supervised learning means detecting with a very small proportion (1% or under 1,000 in quantity) of samples labeled for supervised learning. Firstly, we aim to fully utilize a small set of labeled samples. In addition to the basic node classification loss, we introduce a feature disentanglement module to tackle the inevitable camouflage problem in fraud detection. The feature space of nodes is decoupled into different parts according to the type of edges, allowing them to represent different types of connections as distinctively as possible. Subsequently, we employ a portion of unlabeled samples for auxiliary training. Leveraging the decoupled hidden space features obtained from the labeled samples, the unlabeled samples are subjected to both strong and weak augmentations from the perspectives of normality and abnormality. For a given unlabeled sample, based on consistency regularization, the weak augmentations from both perspectives should output similar classification probabilities, while the strong augmentations from opposite perspectives should yield dissimilar probabilities. Through the consistency regularization loss, we measure the consistency and inconsistency of the weak and strong augmentations of the unlabeled samples. This not only assists the feature learning of the labeled samples, but also allows the labeled samples to participate in the training of the unlabeled samples, thereby leveraging the unlabeled samples to enhance the model's detection capability in the scenario of limited annotated samples. The main contributions of this paper can be summarized as follows:

- A feature disentanglement approach is proposed to address the issue of camouflage in fraud detection, while simultaneously laying the groundwork for augmenting unlabeled samples used for auxiliary training.
- A barely supervised learning approach leveraging consistency regularization is proposed to cope with camouflage with few sample annotations by performing weak and strong augmentation on unlabeled samples from both normal and abnormal perspectives.
- Experiments on real-world datasets demonstrate the superiority of our method in comparison with the state-of-the-art fraud detection methods under the situation of few sample annotations.

Related Works

Graph-based Fraud Detection

By aggregating information from neighboring nodes, GNN effectively represents central nodes in graph-structured data, thus arousing wide research interest. Early on, basic graph neural networks were applied to fraud detection. Different aggregators were designed for modeling different relationships, and homogeneous graphs were constructed based on similarity to enhance embedding learning in GAS (Li et al. 2019). FdGars (Wang et al. 2019b) leveraged a predetermined tagging modality to categorize users based on their content and behavioral attributes and detect fraudulent users with multi-layered GCN.

In more recent research, CARE-GNN (Dou et al. 2020) was proposed to address the issue of inconsistency by employing a reinforcement learning approach to adjust thresholds during the process of aggregating neighbors, thereby selecting the neighbors that should be aggregated. FRAUDRE (Zhang et al. 2021), on the other hand, aggregates different relationship neighbors of a node and tackles the problem of class imbalance by applying an unbalanced loss function. PC-GNN (Liu et al. 2021) addresses class imbalance by utilizing a label-balanced sampler to select nodes for training. Additionally, it designs a neighborhood sampler to oversample fraudulent nodes and undersample normal samples within the neighboring nodes. AMNet (Chai et al. 2022) captures features of both normal and abnormal frequency bands with dual filters based on Bernstein polynomials and aggregates them with attention mechanisms. BWGNN (Tang et al. 2022) employs a GNN model based on Beta kernels to effectively handle exceptional high-frequency features by implementing multiple filters for various frequency bands. GHRN (Gao et al. 2023) deleted harmful heterogeneous connections in a graph with approximate pre-trained labels based on any qualified fraud detection models.

Semi-supervised Learning

In real fraud detection scenarios, labeling all the training data comes at a high cost. Semi-supervised learning offers a way to enhance the performance of detection models by utilizing unlabeled samples when labeled samples are limited. Previous studies (Wang et al. 2019a) have focused on the assumption of similarity in features among neighboring nodes and employed random negative sampling to construct an unsupervised loss. (Kumar, Ghosh, and Verma 2022) iteratively used the predicted results from the previous epoch to construct pseudo-labels for the next epoch of prediction, yet fail to fully exploit the correlation between labeled and unlabeled samples.

In recent years, there has been growing interest in the scenario where only scarce-labeled samples are available(Tian and Yu 2023). It is essential to leverage the connections between unlabeled and labeled samples to achieve qualified model performance. (Sohn et al. 2020) initially constructed pseudo-labels with weakly augmented unlabeled images and then estimated class probabilities for strongly augmented versions of the same image. Then, training was conducted with only a small fraction of labeled samples by measur-

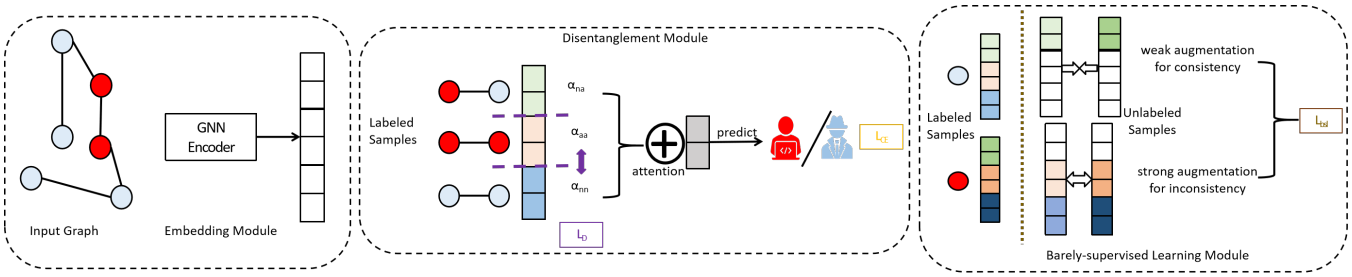


Figure 2: The overall framework of BSL. (a) Embedding module for obtaining node representations based on GNN; (b) Disentanglement module for dividing node representations into different parts that express different types of links; (c) Barely supervised learning module for establishing the relationship between few labeled nodes and the unlabeled nodes.

ing the consistency between the predictions of the weak and strong augmentation through cross-entropy loss. (Lucas, Weinzaepfel, and Rogez 2022; Gui et al. 2022) further explored the challenges of extremely limited supervision. They proposed to enhance model performance by improving the quality of pseudo-labels and categorizing data into superclasses, respectively. In fraud detection, a similar problem of few sample annotations exists. However, due to the sensitivity of the pseudo-label threshold and the complexity of graph structures, the aforementioned methods are not very suitable.

Preliminaries

Hypothesis of Decoupling Node Feature Space

Assuming that the feature space of nodes can be divided into several parts based on different factors, it is highly plausible that the formation of the k-th type of edge is due to a significant correlation between the two nodes in the k-th feature subspace (Ma et al. 2019).

Consistency Regularization

Semi-supervised learning leverages unlabeled data to benefit model training, ensuring that the learned decision boundaries locate in low-density regions, in line with clustering assumptions. In other words, if a slight perturbation is applied to an unlabeled sample, the prediction should not undergo significant changes (Sajjadi, Javanmardi, and Tasdizen 2016). To be more precise, considering an unlabeled sample $x \in D$ and its perturbation $\hat{x}$, our objective is to minimize the distance between two outputs, which can be measured with mean squared error (MSE).

$$d_{MSE}(f_\theta(x), f_\theta(\hat{x})) = \frac{1}{C}\sum_{k=1}^{C}(f_\theta(x)_k - f_\theta(\hat{x})_k)^2 \quad (1)$$

where $f_\theta(x)$ represents the model function and C indicates the number of perturbed samples.

Proposed Method

In this section, we define the problem and present our proposed model, BSL, as shown in Figure 2. Firstly, in the embedding module, a GNN-based encoder is employed to map the node features into a latent space. Subsequently, in the feature disentanglement module, we artificially partition the latent space features obtained from the encoder based on the edge types in fraud detection and design training objectives to express different information. Lastly, in the barely supervised learning module, we establish connections between limited labeled samples and unlabeled samples. Inspired by consistency regularization, weak and strong augmentations for the unlabeled samples from both normal and fraudulent perspectives are constructed for the design of the barely supervised learning loss.

Problem Definition

An attributed graph can be defined as $\mathcal{G} = \{V, X, E, A\}$, where $V = \{v_1, v_2, ..., v_N\}$ is the set of N nodes of G, $E = \{e_{ij}\}$ is the set of edges, and $e_{ij} = (v_i, v_j)$ represents the link between node v_i and v_j. There is a d-dimension origin feature $x_i \in R^d$ for each node v_i, constructing the feature matrix $X \in R^{N\times d}$, while A is the adjacency matrix based on E.

Given an attributed graph G defined above, fraud detection can be transformed into the problem of node classification. Specifically, each node should be predicted as either normal associated with a label of 0, or fraud associated with a label of 1, i.e., $V \rightarrow Y = \{0, 1\}^n$. The task is to predict the labels of the remaining nodes in the graph while few nodes are labeled.

Embedding Module

To express graph-structured data, we employ a GNN to embed the input graph. Specifically, a linear transformation is applied to project the node attribute features into a latent space.

$$\hat{X} = \sigma(FC(X)) \quad (2)$$

where σ is the activation function, and FC is the fully connected layer.

Then GNN is used to obtain the central node feature in the latent space by aggregating neighboring information.

$$Z = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}\hat{X}\theta \quad (3)$$

where $\hat{X} \in R^{n\times d}$ is the node feature matrix in the latent space, n represents the number of nodes, d represents the

dimension of latent space, $\hat{A}$ represents the adjacency matrix with added self-loop, θ is a learnable parameter in the GNN, and $\hat{D}$ is the degree matrix for Laplace transformation.

Disentanglement Module

According to the clustering assumption that samples of the same class have similar features, neighbors of the same class should dominate the aggregation process. Inspired by this notion, we have devised a feature disentanglement module. In the hypothesis of decoupling node feature, we postulate that the formation of the k-th type of edge is likely to be influenced by the sufficient correlation between the two nodes of that edge in the k-th feature subspace. In the fraud detection task, there exist two types of nodes: normal nodes and abnormal nodes, giving rise to three types of edges: normal-abnormal(NA), abnormal-abnormal(AA), and normal-normal(NN). Therefore, we can divide the feature space of the nodes into three equal parts, as illustrated in Figure 2, and then devise a loss function to attempt to achieve decoupled effects in the three feature subspaces of NA, AA, and NN.

We utilize edge classification to ensure that each feature subspace corresponds to a distinct type of connection, and then employ an attention mechanism to learn the weights of different feature subspaces. Since, in essence, each node only has two types of edges, AA and NN are actually mutually exclusive. We introduce a weight contrastive loss to further unravel the correlation between these two weights.

Edge Classification Loss For all the edges of labeled nodes, we calculate the probability of each type of edge and employ classification loss to enable each feature subspace to represent specific types of edges.

$$p_{ij}^k = \sigma(FC(z_i^k - z_j^k)), k \in \{na, aa, nn\}, j \in Neighbor(i) \tag{4}$$

$$P_{ij}^k = softmax(p_{ij}^k) = \frac{exp(p_{ij}^k)}{\sum_m exp(p_{ij}^m)} \tag{5}$$

$$L_{link} = -\sum_k y_{ij} log(P_{ij}^k) \tag{6}$$

where σ is the activation function and FC is a fully-connected layer. The latent space feature of annotated node i is denoted as z_i. Being averagely partitioned, the vectors for NA, AA, and NN are respectively represented by z_i^{na}, z_i^{aa}, and z_i^{nn}. j represents one of the neighbors of node i, and k denotes one of the three distinct feature subspaces.

By inputting the features of two nodes within a certain subspace and applying softmax, we obtain the probability P_{ij}^k of a certain type for a given edge. Then the edge classification loss L_{link} is calculated with cross-entropy, where y_{ij} is the label of the edge between i and j.

Weight Comparison Loss We partitioned the feature space of the nodes into three parts. However, when making predictions on the samples, it is necessary to have a unified vector as input for the classifier. By employing an attention mechanism, we assess the significance of each feature subspace in contributing to the vector that is ultimately used for classification. The specific calculation process is as follows:

$$\omega_i^k = q \cdot \sigma(W^k \cdot (z_i^k)^T + b^k) \tag{7}$$

$$\alpha_i^k = softmax(\omega_i^k) = \frac{exp(\omega_i^k)}{\sum_m exp(\omega_i^m)} \tag{8}$$

where q represents the shared parameter, W^k and b^k are the learnable weight matrix and bias. ω_i^k indicates the attention value, which is input into the softmax. Then attention weight α_i^k is obtained.

In regard to each node, it is impossible for AA and NN edges to coexist. To mitigate the interference caused by the noise of the non-existent third subspace in each node, we implement a contrastive loss to further reduce their coexistence. Specifically, for normal samples, we increase the weight of NN edges while the weight of AA edges is decreased. For abnormal samples, we carry out the above idea conversely. As such, the calculation for L_{attn} is as follows:

$$L_{attn} = (1 - y_i) \cdot (\alpha_i^{nn} - \alpha_i^{aa}) + y_i \cdot (\alpha_i^{aa} - \alpha_i^{nn}) \tag{9}$$

where y_i is the label of node i, and 0 and 1 represent normal and abnormal, respectively.

The disentanglement loss is composed of the edge classification loss L_{link} and the weight contrastive loss L_{attn}, which can be expressed as follows:

$$L_D = L_{link} + L_{attn} \tag{10}$$

For labeled samples, we deal with them with supervised learning. By utilizing attention weights to appropriately weigh the three distinct feature subspaces, an input is generated for the classifier. The classification loss can then be calculated accordingly:

$$\hat{z}_i = \sum_k \alpha_i^k \cdot z_i^k \tag{11}$$

$$\hat{y_i} = predict(\hat{z}_i) = softmax(W \cdot (\hat{z}_i) + b) \tag{12}$$

$$L_{class} = -\sum_c y_i^c log(\hat{y}_i^c) \tag{13}$$

where $\hat{z}_i$ is the input feature for the classifier aggregated with attention weights. $\hat{y_i}$ represents the predicted probability vector for node i, W and b are learnable parameters for the classifier. L_{class} calculates the classification loss with cross-entropy, where y_i^c represents the label of node i on class c, and $\hat{y}_i^c$ is the corresponding predicted probability.

Barely Supervised Learning Module

The disentanglement module partitions a node's feature into three subspaces, with the NA subspace having limited capacity for characterizing a node's category due to its purpose for camouflage. Conversely, the subspace of the same-class nodes, such as AA or NN, plays a decisive role in determining a node's category. Significantly, much less supervised information (e.g. 1% labeled samples or under 1,000 in quantity) is available compared with semi-supervised learning,

which makes it harder for representation learning for these three subspaces. Therefore, an approach with scarce labeled samples (called barely supervised learning), is designed inspired by consistency regularization.

Specifically, we implement the augmentation of an unlabeled sample from both normal and abnormal perspectives through weak and strong augmentations, resulting in four augmented versions. Of the labeled samples, a random normal sample n and an abnormal sample a are selected as tools for the augmentation of unlabeled samples.

$$z_n = [z_n^{na}, z_n^{aa}, z_n^{nn}] \tag{14}$$

$$z_a = [z_a^{na}, z_a^{aa}, z_a^{nn}] \tag{15}$$

where z_n and z_a are features of labeled nodes before aggregation with attention.

As the NA subspace has limited influence on the representation of node categories, we replace z_i^{na} with z_n^{na} and z_a^{na} respectively, to obtain weak augmentation for the unlabeled sample i from the perspectives of both normality and abnormality, as demonstrated in the following equations.

$$z_i^{nw} = [z_n^{na}, z_i^{aa}, z_i^{nn}] \tag{16}$$

$$z_i^{aw} = [z_a^{na}, z_i^{aa}, z_i^{nn}] \tag{17}$$

According to the principle of consistency regularization, the above two slight perturbations will not significantly alter the classification for sample i. Hence, after being mapped by the classifier, the weak augmentations from both perspectives should exhibit close distances. This leads to the derivation of the consistency loss L_{con}:

$$L_{con} = \sum \left(predict(z_i^{nw}) - predict(z_i^{aw})\right)^2 \tag{18}$$

Similarly, given the decisive role that AA and NN feature subspaces play in classification, we substitute z_n^{aa} and z_a^{aa} for z_i^{aa}, and z_n^{nn} and z_a^{nn} for z_i^{nn}, to obtain strong augmentations for unlabeled sample i from both normal and abnormal perspectives.

$$z_i^{ns} = [z_i^{na}, z_n^{aa}, z_n^{nn}] \tag{19}$$

$$z_i^{as} = [z_i^{na}, z_a^{aa}, z_a^{nn}] \tag{20}$$

As fraud detection is a binary classification problem, in conjunction with the principle of consistency regularization, it can be assumed that the strong augmentations from the two different perspectives are opposite. Therefore, after being mapped by the classifier, the distance between the strong augmentations from the two perspectives should be far apart, resulting in the inconsistency loss L_{incon}:

$$L_{incon} = -\sum \left(predict(z_i^{ns}) - predict(z_i^{as})\right)^2 \tag{21}$$

As such, a bridge between labeled and unlabeled samples is established, and the loss function of barely supervised learning is composed of a consistency loss L_{con} and an inconsistency loss L_{incon}.

$$L_{bsl} = L_{con} + L_{incon} \tag{22}$$

Objective Function

The overall loss of our model comprises three components: the supervised classification loss L_{class}, the feature disentanglement loss L_D, and the barely supervised learning loss L_{bsl}. Thus, the computation formula for the total loss is as follows:

$$L = L_{class} + \alpha \cdot L_D + \beta \cdot L_{bsl} \tag{23}$$

where α and β serve as hyperparameters that convey the weights of loss terms. In the testing phase, we utilize the classification outcome $\hat{y}_i$ directly derived from the classifier.

Experiments

Experiment Setup

In this section, we conducted comparative experiments on five real-world datasets using ten currently representative or novel methods to test the performance of our proposed approach. Additionally, we conducted ablation experiments to test the effectiveness of the designed modules. These contents were aimed at addressing the following questions:

Q1: Does BSL outperform the state-of-the-art approaches in the task of fraud detection with few samples annotated?

Q2: Can the incorporation of the feature disentanglement and barely supervised learning modules in BSL lead to an improvement in detection performance?

Q3: Is BSL sensitive to the key hyperparameters?

Q4: Can the feature disentanglement in BSL achieve the desired effect as anticipated?

Datasets We employ five datasets to evaluate the effectiveness of BSL, with their specific statistical characteristics presented in Table I. **Amazon** and **Yelp** (McAuley and Leskovec 2013) are datasets of fraudulent comments, where users or comments are modeled as nodes. **Elliptic** (Weber et al. 2019) constructs a transaction network of bitcoin, where nodes are transactions and edges represent flows of transactions. **T-Finance** and **T-Social** (Tang et al. 2022) are fraud datasets concerned on a transaction network or social network, where accounts are modeled as nodes and two accounts are linked if there is a transaction or a friendship between them.

In the experiment, each dataset is divided into three parts - a small proportion as labeled samples, another part as unlabeled samples for barely supervised learning, and the remainder used as the test set. For Amazon, Yelp, Elliptic, and T-Finance datasets, the split is 1% : 10%: 89%, while that for T-Social is 0.01%: 0.1%: 99.89% due to its large scale. As there are no labeled samples for constructing the validation set, the training loss is used to select the optimal model for

Dataset	Amazon	Yelp	Elliptic	T-Finance	T-Social
Nodes	11,944	45,954	46,564	39,357	5,781,065
Edges	4,398,392	3,846,979	73,248	21,222,543	73,105,508
Features	25	32	93	10	10
Fraud(%)	6.87	14.53	9.76	4.58	3.01

Table 1: The characteristics of the datasets.

Methods \ Dataset	Amazon(1%)			Yelp(1%)			Elliptic(1%)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
GCN	63.86	63.95	63.67	42.17	50.00	45.75	64.02	60.53	61.86
GraphSage	45.23	50.00	47.49	42.17	50.00	45.75	62.77	64.03	63.35
GAT	54.07	51.34	50.89	42.17	50.00	45.75	58.17	54.50	55.35
Semi-GNN	45.23	50.00	47.49	42.17	50.00	45.75	74.60	78.72	75.97
CARE-GNN	76.77	86.19	80.42	56.67	62.33	55.08	67.06	80.54	70.38
FRAUDRE	77.16	86.51	80.29	42.17	50.00	45.75	65.12	77.50	66.86
PC-GNN	91.10	85.69	88.59	59.16	61.15	59.57	64.39	81.31	66.10
AMNet	93.79	84.13	88.56	42.17	50.00	45.75	66.17	79.15	70.03
BWGNN	78.83	78.31	78.28	61.85	62.02	61.92	87.34	79.80	82.99
GTAN	**96.24**	79.21	85.28	42.17	50.00	45.75	79.63	76.54	77.93
ours	95.08	**88.44**	**91.42**	**61.87**	**64.83**	**62.83**	**91.83**	**84.86**	**87.92**

Methods \ Dataset	T-Finance(1%)			T-Social(0.01%)		
	Precision	Recall	F1	Precision	Recall	F1
GCN	58.24	73.38	60.27	60.77	54.32	55.19
GraphSage	61.48	53.54	54.91	49.61	50.31	49.81
GAT	65.20	53.03	54.34	49.47	50.98	50.21
Semi-GNN	65.97	53.80	54.93	49.48	50.98	50.22
CARE-GNN	63.62	83.51	67.87	51.63	61.54	45.79
FRAUDRE	56.05	73.92	55.52	50.80	55.67	44.75
PC-GNN	67.08	**85.19**	72.21	51.31	60.19	43.30
AMNet	66.69	79.45	70.84	OOM	OOM	OOM
BWGNN	84.51	74.62	78.62	69.86	75.48	72.14
GTAN	87.30	75.48	80.08	48.49	50.00	49.43
ours	**88.55**	83.97	**86.10**	**79.32**	**75.58**	**75.67**

Table 2: Performance(%) of fraud detection on five datasets.

testing. As for other baseline models, 10% (or 0.1%) samples are set as the validation set with labels known, which is looser than the scenario faced by our method.

Compared Methods We test our method by comparing it with ten representative and up-to-date GNN-based fraud detection models. Of these, GCN (Kipf and Welling 2016), GraphSage (Hamilton, Ying, and Leskovec 2017), GAT (Veličković et al. 2017) are basic graph neural networks. CARE-GNN (Dou et al. 2020), FRAUDRE (Zhang et al. 2021), PC-GNN (Liu et al. 2021) are recent detection models from the spacial perspective with strategies to choose neighbors and cope with the class imbalance problem. AMNet (Chai et al. 2022) and BWGNN (Tang et al. 2022) analyze the features of normal and anomalous nodes from the perspective of a spectral domain, constructing GNN filters in different frequency bands to extract node features. Semi-GNN (Wang et al. 2019a) and GTAN (Xiang et al. 2023) are semi-supervised models trained with information from the training set.

Experiment Settings and Implementation The parameters are optimized by Adam (Kingma and Ba 2014), while the learning rate is set as 0.01, and the weight decay rate is 1e-5. We set the embedding size to 96, and the batch size to 256 on five datasets.

Our method is implemented in PyTorch 1.13.1 (Paszke et al. 2019) and PyG (Fey and Lenssen 2019) with Python 3.9, and DGL (Wang et al. 2019c) is used for preprocessing the data. Most of the other compared methods are implemented with code published by their authors.

Evaluation metrics We evaluate the effectiveness of BSL with the following three metrics, macro-precision, macro-recall, and macro-F1, because these metrics reflect the performance of models on imbalanced data without overly underestimating the minority (fraud) class.

Overall Results (RQ1)

The detection performance of the proposed BSL with baselines evaluated with macro-precision, macro-recall, and macro-F1 on Amazon, Yelp, Elliptic, T-Finance, and T-Finance datasets is reported in Table 2. It can be observed that basic GNN models, such as GCN, GAT, and GraphSage fail to conduct qualified fraud detection with few samples labeled. When compared to recently proposed models under looser conditions, BSL is superior or comparable according to all three evaluation metrics on the five datasets in most cases.

Macro-F1 weighs the classification performance of fraud detection comprehensively. It can be observed that BSL achieves the best performance on each of the five datasets for overall detection capability. CARE-GNN filters neighbors with a unified adopted threshold, which may lead to the misjudgment of a significant proportion of them. FRAU-

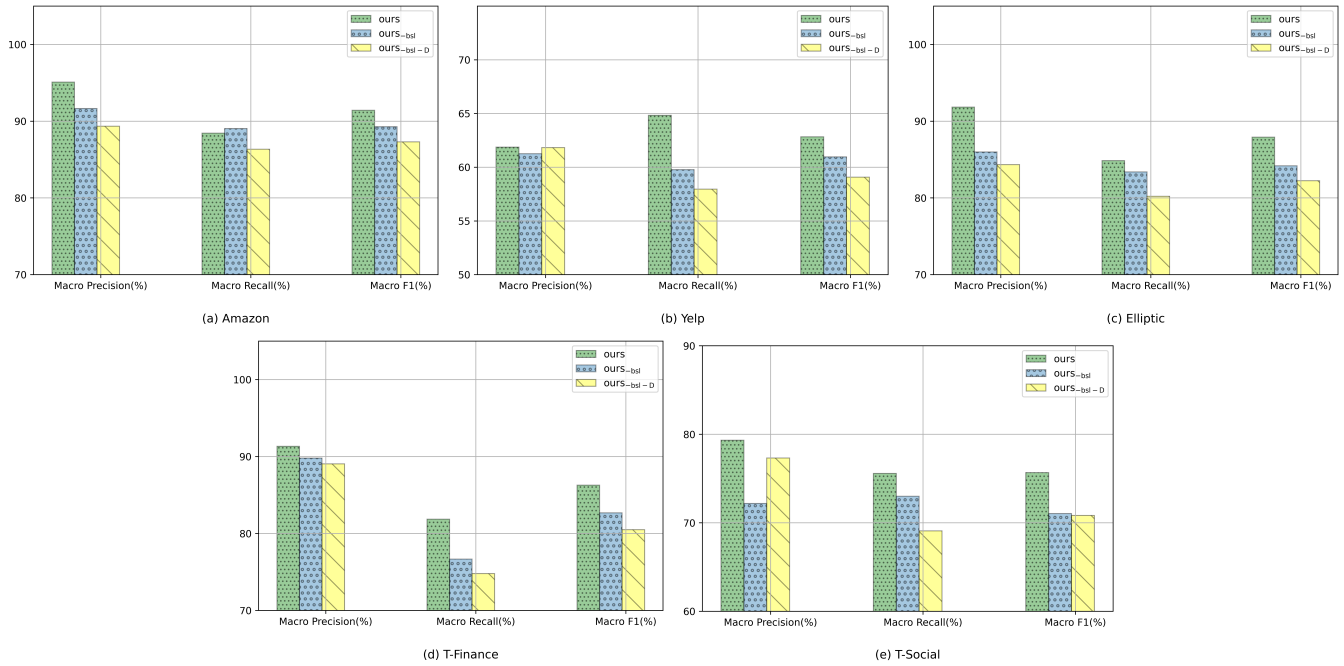


Figure 3: Ablation analysis for five datasets.

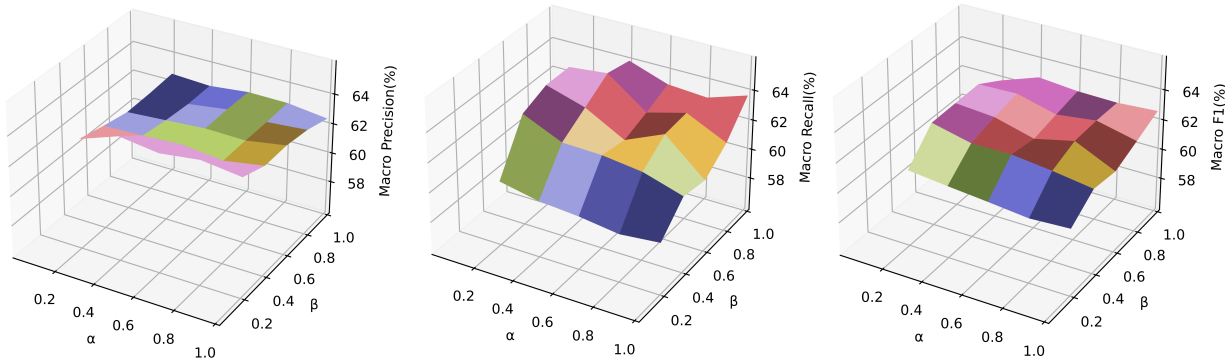
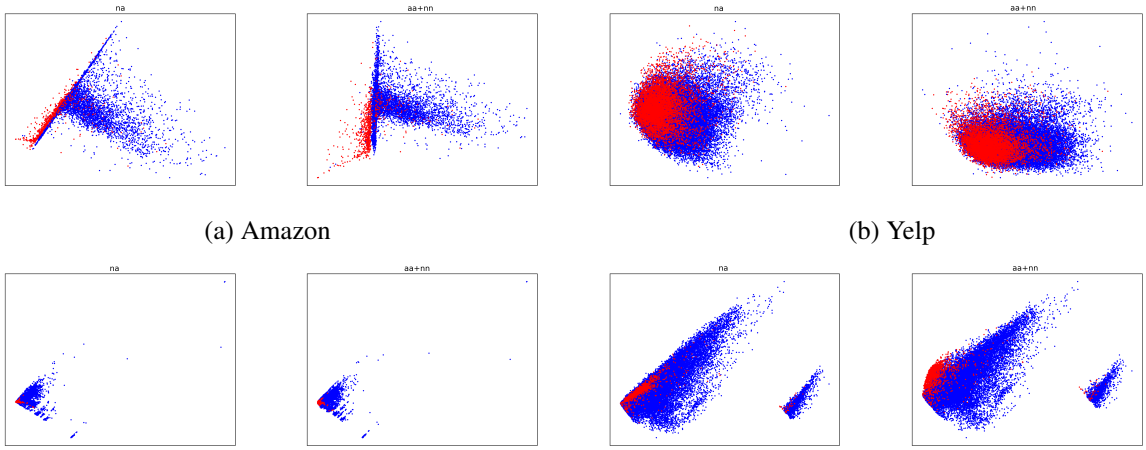


Figure 4: Parameter sensitivity analysis of BSL on Yelp using three metrics: macro-precision, macro-recall, macro-F1.

DRE and PC-GNN are concerned about class imbalance and fail to cope very well with camouflage. AMNet achieves a comparable performance on Amazon and Yelp, but is limited in scalability(OOM, out of memory) because of its requirement of the whole graph as input. BWGNN tries to solve the over-smoothing issue from the spectral domain and achieves good results on most datasets. Semi-GNN utilizes an attention mechanism to aggregate different types of neighbors, but the designed unsupervised loss fails to cope with camouflage in fraud detection. GTAN models risk propagation by adding the transaction label as one of the transaction categorical attributes to detect fraudulent transactions, which may deteriorate obviously in a difficult dataset (Yelp) or under an extremely harsh condition(0.01% labeled samples on T-Social). When it comes to macro-recall and macro-precision, BSL outscores all baseline methods in most cases, while it seldom falls behind a bit on one metric and exceeds more on the others under stricter conditions.

Ablation Study (RQ2)

To evaluate the effectiveness of the two key designed modules of BSL, we conducted a series of ablation experiments by excluding each module, as shown in Figure 3, where ours-BSL excludes the barely supervised learning module, and ours-BSL-D excludes the barely supervised learning module and the disentanglement module. It can be observed that the barely supervised learning module plays a useful role in improving the performance of detection on all five datasets. Meanwhile, the introduction of the feature disentanglement module also makes sense and lays a good foundation for barely supervised learning in most cases.



(a) Amazon (b) Yelp

(c) Elliptic (d) T-Finance

Figure 5: The visualization of disentangled representation with TSNE.

Sensitivity to Hyper-Parameters (RQ3)

Weight factor analysis of the loss function. As shown in Eq.(23), the two factors α and β are critical in balancing the supervised classification loss, the disentanglement loss, and the barely supervised loss. Hence, we evaluate our model's sensitivity to these terms under different settings.

Specifically, we run BSL on the Yelp dataset for $\alpha, \beta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ based on grid search and report the results of all three evaluation metrics in Figure 4. It can be observed that BSL is not sensitive to the two loss weight parameters α and β in terms of macro F1. In particular, BSL achieves better macro-F1 when α and β are around 0.4 and 0.8, respectively; that is to say, the most appropriate ratio of the supervised classification loss, the disentanglement loss, and the barely supervised loss is 1:0.4:0.8 if the best comprehensive detection performance is needed.

Visualization Analysis (RQ4)

T-distributed stochastic neighbor embedding (TSNE) (Van der Maaten and Hinton 2008) is a feature visualization tool that can transform high-dimensional features into two-dimensional features to facilitate observation of the pattern of feature distribution. To evaluate whether the design of the disentanglement module achieves the expected effect, we utilize TSNE on all datasets, except T—Social because of its large scale, to reduce the dimension of the samples' latent feature space for the NA vector and the AA+NN vectors, as shown in Figure 5, which shows the feature distribution after being handled with TSNE. It can be observed that the NA vector spaces of fraudulent and normal nodes are more coupled, while the AA+NN spaces present a more distinguishable border between fraud and normal nodes.

Conclusion

In this paper, we aim to address the issue of camouflage in fraud detection in the presence of scarce sample annotations. Our proposed method, BSL, consists of a feature disentanglement module and a barely supervised learning module. The feature disentanglement module decouples the feature space of nodes into different parts based on types of edges, which is used to address the issue of camouflage and provide a foundation for data augmentation in subsequent barely supervised learning. The barely supervised learning module conducts strong and weak augmentation on unlabeled samples with labeled samples that have undergone feature disentanglement, enabling unlabeled samples to participate in the training of labeled ones. Under the condition of scarce sample annotation, these modules learn distinguishable representations for fraudulent and normal classes. Across five public datasets, BSL demonstrates superior detection capability over baseline models in terms of evaluation metrics.

Acknowledgments

This work was supported by National Natural Science Foundation of China Youth Fund (No.62302287), the National Key Research and Development Program of China (2021YFC3300602) and Shanghai Yangfan Program under Grant 22YF1413600.

References

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.

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.

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*.

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.

Gui, G.; Zhao, Z.; Qi, L.; Zhou, L.; Wang, L.; and Shi, Y. 2022. Improving barely supervised learning by discriminating unlabeled samples with super-class. *Advances in Neural Information Processing Systems*, 35: 19849–19860.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 1024–1034.

Hooi, B.; Song, H. A.; Beutel, A.; Shah, N.; Shin, K.; and Faloutsos, C. 2016. Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 895–904.

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

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kumar, A.; Ghosh, S.; and Verma, J. 2022. Guided Self-Training based Semi-Supervised Learning for Fraud Detection. In *Proceedings of the Third ACM International Conference on AI in Finance*, 148–155.

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, 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*, (01): 1–13.

Liang, C.; Liu, Z.; Liu, B.; Zhou, J.; Li, X.; Yang, S.; and Qi, Y. 2019. Uncovering insurance fraud conspiracy with network learning. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1181–1184.

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, Z.; Chen, C.; Li, L.; Zhou, J.; Li, X.; Song, L.; and Qi, Y. 2019. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4424–4431.

Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; and Song, L. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2077–2085.

Liu, Z.; Dou, Y.; Yu, P. S.; Deng, Y.; and Peng, H. 2020. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd international ACM SIGIR Conference on Research and Development in Information Retrieval*, 1569–1572.

Lucas, T.; Weinzaepfel, P.; and Rogez, G. 2022. Barely-supervised learning: Semi-supervised learning with very few labeled images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1881–1889.

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.

Ma, J.; Zhang, D.; Wang, Y.; Zhang, Y.; and Pozdnoukhov, A. 2018. GraphRAD: a graph-based risky account detection system. In *Proceedings of ACM SIGKDD Conference, London, UK*, volume 9.

Maes, S.; Tuyls, K.; Vanschoenwinkel, B.; and Manderick, B. 2002. Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies*, volume 261, 270.

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.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Phua, C.; Lee, V.; Smith, K.; and Gayler, R. 2010. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

Ren, L.; Chen, J.; Liu, T.; and Yu, H. 2023. OD-Enhanced Dynamic Spatial-Temporal Graph Convolutional Network for Metro Passenger Flow Prediction. In *International Conference on Neural Information Processing*, 72–85. Springer.

Sajjadi, M.; Javanmardi, M.; and Tasdizen, T. 2016. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in Neural Information Processing Systems*, 29.

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.

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.

Tian, P.; and Yu, H. 2023. Can we improve meta-learning model in few-shot learning by aligning data distributions? *Knowledge-Based Systems*, 277: 110800.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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

Wang, J.; Wen, R.; Wu, C.; Huang, Y.; and Xiong, J. 2019b. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Companion Proceedings of the 2019 World Wide Web Conference*, 310–316.

Wang, M.; Zheng, D.; Ye, Z.; Gan, Q.; Li, M.; Song, X.; Zhou, J.; Ma, C.; Yu, L.; Gai, Y.; et al. 2019c. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

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 preprint arXiv:1908.02591*.

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*, 14557–14565.

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.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33: 7793–7804.