Voucher Abuse Detection with Prompt-based Fine-tuning on Graph Neural Networks

Zhihao Wen*
zhwen.2019@smu.edu.sg
Singapore Management University
Singapore

Yuan Fang†
yfang@smu.edu.sg
Singapore Management University
Singapore

Yihan Liu
yihan.liu@lazada.com
Lazada Inc.
Singapore

Yang Guo
yg.357086@lazada.com
Lazada Inc.
Singapore

Shuji Hao†
hao.shuji@gmail.com
Lazada Inc.
Singapore

ABSTRACT

Voucher abuse detection is an important anomaly detection problem in E-commerce. While many GNN-based solutions have emerged, the supervised paradigm depends on a large quantity of labeled data. A popular alternative is to adopt self-supervised pre-training using label-free data, and further fine-tune on a downstream task with limited labels. Nevertheless, the "pre-train, fine-tune" paradigm is often plagued by the objective gap between pre-training and downstream tasks. Hence, we propose VPGNN, a prompt-based fine-tuning framework on GNNs for voucher abuse detection. We design a novel graph prompting function to reformulate the downstream task into a similar template as the pretext task in pre-training, thereby narrowing the objective gap. Extensive experiments on both proprietary and public datasets demonstrate the strength of VPGNN in both few-shot and semi-supervised scenarios. Moreover, an online evaluation of VPGNN shows a 23.4% improvement over two existing deployed models.

CCS CONCEPTS

• **Information systems** → **Clustering and classification**; **Enterprise applications**.

KEYWORDS

Anomaly detection, graph neural networks, pre-training, prompt.

ACM Reference Format:
Zhihao Wen*, Yuan Fang†, Yihan Liu, Yang Guo, and Shuji Hao†. 2023. Voucher Abuse Detection with Prompt-based Fine-tuning on Graph Neural Networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615505

*Part of the work done during an internship at Lazada Inc. †Corresponding authors.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615505

1 INTRODUCTION

Amidst vigorous market rivalry, user acquisition has been a crucial metric for E-commerce platforms. One of the primary strategies is to introduce electronic *vouchers*, which could help attract more new users or encourage existing users to buy more. However, the widespread use of vouchers in E-commerce also provides opportunities for abusers. They usually start by registering a massive number of accounts and placing orders just to exploit the vouchers intended for new users. They either resell the goods (often bought with a deep discount after applying the vouchers) at a higher market price or collude with the sellers by placing orders with vouchers in the sellers' own stores. Such behaviors not only cause losses for E-commerce platforms but also damage the ecosystem for legitimate users. Thus, it is of great significance to detect orders of these abusive users and prevent them from collecting vouchers.

In this paper, we study the problem called *Voucher Abuse Detection*, which aims to detect the orders of abusive users in the E-commerce industry. A key insight into voucher abuse detection is the network structures among orders. For example, even when orders are placed through different accounts, they can still be related if they share the same device or the same delivery address. As illustrated in Fig. 1(a), the order graph encodes rich relationships and patterns between orders, which can help differentiate the behaviors of legitimate and abusive users. As shown in Fig. 1(b), a legitimate user typically only logs into one account on one or two devices, and applies one voucher on their first order to utilize the new buyer incentive. In contrast, as shown in Fig. 1(c), an abusive user often employs a large number of devices, and in each device, they create multiple accounts with multiple sets of information (*e.g.*, e-mail and mobile number). In each account, they collect the voucher for new buyer incentives and just place one order with that voucher. By distinguishing the graph structures related to legitimate and abusive orders, we cast the problem of voucher abuse detection as *binary node classification* on graphs.

While voucher abuse detection is a subclass of anomaly detection, existing solutions are not ideal choices for the specific problem of voucher abuse detection. On one hand, although traditional machine learning methods [5–7] are widely used in industry, they do not leverage important graph structure information. Hence, many recent attempts for anomaly detection [30] turn to graph neural networks (GNNs) [43] to exploit structure information. On the

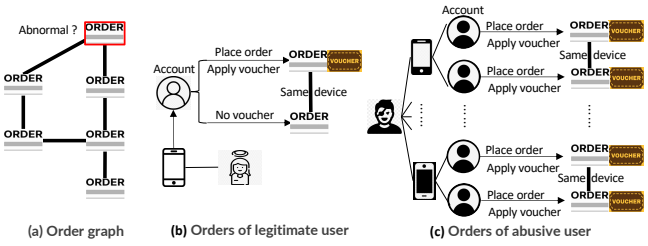


Figure 1: Illustration of voucher abuse detection.

other hand, voucher abusers often adopt *variable strategies* from time to time to reduce detection by the platform. Thus, *timely* and *high-quality* labeled instances of abusive order are crucial for detection, but they are very limited in the production environment of real business scenarios. Hence, most GNN-based approaches cannot cope well in a supervised manner. Meanwhile self-supervised GNNs [14, 15, 32] become promising as they aim to capture intrinsic graph patterns without requiring any annotated label. However, most self-supervised approaches follow the "pre-train, fine-tune" paradigm [14, 15], which suffers from a major drawback: There exists an objective gap between the pre-training and downstream tasks, impairing the generalization of the pre-trained model.

Challenges and our work. To bridge the gap between GNN pre-training and downstream tasks, we propose a framework for Voucher abuse detection with Prompt-based fine-tuning on Graph Neural Networks (VPGNN). We introduce prompting [12, 20] to graph-based tasks, which reformulates the downstream node classification into a similar form as the pretext task in pre-training. While increasingly popular in NLP, prompt-based learning for GNNs still presents two major challenges.

First, we cannot directly apply the textual prompting function to bridge various graph-based tasks, as textual prompts are incompatible with fundamental graph elements (*e.g.*, nodes and edges). Therefore, in this paper, we propose a *graph prompting function* that reformulates the downstream node classification problem into a pairwise matching task between node tokens and context tokens. Specifically, each context token represents a node class, and node classification is achieved by selecting the most likely class (*i.e.*, context token) that matches the node token. This pairwise template would be consistent with the paired formulation of many popular pretext tasks in graph pre-training [13–15, 38].

Second, it is still unclear how to initialize the context tokens before any tuning. The context tokens should be 1) informative, to fully exploit prior knowledge learned during pre-training, which is also the initial intention of prompting; 2) robust, especially in low-resource scenarios with limited task labels. For informativeness, we reuse the graph readout function from pre-training to initialize the context tokens downstream; for robustness, we augment the limited labeled nodes with their local subgraphs. Finally, the context tokens and the pre-trained GNN model are fine-tuned together for downstream classification.

Contributions. We summarize the contributions of this work. (1) We propose a novel framework called VPGNN, with a complete pipeline to pre-train and generalize GNNs for voucher abuse detection. (2) We design a graph prompting function to reformulate the downstream node classification task, which aligns better with the pretext task. (3) In offline experiments, VPGNN can outperform state-of-the-art baselines by up to 4.4% in the 10-shot setting; in online evaluation, VPGNN shows a 23.4% improvement over two existing deployed models.

2 RELATED WORK

Recently, GNNs [13, 17, 37, 43, 44] have enjoyed widespread application in industry. Motivated by their success, many GNN-based anomaly detection algorithms have emerged, including GAS [18], FdGars [40], GraphConsis [27] and CARE-GNN [10] for review fraud detection, GeniePath [25] and SemiGNN [39] for financial fraud detection, FANG [31] for fake news detection, ASA [42] for mobile fraud detection, and MTAD-GAT [45] for time-series anomaly detection. There are also some unsupervised anomaly detection GNNs [1, 9, 16, 23], but they are often less reliable since they do not make use of any labeled data. To reduce labeling requirements, GNN pre-training [14, 15, 22, 29, 32] has become a popular alternative, which aims to capture general patterns on label-free graphs. Nevertheless, a considerable amount of labeled data are still required for fine-tuning on downstream tasks.

In NLP, prompting [20] has emerged to overcome the objective gap between the pretext and downstream tasks. Prompting reformulates the downstream task to follow a similar template as the pretext task so that the downstream task can be optimized with a light tuning or even without tuning [3]. On graphs, there have also been some attempts to leverage prompt-based learning. GPF [11] only trains a prefix prompt vector appended to the node features, lacking a unified template for pretext and downstream tasks. GPPT [35] and GraphPrompt [28] attempt to unify the pretext task of link prediction and downstream classification, but the unification is incompatible with voucher abuse detection, where abusive orders are linked with many legitimate orders.

3 PRELIMINARIES

Pre-training. Many pretext tasks [13, 14, 38] have been proposed to pre-train GNNs. In voucher abuse detection, abusive orders are the minority and the majority are legitimate orders. Hence, we utilize DGI [38] to maximize the local-global mutual information, whereby the graph-level global information captures the "normal" patterns manifested by the majority, which helps indicate the extent of a node deviating from what is normal.

Specifically, let $\mathbf{H}$ be the node representation matrix in which each row $\mathbf{h}_i$ is the representation of node i generated by a GNN encoder, parameterized by θ. Furthermore, let $\mathbf{h}_G$ be the global representation of G, given by $\mathbf{h}_G = \Omega(\mathbf{H}; \omega) = \text{POOL}(\{\mathbf{h}_i \mid i \in V\})$ where Ω is a readout function with parameters ω. In pre-training, DGI [38] aims to minimize the following loss:

$$\arg\min_{\theta,\omega,\phi} \sum_{(i,G)} \mathcal{L}^{\text{pre}}\left(\Phi^{\text{pre}}\left(\mathbf{h}_i, \mathbf{h}_G; \phi\right), \text{match}(i, G)\right), \quad (1)$$

where Φ^{pre}, parameterized by ϕ, is a projection head to evaluate the matching score of a node-graph pair (i, G), which measures the local-global consistency. Note that the supervision comes from $\text{match}(i, G)$, which is an indicator function: 1 if node i is from the original graph G; 0 if node i is from a corrupted version of G. Hence, the pre-training process does not require any human annotation,

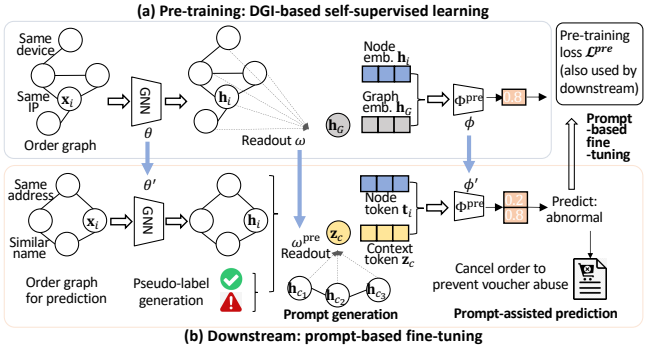


Figure 2: Overall framework of VPGNN.

updating model parameters including θ (GNN), ω (readout), and ϕ (projection head) in a self-supervised manner.

Fine-tuning. The pre-trained GNN parameters θ^{pre}, optimized by Eq. (1), serve as a good initialization for downstream classification tasks. Following the "pre-train, fine-tune" paradigm, the initialization is further fine-tuned together with a new set of classification weights ψ, by optimizing the following:

$$\arg\min_{\theta',\psi} \sum_{i \in V} \mathcal{L}^{\text{down}}\big(\Phi^{\text{down}}(\mathbf{h}_i;\psi), y_i\big), \tag{2}$$

where Φ^{down} is a new projection head with randomly initialized parameters ψ, replacing the pretext projection head Φ^{pre}. $\mathcal{L}^{\text{down}}$ is the downstream classification loss (*e.g.*, cross entropy), and y_i is the task-specific label of node i.

4 PROPOSED APPROACH

In this section, we give an overview of our approach and elaborate on our prompt-based fine-tuning.

4.1 Overview of VPGNN

Unlike textual prompts in NLP, on graph data, it is non-trivial to design the prompts due to the incompatibility between traditional textual prompts and graph elements. To materialize prompt-based fine-tuning on graphs, our VPGNN has two major stages, as shown in Fig. 2. First, in Fig. 2(a), we conduct pre-training based on DGI. Then, in (b), we perform prompt-based fine-tuning for the downstream voucher abuse detection.

In particular, prompt-based fine-tuning consists of three key modules: (1) *Prompt generation*: Our graph prompting function generates a set of token pairs for each input node; (2) *Prompt-assisted prediction*: For each input node, the matching probabilities of its token pairs can be scored by the same projection head in pre-training. The matching probabilities will be used for the prediction of legitimate/abusive orders. (3) *Prompt-based fine-tuning*: The context tokens in the prompt will be fine-tuned together with the pre-trained GNN, and the key difference from traditional fine-tuning is that the same pretext projection head Φ^{pre} and pretext loss function $\mathcal{L}^{\text{pre}}$ employed in pre-training will be *reused* for the downstream task, without needing a new projection head or task loss.

In the following, we will first briefly introduce the data preparation process, then focus on the downstream stage in Fig. 2(b), as the pre-training stage has been introduced in the preliminaries.

4.2 Data preparation

Our model VPGNN requires two key input elements, including an order graph and a small number of labels, as we lay out below.

Graph construction. We work with two categories of proprietary raw data: (1) User profiles with details such as email, IP and shipping addresses, and (2) Buyer journey logs such as logins, orders, payments, *etc.* We construct an order graph based on various shared attributes like the same device or address, similar usernames, *etc.*, as shown in Fig. 1(a), to capture collusive patterns between orders. Note that different order graphs could be created based on different markets or days, where pre-training may be conducted on one order graph for downstream prediction on a different order graph, enabling generalization across markets or time.

Limited pseudo-label generation. Due to the fluid nature of voucher abuse, timely and high-confidence labels are preferred. Hence, we generate pseudo-labels by employing a set of predefined business rules, which have been crafted by Lazada Inc.'s internal experts. In particular, the rules are designed to be conservative to avoid upsetting legitimate users, so that only the most conspicuous abusive orders would be flagged out. Hence, these pseudo-labels hold high confidence, but their availability is limited. These few labeled examples will be employed for our prompt-based fine-tuning in the downstream prediction.

4.3 Prompt generation, prediction, and tuning

Instead of introducing a new projection head and loss for downstream task, our prompt-based fine-tuning framework generates and tunes prompts, and makes predictions based on the prompts.

First, we propose a graph prompting function $\mathcal{P}$, which transforms an input node i into a prompt $\mathbf{p}_i$. The prompt $\mathbf{p}_i$ is a continuous embedding vector and has the same shape as the input to the pretext projection head in pre-training. Given that our pretext task in DGI is to maximize the mutual information in node-graph pairs, our prompt $\mathbf{p}_i$ also assumes a pairwise template, consisting of a pair of node token and context token. For a node $i \in V$, we have

$$\mathbf{p}_i = \mathcal{P}(i) = [\mathbf{t}_i, \mathbf{z}_c], \tag{3}$$

where the node token $\mathbf{t}_i$ is a vector representation of node i that can be encoded by a GNN, and the context token $\mathbf{z}_c$ is a learnable embedding for class y_c in the downstream task, as explained below.

Node token. The node token captures the node information. Corresponding to the input text embedding in NLP, our node token $\mathbf{t}_i$ is an embedding of the input node i. It can simply be $\mathbf{h}_i$, the representation of node i as encoded by a GNN, or an aggregation of the embedding vectors of adjacent nodes of i. In our work, we simply implement $\mathbf{t}_i = \mathbf{h}_i$, which is the direct output of the GNN.

Context token. The context token is designed to capture the contextual information about a class. Inspired by prompt tuning [19, 21], we model our context token with a learnable vector $\mathbf{z}_c$ for each class y_c in the downstream task. Suppose there is a set of C classes $\{1, 2, \ldots, C\}$. Hence, for each input node i, we can pair its node token with C different context tokens, to form C token pairs, namely, $(\mathbf{t}_i, \mathbf{z}_1), (\mathbf{t}_i, \mathbf{z}_2), \ldots, (\mathbf{t}_i, \mathbf{z}_C)$. The context tokens can be represented by a learnable prompt matrix $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_C]^\top \in \mathbb{R}^{C \times d}$, where d is also the embedding dimension of node representations.

Prompt initialization. Since the context tokens are learnable vectors, we need to address their initialization. The traditional way is random initialization, *i.e.*, trained from scratch, which is uninformative and fails to exploit our pre-trained model.

An optimal context token $\mathbf{z}_c$ is designed to capture the contextual information about class $c \in \{1, \ldots, C\}$. One straightforward option is to take the mean of the representation vectors of all labeled nodes in class c, since information about the class should ideally relate to class centroid. This is more informative than random initialization, by making use of the embedding vectors encoded by the pre-trained GNN. However, it still suffers from two problems. Firstly, in low-resource scenarios where each class has very few labeled nodes, the sample mean could be an unreliable estimation of the class centroid. Secondly, a simple mean cannot capture complex non-linear relationship between the contextual information about the class and the class centroid.

To improve the robustness of the initialization, we augment the labeled nodes with their neighboring nodes, which can provide additional local information around a labeled node. To improve the informativeness, we reuse the graph Readout function from pre-training. As the Readout is designed to pool the nodes in a graph to derive a graph-level summary representation, it can be similarly applied downstream to pool the nodes related to a class to derive a class-level summary representation. The class-level summary can serve as a more informative initialization for the context token, which aims to capture information about the class. Specifically, for each class c, we gather the set of labeled nodes and their neighboring nodes $V_c = \{c_1, c_2, \ldots, c_N\}$, where $N = |V_c|$. Then, we construct an input embedding matrix $\mathbf{H}_c = [\mathbf{h}_{c_1}, \mathbf{h}_{c_2}, \ldots, \mathbf{h}_{c_N}]^\top \in \mathbb{R}^{N \times d}$ for the nodes in V_c, where $\mathbf{h}_{c_i}$ is the representation of node $c_i \in V_c$ as encoded by the pre-trained GNN. Subsequently, the context token $\mathbf{z}_c$ can be initialized by $\Omega(\mathbf{H}_c; \omega^{\text{pre}})$, where ω^{pre} is the pre-trained parameters of the Readout function. For efficiency, we only sample η neighbors from each labeled node.

Based on our prompt design, we outline prompt-based learning for voucher abuse detection, which involves prompt-assisted prediction and prompt-based fine-tuning.

Prompt-assisted prediction. For each input node i, the prompting function generates two token pairs $(\mathbf{t}_i, \mathbf{z}_0)$ and $(\mathbf{t}_i, \mathbf{z}_1)$, given classes {0 = legitimate, 1 = abusive} for the binary voucher abuse detection. We can leverage the same pretext projection head Φ^{pre} to score the matching probability of each token pair, similar to scoring the node-graph pairs in pre-training. Thus, we predict the order represented by node i as abusive if and only if

$$\Phi^{\text{pre}}(\mathbf{t}_i, \mathbf{z}_1; \phi') > \Phi^{\text{pre}}(\mathbf{t}_i, \mathbf{z}_0; \phi'), \tag{4}$$

since $\mathbf{z}_0, \mathbf{z}_1$ capture the contextual information about the two classes. Here ϕ' can be initialized by ϕ^{pre}, the pre-trained parameters of the projection head, and further fine-tuned based on task-specific labels as we show next.

Prompt-based fine-tuning. Our prompt design allows us to reuse not only the pretext projection head without introducing a new classification head, but also the pretext task loss without formulating a new task loss.

$$\arg\min_{\theta', \phi', \mathbf{Z}} \sum_{(i,c)} \mathcal{L}^{\text{pre}}(\Phi^{\text{pre}}(\mathbf{t}_i, \mathbf{z}_c; \phi'); \text{match}(i, c)) + \lambda \mathcal{L}^o, \tag{5}$$

where $\text{match}(i, c)$ is a reloaded indicator function: 1 if node i is labeled c; 0 otherwise. $\mathcal{L}^o = \left\| \mathbf{Z}\mathbf{Z}^\top - \mathbf{I} \right\|_2^2$ is the orthogonal constraint on the prompt matrix to promote separability of each class, and $\lambda \geq 0$ is a co-efficient to control the importance of the constraint.

Compared to the traditional fine-tuning in Eq. (2), we still optimize w.r.t. the pre-training loss function $\mathcal{L}^{\text{pre}}$ with the same projection head Φ^{pre}. Meanwhile, we optimize the prompt matrix $\mathbf{Z}$ in Eq. (5) instead of the new classification weights ψ in Eq. (2). Hence, our prompt-based approach unifies the pretext and downstream task, narrowing the gap between pre-training and downstream objectives, where θ' and ϕ' are the parameters of pre-trained GNNs and projection head, $\mathbf{Z}$ are the well-initialized context tokens, λ is the hyper-parameter, controlling the orthogonal constraint.

5 EXPERIMENTS

We conduct a comprehensive suite of offline experiments first, followed by an online evaluation of our deployed model.

5.1 Offline setup and results

Datasets. First, we collect four proprietary large-scale datasets for voucher abuse detection, named **VN0909**, **VN1010**, **ID0909**, and **ID1023**, from an e-commerce platform provided by Lazada Inc. on the premise of complying with all the security and privacy policies. Each dataset is a graph with millions of nodes, where the nodes represent the orders with pre-defined features, and the edges are pre-defined relationships between them. VN0909 and VN1010 are from Lazada's Vietnam market, collected on Sep. 9 and Oct. 10 2022, while ID0909 and ID1023 are from Lazada's Indonesia market, collected on Sep. 9 and Oct. 23. Note that VN0909 is only used for *pre-training*, and we do test on three other datasets. Second, we also use a public dataset on anomaly detection, namely **Amazon** [10]. We perform pre-training on itself.

Task setup. We construct downstream tasks in *10-shot*, *20-shot* and semi-supervised settings. In 10- or 20-shot settings, we sample 10 or 20 nodes labeled as anomaly for training, respectively. In the semi-supervised setting, we sample 100 anomaly labels on Amazon and 5000 anomaly labels on others for training. As anomaly detection is an imbalanced binary classification problem, we further sample negative nodes (*i.e.*, normal nodes without anomaly) for training so that the anomaly rate in training is similar to the overall rate. For all settings, a validation set with an equal size to the training set is also sampled, and the remaining will be used for testing. Finally, we randomly generate 10 different splits of train/validation/test in each setting, and report the average performance with a 95% confidence interval.

Baselines. We consider competitive baselines from four categories. (1) *Classical machine learning*, widely used in industry, including **SVM** [7], **XGBoost** [5], and **MLP** [6]; (2) *General semi-supervised GNNs*, which use both node features and graph structures and train end to end, including **GCN** [17], **GAT** [37], and $\textbf{SAGE}_{\textbf{sup}}$, the supervised version of GraphSAGE [13]; (3) *Anomaly detection GNNs*, which are special-purpose GNNs designed for anomaly detection and trained in a supervised fashion. **CARE-GNN** [10] enhances the GNN aggregation process with three unique modules against camouflages. **GeniePath** [25] is an approach learning adaptive

Table 1: Performance comparison between VPGNN and the baselines, in percent, with 95% confidence intervals.

In each row, the best result is bolded and the runner-up is underlined. "/" indicates no result obtained due to out-of-memory issue or excessively long training time (>72 hours).

	SVM	XGBoost	MLP	GCN	$SAGE_{sup}$	GAT	CARE-GNN	GeniePath	AMNet	DCI	$SAGE_{unsup}$	Pre-train	VPGNN
Number of shots = 10													
VN1010	37.1±8.9	65.7±5.5	62.9±2.8	59.1±6.7	61.9±3.6	60.9±4.6	/	58.0±4.5	/	/	61.8±4.6	64.8±3.6	**67.1**±3.1
ID0909	28.1±11.0	51.3±9.9	61.6±3.8	64.1±3.9	61.2±7.3	65.6±3.7	/	62.1±2.8	/	/	62.2±3.7	66.1±3.0	**69.0**±3.7
ID1023	38.7±8.3	73.5±6.1	69.3±2.1	69.3±4.8	71.3±5.2	73.7±3.6	73.0±2.9	72.0±5.0	70.0±3.5	73.4±1.6	67.5±5.3	71.8±5.2	**75.1**±1.9
Amazon	41.4±9.2	62.5±11.5	63.3 ±5.7	16.5±4.9	59.9±9.1	20.5±6.1	38.6±2.9	30.7±2.8	64.8±6.2	18.5±4.0	36.7±6.0	62.3±8.1	**70.0**±2.7
Number of shots = 20													
VN1010	59.2±3.8	73.1±3.9	69.2±2.2	71.6±2.8	73.6±3.2	74.5±4.4	/	69.5±5.7	/	/	72.8±2.0	75.7±3.0	**75.9**±2.8
ID0909	53.1±6.1	64.9±3.8	64.9±1.7	68.7±2.5	70.3±2.8	71.0±3.4	/	61.4±9.1	/	/	67.5±2.2	71.4±2.5	**72.7**±2.8
ID1023	65.2±3.6	78.9±1.4	74.7±1.6	79.0±1.7	81.5±1.3	81.1±1.1	74.2±0.9	80.7±4.2	75.6±1.9	79.4±1.3	78.1±2.1	81.3±1.4	**81.8**±1.1
Amazon	60.3±3.6	72.9±8.2	70.4±4.4	16.8±8.0	63.0±8.5	48.8±10.1	42.2±6.7	30.8±4.4	75.1±3.1	21.8±2.6	54.5±2.8	73.1±3.9	**76.6**±2.5
Semi-supervised													
VN1010	86.7±0.1	87.8±0.1	86.7±0.1	91.8±0.1	94.1±0.1	91.9±0.0	/	91.7±0.4	/	/	89.3±0.0	94.1±0.1	**95.2**±0.1
ID0909	86.0±0.2	89.2±0.3	86.8±0.3	92.2±0.2	93.4±0.2	92.3±0.2	/	91.1±0.4	/	/	86.3±0.2	93.3±0.2	**94.1**±0.2
ID1023	89.3±0.1	89.9±0.2	88.8±0.2	94.4±0.1	95.6±0.1	94.5±0.1	87.0±0.3	94.5±0.1	94.1±0.3	93.7±1.0	92.6±0.1	95.6±0.1	**96.2**±0.1
Amazon	78.8±1.1	75.6±2.7	78.1±1.8	34.4±3.6	81.1±1.5	73.3±2.9	45.2±5.6	30.9±4.5	**81.7**±1.1	26.1±4.5	76.1±0.9	80.9±0.9	80.6±0.7

receptive fields of GNN, with an adaptive path layer consisting of two complementary functions designed for breadth and depth exploration respectively. **AMNet** [4] is an adaptive multi-frequency GNN, capturing both low-frequency and high-frequency signals, and adaptively combine signals of different frequencies; (4) *Pre-trained GNNs*, which perform pre-training on label-free graphs, including **DCI** [41], **$SAGE_{unsup}$**, the unsupervised version of GraphSAGE [13] which employs a form of linear probe that is known to be a strong few-shot learner [36], and **Pre-train** which uses the same pre-training strategy as VPGNN, and is then fine-tuned with a new projection head.

Settings. We set the number of hidden units to 128, using two layers with ReLU activation for all GNNs, except for GeniePath and CareGNN. Geniepath uses 16 hidden units and 7 layers, while CareGNN has 64 hidden units and 1 layer, as recommended in [10, 25]. The Adam optimizer is applied to both pre-training and fine-tuning. The learning rate is 0.01 on the Lazada datasets and 0.001 on Amazon. For VPGNN, we use a 128×128 fully connected layer as the Readout function, inner product as the pretext task projection head, set $\eta = 5$ for the number of neighbors sampled per labeled node for context token initialization, and $\lambda = 0.01$ for the coefficient of the orthogonal constraint.

Classification performance. In Tab. 1, we compare the performance of VPGNN with the baselines. First, given that pre-training is done on VN0909, the superior performance of VPGNN on VN1010, ID0909 and ID1023 shows its generalization ability across time and/or markets. Furthermore, VPGNN is a strong few-shot learner, as it attains larger improvements relative to the runner-up under the 10-shot setting. In general, other pre-trained GNNs also tend to perform better under the few-shot settings. Second, classical machine learning methods are generally inferior to GNN-based methods, demonstrating that graph structures can complement node features in our problem. Third, those GNNs specifically designed for anomaly detection only achieve similar performance to vanilla GNNs on the Lazada datasets, showing that these special-purpose GNNs cannot play to their strengths in voucher abuse detection due to the fluidity and complexity of the problem.

Ablation study. To better understand the contribution of each component in VPGNN, we compare VPGNN with the following ablated models. 1) *No prompt*, which follows the traditional "pre-train, fine-tune" paradigm without prompting; 2) *Random init.*, which randomly initializes the context tokens; 3) *No constr.*, which removes the orthogonal prompt constraint in Eq. (5). We report the results on ID1023 with different shots in Tab. 2, and under the semi-supervised setting on different datasets in Fig. 3. VPGNN outperforms all the ablated models consistently, demonstrating the overall benefit of integrating various components. Among the ablated models, *No prompt* performs rather poorly in most cases, especially under the 10-shot setting. This demonstrates the superiority of prompt-based fine-tuning compared with traditional fine-tuning in low-resource setting. *Random init.* also performs not as well, showing that the token initialization in VPGNN is crucial.

Analysis of token initialization. When initializing the context tokens, we sample η neighbors of the labeled nodes. To analyze the impact of the sampling, we experiment with different η values under the 10-shot setting, as shown in Fig. 4. Naturally, when more neighbors are sampled, we observe somewhat better performance. But too many neighbors will also bring more noise. Therefore, it is proper to sample a not big or small amount of neighbors. Besides, we notice that when $\eta = 0$, *i.e.*, no neighbor information is used, the performance is worse, reflecting that utilizing neighbor information will make an informative and robust token initialization.

Prompt tuning only. For large-scale graphs in E-commerce, it is costly to fine-tune the pre-trained GNN model. Some studies [19, 21] have shown that prompt tuning only while freezing the pre-trained model can still outperform traditional fine-tuning. To evaluate the effect of only tuning the prompt in VPGNN, we compare it with DCI and Pre-train, two models with fine-tuning on ID1023, across different shots. Specifically, DCI and Pre-train are fine-tuned with 10 epochs, while we fix all pre-trained parameters of VPGNN and only tune the prompt vectors. As shown in Fig. 5, VPGNN still achieves significant improvements over DCI and Pre-train, even when no fine-tuning is done on the pre-trained model.

5.2 Online deployment and evaluation

To further demonstrate the effectiveness of VPGNN in production, we deploy it in the Lazada production line to detect voucher abuse orders, and compare it against existing deployed models.

Deployment details. The deployed model is a *D+1* model, which means that we process and predict the orders placed in the past

Table 2: Ablation study on ID1023 under different shots.

Model \ shots	10	20	Semi
No prompt	71.82	81.31	95.61
Random init.	73.76	79.07	95.55
No constr.	74.72	81.81	96.12
VPGNN	**75.09**	**81.84**	**96.21**

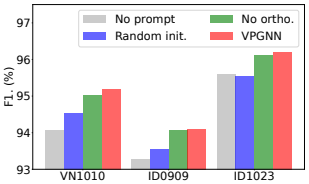


Figure 3: Ablation study on semi-supervised setting.

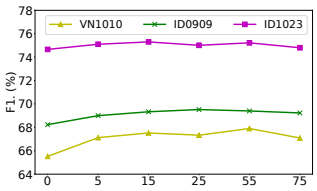


Figure 4: Impact of neighbors for token initialization

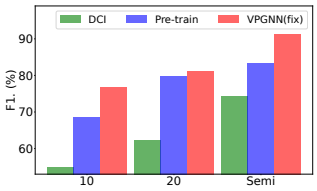
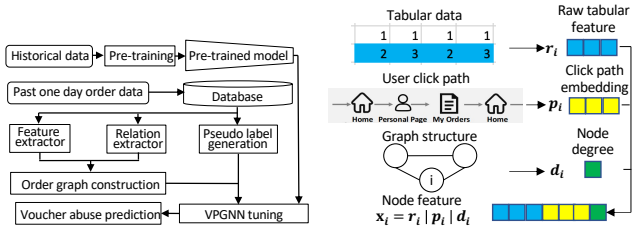


Figure 5: Prompt tuning only without fine-tuning



(a) Deployment (b) Feature engineering

Figure 6: Deployment pipeline and feature engineering.

Table 3: Online performance in the Indonesia market over the Double 12 Campaign, measured in BPWC.

Model	11 Dec 2022	12 Dec 2022	13 Dec 2022	Overall
BCP	100.0%	100.0%	100.0%	100.0%
LPA	816.9%	1784.1%	330.3%	809.3%
VPGNN	**1964.1%**	**1990.3%**	**469.1%**	**998.7%**
(% ↑ over LPA)	(140.4%)	(11.6%)	(42.0%)	(23.4%)

one day in the production environment. Figure 6(a) describes the deployment pipeline, and (b) zooms into feature engineering.

First, we pre-train a model using historical order data. Next, given the past one-day order data flown into the deployed model, they first pass through the feature extractor and relation extractor to exploit rich attributes and various relations for the order graph construction. Meanwhile, high-confidence pseudo-labels are generated in limited quantity (see Sect. 4.2). Finally, the pre-trained model is loaded and further fine-tuned by VPGNN using the pseudo labels. After tuning, we output the predictions for abusive orders, and a downstream team will handle the predictions appropriately (*e.g.*, canceling orders or suspending accounts).

Moreover, feature engineering still plays a vital role in modern production systems. First, each order is accompanied by a user click path, in the form of a sequence of actions, *e.g.*, "`Home, Personal Page, My orders, Home`", which is, in fact, a natural language sentence. Hence, we choose FastText [2] to model and generate the click path embedding $\mathbf{p}_i$ for order i, which is a fast implementation based on the unsupervised skip-gram model, more feasible than large pre-trained language models [8, 24] in our high-throughput production setting. Second, voucher abuse orders tend to have the characteristics of aggregation [26], *i.e.*, often have a high degree. Due to economic constraints, voucher abusers often place a large number of orders from multiple accounts on shared devices and/or IP addresses. Hence, we also take node degree as an important feature. Finally, we concatenate the raw tabular feature $\mathbf{r}_i$, the generated FastText embedding of click path $\mathbf{p}_i$ and node degree d_i as the node feature $\mathbf{x}_i$ for order i.

Online performance. VPGNN is deployed in the *Indonesia* market between *11 and 13 Dec, 2022*, for the Double 12 Campaign. We compare VPGNN with two existing deployed models: BCP, a distributed implementation of an unsupervised K-means clustering model [34], and LPA [33], an efficient unsupervised community detection algorithm.

As ground truth is only judged on the detected orders, recall cannot be computed since the total number of positive orders is unknown. Besides, the exact precision and the number of true positives cannot be disclosed as they are commercially sensitive. Hence, we define a metric called *BCP-normalized Precision Weighted Coverage* (BPWC). Given a model A under evaluation, this metric is calculated by

$$\frac{\text{Precision of model A} \times \#\text{ True positives detected by model A}}{\text{Precision of BCP} \times \#\text{ True positives detected by BCP}}. \quad (6)$$

Here BCP is treated as the base unit (*i.e.*, 100%). This metric considers both precision and true positives (which is proportional to recall), and hence is a good trade-off between more detections and fewer false alerts, without disclosing sensitive information.

As shown in Tab. 3, compared to BCP and LPA, our proposed VPGNN achieves significant improvements in the production setting: over the entire period, VPGNN shows a 23.4% increase over LPA, and almost a 9-fold increase over BCP. The results demonstrate the advantage of prompt-based fine-tuning on GNNs when dealing with voucher abuse detection.

6 CONCLUSION

In this paper, we proposed a prompt-based fine-tuning approach for GNNs, called VPGNN, to address the problem of voucher abuse detection. We attempted to bridge the gap between pretext and downstream tasks by proposing a graph prompting function that reformulates the downstream task to follow a similar template as the pretext task. The pre-trained GNN model could then be applied with a relatively light fine-tuning given limited downstream labels for abusive orders. Extensive offline experiments on both proprietary and public datasets show that VPGNN outperforms state-of-the-art baselines in few-shot and semi-supervised scenarios. Moreover, an online evaluation also demonstrates that VPGNN achieves a 23.4% improvement over two existing deployed models.

ACKNOWLEDGMENTS

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

REFERENCES

[1] Sambaran Bandyopadhyay, N Lokesh, and M Narasimha Murty. 2019. Outlier aware network embedding for attributed networks. In *AAAI conference on artificial intelligence*. 12–19.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[4] Ziwei Chai, Siqi You, Yang Yang, Shiliang Pu, Jiarong Xu, Haoyang Cai, and Weihao Jiang. 2022. Can Abnormality be Detected by Graph Neural Networks?. In *International Joint Conference on Artificial Intelligence*. 1945–1951.

[5] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 785–794.

[6] Ronan Collobert and Samy Bengio. 2004. Links between perceptrons, MLPs and SVMs. In *International Conference on Machine learning*.

[7] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

[9] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *SIAM International Conference on Data Mining*. SIAM, 594–602.

[10] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *ACM International Conference on Information and Knowledge Management*. 315–324.

[11] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. 2022. Universal Prompt Tuning for Graph Neural Networks. *arXiv preprint arXiv:2209.15240* (2022).

[12] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *The Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 3816–3830.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems* 30 (2017).

[14] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2019. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

[15] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. GPT-GNN: Generative pre-training of graph neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1857–1867.

[16] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. 2021. Anemone: graph anomaly detection with multi-scale contrastive learning. In *ACM International Conference on Information and Knowledge Management*. 3122–3126.

[17] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[18] Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. 2019. Spam review detection with graph convolutional networks. In *ACM International Conference on Information and Knowledge Management*. 2703–2711.

[19] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *The Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 4582–4597.

[20] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023).

[21] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *The Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 61–68.

[22] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip Yu. 2023. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2023), 5879–5900.

[23] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (2021), 2378–2392.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[25] Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. 2019. GeniePath: Graph neural networks with adaptive receptive paths. In *AAAI Conference on Artificial Intelligence*. 4424–4431.

[26] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *ACM International Conference on Information and Knowledge Management*. 2077–2085.

[27] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. 2020. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. 1569–1572.

[28] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *The Web Conference*. 417–428.

[29] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. 2021. Learning to pre-train graph neural networks. In *AAAI Conference on Artificial Intelligence*. 4276–4284.

[30] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[31] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *ACM International Conference on Information and Knowledge Management*. 1165–1174.

[32] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph contrastive coding for graph neural network pre-training. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1150–1160.

[33] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E* 76, 3 (2007), 036106.

[34] Kristina P Sinaga and Miin-Shen Yang. 2020. Unsupervised K-means clustering algorithm. *IEEE Access* 8 (2020), 80716–80727.

[35] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. GPPT: Graph pre-training and prompt tuning to generalize graph neural networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1717–1727.

[36] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. 2020. Rethinking few-shot image classification: a good embedding is all you need?. In *European Conference on Computer Vision, Part XIV*. 266–282.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[38] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*.

[39] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A semi-supervised graph attentive network for financial fraud detection. In *IEEE International Conference on Data Mining*. 598–607.

[40] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. 2019. FdGars: Fraudster detection via graph convolutional networks in online app review system. In *The Web Conference (Companion)*. 310–316.

[41] Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Decoupling representation learning and classification for GNN-based anomaly detection. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. 1239–1248.

[42] Rui Wen, Jianyu Wang, Chunming Wu, and Jian Xiong. 2020. ASA: Adversary situation awareness via heterogeneous graph convolutional networks. In *The Web Conference (Companion)*. 674–678.

[43] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2020), 4–24.

[44] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.

[45] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate time-series anomaly detection via graph attention network. In *IEEE International Conference on Data Mining*. 841–850.