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

Zhihao Wen and Yuan Fang
Singapore Managent University
Singapore

Yihan Liu, Yang Guo and Shuji Hao
Lazada Inc.
Singapore

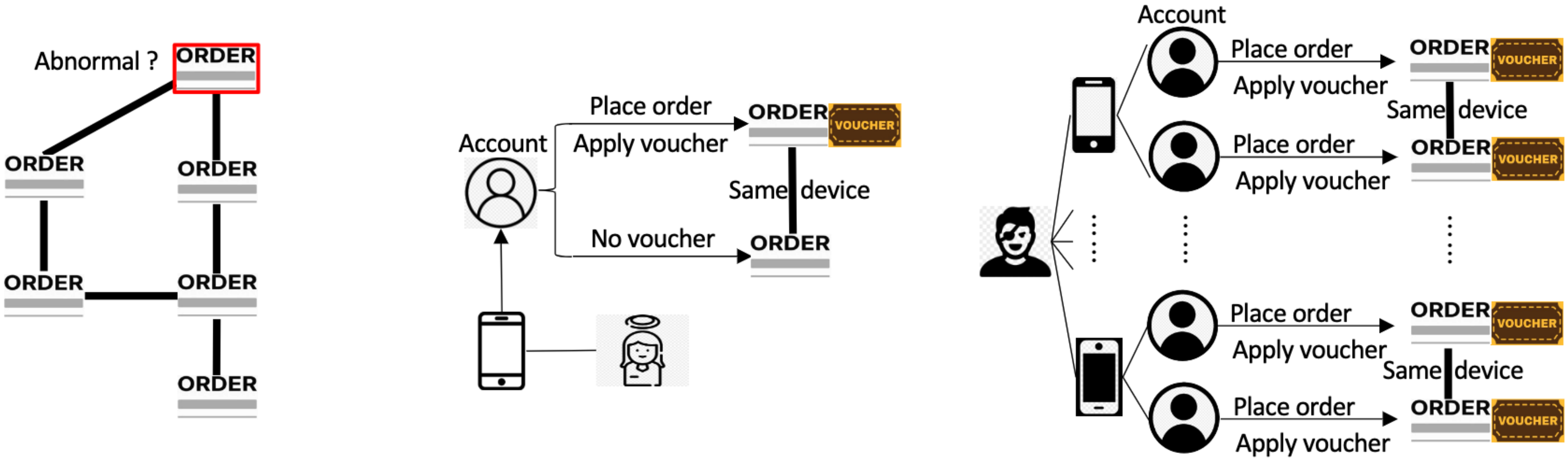
SMU
SINGAPORE MANAGEMENT UNIVERSITY
School of
Computing and
Information Systems

Lazada

Outline

- ❑ **Introduction**
- ❑ Methodology
- ❑ Experiment
- ❑ Conclusion

Voucher Abuse Detection on Order Graph



(a) Order graph

(b) Orders of legitimate user

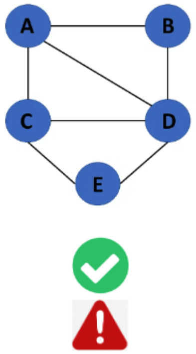
(c) Orders of abusive user

Order graph encodes rich **relationships** and **patterns** between orders

A **legitimate user** typically only logs into **one** account on **one** device and applies **one** voucher

An **abusive user** often employs **many** devices, and in each device, they create **multiple** accounts

Existing works



- Traditional ML methods [1][2][3], do not leverage **graph structure** information
- Supervised GNN-based approaches [4][5][6], cannot perform well with **limited labels**
- Self-supervised GNNs [7][8] become promising for capture **intrinsic** graph **patterns without** annotated **label**.
- Self-supervised GNNs suffer from a major **drawback**: the **objective gap** between the pre-training and downstream tasks

[1] Xgboost: A scalable tree boosting system. In KDD 2016.
[2] Links between perceptrons, MLPs and SVMs. In ICML 2004.
[3] Support-vector networks. In Machine learning 1995.
[4] Semi-Supervised Classification with Graph Convolutional Networks. In ICLR 2017.
[5] Inductive Representation Learning on Large Graphs. In NeurIPS 2017.
[6] Graph Attention Networks. In ICLR 2018.
[7] Strategies for Pre-training Graph Neural Networks. In ICLR 2019.
[8] GPT-GNN: Generative pre-training of graph neural networks. In KDD 2020.

Challenges of bridging the gap between GNN pre-training and downstream tasks

Q1: Cannot directly apply the **textual prompting** function to bridge various **graph-based** tasks

→ We propose a **graph prompting function** that reformulates the downstream **node classification** problem into a **pairwise matching** task between node tokens and context tokens

Q2: How to **initialize** the context tokens

→ We **reuse** the **graph readout** function from pre-training to initialize the context tokens downstream

Outline

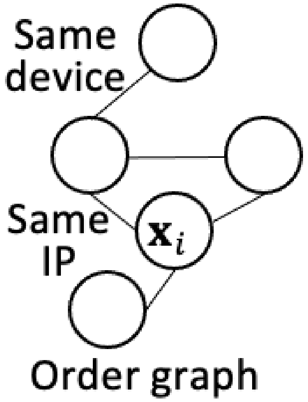
- ❑ Introduction
- ❑ **Methodology**
- ❑ Experiment
- ❑ Conclusion

Data preparation

Two key inputs: 1) an **order graph**; 2) a small number of **labels**.

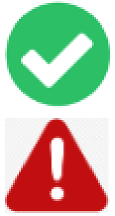
Graph construction.

- Two categories of raw data: (1) **User profiles**, and (2) **Buyer journey logs**.
- Construct an order graph based on various **shared attributes**

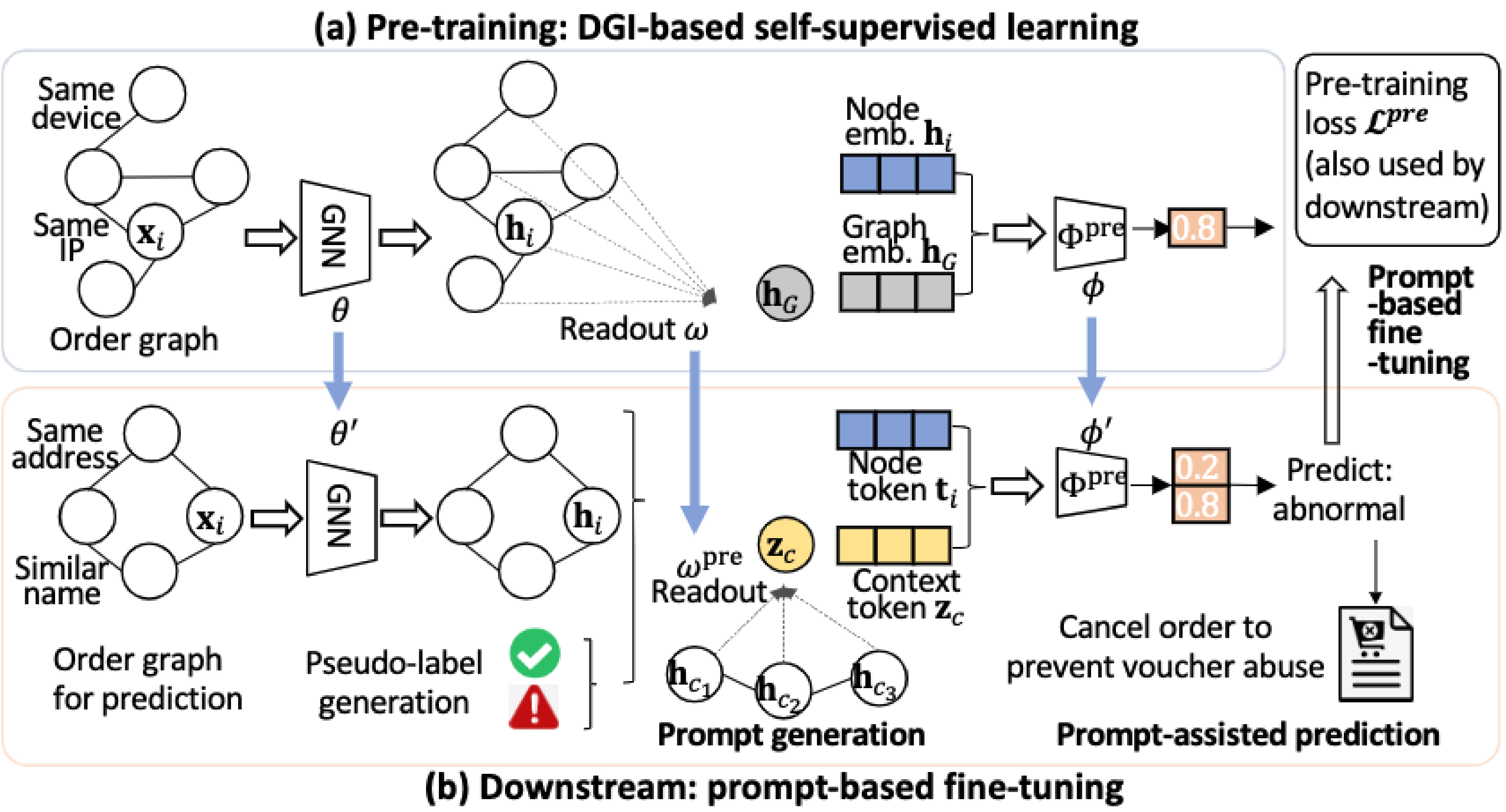


Limited pseudo-label generation.

- Generate limited pseudo-labels by employing a set of **predefined business rules**

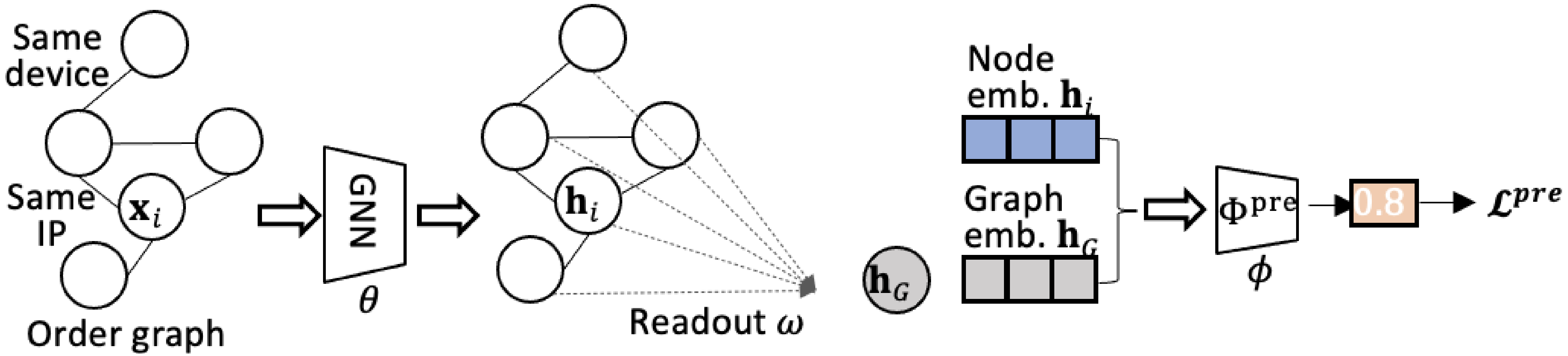


Overall framework of our proposed VPGNN



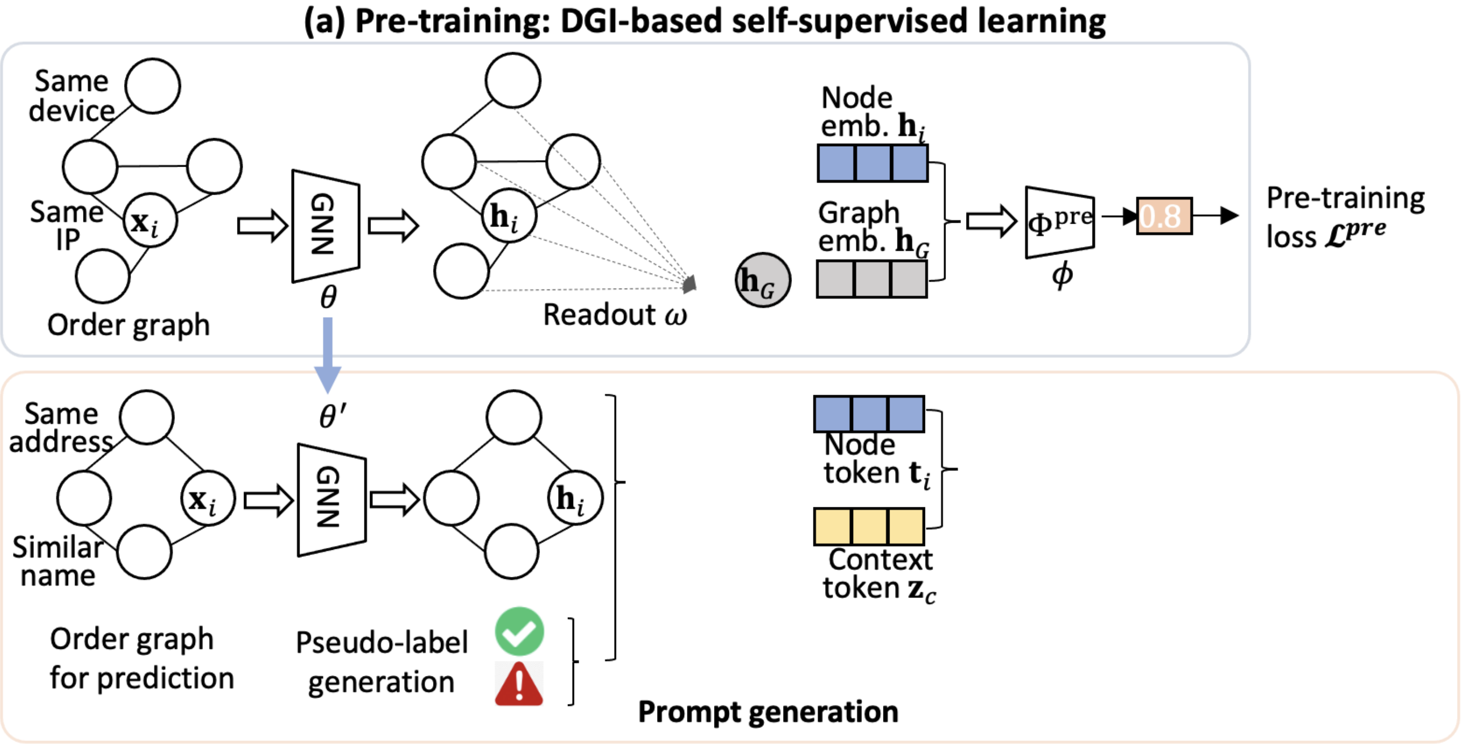
Overall framework of VPGNN. (a) We conduct **self-supervised pre-training** based on **DGI**. (b) we perform **prompt-based fine-tuning** for the downstream voucher abuse detection.

Pre-training based on DGI



- In voucher abuse detection, **abusive** orders are the **minority** and the **majority** are **legitimate** orders.
- Utilize **DGI** to maximize the **local-global** mutual information， whereby the **graph-level global** information captures the “normal” patterns manifested by the **majority**

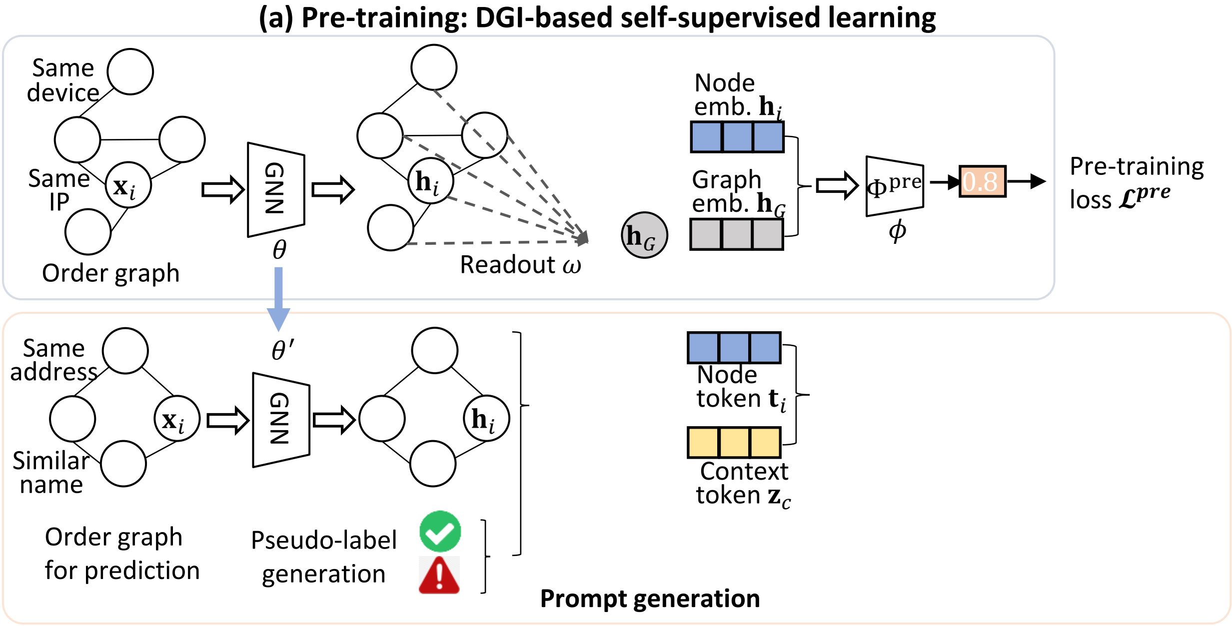
Prompt generation



- Our prompt-based fine-tuning framework **generates** and **tunes** prompts, and **makes predictions** based on the prompts.

- Propose a **graph prompting function** $\mathcal{P}$, transforming an input node i into a prompt p_i consisting of a **pair** of **node token $\mathbf{t}_i$** and **context token $\mathbf{z}_c$**

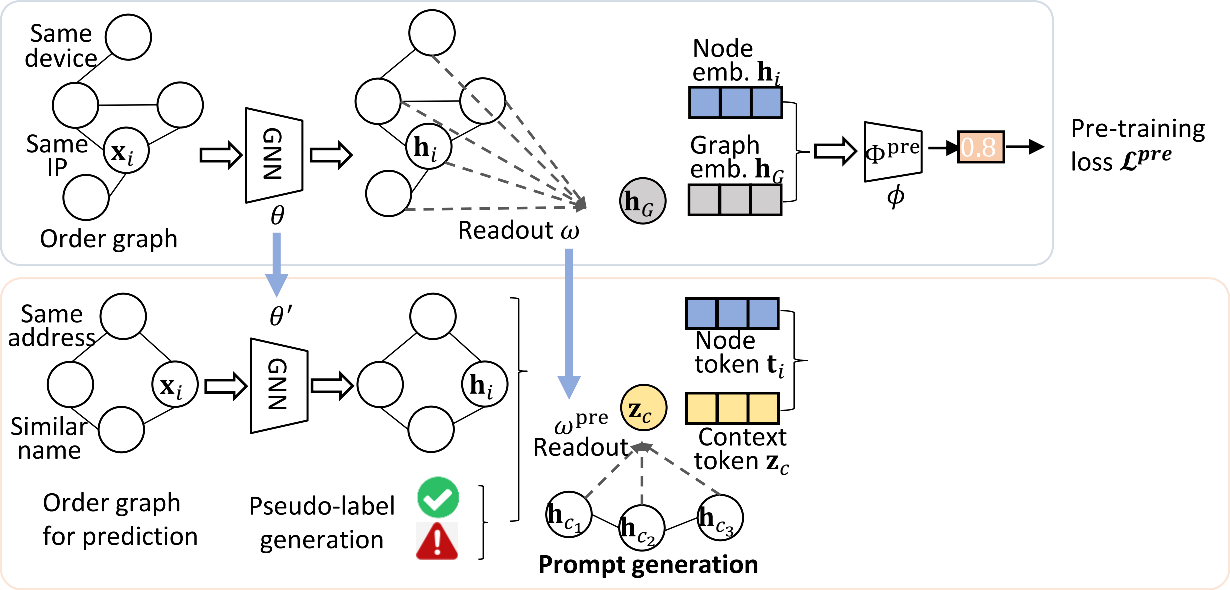
Prompt generation



- Our prompt-based fine-tuning framework **generates** and **tunes** prompts, and **makes predictions** based on the prompts.
- Propose a **graph prompting function** $\mathcal{P}$, transforming an input node i into a prompt p_i consisting of a **pair** of **node token $\mathbf{t}_i$** and **context token $\mathbf{z}_c$**

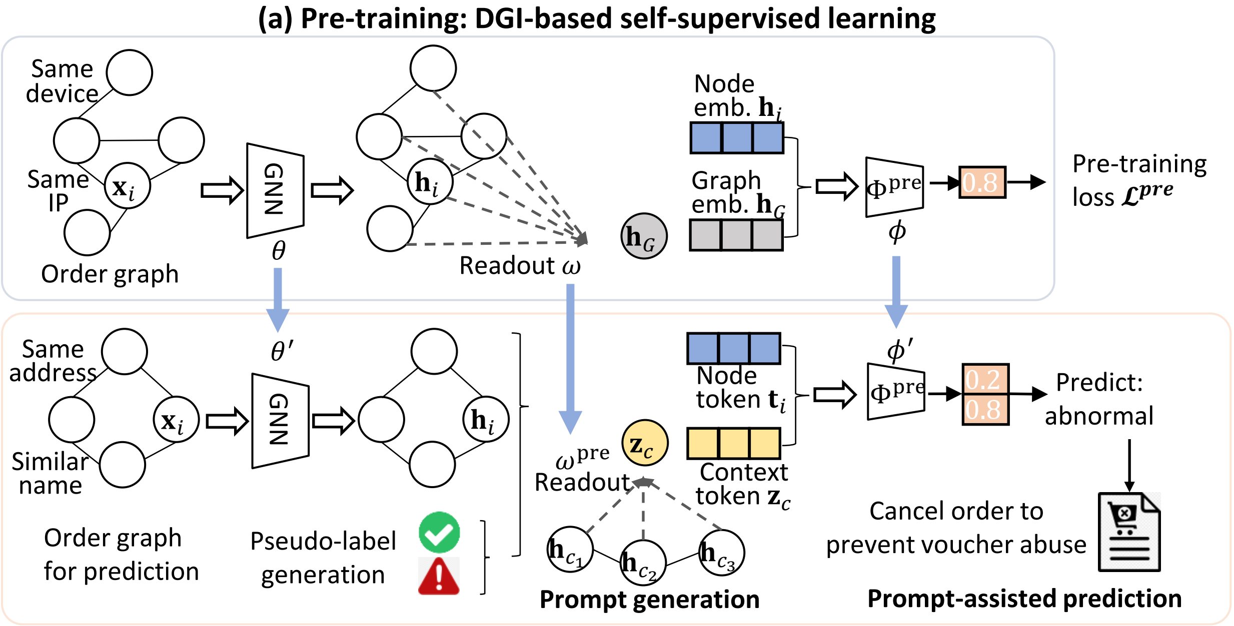
Prompt initialization

(a) Pre-training: DGI-based self-supervised learning



- Context tokens are learnable vectors, needing **initialization**.
- To improve the **robustness** of the initialization, we augment the labeled nodes with their **neighboring nodes**.
- To improve the **informativeness**, we reuse the graph **Readout** function from **pre-training** to pool the labeled nodes with their subgraphs.

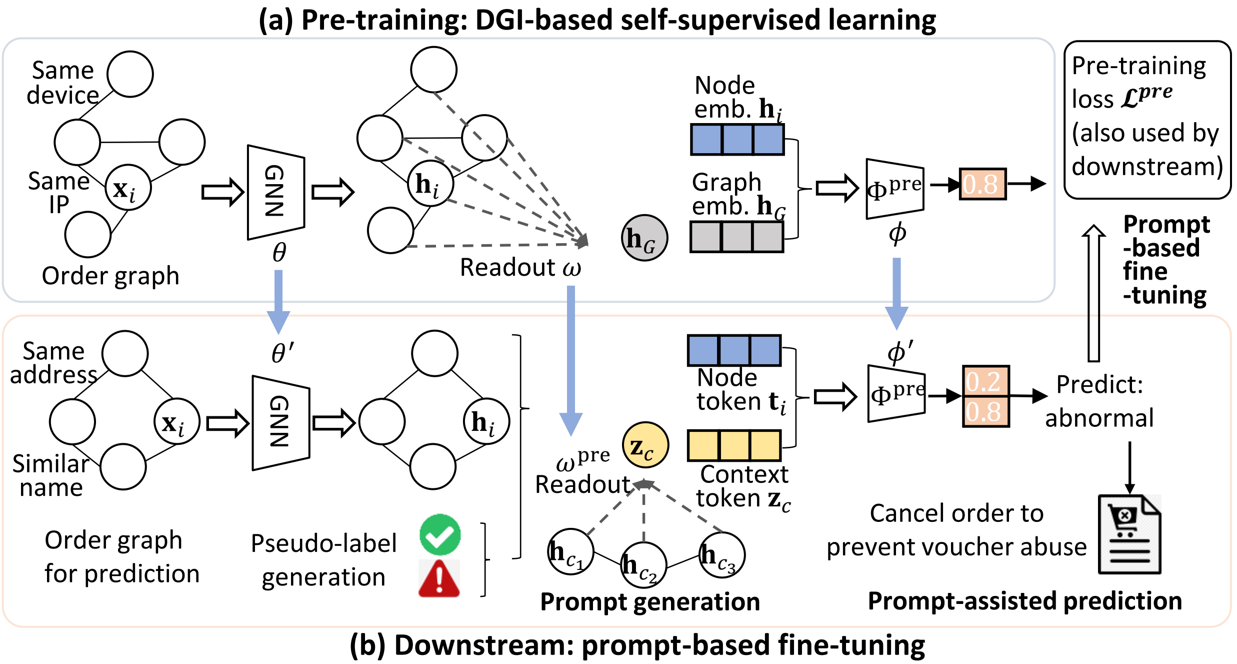
Prompt-assisted prediction



- Leverage the **same** pretext projection head Φ^{pre} to score the **matching probability** of each token pair.
- Predict the order represented by node i as **abusive** if

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

Prompt-based fine-tuning



- Our prompt design allows us to reuse not only the pretext projection head, but also the **pretext task loss**.
- Our prompt-based approach **unifies** the **pretext** and **downstream** task, narrowing the gap between pre-training and downstream objectives.

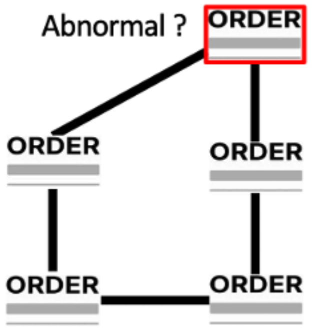
Outline

- ❑ Introduction
- ❑ Methodology
- ❑ **Experiment**
- ❑ Conclusion

Datasets

- We collect four proprietary large-scale datasets, named VN0909, VN1010, ID0909, and ID1023, from an e-commerce platform provided by **Lazada Inc**.

- Each dataset is a huge **order graph**, where the nodes represent the **orders** with pre-defined features, and the edges are pre-defined **relationships** between them.



- VN0909 is only used for **pre-training**, and we do **test** on **three other datasets**.

- We also use a public dataset , namely **Amazon**.

Offline performance comparison with baselines

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
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**
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

Classical machine learning General semi-supervised GNNs Anomaly detection GNNs Pre-trained GNNs

- 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.
- VPGNN is a strong **few-shot** learner, as it attains larger improvements relative to the runner-up under the 10-shot setting.

Online performance

Online performance in the Indonesia market over the Double 12 Campaign.

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 (% ↑ over LPA)	**1964.1%** (140.4%)	**1990.3%** (11.6%)	**469.1%** (42.0%)	**998.7%** (23.4%)

$$\text{Metric} = \frac{\text{Precision of model A} \times \#\ \text{True positives detected by model A}}{\text{Precision of BCP} \times \#\ \text{True positives detected by BCP}}.$$

- VPGNN shows a **23.4%** increase over LPA, demonstrating the advantage of prompt-based fine-tuning on GNNs when dealing with voucher abuse detection.

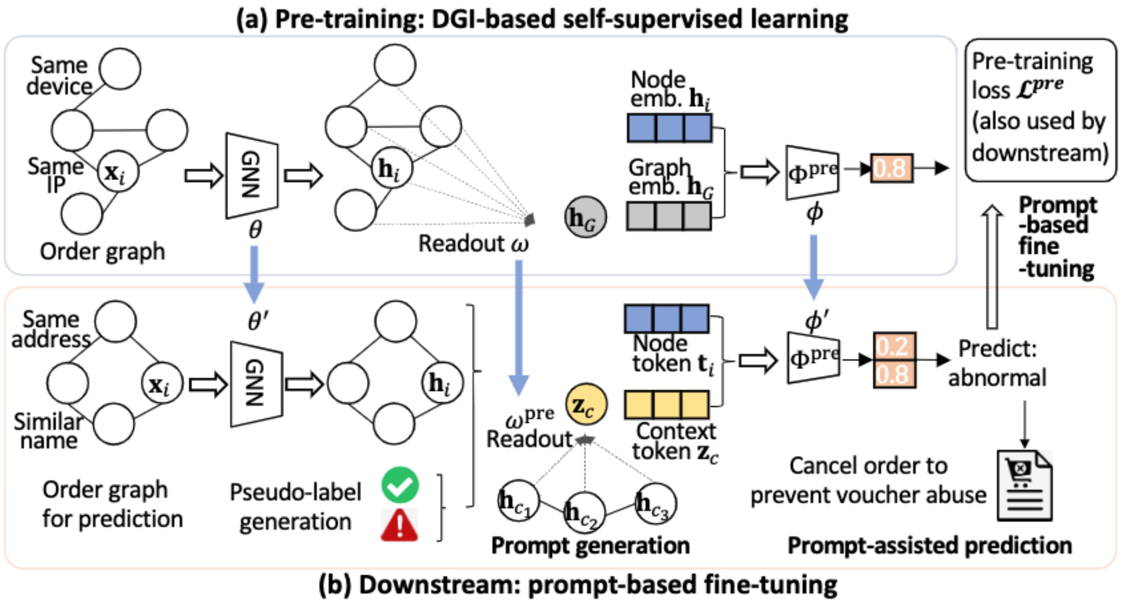
Outline

- ❑ Introduction
- ❑ Methodology
- ❑ Experiment
- ❑ **Conclusion**

Conclusion

Key contributions

- Addressed the problem of **voucher abuse detection** on e-commerce order graph;
- 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.



THANK YOU FOR YOUR ATTENTION

Paper, code, data…
www.yfang.site

SMU
SINGAPORE MANAGEMENT UNIVERSITY
School of
Computing and
Information Systems

We are hiring! Looking for a PhD student for Fall 2024 admission!
www.yfang.site/hiring