Graph Foundation Models For Generalizable Anomaly Detection Across Diverse Graph Domain

다양한 그래프 도메인에서 일반화 가능한 이상탐지를 위한 그래프 파운데이션 모델

Bonyou Koo(구본유)

Supervisor: Byungkook Oh

Graph & Language Intelligence Laboratory
Department of Computer Science and Engineering
Konkuk University

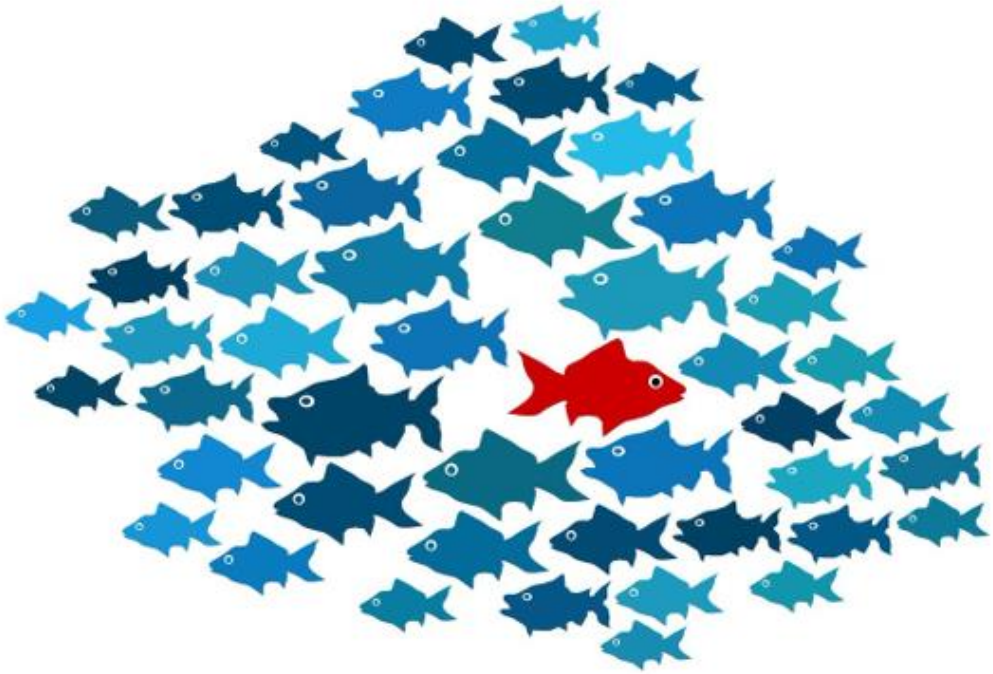
2026.08.05

CONTENTS

CONTENTS

Introduction

Anomaly Detection

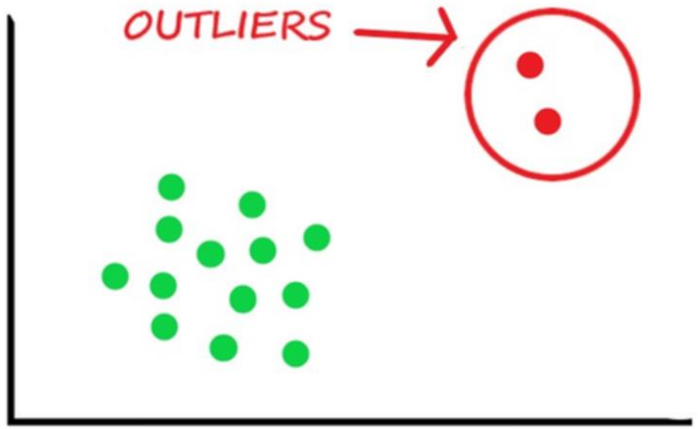


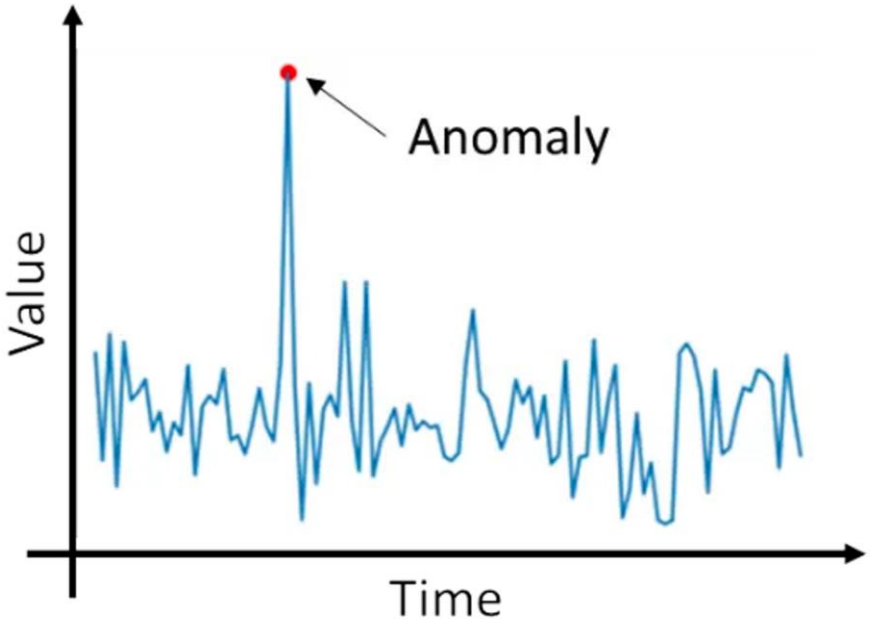
- Anomalies(outliers, novelties)
  - ✓ Not conform to a notion of normal behavior.
  - ✓ Different from most of the data.

- Anomaly detection
  - ✓ we try to distinguish anomalous samples
    - Deviate from normal samples.

Anomaly Detection

Intro





- Anomalies(outliers, novelties)
  - ✓ Not conform to a notion of normal behavior.
  - ✓ Different from most of the data.

- Anomaly detection
  - ✓ we try to distinguish anomalous samples
    - Deviate from normal samples.

Anomaly Detection

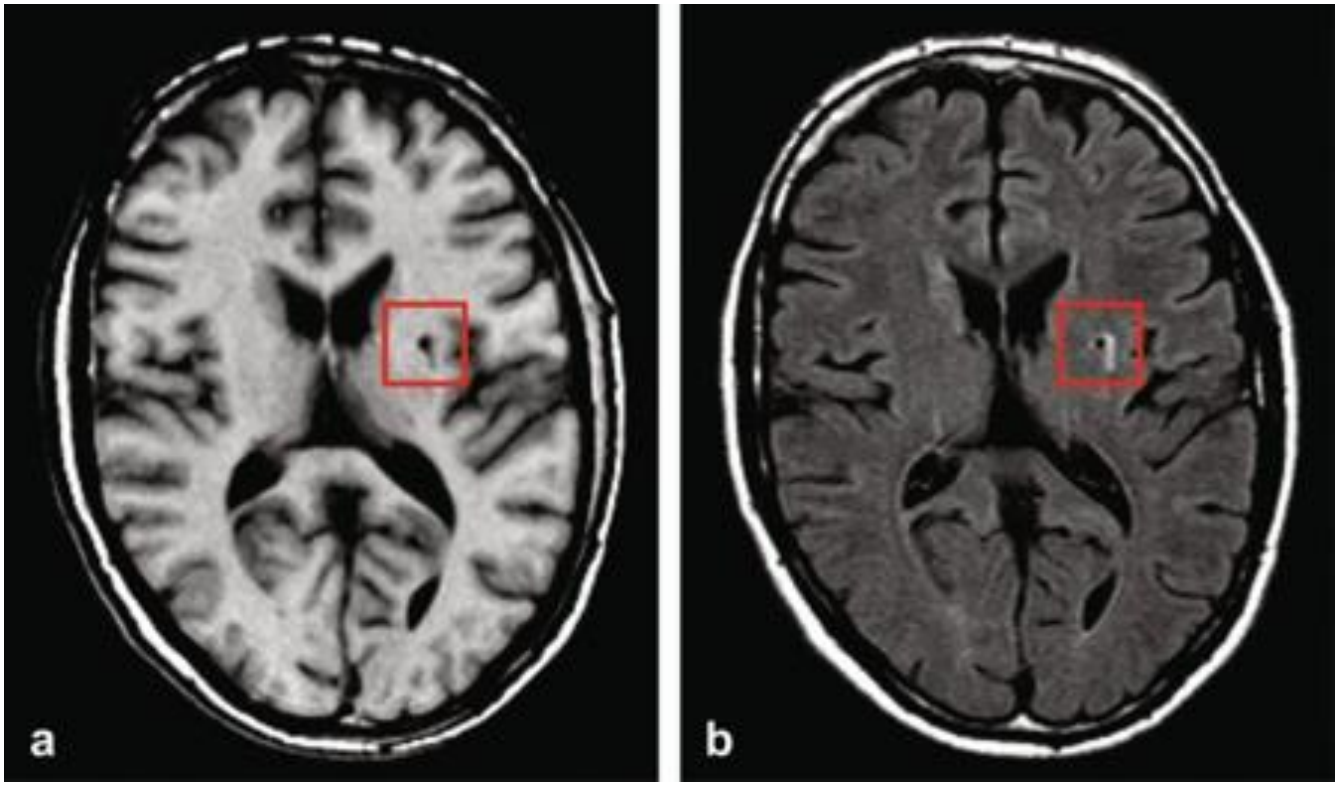
Application of Anomaly Detection

- Security and Surveillance

Anomaly Detection

Application of Anomaly Detection

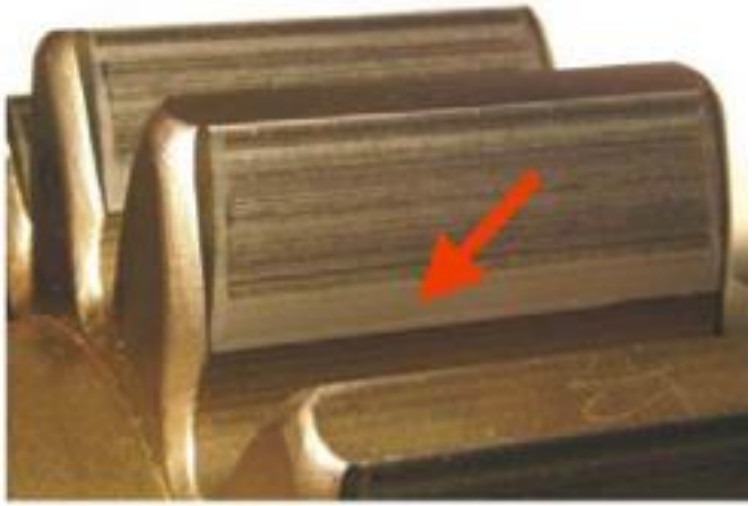
- Biomedical Application



Anomaly Detection

Application of Anomaly Detection

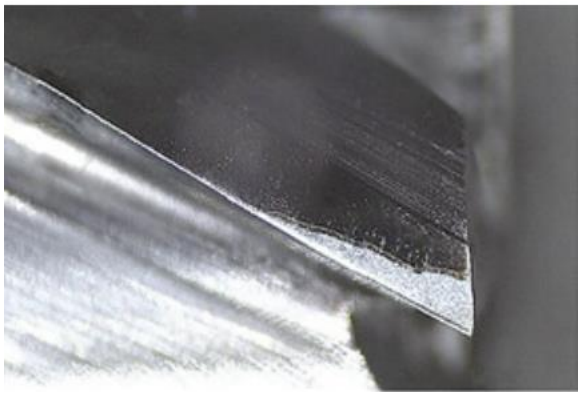
- Machinery Defects Diagnostics
  - Early alarm of malfunctioning



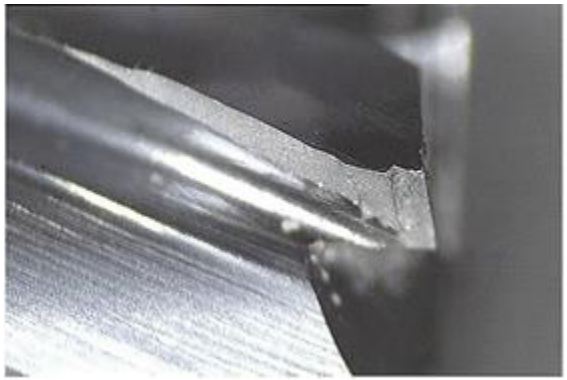
Anomaly Detection

Application of Anomaly Detection

- Industrial Damage Detection



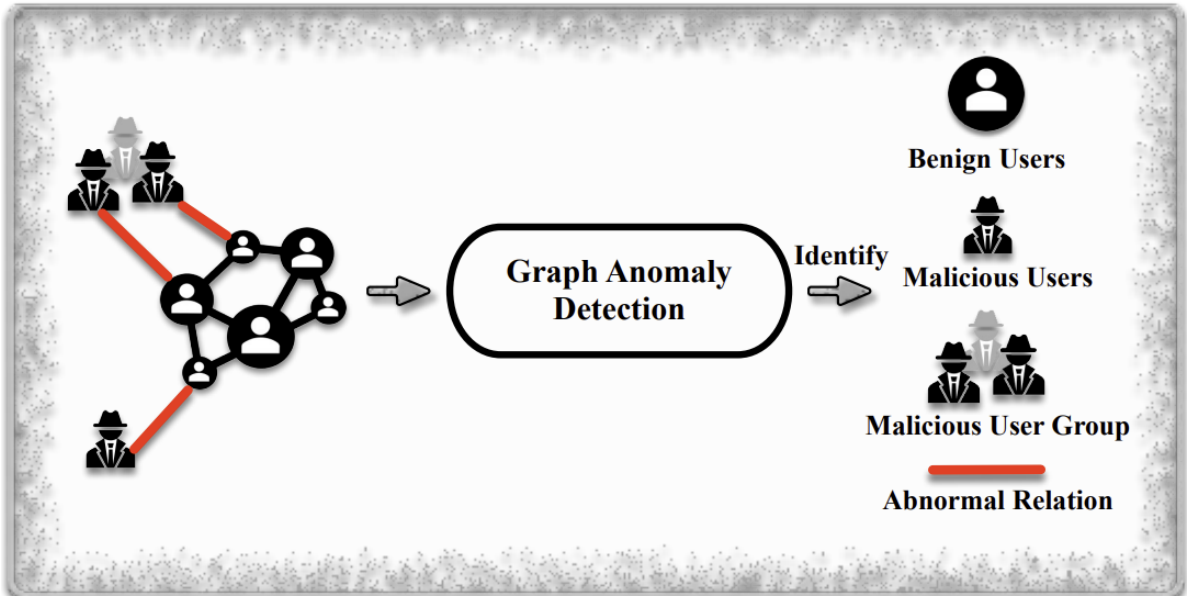
y : No Wear



y : Wear

Background

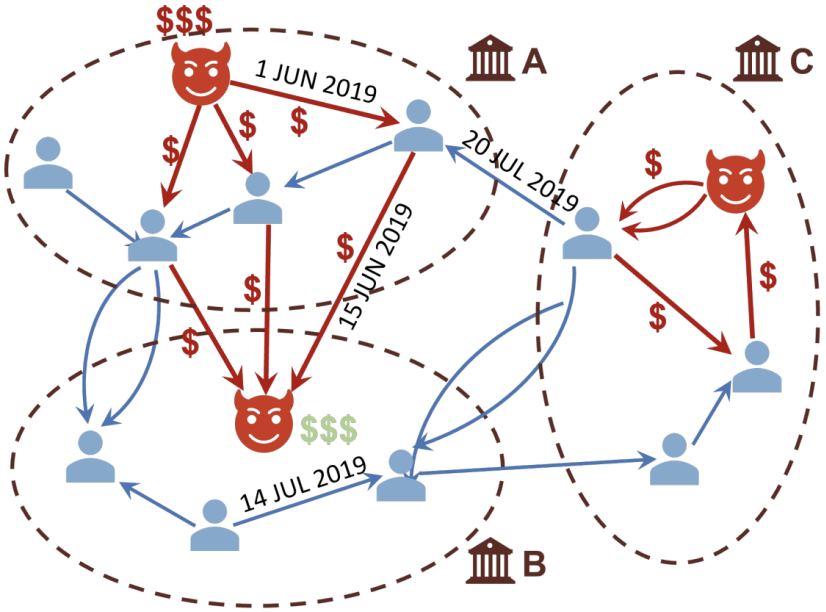
Graph Anomaly detection



- Graph anomaly detection
  - Unexpected irregular suspicious instance in graph structure.
  - Aims to identify anomalous graph objects that different from most of data.
    - Nodes(i.e., malicious users)
    - Edges(i.e., abnormal relations)
    - Sub-graphs(i.e., malicious user group)

Background

Graph Anomaly detection



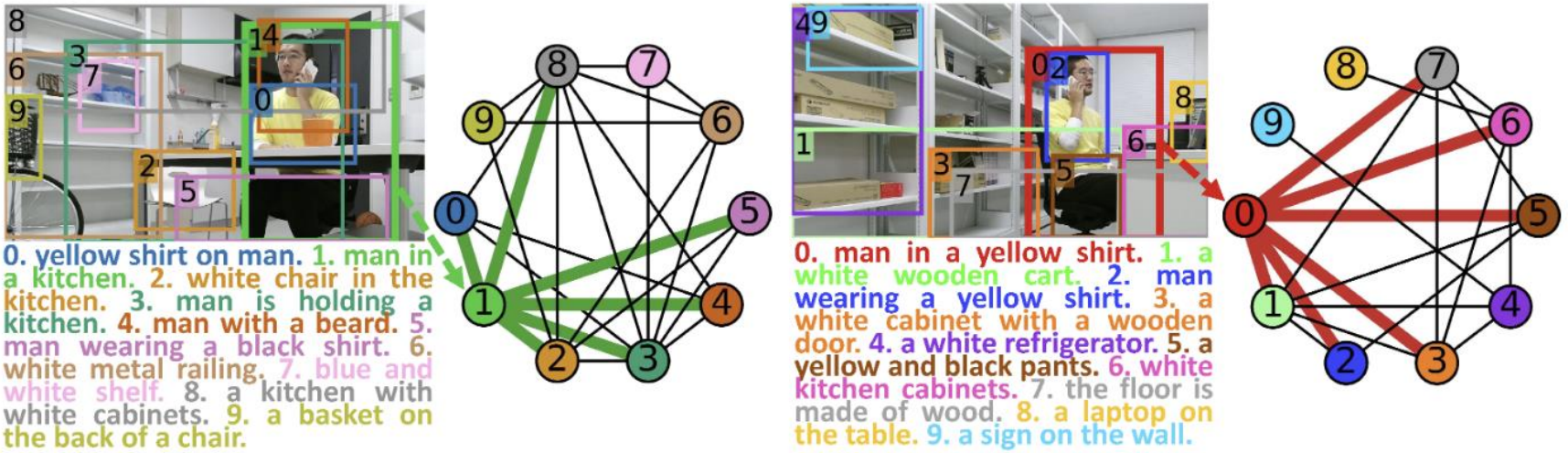
(b) Transactions in Graph Format

- Graph anomaly detection
  - Aims to identify anomalous graph objects
    - Social network
    - Financial fraud
    - Network intrusion, cybersecurity, sensor network …

FraudGT: A Simple, Effective, and Efficient Graph Transformer for Financial Fraud Detection, ICAIF' 2024

Background

Graph Anomaly detection



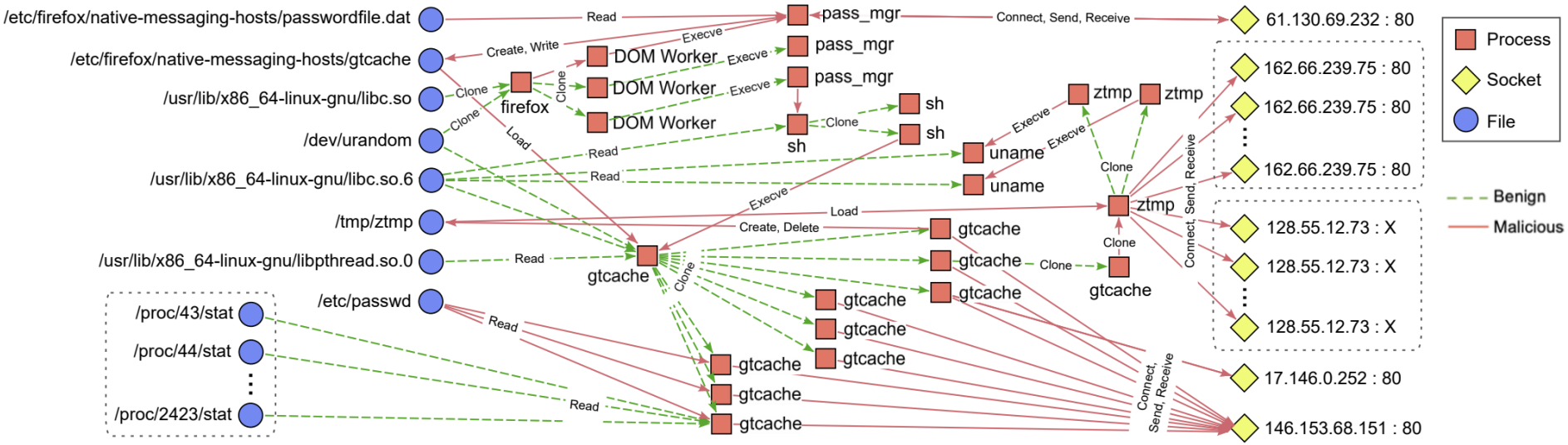
(a) Normal example. (b) Abnormal example.

- Graph anomaly detection
  - Aims to identify anomalous graph objects
    - Social network
    - Financial fraud
    - Network intrusion, cybersecurity, sensor network …

Region Anomaly Detection via Spatial and Semantic Attributed Graph in Human Monitoring , Sensors' 2023

Background

Graph Anomaly detection



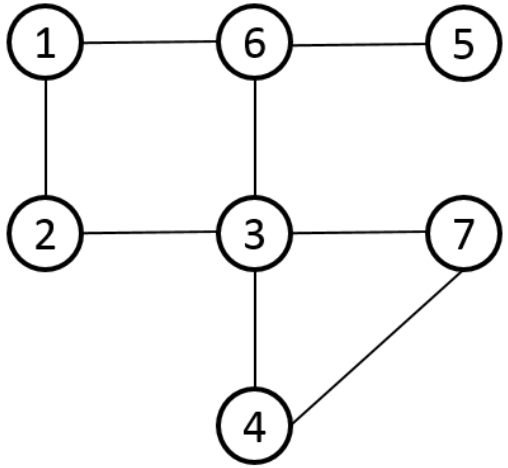
- Graph anomaly detection
  - Aims to identify anomalous graph objects
    - Social network
    - Financial fraud
    - Network intrusion, cybersecurity, sensor network …

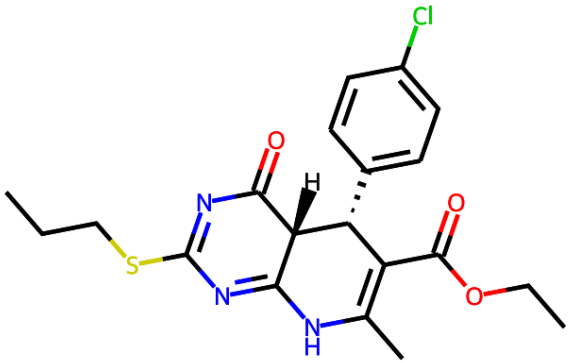
Anomaly Detection in Continuous-Time Temporal Provenance Graphs, NIPS' 2023

CONTENTS

Preliminary

Graph





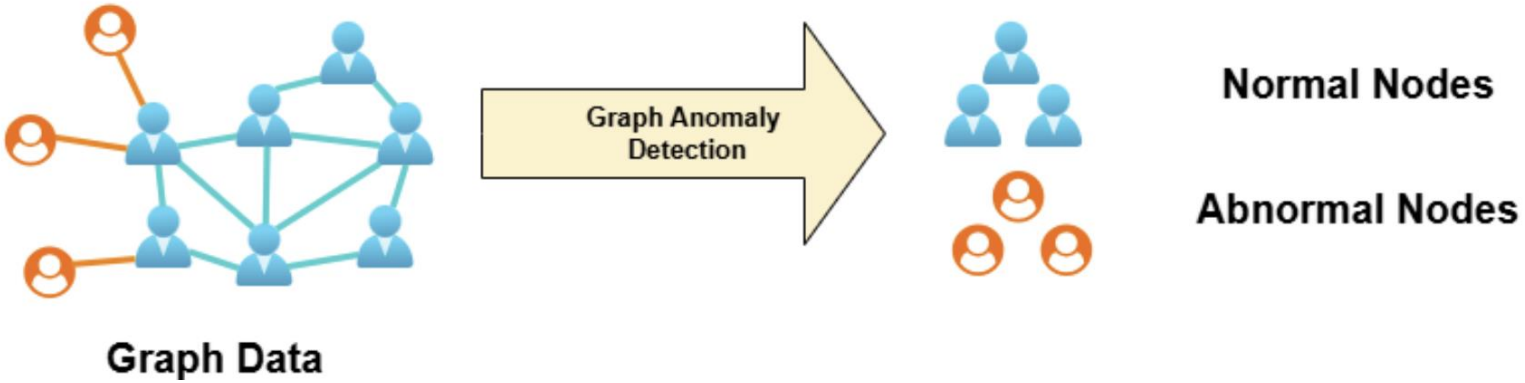
Himol (nature comm,2023)

- Abstract relations, topology, or connectivity
- Graphs G=(V,E)
  - V : a set of vertices (node)
  - E : a set of edges (links, relations)
    + weight (distance, strength, connection)
  - Social media, computer network, finance transaction, …

Preliminary

Graph Anomaly Detection

- Identifying nodes whose structural connectivity or attribute pattern
  - ✓ Deviate from normal graph pattern



A Zero-shot Generalized Graph Anomaly Detection Framework via Node Reconstruction, arXiv, 2026

Preliminary

Graph Anomaly Detection

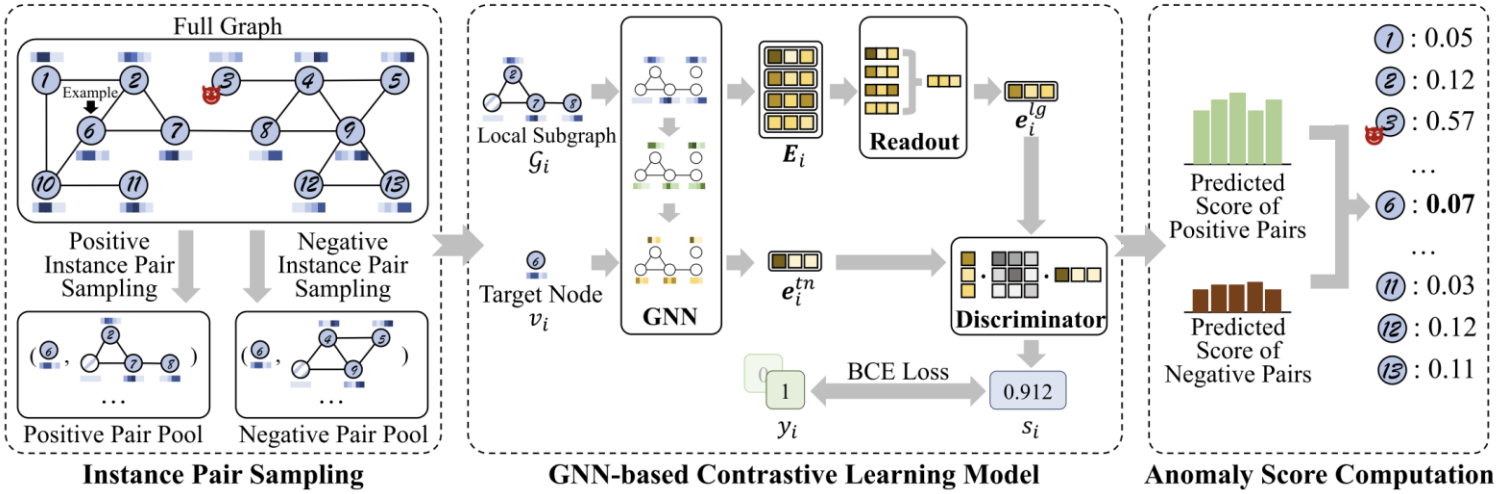
- $\mathcal{G} = (\mathrm{A}, X)$
  - Attributed graph
- $\mathrm{A} \in \mathbb{R}^{N \times N}$ with $N = |V|$
  - Adjacency matrix
    - $A_{ij} = 1\ iff$ exist an edge between nodes i and j
    - $A_{ij} = 0$; otherwise
    - If with self loops $\tilde{A} = A + I_n$
      - $I_n \in \mathbb{R}^{N \times N}$
- X
  - Features $\in \mathbb{R}^{N \times d}$
- $\mathcal{V}_n, \mathcal{V}_a$
  - Normal node set, abnormal node set
- $y \in \{0,1\}^N$
  - $y_i = 0$: normal node label
  - $y_i = 1$: abnormal node label

Preliminary

Node level GAD

- $f: \mathbb{R}^{N\times \mathrm{N}} \times \mathbb{R}^{N\times D} \rightarrow \mathbb{R}^{N}$
  - $f(v) < f(v'), \forall v \in \mathcal{V}_n, \forall v' \in \mathcal{V}_a$
  - $|\mathcal{V}_n| \gg |\mathcal{V}_a|$

- Evaluation
  - AUROC, AUPRC



Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning, IEEE TNNLS, 2022

Preliminary

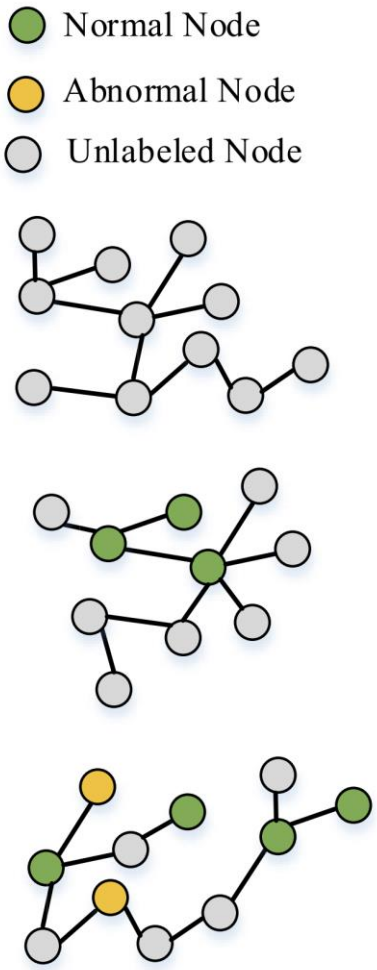
Categorization by supervision level

- Unsupervised
  - No label both normal and abnormal during training
  - Using graph structure and attribute
  - Learning normal nodes characteristic

- Semi-supervised
  - Only normal labeled node
  - Like one-class classification

- Fully-supervised (open-set supervised GAD)
  - some/few labeled anomalies
    - Train : normal + known anomaly
    - Test : Known anomaly + Unknown anomaly

  - What is the difference compared to binary classification?
    - Anomalies are not a class??



Preliminary

Major problem

- Graph problem
  - Structural dependency
  - Diverse types of graph
  - Computational complexity
- Anomaly Detection Problem
  - Unwariness of abnormality
  - Data imbalance
  - Abnormality camouflage
- GAD problem
  - Diverse graph anomaly instances
  - Large variation in graph abnormality

Deep Graph Anomaly Detection: A Survey and New Perspectives, IEEE TKDE, 2025

Preliminary

Challenge

- Class imbalance
- Abnormal sample are difficult to label
- Unknown and camouflage of anomalies
- Contamination of normal nodes
- Abnormally interpretability

- Graph structure-aware
- GAD at scales
- Generalization to different graph anomalies
- Balanced GAD
- Robust and interpretable GAD

Deep Graph Anomaly Detection: A Survey and New Perspectives, IEEE TKDE, 2025

CONTENTS

AnomalyGFM: Graph Foundation Model for Zero/Few-shot Anomaly Detection

Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2

Hezhe Qiao[1)], Chaoxi Niu[1)], Ling Chen[1)], Guansong Pang[2)]

1) School of Computing and Information Systems, Singapore Management University, Singapore
2) Australian Artificial Intelligence Institute University of Technology Sydney Sydney, Australia

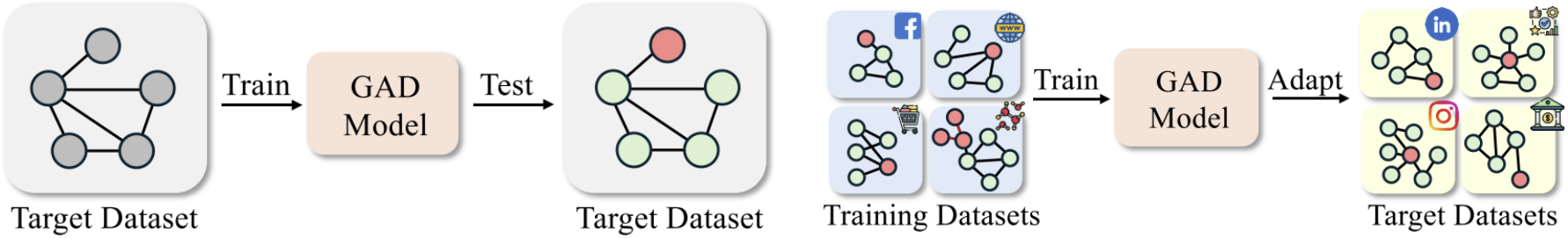
AnomalyGFM

Existing methods, either supervised or unsupervised method

- Follow a paradigm **of training and inference on the same graph**
- Due to the distribution shift between training and testing graph.
- Existing method **struggle to generalize to new/unseen graphs**
- And inapplicable in applications where graph data is not accessible during training

- **Foundation model** achieved in graph classification and link prediction
  - **But difficult to the GAD task**
    - Generic methods **inclines to the prevalent regular pattens**
    - But differ from abnormality patterns across the graphs

- **GAD generalist model not support both zero and few shot**



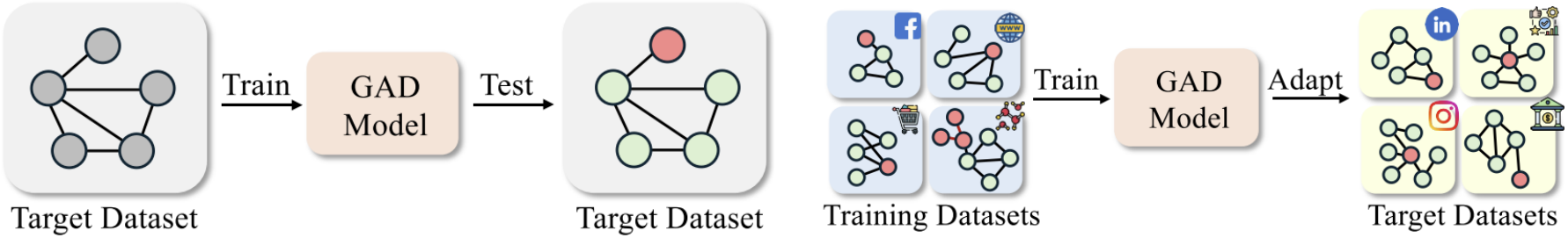
AnomalyGFM

To fill this gap, we introduce AnomalyGFM

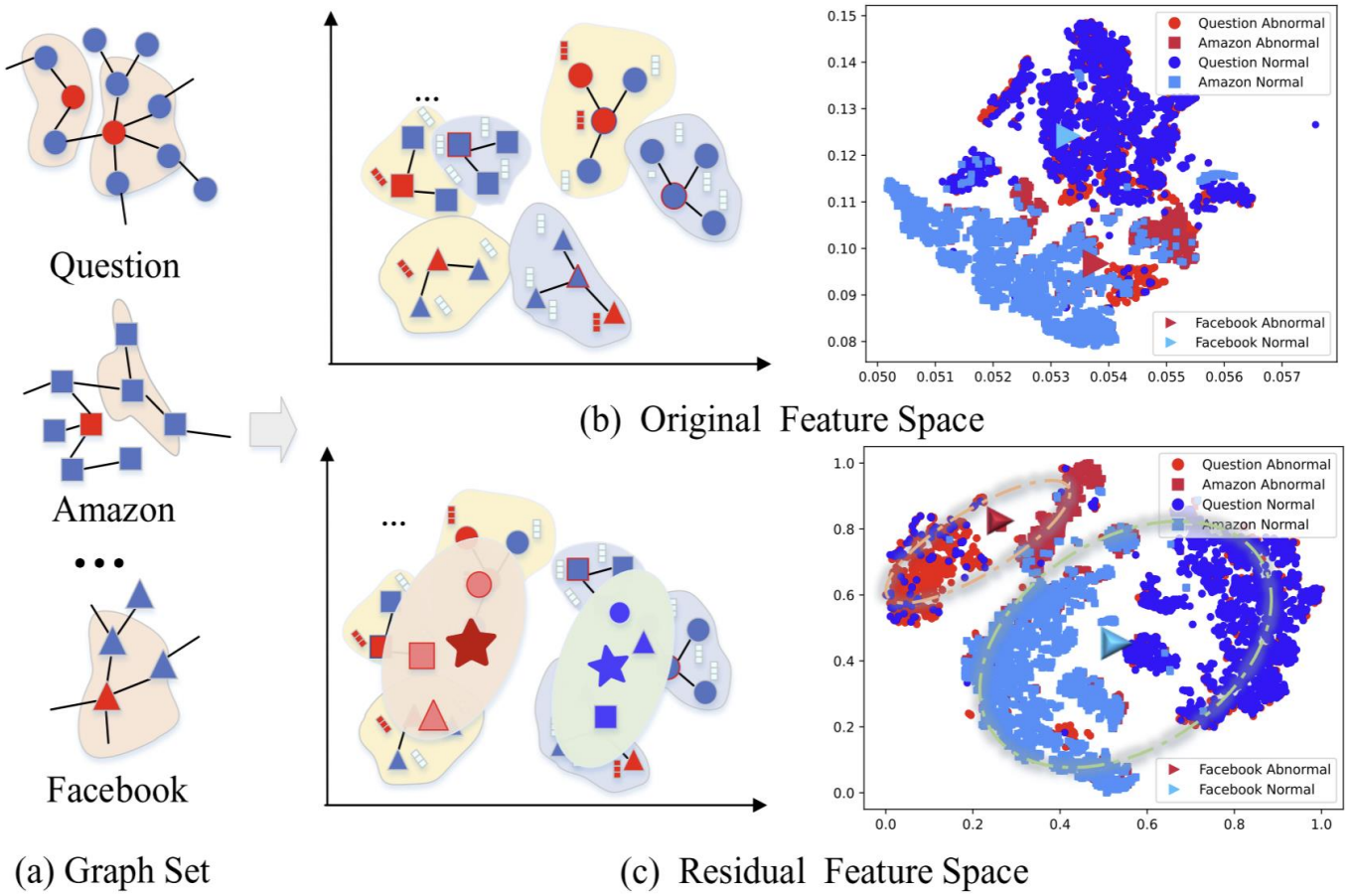
- **GAD-oriented graph foundation model**
- **That is capable of both zero shot and few shot**

- **Key insight is that to avoid overfitting the pre-training graph data**
- **Graph-agnostic representation for normal and abnormal class**
- **Required to support zero/fewshot GAD across different graphs**

- **To align data-independent, learnable normal and abnormal class prototype**



AnomalyGFM



Residual feature

- Normal and abnormal pattern are irregular and heterogeneously distributed in different datasets.
- we need to project node information into unified feature space

AnomalyGFM

Residual feature

- Normal and abnormal pattern are irregular and heterogeneously distributed in different datasets.
- we need to project node information **into unified feature space**

- **Where can measure the abnormality of nodes from different graphs**
  - In consistently identical ways

- Normal nodes are expected to small residual from neighbor
- Abnormal nodes are large

- By learning to align such residual feature with prototype
  - Distills the **residual feature discriminability** into **graph-agnostic** prototype

- Employing **graph-agnostic** class prototypes, facilitating **strong generalization to unseen/new graph**
  - without additional tuning and training

- To further support few shot, leveraging graph prompt tuning

AnomalyGFM

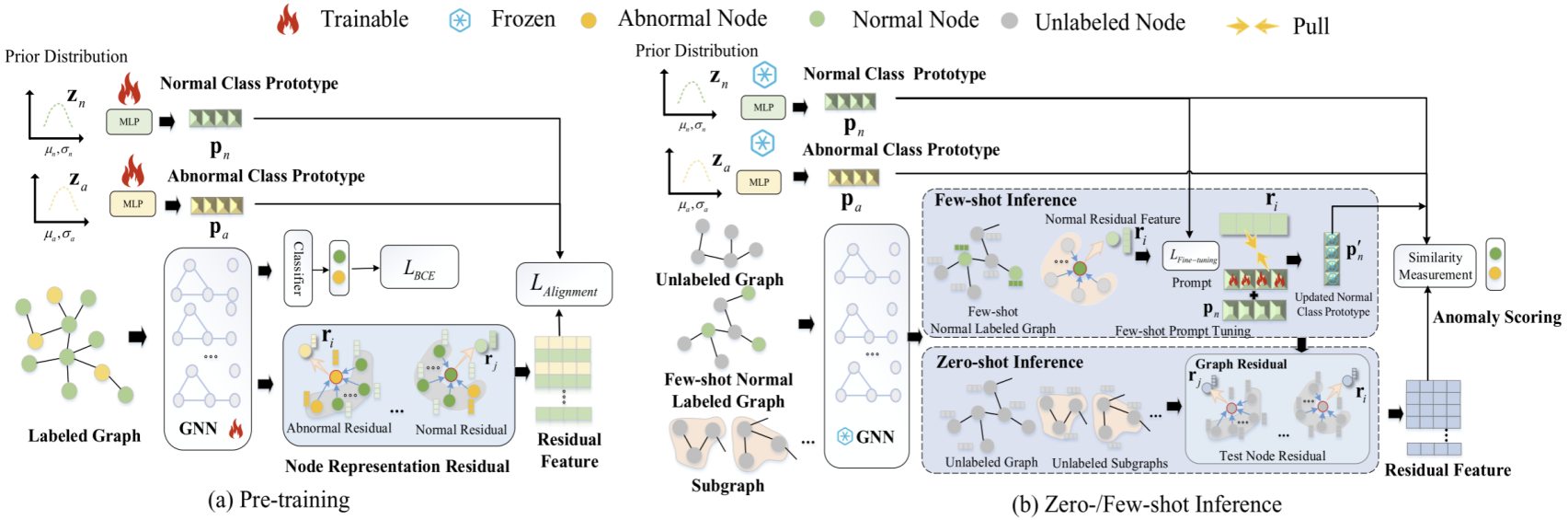


Figure 2: Overview of AnomalyGFM. (a) During the pre-training, AnomalyGFM aims to align two learnable class-level prototypes with the representation residuals of normal and abnormal nodes to learn discriminative graph-agnostic prototypes on an auxiliary labeled graph. (b) During inference, the anomaly score is determined in a consistent way using similarity between node representation residual and prototypes where all learnable parameters are frozen. Given a target graph with few-shot labeled normal nodes, these nodes are used to better adapt the pre-trained normal class prototypes to the target graph through a one-class prompt tuning method.

3.1 Preliminaries

Feature Unification

- Due to the feature difference across the graph
- Necessity to align the node feature into shared feature space
- We employ singular value decomposition
  - Project high-dim into low dim

$$\mathbf{X}^{(i)} \in \mathbb{R}^{N^{(i)} \times d^{(i)}} \xrightarrow[\text{Projection}]{\text{Feature}} \tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^{N^{(i)} \times d'},$$

3.3 Pre-training via Prototype Alignment in a Unified Feature Space

Node Representation Residual

- Representation between target node and its neighbor
  - Normal node expected be small
  - Abnormal is relatively large

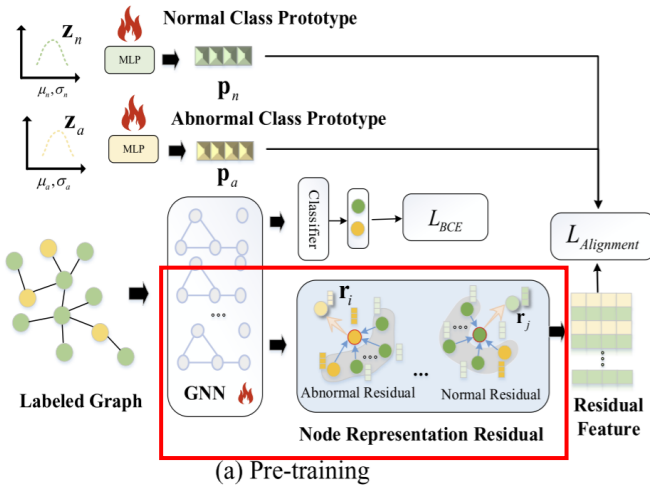
  - Offer consistent method to capture heterogeneous abnormalities across graphs

Node representation Outputted by the GNN $\in \mathbb{R}^d$

- $r_i = h_i - \frac{1}{|N(v_i)|} \sum_{v_j \in N(v_i)} h_j$

Residual feature $\in \mathbb{R}^d$

Neighbor set of v_i



(a) Pre-training

3.3 Pre-training via Prototype Alignment in a Unified Feature Space

Graph-agnostic Prototype

- To exploit the anomaly discriminability

- Learn two data-independent learnable prototype

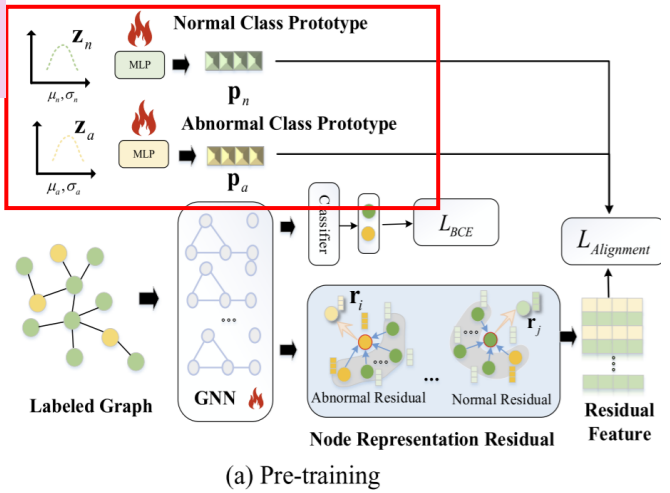
learnable mapping function

- $p_n = \Phi(z_n; \Theta_n) \in \mathbb{R}^T$ — dimension of learnable prototype
- $p_a = \Phi(z_a; \Theta_a) \in \mathbb{R}^T$

prototype

Initialization drawn from a $\mathcal{N}(\mu, \sigma)$
- μ : mean of gaussian
- σ : covariance of gaussian

Θ_n, Θ_a: learnable parameter



(a) Pre-training

3.3 Pre-training via Prototype Alignment in a Unified Feature Space

Graph-agnostic Prototype

- To exploit the anomaly discriminability

$$L_{BCE} = \sum_{i}^{v} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

Total loss

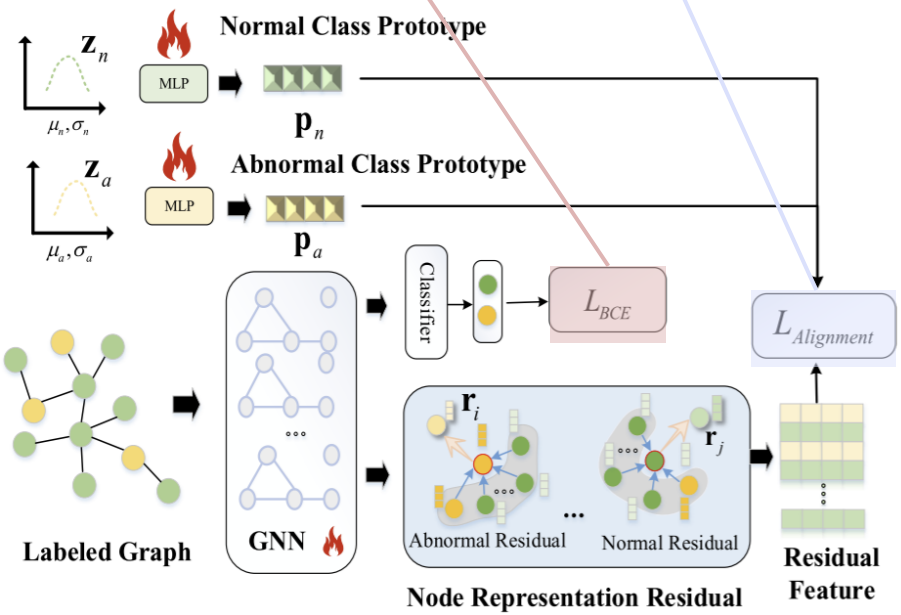
Binary classification loss

Hyperparameter align module

$$L_{total} = L_{BCE} + \alpha L_{alignment}$$

Alignment loss

$$L_{alignment} = I_{y_i=0} ||r_i - p_n||_2^2 + I_{y_i=1} ||r_i - p_a||_2^2$$



(a) Pre-training

3.4 Few-shot Prompt Tuning

Normal Prototype-based Prompting

- Introduce small learnable parameters and adaption layer for flexibility

New normal prompt

Adaptation layer

$$p'_n = p_n + g(p_n; \phi) + \Psi_{p_n}$$

Pretrained normal prompt

learnable parameter

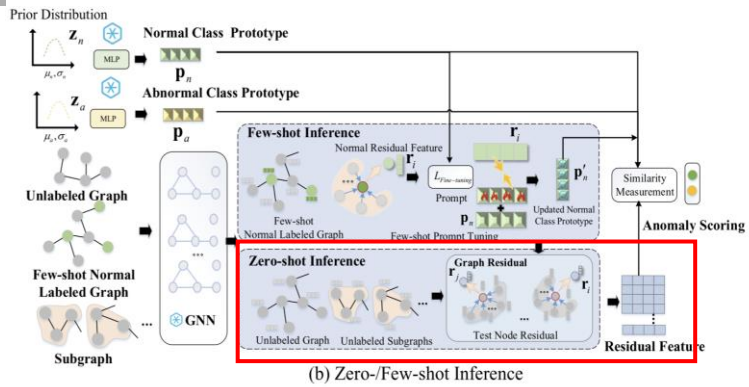
One Class Prompt Tuning

- **calculate the residual and optimize**

Labeled normal node

$$L_{pt} = \sum_{i=1}^{|V_l^{Test}|} \left|\left| r_i - p'_n \right|\right|_2^2$$

Residual



(b) Zero-/Few-shot Inference

3.5 GAD Using AnomalyGFM

Zero-shot inference

- Trainable parameter are all frozen
- obtain residual feature for each node
- Anomaly score is determined by the similarity

Target nodes' representation

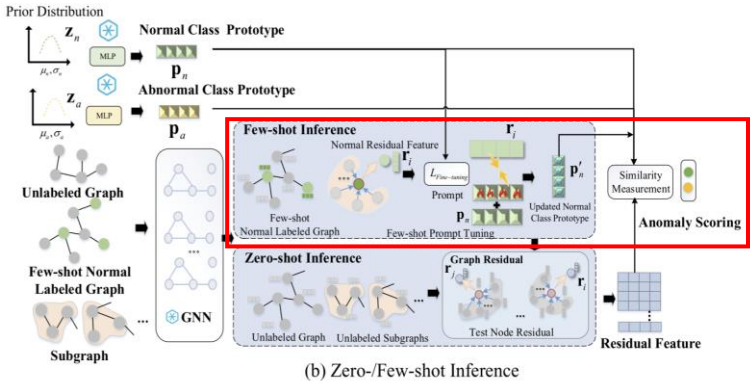
Anomaly score $s_i = \exp(\mathrm{r_i^T p_a}) + \beta \exp(-\mathrm{r_i^T p'_n})$

abnormal prototype

normal prototype

Hyperparameter

- Anomaly scoring calculation
  - Direct similarity Normal prototype
  - Inverse similarity abnormal prototype



(b) Zero-/Few-shot Inference

3.5 GAD Using AnomalyGFM

Few shot inference

- Trainable parameter are all frozen
- obtain residual feature for each node
- Anomaly score is determined by the similarity

Target nodes' representation

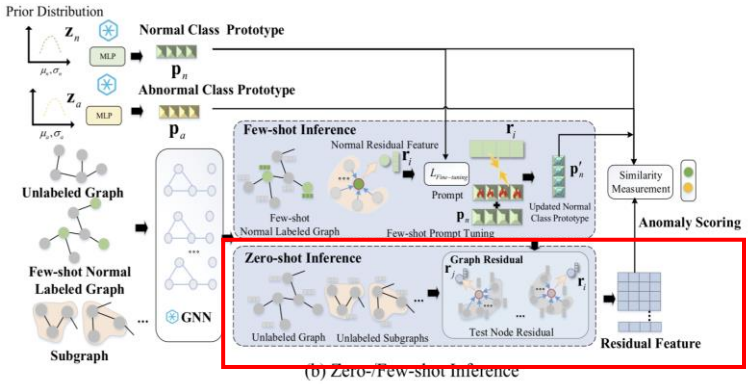
Anomaly score $s_i = \exp(\mathrm{r_i^T p_a}) + \beta \exp(-\mathrm{r_i^T p_n})$

abnormal prototype

normal prototype

Hyperparameter

- Anomaly scoring calculation
  - Direct similarity Normal prototype
  - Inverse similarity abnormal prototype



3.5 GAD Using AnomalyGFM

Inference on Very Large-scale Graphs

- Esting GAD require loading all nodes of the grash
- Leads to poor scalability on large graph during inference
- Through the subgraph-based inference
  - We can generalize to large-scale graphs

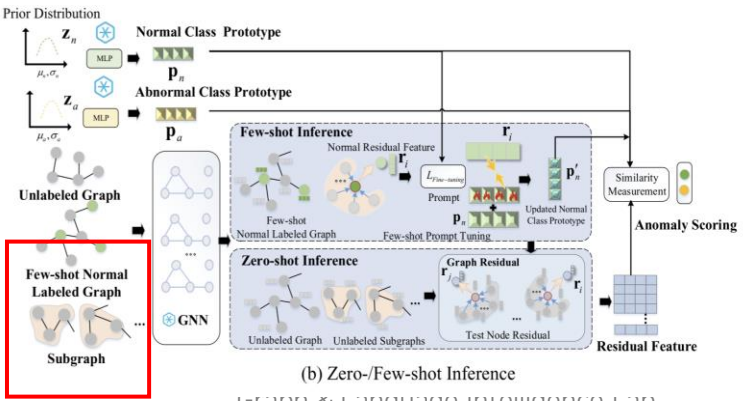
Node representation Outputted by the GNN $\in \mathbb{R}^d$

- $r_i = h_i - \frac{1}{|S(v_i)|} \sum_{v_j \in S(v_i)} h_j$

Residual feature $\in \mathbb{R}^d$

subgraph of v_i

Extract using RW



4 Experiments

4.1 Experimental Setup

- Utilize 11 benchmark dataset
  - From social networks, finance and co-review domains

Table 1: Key statistics of GAD datasets used in our experiments. Ano. indicates the number (rate) of anomalies in the graph and Sim. indicates the global average edge similarity.

Dataset	Domain	#Nodes	#Edges	#Feat.	Ano.	Sim.
Facebook	Social Networks	1,081	55,104	576	25(2.31%)	0.690
Reddit	Social Networks	10,984	168,016	64	366(3.33%)	0.997
Amazon	Co-review	10,244	175,608	25	693(6.66%)	0.645
Disney	Co-purchase	124	335	28	6(4.8%)	0.804
Amazon-all	Co-review	11,944	4,398,392	25	821(6.87%)	0.645
YelpChi-all	Co-review	45,941	3,846,979	32	6,674(14.52%)	0.905
Tolokers	Work Collaboration	11,758	519,000	10	2,566(21.8%)	0.814
Question	Social Networks	48,921	153,540	301	1,460(2.98%)	0.679
Elliptic	Bitcoin Transaction	46,564	73,248	93	4,545 (9.8%)	0.356
T-Finance	Transaction Record	39,357	21,222,543	10	1,803(4.6%)	0.107
T-Social	Social Friendship	5,781,065	73,105,508	10	174,280(3.0%)	0.307

4 Experiments

4.2 Zero-shot GAD performance

Table 2: AUROC and AUPRC results for zero-shot GAD on nine real-world datasets, with the models trained on Facebook only. For each dataset, the best performance per column within each metric is boldfaced, with the second-best underlined. Avg. denotes the average performance. P-values are from one-tailed Wilcoxon signed rank tests.

Metric	Method	Dataset								Avg.	p-value
		Reddit	Amazon	Disney	Aamzon-all	YelpChi-all	Tolokers	Question	Elliptic	T-Finance		
AUROC	Unsupervised Methods											
	AnomalyDAE (ICASSP'20)	0.5016	0.5818	0.4853	0.7228	0.5002	0.5948	0.4311	0.4197	0.2324	0.4966	0.007
	CoLA (TNNLS'21)	0.4623	0.4580	0.4696	0.4091	0.4879	0.4501	0.4945	0.5572	0.4889	0.4752	0.003
	TAM (NeurIPS'23)	0.5725	0.4720	0.4773	0.7543	0.4216	0.5351	0.5119	0.3282	0.2990	0.4857	0.003
	GADAM (ICLR'24)	0.4532	0.6646	0.4288	0.5959	0.4829	0.4832	**0.5594**	0.3922	0.1382	0.4664	0.007
	Supervised Methods											
	GCN (ICLR'17)	0.5645	0.5988	0.5000	0.7195	0.5486	0.5319	0.5161	**0.7640**	0.2345	0.5531	0.039
	GAT (ICLR'18)	0.5000	0.4981	0.5175	0.5005	0.4802	0.5030	0.4577	0.6588	0.5072	0.5136	0.007
	BWGNN (ICML'22)	0.5208	0.4769	0.6073	0.3648	0.5282	0.4877	0.4404	0.5843	0.5457	0.5062	0.003
	GHRN (WebConf'23)	0.5253	0.4560	0.5336	0.3382	0.5125	0.4860	0.4535	0.5400	0.5324	0.4863	0.003
	XGBGraph (NeurIPS'23)	0.4601	0.4179	0.6692	0.7950	0.4945	0.5462	0.5095	0.4274	0.3402	0.5177	0.003
	Generalist Methods											
	GraphPrompt (WebConf'23)	0.4677	0.4904	0.5192	0.3215	0.4976	0.4779	0.4204	0.3221	0.5405	0.4508	0.003
	UNPrompt (Arxiv'24)	0.5337	0.7525	0.6412	0.7962	0.5558	**0.6853**	0.4757	0.5901	0.2318	0.5847	0.074
	AnomalyGFM	**0.5974**	**0.8417**	**0.6751**	**0.9032**	**0.5791**	0.5843	0.5280	0.6195	**0.5614**	0.6544	/
AUPRC	Unsupervised Methods											
	AnomalyDAE (ICASSP'20)	0.0327	0.0833	0.0566	0.1921	0.1484	0.1876	0.0241	0.0798	0.0274	0.0924	0.003
	CoLA (TNNLS'21)	0.0391	0.0669	0.0701	0.0861	0.1466	0.0848	0.0292	0.0998	0.0430	0.0739	0.007
	TAM (NeurIPS'23)	0.0413	0.0666	0.0628	0.1736	0.1240	0.0970	0.0307	0.0697	0.0332	0.0776	0.007
	GADAM (ICLR'24)	0.0293	0.1562	0.0651	0.1595	0.1371	0.1001	0.0395	0.0733	0.0261	0.0873	0.003
	Supervised Methods											
	GCN (ICLR'17)	**0.0439**	0.0891	0.0484	0.1536	0.1735	0.1060	0.0387	**0.1963**	0.0274	0.0974	0.074
	GAT (ICLR'18)	0.0329	0.0688	0.0530	0.0696	0.1400	0.0822	0.0259	0.1366	0.0463	0.0728	0.003
	BWGNN (ICML'22)	0.0389	0.0652	0.0624	0.0586	0.1605	0.1030	0.0257	0.1158	0.0479	0.0753	0.007
	GHRN (WebConf'23)	0.0407	0.0633	0.0519	0.0569	0.1505	0.0957	0.0259	0.1148	0.0457	0.0717	0.007
	XGBGraph (NeurIPS'23)	0.0330	0.0536	0.1215	0.2307	0.1449	0.1256	0.0306	0.0816	**0.0754**	0.0996	0.027
	Generalist Methods											
	GraphPrompt (WebConf'23)	0.0334	0.0661	0.0610	0.0666	0.1542	0.2070	0.0266	0.0664	0.0492	0.0811	0.003
	UNPrompt (Arxiv'24)	0.0351	0.1602	0.1236	0.2430	0.1810	0.2219	0.0348	0.1278	0.0279	0.1283	0.003
	AnomalyGFM	0.0387	**0.5790**	**0.1242**	**0.6820**	**0.1819**	**0.2749**	**0.0397**	0.1371	0.0593	**0.2352**	/

1) Consistency outperforms the three type competing methods
2) Unsupervised method often fail to good result
   - Focus on some specific abnormal pattern

4 Experiments

4.2 Zero-shot GAD performance

	Generalist Methods										
	GraphPrompt (WebConf'23)	0.4677	0.4904	0.5192	0.3215	0.4976	0.4779	0.4204	0.3221	0.5405	0.4508	0.003
	UNPrompt (Arxiv'24)	0.5337	0.7525	0.6412	0.7962	0.5558	**0.6853**	0.4757	0.5901	0.2318	0.5847	0.074
	AnomalyGFM	**0.5974**	**0.8417**	**0.6751**	**0.9032**	**0.5791**	0.5843	0.5280	0.6195	**0.5614**	**0.6544**	/
AUPRC	Unsupervised Methods											
	AnomalyDAE (ICASSP'20)	0.0327	0.0833	0.0566	0.1921	0.1484	0.1876	0.0241	0.0798	0.0274	0.0924	0.003
	CoLA (TNNLS'21)	0.0391	0.0669	0.0701	0.0861	0.1466	0.0848	0.0292	0.0998	0.0430	0.0739	0.007
	TAM (NeurIPS'23)	0.0413	0.0666	0.0628	0.1736	0.1240	0.0970	0.0307	0.0697	0.0332	0.0776	0.007
	GADAM (ICLR'24)	0.0293	0.1562	0.0651	0.1595	0.1371	0.1001	0.0395	0.0733	0.0261	0.0873	0.003
	Supervised Methods											
	GCN (ICLR'17)	0.0439	0.0891	0.0484	0.1536	0.1735	0.1060	0.0387	0.1963	0.0274	0.0974	0.074
	GAT (ICLR'18)	0.0329	0.0688	0.0530	0.0696	0.1400	0.0822	0.0259	0.1366	0.0463	0.0728	0.003
	BWGNN (ICML'22)	0.0389	0.0652	0.0624	0.0586	0.1605	0.1030	0.0257	0.1158	0.0479	0.0753	0.007
	GHRN (WebConf'23)	0.0407	0.0633	0.0519	0.0569	0.1505	0.0957	0.0259	0.1148	0.0457	0.0717	0.007
	XGBGraph (NeurIPS'23)	0.0330	0.0536	0.1215	0.2307	0.1449	0.1256	0.0306	0.0816	**0.0754**	0.0996	0.027
	Generalist Methods											
	GraphPrompt (WebConf'23)	0.0334	0.0661	0.0610	0.0666	0.1542	0.2070	0.0266	0.0664	0.0492	0.0811	0.003
	UNPrompt (Arxiv'24)	0.0351	0.1602	0.1236	0.2430	0.1810	0.2219	0.0348	0.1278	0.0279	0.1283	0.003
	AnomalyGFM	0.0387	**0.5790**	**0.1242**	**0.6820**	**0.1819**	**0.2749**	**0.0397**	0.1371	0.0593	**0.2352**	/

3) Most supervised methods perform well in zero-shot inference,
- achieves top performance on the Elliptic dataset
- simple model design, avoiding heavy overfitting to the training graph.
- However, the performance still underperforms AnomalyGFM
  - demonstrating that the node representation
  - **residual** is more generalized and our prototype alignment module
  - provides a strong add-on to the binary loss function in our method.

4 Experiments

4.2 Zero-shot GAD performance

	Generalist Methods										
	GraphPrompt (WebConf'23)	0.4677	0.4904	0.5192	0.3215	0.4976	0.4779	0.4204	0.3221	0.5405	0.4508	0.003
	UNPrompt (Arxiv'24)	0.5337	0.7525	0.6412	0.7962	0.5558	**0.6853**	0.4757	0.5901	0.2318	0.5847	0.074
	AnomalyGFM	**0.5974**	**0.8417**	**0.6751**	**0.9032**	**0.5791**	0.5843	0.5280	0.6195	**0.5614**	**0.6544**	/
	Unsupervised Methods											
	AnomalyDAE (ICASSP'20)	0.0327	0.0833	0.0566	0.1921	0.1484	0.1876	0.0241	0.0798	0.0274	0.0924	0.003
	CoLA (TNNLS'21)	0.0391	0.0669	0.0701	0.0861	0.1466	0.0848	0.0292	0.0998	0.0430	0.0739	0.007
	TAM (NeurIPS'23)	0.0413	0.0666	0.0628	0.1736	0.1240	0.0970	0.0307	0.0697	0.0332	0.0776	0.007
	GADAM (ICLR'24)	0.0293	0.1562	0.0651	0.1595	0.1371	0.1001	0.0395	0.0733	0.0261	0.0873	0.003
	Supervised Methods											
	GCN (ICLR'17)	**0.0439**	0.0891	0.0484	0.1536	0.1735	0.1060	0.0387	**0.1963**	0.0274	0.0974	0.074
AUPRC	GAT (ICLR'18)	0.0329	0.0688	0.0530	0.0696	0.1400	0.0822	0.0259	0.1366	0.0463	0.0728	0.003
	BWGNN (ICML'22)	0.0389	0.0652	0.0624	0.0586	0.1605	0.1030	0.0257	0.1158	0.0479	0.0753	0.007
	GHRN (WebConf'23)	0.0407	0.0633	0.0519	0.0569	0.1505	0.0957	0.0259	0.1148	0.0457	0.0717	0.007
	XGBGraph (NeurIPS'23)	0.0330	0.0536	0.1215	0.2307	0.1449	0.1256	0.0306	0.0816	**0.0754**	0.0996	0.027
	Generalist Methods											
	GraphPrompt (WebConf'23)	0.0334	0.0661	0.0610	0.0666	0.1542	0.2070	0.0266	0.0664	0.0492	0.0811	0.003
	UNPrompt (Arxiv'24)	0.0351	0.1602	0.1236	0.2430	0.1810	0.2219	0.0348	0.1278	0.0279	0.1283	0.003
	AnomalyGFM	0.0387	**0.5790**	**0.1242**	**0.6820**	**0.1819**	**0.2749**	**0.0397**	0.1371	0.0593	**0.2352**	/

4) Generalist method such as Graphprompt

- Design for general nod/graph classification
- Struggle to perform on GAD
  - Irregular nature abnormalities in different GAD datasets
  - Hindering the applicability of the regular pattern/knowledge

4 Experiments

4.3 Few shot GAD performance

Table 3: Results on the nine real-world GAD datasets under 1/5/10-shot with models tuned on the few-shot data.

Metric	Setting	Method	Dataset									Avg.	p-value
			Reddit	Amazon	Disney	Aamzon-all	YelpChi-all	Tolokers	Question	Elliptic	T-Finance		
AUROC	1-shot	GPPT (KDD'22)	0.5000	0.5303	0.4997	0.5010	0.5000	0.5061	0.4921	0.6162	0.3647	0.5011	0.003
		GraphPrompt (WebConf'23)	0.4216	0.4882	0.4223	0.2631	0.4811	0.5328	0.4086	0.6001	0.4000	0.4464	0.004
		ARC (NeurIPS'24)	0.4899	0.4571	0.3578	0.4570	0.4910	0.4667	**0.5865**	0.2904	0.2484	0.4272	0.008
		AnomalyGFM	**0.5922**	**0.8531**	**0.6649**	**0.8972**	**0.5872**	**0.5898**	0.5303	**0.6199**	**0.5916**	**0.6584**	/
	5-shot	GPPT (KDD'22)	0.5000	0.5098	0.5000	0.5051	0.5000	0.5181	0.4959	0.5736	0.2609	0.4848	0.003
		GraphPrompt (WebConf'23)	0.4406	0.4900	0.6497	0.4726	0.5359	0.5381	0.4069	0.6012	0.4069	0.5046	0.003
		ARC (NeurIPS'24)	0.4720	0.4458	0.4435	0.4473	0.5112	0.4746	**0.5906**	0.2714	0.2168	0.4303	0.007
		AnomalyGFM	**0.6023**	**0.8600**	**0.6613**	**0.9011**	**0.5951**	**0.6095**	0.5426	**0.6119**	**0.6248**	**0.6676**	/
	10-shot	GPPT (KDD'22)	0.5000	0.5087	0.4769	0.5023	0.5000	0.4971	0.5047	0.4212	0.5539	0.4961	0.003
		GraphPrompt (WebConf'23)	0.4321	0.4906	0.6314	0.7167	0.5367	0.5329	0.3826	0.6221	0.4260	0.5301	0.007
		ARC (NeurIPS'24)	0.4867	0.4323	0.4769	0.4467	0.5145	0.4786	**0.5901**	0.2644	0.2298	0.4355	0.003
		AnomalyGFM	**0.6252**	**0.8649**	**0.6649**	**0.9215**	**0.6064**	**0.6140**	0.5611	**0.6303**	**0.6283**	**0.6796**	/
AUPRC	1-shot	GPPT (KDD'22)	0.0333	0.0766	0.0488	0.0687	0.1453	0.2204	0.0294	0.1239	0.0432	0.0877	0.003
		GraphPrompt (WebConf'23)	0.0283	0.0680	0.0486	0.0426	0.1113	0.2321	**0.0448**	0.1108	0.0302	0.0796	0.012
		ARC (NeurIPS'24)	0.0332	0.0581	0.0453	0.0590	0.1402	0.2122	0.0468	0.0701	0.0277	0.0769	0.011
		AnomalyGFM	**0.0398**	**0.5801**	**0.1223**	**0.6921**	**0.1852**	**0.2786**	0.0332	**0.1401**	**0.0601**	**0.2368**	/
	5-shot	GPPT (KDD'22)	0.0333	0.0692	0.0504	0.0716	0.1453	0.2265	0.0297	0.1127	0.0365	0.0861	0.004
		GraphPrompt (WebConf'23)	0.0285	0.0681	0.0892	0.0600	0.1661	**0.2957**	0.0327	0.1416	0.0360	0.1019	0.009
		ARC (NeurIPS'24)	0.0312	0.0571	0.0546	0.0572	0.1464	0.2150	**0.0471**	0.0726	0.0267	0.0786	0.011
		AnomalyGFM	**0.0401**	**0.5831**	**0.1257**	**0.6985**	**0.1918**	0.2866	0.0336	**0.1437**	**0.0622**	**0.2405**	/
	10-shot	GPPT (KDD'22)	0.0334	0.0691	0.0526	0.0698	0.1453	0.2178	0.0301	0.0905	0.0511	0.0844	0.004
		GraphPrompt (WebConf'23)	0.0278	0.0681	0.0848	0.1427	0.1649	**0.2922**	0.0263	0.1421	0.0382	0.1096	0.007
		ARC (NeurIPS'24)	0.0327	0.0557	0.0743	0.0583	0.1491	0.2168	**0.0463**	0.0677	0.0272	0.0809	0.011
		AnomalyGFM	**0.0444**	**0.5895**	**0.1399**	**0.7124**	**0.1990**	0.2897	0.0346	**0.1570**	**0.0644**	**0.2478**	/

- AnomalyGfM achieve better sample-efficiency
- Class-level prototypes that agnostic to different domain of graphs
- Enhancing the generalization yielding the best performance

4 Experiments

4.4 Scale Up to Very Lage Graphs

Table 4: Subgraph inference on the very large-scale graph '/' indicates that the model cannot handle the dataset.

Metric	Method	Dataset	
		T-Finance	T-Social
AUROC	Unsupervised Methods		
	TAM (NeurIPS'23)	0.2990	/
	GADAM (ICLR'24)	0.1382	0.5155
	Supervised Methods		
	BWGNN (ICML'22)	0.5457	0.4964
	GHRN (WebConf'23)	0.5324	0.4934
	XGBGraph (NeurIPS'23)	0.3402	0.4602
	AnomalyGFM	**0.7852**	**0.5991**
AUPRC	Unsupervised Methods		
	TAM (NeurIPS'23)	0.0332	/
	GADAM (ICLR'24)	0.0261	0.0285
	Supervised Methods		
	BWGNN (ICML'22)	0.0479	0.0301
	GHRN (WebConf'23)	0.0457	0.0303
	XGBGraph (NeurIPS'23)	0.0754	0.0305
	AnomalyGFM	**0.1059**	**0.0398**

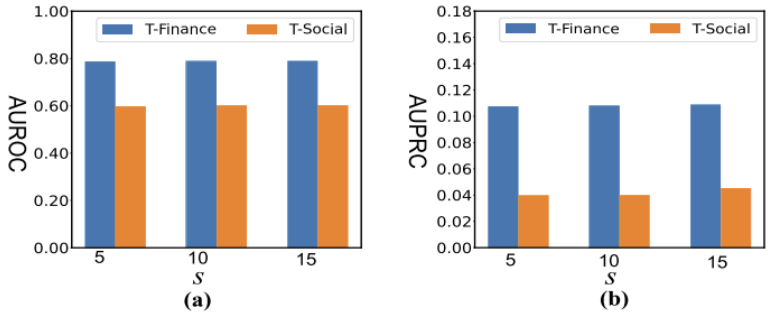


Figure 3: AnomalyGFM performance w.r.t subgraph size s.

AnomalyGFM outperform the unsupervised methods and supervised learning

- Data-independent prototypes
- Even in the presence of **subgraphs**, residual representation is performed well

4 Experiments

4.4 Scale Up to Very Lage Graphs

Table 4: Subgraph inference on the very large-scale graph
'/' indicates that the model cannot handle the dataset.

Metric	Method	Dataset	
		T-Finance	T-Social
AUROC	Unsupervised Methods		
	TAM (NeurIPS'23)	0.2990	/
	GADAM (ICLR'24)	0.1382	0.5155
	Supervised Methods		
	BWGNN (ICML'22)	0.5457	0.4964
	GHRN (WebConf'23)	0.5324	0.4934
	XGBGraph (NeurIPS'23)	0.3402	0.4602
	AnomalyGFM	**0.7852**	**0.5991**
AUPRC	Unsupervised Methods		
	TAM (NeurIPS'23)	0.0332	/
	GADAM (ICLR'24)	0.0261	0.0285
	Supervised Methods		
	BWGNN (ICML'22)	0.0479	0.0301
	GHRN (WebConf'23)	0.0457	0.0303
	XGBGraph (NeurIPS'23)	0.0754	0.0305
	AnomalyGFM	**0.1059**	**0.0398**

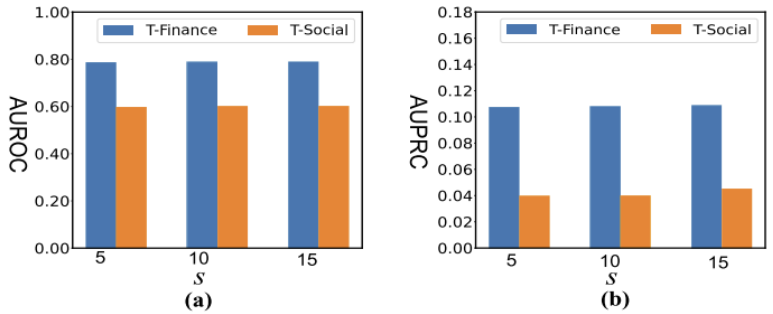


Figure 3: AnomalyGFM performance w.r.t subgraph size s.

Also evaluate the sensitivity of subgraph size

- Remain stable with varying subgraph sizes

AnomalyGFM

4 Experiments

4.5 Ablation study

Table 5: Ablation study results for AnomalyGFM and its variants.

Metric	Method	Dataset									
		Reddit	Amazon	Disney	Amazon-all	YelpChi-all	Tolokers	Question	Elliptic	T-Finance	**Avg.**
AUROC	FA	0.5697	0.5680	0.6073	0.5097	0.5211	0.5594	0.5326	**0.7765**	**0.6674**	0.5901
	BCE	0.5445	0.4308	**0.7542**	0.5545	0.5232	0.5187	0.4593	0.2581	0.6581	0.5223
	BCE-R	0.5108	0.5314	0.4887	0.5777	0.4590	0.4788	0.4601	0.5414	0.5380	0.5095
	AnomalyGFM	**0.5974**	**0.8417**	0.6751	**0.9032**	**0.5791**	**0.5843**	**0.5380**	0.6195	0.5614	**0.6555**
AUPRC	FA	**0.0400**	0.0776	0.0960	0.0684	0.1555	0.2478	0.0356	0.1199	**0.1495**	0.1100
	BCE	0.0394	0.0544	**0.2254**	0.0718	0.1594	0.2264	0.0272	0.0287	0.1389	0.1079
	BCE-R	0.0314	0.0677	0.0685	0.0787	0.1294	0.2122	0.0305	0.0658	0.0971	0.0868
	AnomalyGFM	0.0387	**0.5790**	0.1242	**0.6820**	**0.1819**	**0.2749**	**0.0397**	**0.1371**	0.0593	**0.2352**

FA : Feature Alignment : learning prompts from the original feature also effective but the are sub-optimal performance

BCE : not use similarity loss, using BCE for anomaly scoring

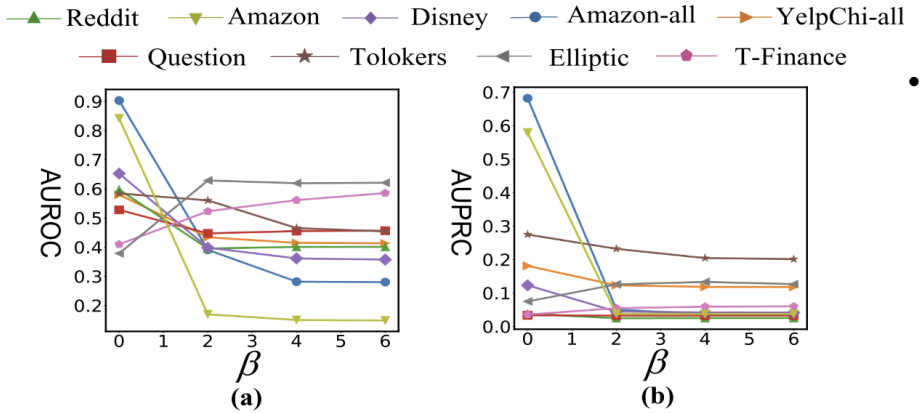
- Does not perform well except on the small dataset
- Struggle to generalize to new graphs due to the difference between train and test

BCE-R : using residual feature for prediction, use BCE for anomaly scoring

- Not achieve good performance

4 Experiments

4.5 Ablation study



- High global similarity
  - Node feature relatively small
  - Normal prototype score affect anomaly score
  - Anomaly scoring is set to zero by default

Figure 7: AUROC and AUPRC of AnomalyGFM w.r.t β.

Dataset	Domain	#Nodes	#Edges	#Feat.	Ano.	Sim.
Facebook	Social Networks	1,081	55,104	576	25(2.31%)	0.690
Reddit	Social Networks	10,984	168,016	64	366(3.33%)	0.997
Amazon	Co-review	10,244	175,608	25	693(6.66%)	0.645
Disney	Co-purchase	124	335	28	6(4.8%)	0.804
Amazon-all	Co-review	11,944	4,398,392	25	821(6.87%)	0.645
YelpChi-all	Co-review	45,941	3,846,979	32	6,674(14.52%)	0.905
Tolokers	Work Collaboration	11,758	519,000	10	2,566(21.8%)	0.814
Question	Social Networks	48,921	153,540	301	1,460(2.98%)	0.679
Elliptic	Bitcoin Transaction	46,564	73,248	93	4,545 (9.8%)	0.356
T-Finance	Transaction Record	39,357	21,222,543	10	1,803(4.6%)	0.107
T-Social	Social Friendship	5,781,065	73,105,508	10	174,280(3.0%)	0.307

6. Conclusion

AnomayGFM

- GAD-oriented graph foundation model
- Work under few-shot and zero-shot

- Learn discriminative and data-independent prototype
  - By aligning with graph-agnostic node residuals
  - This provides consistent and identical way for abnormality measurement

- Extensive experiments demonstrated the effectiveness and generalization

CONTENTS

Zero-shot Generalist Graph Anomaly Detection with Unified Neighborhood Prompts

Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence

Chaoxi Niu[1)], Hezhe Qiao[1)], Changlu Chen[1)]
Ling Chen[1)], Guansong Pang[2)]

1) AAII, University of Technology Sydney, Sydney, Australia
2) School of Computing and Information Systems, Singapore Management University, Singapore
3) Faculty of Data Science, City University of Macau, Macau, China

UNPrompt

3.3 Node Attribute Unification

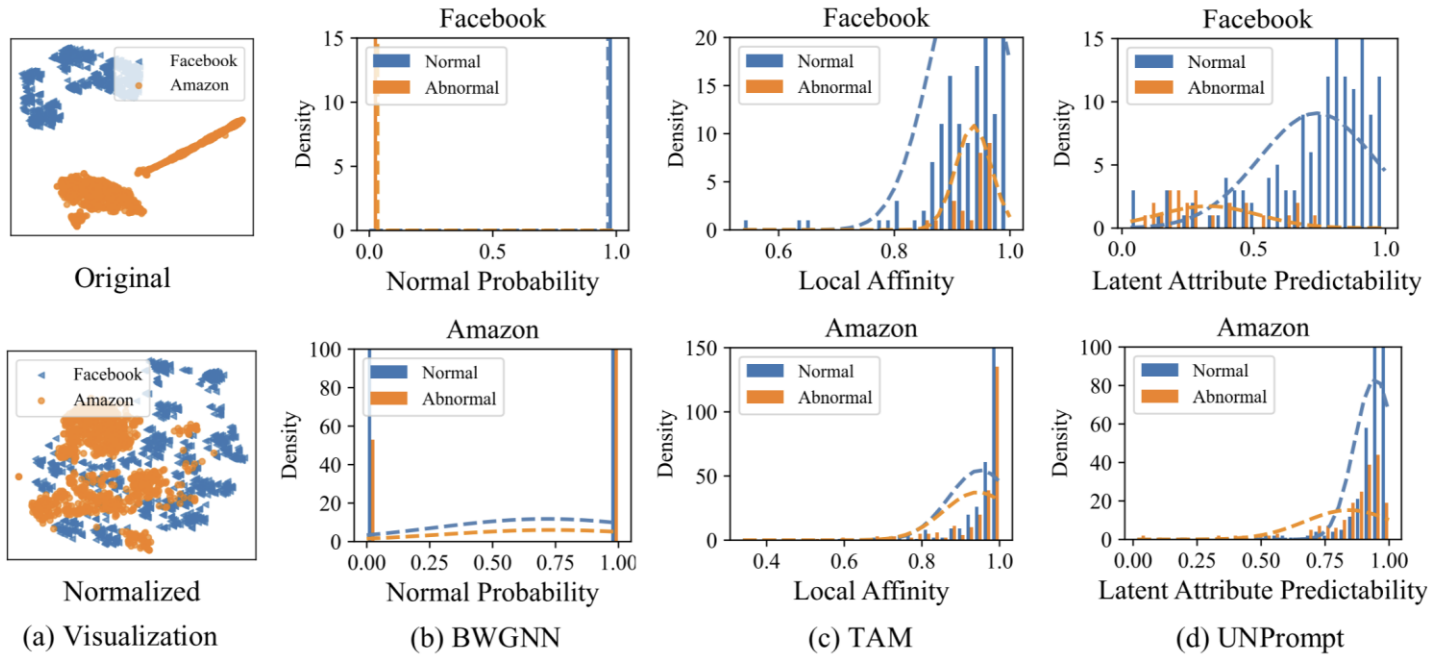


Figure 1: **(a)** Visualization of two popular GAD datasets: Facebook and Amazon, where the node attributes are unified into a common semantic space via our proposed normalization compared to the original heterogeneous raw attributes. **(b)-(d)** The anomaly scores of BWGNN (normal probability), TAM (local affinity), and UNPrompt (latent attribute predictability) on the two datasets, where the methods are all trained on Facebook and tested on Amazon under the zero-shot setting. It is clear that BWGNN and TAM struggle to generalize from Facebook to Amazon, while UNPrompt can learn well to generalize across the datasets.

UNPrompt

3.3 Node Attribute Unification

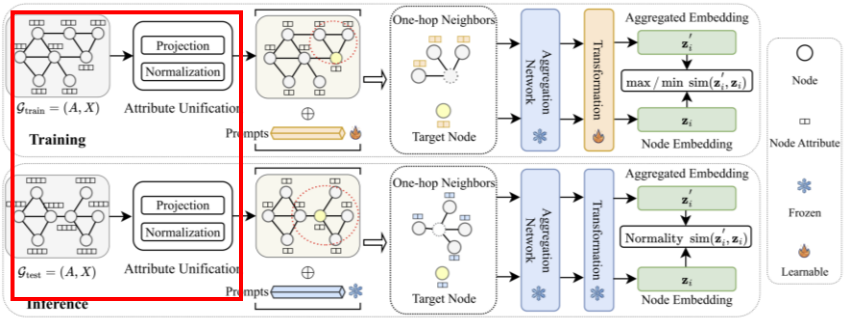
- Graph from different distribution and domains
  - ✓ Differ in the dimensionality and semantics of node attributes.

Feature Projection.

$$X^{(i)} \in \mathbb{R}^{N^{(i)} \times d^{(i)}} \xrightarrow[\text{Projection}]{\text{Feature}} \tilde{X}^{(i)} \in \mathbb{R}^{N^{(i)} \times d'} .$$

Coordinate-wise Normalization

$$\bar{X}^{(i)} = \frac{\tilde{X}^{(i)} - \boldsymbol{\mu}^{(i)}}{\boldsymbol{\sigma}^{(i)}},$$



UNPrompt

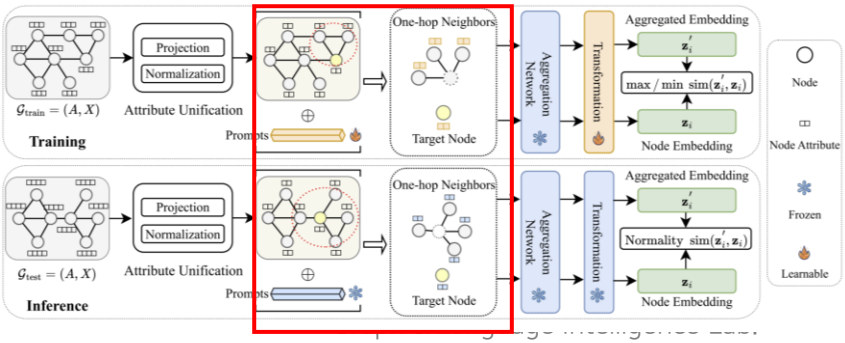
3.4 Neighborhood Prompt Learning Latent Node Attribute Predictability as Anomaly Score.

- we reveal that the predictability of latent node attributes can serve as a generalized anomaly measure.
  - ✓ Normal nodes tend to have more connection with normal nodes
    - Due to homophily relations
    - **So target node embedding is predictable**

$$\hat{Z} = g(\mathcal{G}) = \hat{A}\bar{X}W$$

- $\hat{Z}$: aggregated repression of neighbors
- $\hat{A} = (D)^{-1}A$: normalized adjacency matrix
  - A not require self-looped and symmetric
- W : learnable parameter

- $Z = \bar{X}W$
  - A is discarded



UNPrompt

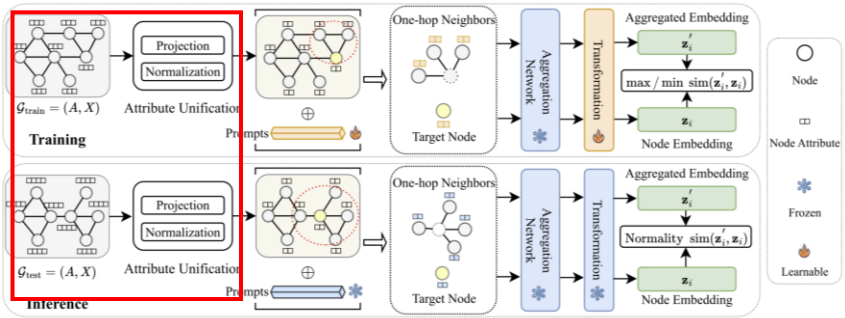
3.4 Neighborhood Prompt Learning Latent Node Attribute Predictability as Anomaly Score.

- we reveal that the predictability of latent node attributes can serve as a generalized anomaly measure.
  - ✓ Normal nodes tend to have more connection with normal nodes
    - Due to homophily relations
    - So target node embedding is predictable

$$\hat{Z} = g(\mathcal{G}) = \hat{A}\bar{X}W$$

- $\hat{Z}$: aggregated repression of neighbors
- $\hat{A} = (D)^{-1}A$: normalized adjacency matrix
  - A not require self-looped and symmetric
- W : learnable parameter

- $Z = \bar{X}W$
  - A is discarded



UNPrompt

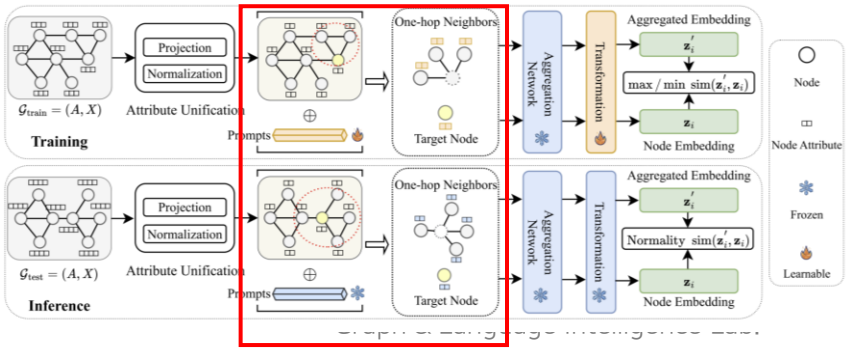
3.4 Neighborhood Prompt Learning Latent Node Attribute Predictability as Anomaly Score.

$$s_i = sim(z_i, \widetilde{z}_i) = \frac{z_i(\widetilde{z}_i)^T}{||z_i|| ||\widetilde{z}_i||}$$

- ✓ Higher similarity
  - Can be well predicted by its neighbor

GNN Pre-training

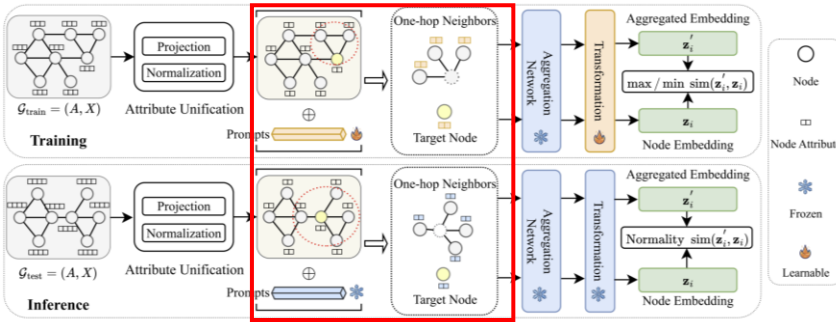
- $\min_{h} \sum \ell(z_i, \widetilde{z}_i)$
  - $\ell(z_i, \widetilde{z}_i) = -sim(z_i, \widetilde{z}_i)\ if\ y_i = 0$
  - $\ell(z_i, \widetilde{z}_i) = sim(z_i, \widetilde{z}_i)\ if\ y_i = 1$



UNPrompt

3.4 Neighborhood Prompt Learning Latent Node Attribute Predictability as Anomaly Score.

- Neighborhood Prompting via Latent Attribute Prediction.
  - ✓ We aim to learn more generalized normal and abnormal pattern via prompt tuing
    - Neighborhood prompt : Learnable prompts appended to neighbors' attributes
      - Help maximize the neighbor-based latent prediction
      - Shared and learnable token
      - $P = [p1, \dots p_k]^T$
    - Unified space are augmented by the weighted combination of these token
      - $\hat{x_i} = \bar{x_i} + \sum_j^K a_j p_j$
        - » $\alpha_j = \frac{e_j^{(w_j)^T x_i^t}}{\sum_l^K e_j^{(w_j)^T x_i^t}}$
          - α_j : importance of token
          - w_j : learnable projection
    - $\tilde{\mathcal{G}} = (A, \bar{X} + P)$

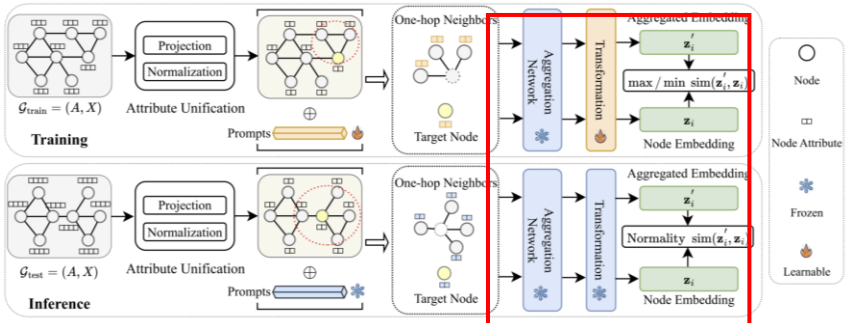


UNPrompt

3.5 Training and Inference of UNPrompt

- Transformation
  - ✓ To further enhance the representation discrimination
    - $\tilde{Z} = h(\tilde{Z})$, $\mathrm{Z} = h(Z)$

- Training
  - ✓ G is optimized via graph contrastive learning
  - ✓ Neighbor prompt and transformation layer are optimized to
- Inference
  - ✓ $s_i = sim(z_i, \tilde{z_i}) = \frac{z_i(\tilde{z_i})^T}{||z_i|| ||\tilde{z_i}||}$



UNPrompt

5 Experiments

Metric	Method	Dataset: Amazon	Reddit	Weibo	YelpChi	Aamzon-all	YelpChi-all	Disney	Avg.
AUROC	Unsupervised Methods								
	AnomalyDAE (ICASSP'20)	$0.5818_{\pm 0.039}$	$0.5016_{\pm 0.032}$	$\underline{0.7785}_{\pm 0.058}$	$0.4837_{\pm 0.094}$	$0.7228_{\pm 0.023}$	$0.5002_{\pm 0.018}$	$0.4853_{\pm 0.003}$	$\underline{0.5791}$
	CoLA (TNNLS'21)	$0.4580_{\pm 0.054}$	$0.4623_{\pm 0.005}$	$0.3924_{\pm 0.041}$	$0.4907_{\pm 0.017}$	$0.4091_{\pm 0.052}$	$0.4879_{\pm 0.010}$	$0.4696_{\pm 0.065}$	0.4529
	HCM-A (ECML-PKDD'22)	$0.4784_{\pm 0.005}$	$0.5387_{\pm 0.041}$	$0.5782_{\pm 0.048}$	$0.5000_{\pm 0.000}$	$0.5056_{\pm 0.059}$	$0.5023_{\pm 0.005}$	$0.2014_{\pm 0.015}$	0.4721
	TAM (NeurIPS'23)	$0.4720_{\pm 0.005}$	$\mathbf{0.5725}_{\pm 0.004}$	$0.4867_{\pm 0.028}$	$0.5035_{\pm 0.014}$	$0.7543_{\pm 0.002}$	$0.4216_{\pm 0.002}$	$0.4773_{\pm 0.003}$	0.5268
	GADAM (ICLR'24)	$\underline{0.6646}_{\pm 0.063}$	$0.4532_{\pm 0.024}$	$0.3652_{\pm 0.052}$	$0.3376_{\pm 0.012}$	$0.5959_{\pm 0.080}$	$0.4829_{\pm 0.016}$	$0.4288_{\pm 0.023}$	0.4755
	Supervised Methods								
	GCN (ICLR'17)	$0.5988_{\pm 0.016}$	$\underline{0.5645}_{\pm 0.000}$	$0.2232_{\pm 0.074}$	$0.5366_{\pm 0.019}$	$0.7195_{\pm 0.002}$	$\underline{0.5486}_{\pm 0.001}$	$0.5000_{\pm 0.000}$	0.5273
	GAT (ICLR'18)	$0.4981_{\pm 0.008}$	$0.5000_{\pm 0.025}$	$0.4521_{\pm 0.101}$	$\underline{0.5871}_{\pm 0.016}$	$0.5005_{\pm 0.012}$	$0.4802_{\pm 0.004}$	$0.5175_{\pm 0.054}$	0.5051
	BWGNN (ICML'22)	$0.4769_{\pm 0.020}$	$0.5208_{\pm 0.016}$	$0.4815_{\pm 0.108}$	$0.5538_{\pm 0.027}$	$0.3648_{\pm 0.050}$	$0.5282_{\pm 0.015}$	$0.6073_{\pm 0.026}$	0.5048
	GHRN (WebConf'23)	$0.4560_{\pm 0.033}$	$0.5253_{\pm 0.006}$	$0.5318_{\pm 0.038}$	$0.5524_{\pm 0.020}$	$0.3382_{\pm 0.085}$	$0.5125_{\pm 0.016}$	$0.5336_{\pm 0.030}$	0.4928
	XGBGraph (NeurIPS'23)	$0.4179_{\pm 0.000}$	$0.4601_{\pm 0.000}$	$0.5373_{\pm 0.000}$	$0.5722_{\pm 0.000}$	$\underline{0.7950}_{\pm 0.000}$	$0.4945_{\pm 0.000}$	$\mathbf{0.6692}_{\pm 0.000}$	0.5637
	GraphPrompt (WebConf'23)	$0.4904_{\pm 0.001}$	$0.4677_{\pm 0.001}$	$0.6135_{\pm 0.008}$	$0.3935_{\pm 0.002}$	$0.3215_{\pm 0.001}$	$0.4976_{\pm 0.000}$	$0.5192_{\pm 0.002}$	0.4719
	UNPrompt (Ours)	$\mathbf{0.7525}_{\pm 0.016}$	$0.5337_{\pm 0.002}$	$\mathbf{0.8860}_{\pm 0.007}$	$\mathbf{0.5875}_{\pm 0.016}$	$\mathbf{0.7962}_{\pm 0.022}$	$\mathbf{0.5558}_{\pm 0.012}$	$\underline{0.6412}_{\pm 0.030}$	**0.6790**
AUPRC	Unsupervised Methods								
	AnomalyDAE (ICASSP'20)	$0.0833_{\pm 0.015}$	$0.0327_{\pm 0.004}$	$\underline{0.6064}_{\pm 0.031}$	$0.0624_{\pm 0.017}$	$0.1921_{\pm 0.026}$	$0.1484_{\pm 0.009}$	$0.0566_{\pm 0.000}$	$\underline{0.1688}$
	CoLA (TNNLS'21)	$0.0669_{\pm 0.002}$	$0.0391_{\pm 0.004}$	$0.1189_{\pm 0.014}$	$0.0511_{\pm 0.000}$	$0.0861_{\pm 0.019}$	$0.1466_{\pm 0.003}$	$0.0701_{\pm 0.023}$	0.0827
	HCM-A (ECML-PKDD'22)	$0.0669_{\pm 0.002}$	$0.0391_{\pm 0.004}$	$0.1189_{\pm 0.014}$	$0.0511_{\pm 0.000}$	$0.0861_{\pm 0.019}$	$0.1466_{\pm 0.003}$	$0.0355_{\pm 0.001}$	0.0777
	TAM (NeurIPS'23)	$0.0666_{\pm 0.001}$	$\underline{0.0413}_{\pm 0.001}$	$0.1240_{\pm 0.014}$	$0.0524_{\pm 0.002}$	$0.1736_{\pm 0.004}$	$0.1240_{\pm 0.001}$	$0.0628_{\pm 0.001}$	0.0921
	GADAM (ICLR'24)	$0.1562_{\pm 0.103}$	$0.0293_{\pm 0.001}$	$0.0830_{\pm 0.005}$	$0.0352_{\pm 0.001}$	$0.1595_{\pm 0.121}$	$0.1371_{\pm 0.006}$	$0.0651_{\pm 0.012}$	0.0951
	Supervised Methods								
	GCN (ICLR'17)	$\underline{0.0891}_{\pm 0.007}$	$\mathbf{0.0439}_{\pm 0.000}$	$0.1109_{\pm 0.020}$	$0.0648_{\pm 0.009}$	$0.1536_{\pm 0.002}$	$\underline{0.1735}_{\pm 0.000}$	$0.0484_{\pm 0.000}$	0.0977
	GAT (ICLR'18)	$0.0688_{\pm 0.002}$	$0.0329_{\pm 0.002}$	$0.1009_{\pm 0.017}$	$\mathbf{0.0810}_{\pm 0.005}$	$0.0696_{\pm 0.001}$	$0.1400_{\pm 0.002}$	$0.0530_{\pm 0.004}$	0.0780
	BWGNN (ICML'22)	$0.0652_{\pm 0.002}$	$0.0389_{\pm 0.003}$	$0.2241_{\pm 0.046}$	$0.0708_{\pm 0.018}$	$0.0586_{\pm 0.003}$	$0.1605_{\pm 0.005}$	$0.0624_{\pm 0.003}$	0.0972
	GHRN (WebConf'23)	$0.0633_{\pm 0.003}$	$0.0407_{\pm 0.002}$	$0.1965_{\pm 0.059}$	$0.0661_{\pm 0.010}$	$0.0569_{\pm 0.006}$	$0.1505_{\pm 0.005}$	$0.0519_{\pm 0.003}$	0.0894
	XGBGraph (NeurIPS'23)	$0.0536_{\pm 0.000}$	$0.0330_{\pm 0.000}$	$0.2256_{\pm 0.000}$	$0.0655_{\pm 0.000}$	$\underline{0.2307}_{\pm 0.000}$	$0.1449_{\pm 0.000}$	$\underline{0.1215}_{\pm 0.000}$	0.1250
	GraphPrompt (WebConf'23)	$0.0661_{\pm 0.000}$	$0.0334_{\pm 0.000}$	$0.2014_{\pm 0.004}$	$0.0382_{\pm 0.000}$	$0.0666_{\pm 0.000}$	$0.1542_{\pm 0.000}$	$0.0617_{\pm 0.001}$	0.0888
	UNPrompt (Ours)	$\mathbf{0.1602}_{\pm 0.013}$	$0.0351_{\pm 0.000}$	$\mathbf{0.6406}_{\pm 0.026}$	$\underline{0.0712}_{\pm 0.008}$	$\mathbf{0.2430}_{\pm 0.028}$	$\mathbf{0.1810}_{\pm 0.012}$	$\mathbf{0.1236}_{\pm 0.031}$	**0.2078**

UNPrompt

- Coordinate-wise normalization-based node attribute unification
- Neighborhood prompt learning
  - ✓ Predictability of the latent attributes
  - ✓ Minimize predictability of abnormal nodes

CONTENTS

OWLEYE: ZERO-SHOT LEARNER FOR CROSS-DOMAIN GRAPH DATA ANOMALY DETECTION

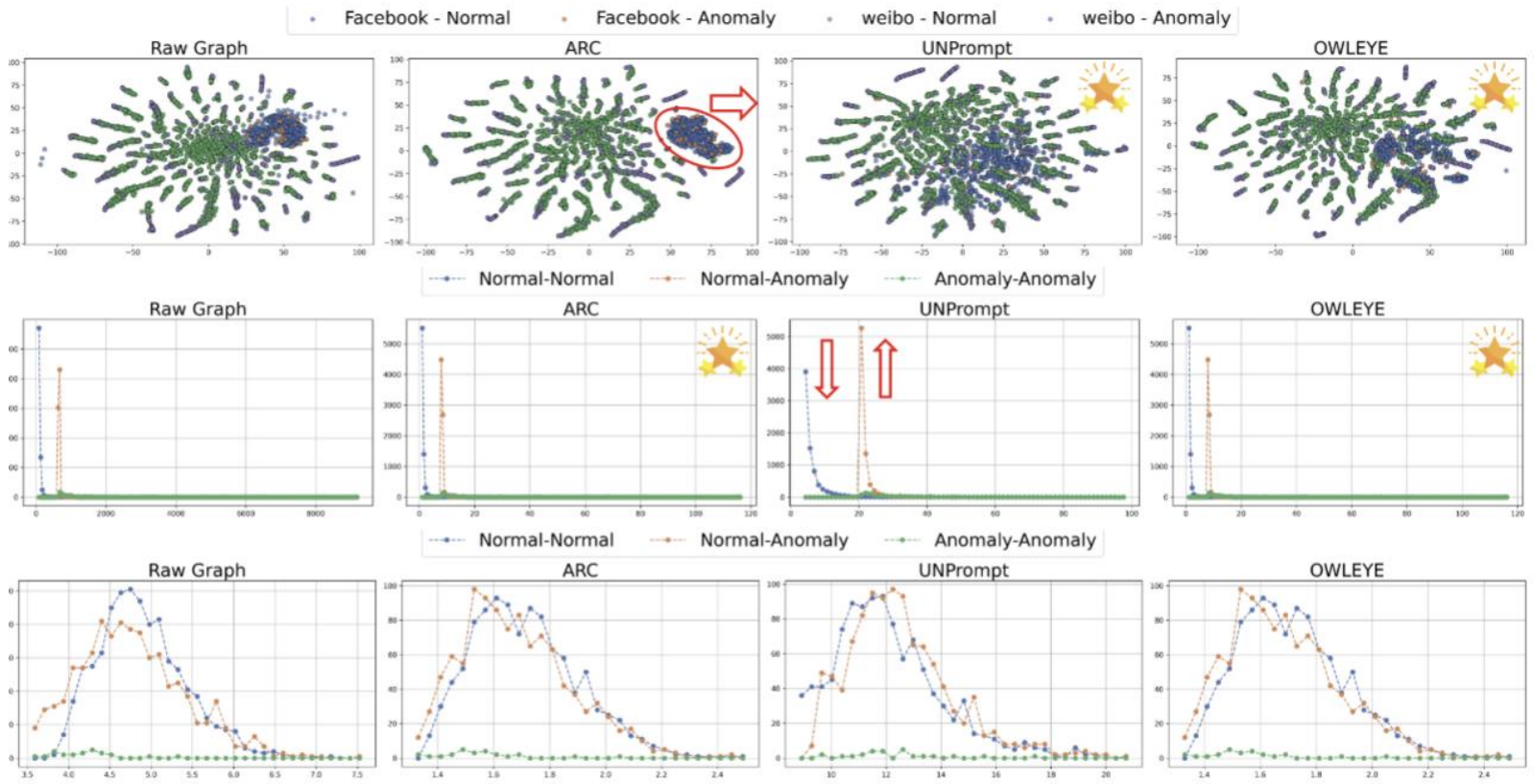
International Conference on Learning Representations 2026

Lecheng Zheng[1)], Dongqi Fu[2)], Zihao Li[3)] Jingrui He[3)]

1) Virginia Tech
2) Meta AI
3) University of Illinois Urbana-Champaign

OWLEYE

- graphs from different domains often have inherently different feature spaces and semantic interpretations.
- current approaches lack mechanisms for continual capabilities
- Not support unsupervised learning



OWLEYE

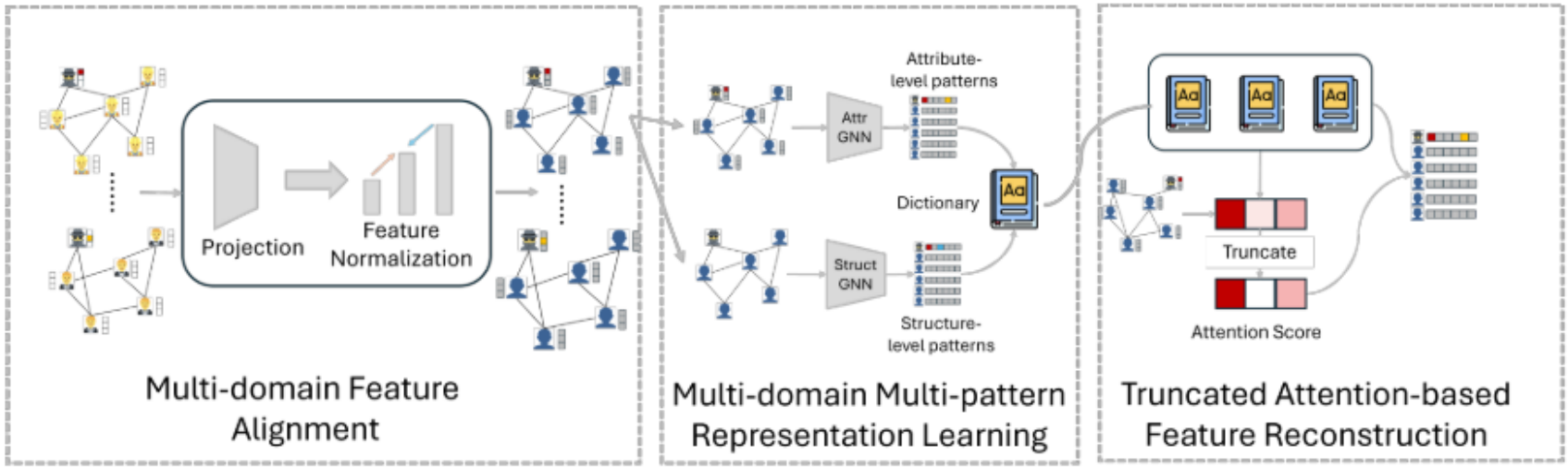


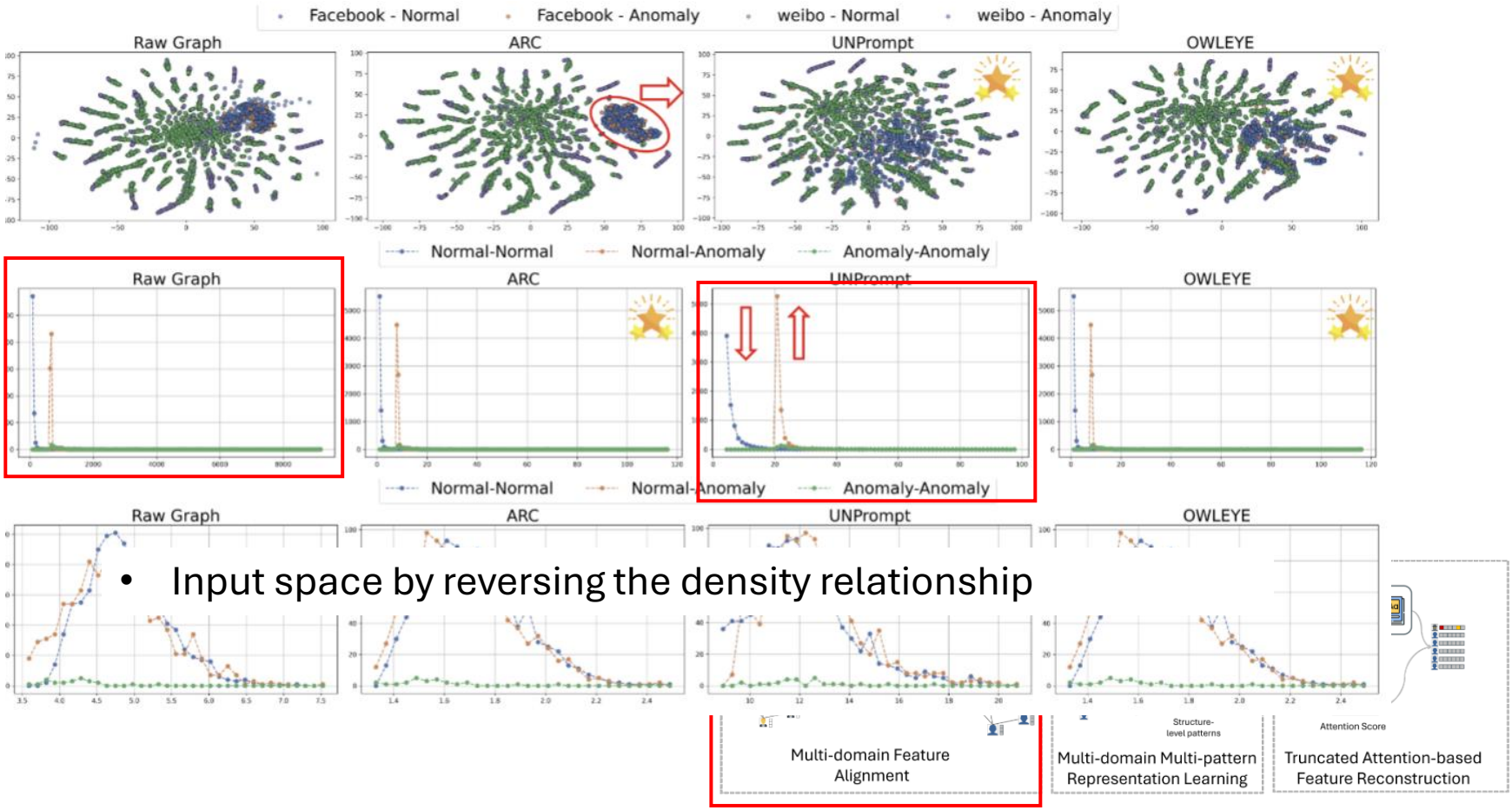
Figure 2: Overview of OWLEYE.

- Cross-domain feature alignment
  - $\mathcal{T}_{train} = \{\mathcal{D}^1_{train}, \mathcal{D}^2_{train}, \cdots, \mathcal{D}^m_{train}\}$
    - Set of training dataset with m graphs
- Multi domain pattern learning
- Truncated attention-based reconstruction

OWLEYE

2.1 PRESERVING DOMAIN-SPECIFIC SEMANTICS IN CROSS-DOMAIN FEATURE ALIGNMENT.

- Cross-domain Feature Normalization
  - ✓ Semantic meaning across the dataset remain distinct



- Input space by reversing the density relationship

Figure 2: Overview of OWLEYE.

2.1 PRESERVING DOMAIN-SPECIFIC SEMANTICS IN CROSS-DOMAIN FEATURE ALIGNMENT.

- Cross-domain Feature Normalization
  - ✓ Semantic meaning across the dataset remain distinct
    - $\tilde{X}^i \leftarrow \frac{\tilde{X}^i}{N^i} \cdot \max(f, \tau)$
      - $N^i = \frac{1}{n^i} \sum_{v_j \in V^i} \sqrt{\sum_{k=1}^{d} (\tilde{x}_{jk}^i)^2}$
      - $f = \sqrt{\frac{\text{dist}^{\text{med}} \, \text{dist}_N^i}{\text{dist}^i \, \text{dist}_N^{\text{med}}}}$
        - » $\text{dist}^{\text{med}} = \text{median}(\text{dist}^1, \ldots, \text{dist}^m)$
        - » $\text{dist}^i = \frac{1}{(n^i)^2} \sum_{v_j, v_k \in V^i} \left\| \tilde{\mathbf{x}}_{v_j}^i - \tilde{\mathbf{x}}_{v_k}^i \right\|_2$
          - Pair wise node distance for Normal/Abnormal and Normal Abnormal pair

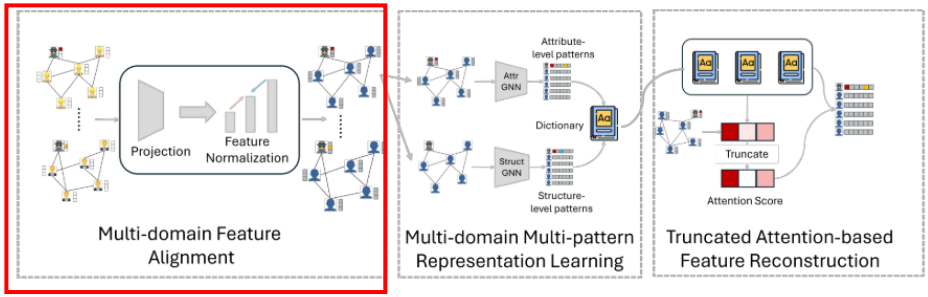


Figure 2: Overview of OWLEYE.

2.2 MULTI-DOMAIN MULTI-PATTERN DICTIONARY LEARNING

- Existing methods not support integration of knowledge extracted from new graphs. As well as the **incremental updating of normal and abnormal patterns**
- **Dynamic dictionary**
  - ✓ **Continual learning by accumulating and updating knowledge**
  - ✓ **Attribute-level, structure-level**
    - Attribute-level Representation Learning.
      - $\mathbf{H}_{\text{attr}}^{i,l+1} = \sigma\left(\mathbf{A}^i \mathbf{H}_{\text{attr}}^{i,l} \mathbf{W}_{\text{attr}}^{l}\right), \qquad \mathbf{H}_{\text{attr}}^{i,0} = \tilde{\mathbf{X}}^i$
      - $\mathbf{H}^i = \left[\mathbf{H}_{\text{attr}}^{i,2} - \mathbf{H}_{\text{attr}}^{i,1}, \ldots, \mathbf{H}_{\text{attr}}^{i,L} - \mathbf{H}_{\text{attr}}^{i,1}\right]$
    - Structure-level Representation Learning.
      - $\mathbf{R}_{\text{struc}}^{i,l+1} = \sigma\left(\mathbf{A}^i \mathbf{R}_{\text{struc}}^{i,l} \mathbf{W}_{\text{struc}}^{l}\right), \qquad \mathbf{R}_{\text{struc}}^{i,0} = \mathbf{1}$
      - $\mathbf{R}^i = \left[\mathbf{R}_{\text{struc}}^{i,2} - \mathbf{R}_{\text{struc}}^{i,1}, \ldots, \mathbf{R}_{\text{struc}}^{i,L} - \mathbf{R}_{\text{struc}}^{i,1}\right]$
    - Cross-domain Pattern Extraction.
      - $\text{Dict}_H^j = \mathbf{H}^j\left[\text{idx}^j\right], \qquad \text{Dict}_R^j = \mathbf{R}^j\left[\text{idx}^j\right]$
      - $\text{sim}\left(\mathcal{G}^i, \text{Dict}_R^j\right) = \max\left[\text{softmax}\left(\mathbf{R}^i \mathbf{W}_1 \left(\mathbf{R}^j[\text{idx}]\right)^\top\right)\right]$

2.2 MULTI-DOMAIN MULTI-PATTERN DICTIONARY LEARNING

- Existing methods not support integration of knowledge extracted from new graphs. As well as the **incremental updating of normal and abnormal patterns**
- **Dynamic dictionary**
  - ✓ **Continual learning by accumulating and updating knowledge**
  - ✓ **Attribute-level, structure-level**

    - Attribute-level Representation Learning.
      - $\mathbf{H}_{\text{attr}}^{i,l+1} = \sigma\left(\mathbf{A}^i \mathbf{H}_{\text{attr}}^{i,l} \mathbf{W}_{\text{attr}}^{l}\right), \qquad \mathbf{H}_{\text{attr}}^{i,0} = \tilde{\mathbf{X}}^i$
      - $\mathbf{H}^i = \left[\mathbf{H}_{\text{attr}}^{i,2} - \mathbf{H}_{\text{attr}}^{i,1}, \ldots, \mathbf{H}_{\text{attr}}^{i,L} - \mathbf{H}_{\text{attr}}^{i,1}\right]$
        - » Capture high-order neighborhood information
        - » Concat the multi-hop information with residual

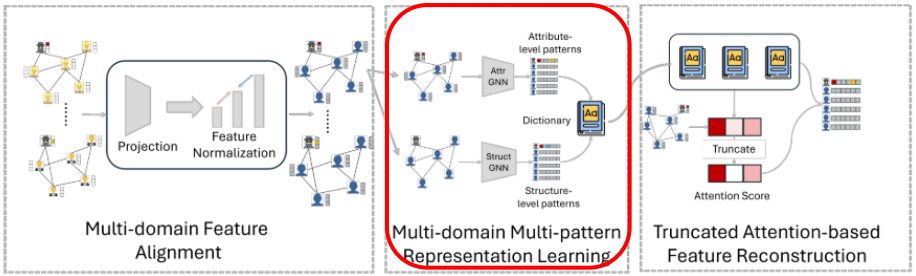


Figure 2: Overview of OWLEYE.

2.2 MULTI-DOMAIN MULTI-PATTERN DICTIONARY LEARNING

- Existing methods not support integration of knowledge extracted from new graphs. As well as the **incremental updating of normal and abnormal patterns**
- **Dynamic dictionary**
  - ✓ **Continual learning by accumulating and updating knowledge**
  - ✓ **Attribute-level, structure-level**

    - Structure-level Representation Learning.
      - While fails to capture relation and interaction.
        - » When nodes that similar attributes and connect but labels are normal and abnormal
      - $\mathbf{R}_{\text{struc}}^{i,l+1} = \sigma\left(\mathbf{A}^i \mathbf{R}_{\text{struc}}^{i,l} \mathbf{W}_{\text{struc}}^{l}\right), \qquad \mathbf{R}_{\text{struc}}^{i,0} = \mathbf{1}$
      - $\mathbf{R}^i = \left[\mathbf{R}_{\text{struc}}^{i,2} - \mathbf{R}_{\text{struc}}^{i,1}, \ldots, \mathbf{R}_{\text{struc}}^{i,L} - \mathbf{R}_{\text{struc}}^{i,1}\right]$

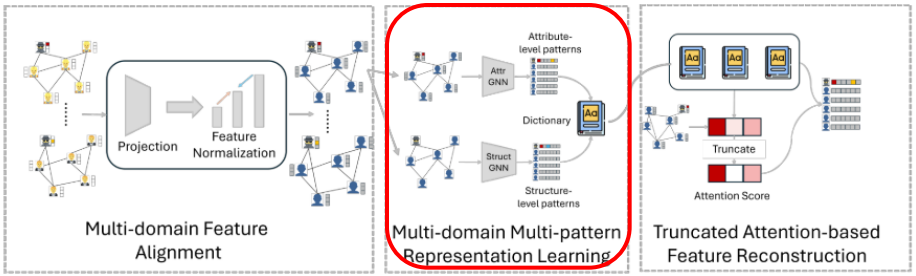


Figure 2: Overview of OWLEYE.

2.2 MULTI-DOMAIN MULTI-PATTERN DICTIONARY LEARNING

- Dynamic dictionary
  - ✓ Continual learning by accumulating and updating knowledge
  - ✓ Attribute-level, structure-level

    - Cross-domain Pattern Extraction.
      - Randomly extract n_{sup} pattern from $\mathcal{G}_j$ and store in two dictionary.
      - $\mathrm{Dict}_H^j = \mathbf{H}^j[\mathrm{idx}^j], \quad \mathrm{Dict}_R^j = \mathbf{R}^j[\mathrm{idx}^j]$
        - » Idx : random node index

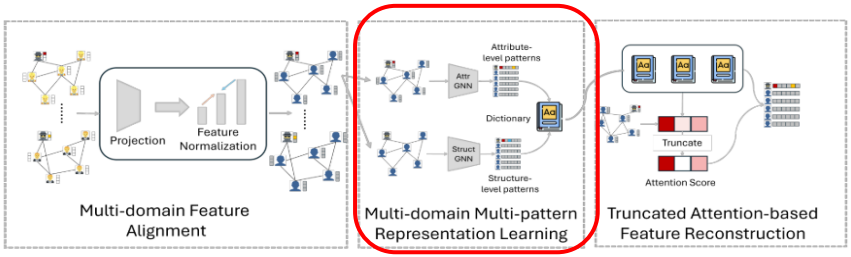


Figure 2: Overview of OwlEye.

2.2 MULTI-DOMAIN MULTI-PATTERN DICTIONARY LEARNING

- Dynamic dictionary
  - ✓ Continual learning by accumulating and updating knowledge
  - ✓ Attribute-level, structure-level
    - $\mathrm{sim}(\mathcal{G}^i, \mathrm{Dict}_R^j) = \max\left[\mathrm{softmax}\left(\mathbf{R}^i\mathbf{W}_1\left(\mathbf{R}^j[\mathrm{idx}]\right)^\top\right)\right] \in \mathbb{R}^{n^i}$
      - » $\mathbf{R}^i \in \mathbb{R}^{n^i \times ld}$, $\mathbf{W}_1 \in \mathbb{R}^{ld \times ld}$, $\mathbf{R}^j[\mathrm{idx}] \in \mathbb{R}^{n_{sup} \times ld}$
      - » $\mathrm{softmax}\left(\mathbf{R}^i\mathbf{W}_1\left(\mathbf{R}^j[\mathrm{idx}]\right)^\top\right) \in \mathbb{R}^{n^i \times n_{sup}}$
      - » Maximal node similarity between graph and pattern stored in Dict
        - Structural similarity $G^{i'}$s v_{th} node, G^j dictionary p_{th} pattern
        - Node similarity between graph and pattern stored in dictionary
      - » In structural level
        - ∵ Camouflaged anomaly

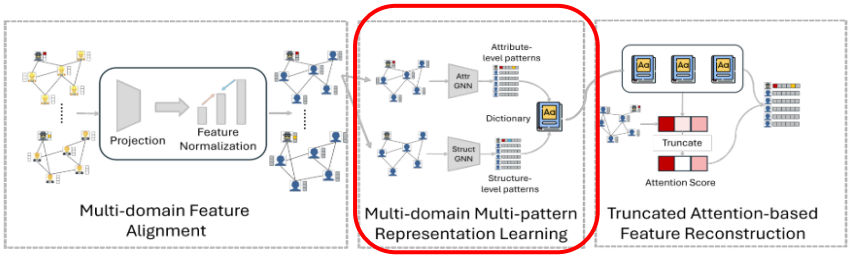


Figure 2: Overview of OwlEye.

2.3 TRUNCATED ATTENTION-BASED FEATURE RECONSTRUCTION FOR IN-CONTEXT LEARNING AND INFERENCE

- To support zero-shot with relying on any labeled data during test
  - ✓ Naïve approach is sampled pseudo support nodes
    - Majority of the nodes are normal
  - ✓ Filter out the potential abnormal data and select most representative nodes to reconstruct
    - $\boldsymbol{\alpha}(\mathcal{G}^i, \mathrm{Dict}_H^j) = \sqrt{\frac{(\mathbf{W}^Q \mathbf{H}^i)(\mathbf{W}^K(\mathbf{H}^j)^\top)}{\sqrt{Ld}}} \in \mathbb{R}^{n^i \times n_{sup}}$
      - I : query node index from graph, j: dictionary pattern index
    - $\boldsymbol{\alpha}[\mathrm{idx}_\alpha] = -\infty,\ where\ \mathrm{idx}_\alpha = \mathrm{Top}\left(-\boldsymbol{\alpha}(\mathcal{G}^i, \mathrm{Dict}_H^j), k\right)$
      - Truncated the less representative nodes
    - $\alpha_H^{ij} = \mathrm{softmax}\left(\boldsymbol{\alpha}(\mathcal{G}^i, \mathrm{Dict}_R^j)/\tau_a\right) \in \mathbb{R}^{n^i \times n_{sup}}$
      - Both structural level
    - $\hat{\mathbf{H}}^i = \frac{1}{m}\sum_{j=1}^{m} \mathrm{sim}\left(\mathcal{G}^i, \mathrm{Dict}_H^j\right) \odot \left(\alpha_H^{ij} \mathrm{Dict}_H^j\right) \in \mathbb{R}^{n^i} \odot \mathbb{R}^{n^i \times n_{sup}}\ \mathbb{R}^{n_{sup} \times ld} \in \mathbb{R}^{n^i \times ld}$
      - Graph-dictionary level
      - ⊙ Pattern level similarity
        - » Weighted retrieval
        - » Reconstruction representation that weighted pattern representation

2.3 TRUNCATED ATTENTION-BASED FEATURE RECONSTRUCTION FOR IN-CONTEXT LEARNING AND INFERENCE

Training.

- $\mathcal{L}_{\text{recon}} = \sum_{i=1}^{m} \left[\sum_{v_k \in \mathcal{A}^i} \cos\left(\mathbf{H}_{v_k}^i, \widehat{\mathbf{H}}_{v_k}^i\right) - \sum_{v_j \in \mathcal{N}^i} \cos\left(\mathbf{H}_{v_j}^i, \widehat{\mathbf{H}}_{v_j}^i\right) \right]$
  - ✓ $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\mathbf{b}^\top}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$
  - ✓ $\mathcal{A}^i$: abnormal node set , $\mathcal{N}^i$: normal node set
  - ✓ Encourage the Normal nodes are reconstruction using normal dictionary

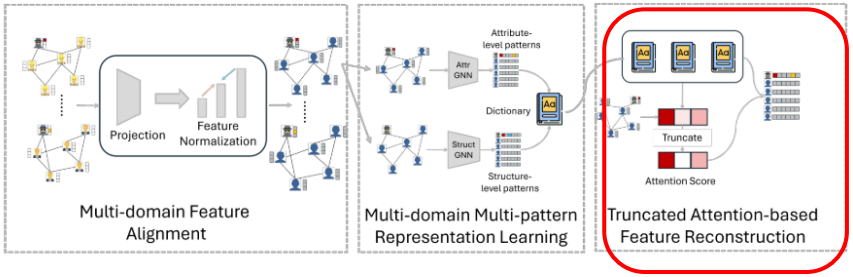


Figure 2: Overview of OWLEYE.

2.3 TRUNCATED ATTENTION-BASED FEATURE RECONSTRUCTION FOR IN-CONTEXT LEARNING AND INFERENCE

Training.

- $\mathcal{L}^{H}_{\text{triplet}} = \sum_{i=1}^{m} \sum_{\substack{v_j \in \mathcal{A}^i \\ v_k \in \mathcal{N}^i}} \max\left(\left\| \widehat{\mathbf{H}}^i_{v_j} - \mathbf{H}^i_{v_j} \right\|_2^2 - \left\| \widehat{\mathbf{H}}^i_{v_j} - \widehat{\mathbf{H}}^i_{v_k} \right\|_2^2 + \lambda, 0 \right)$
- $\mathcal{L}^{R}_{\text{triplet}} = \sum_{i=1}^{m} \sum_{\substack{v_j \in \mathcal{A}^i \\ v_k \in \mathcal{N}^i}} \max\left(\left\| \widehat{\mathbf{R}}^i_{v_j} - \mathbf{R}^i_{v_j} \right\|_2^2 - \left\| \widehat{\mathbf{R}}^i_{v_j} - \widehat{\mathbf{R}}^i_{v_k} \right\|_2^2 + \lambda, 0 \right)$
  - ✓ Minimize the distance between normal and abnormal
  - ✓ Minimize the distance between raw feature and reconstruction feature
    - While separation the reconstructed normal and abnormal

- $\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{triplet}}$

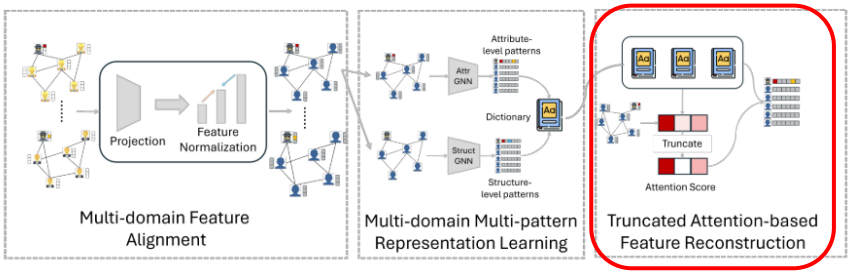


Figure 2: Overview of OWLEYE.

2.3 TRUNCATED ATTENTION-BASED FEATURE RECONSTRUCTION FOR IN-CONTEXT LEARNING AND INFERENCE

Inference.

- $S_{v_j} = \left\| \widehat{\mathbf{H}}^i_{v_j} - \mathbf{H}^i_{v_j} \right\|_2^2 + \beta \left\| \widehat{\mathbf{R}}^i_{v_j} - \mathbf{R}^i_{v_j} \right\|_2^2$

  - ✓ $\widehat{\mathbf{H}}^i = \frac{1}{m} \sum_{j=1}^{m} \mathrm{sim}\left(\mathcal{G}^i, \mathrm{Dict}_H^j\right) \odot \left(\boldsymbol{\alpha}_H^{ij} \mathrm{Dict}_H^j\right)$
  - ✓ $\widehat{\mathbf{R}}^i = \frac{1}{m} \sum_{j=1}^{m} \mathrm{sim}\left(\mathcal{G}^i, \mathrm{Dict}_R^j\right) \odot \left(\boldsymbol{\alpha}_R^{ij} \mathrm{Dict}_R^j\right)$

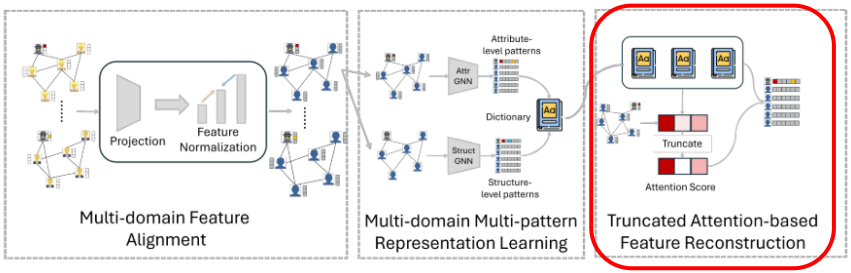


Figure 2: Overview of OWLEYE.

3 EXPERIMENTS

Table 1: Anomaly detection performance w.r.t AUPRC. We highlighted the results ranked first and second. "Average" indicates the average AUPRC over 8 datasets.

Method	Cora	Flickr	ACM	BlogCatalog	Facebook	Weibo	Reddit	Amazon	Average
Supervised (10-shot)									
BWGNN	9.57±2.40	12.39±2.68	13.37±6.03	12.97±3.15	5.81±1.17	9.55±2.12	3.21±2.32	12.40±1.86	9.80±2.91
GHRN	14.04±0.73	16.45±2.59	16.29±1.41	13.58±2.19	6.24±1.12	17.51 ±1.52	4.44±1.15	13.84±2.63	12.80±1.67
Unsupervised (zero-shot)									
DOMINANT	31.77±0.34	28.76±1.52	32.49±4.97	29.51±3.44	3.42±0.86	29.63±0.86	3.28±0.37	36.80±8.37	24.46±3.11
SLGAD	18.27±1.01	16.93±8.20	1.33±0.23	9.47±3.00	0.93±0.23	35.80±1.41	4.00±2.27	5.33±2.20	11.51±2.38
TAM	9.43±0.27	23.34±1.42	40.68±2.58	25.59±4.76	12.18±3.14	23.01±15.14	4.22±0.22	45.26±4.34	22.96±3.96
CARE	35.12±0.23	25.64±0.16	37.76±0.35	25.06±0.10	5.52±0.34	40.70±0.74	3.17±0.17	56.76±1.44	28.72±0.44
ARC	45.20±1.08	35.13±0.20	39.02±0.08	33.43±0.15	4.25±0.47	64.18±0.68	4.20±0.25	20.48±6.89	30.74±1.23
UNPrompt	9.84±2.90	25.21±1.84	11.18±1.67	18.24±13.05	4.32±0.55	20.58±5.62	3.77±0.32	9.41±2.69	12.82±3.58
OwlEye	43.94±0.46	37.69±0.25	39.75±0.13	34.99±0.31	5.62±1.17	60.90±0.21	4.25±0.11	62.20±3.18	36.17±0.73

Table 2: Anomaly detection performance in 10-shot setting w.r.t AUPRC. We highlighted the results ranked first and second. "Average" indicates the average AUPRC over 8 datasets.

Method	Cora	Flickr	ACM	BlogCatalog	Facebook	Weibo	Reddit	Amazon	Average
Supervised (10-shot)									
BWGNN	9.57±2.40	12.39±2.68	13.37±6.03	12.97±3.15	5.81±1.17	9.55±2.12	3.21±2.32	12.40±1.86	9.80±2.91
GHRN	14.04±0.73	16.45±2.59	16.29±1.41	13.58±2.19	6.24±1.12	17.51±1.52	4.44±1.15	13.84±2.63	12.80±1.67
Unsupervised & Finetune (10-shot)									
DOMINANT	22.35±0.81	30.42±1.35	24.76±0.84	34.82±0.78	4.12±0.23	78.63±1.28	4.18±0.64	8.86±0.69	26.02±0.83
SLGAD	19.38±1.46	17.46±5.62	5.33±1.39	11.67±2.22	3.81±0.23	36.23±1.41	4.32±2.13	7.69±2.82	13.24±2.16
TAM	14.27±0.65	27.68±1.45	57.32±5.26	27.49±1.09	11.73±2.34	26.78±0.28	3.67±0.16	52.62±3.17	27.70±1.80
CARE	39.52±2.90	27.19±0.24	38.12±0.43	27.75±0.66	5.86±0.69	44.37±0.96	3.62±0.15	59.51±1.28	30.74±0.91
ARC	48.02±0.83	37.15±0.24	39.13±0.15	34.20±0.27	4.92±0.75	63.83±2.55	4.32±0.16	21.90±6.32	31.68±1.41
UNPrompt	11.40±0.55	22.65±0.30	14.80±0.70	18.01±12.88	4.04±1.07	22.23±4.83	3.85±0.21	11.08±0.46	13.51±2.62
OwlEye	44.40±1.58	38.32±0.15	39.16±0.10	35.00±0.34	6.83±1.38	64.14±1.25	4.96±0.10	63.02±5.71	36.73±1.33

OWLEYE-T (with standard attention instead of truncated attention)

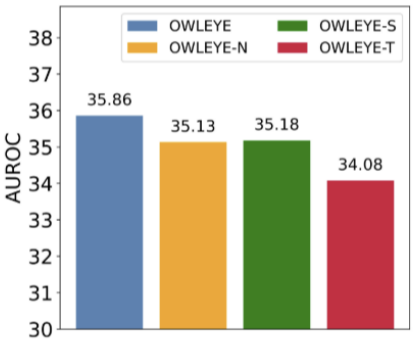
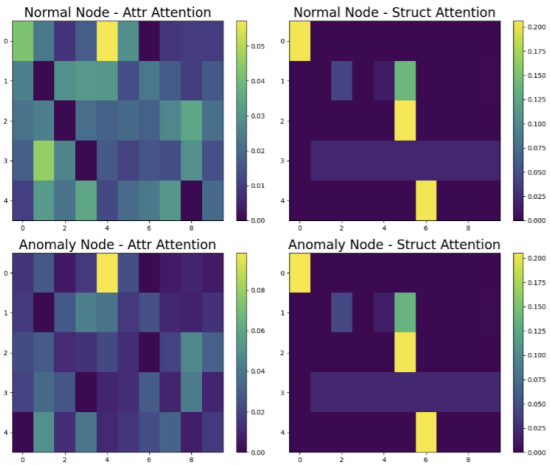
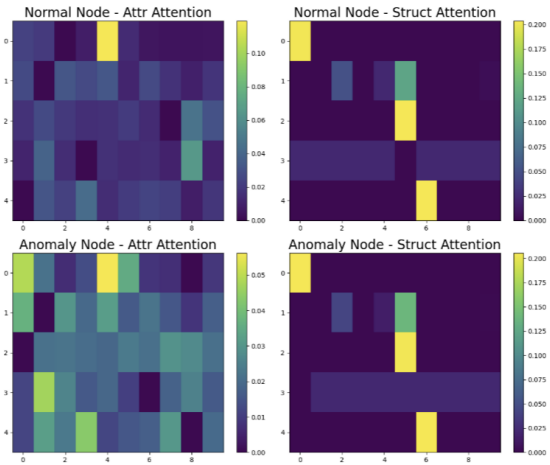


Figure 3: **Left:** Ablation Study

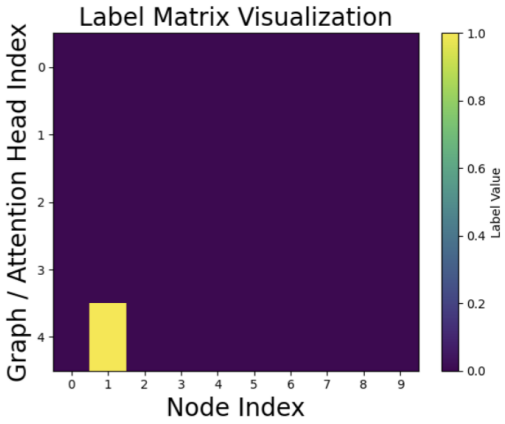
3 EXPERIMENTS



(a) Correctly Predicted Nodes



(b) Incorrectly Predicted Nodes



(c) Label Matrix

Figure 4: Visualization of cross-attention map on the Cora dataset. Subfigures (a) and (b) display the cross-attention scores across the graphs, where the y-axis corresponds to the graph indices (0–3 indicating the four training graphs and 4 representing the test graph) and the x-axis denotes the ten extracted patterns learned for each graph. The top row shows the attribute attention and structural attention for a normal node and the bottom row shows the attribute attention and structural attention for an anomalous one. Subfigure (c) in both figures presents the ground-truth label matrices that specify whether each node is normal or abnormal.

OWLEYE

- zero-shot graph anomaly detection across multiple domains
  - ✓ feature alignment
  - ✓ lack of continual evolving capabilities,
  - ✓ reliance on labeled target data

- Extensive experiments on real-world datasets
  - ✓ validate the superior performance and transferability over existing sota approaches

comparison

Method	Generalized knowledge	abnormal criteria
AnomalyGFM	class prototype	prototype similarity
UNPrompt	Prompt	Neighborhood predictability
OWLEYE	multi-domain pattern dictionary	Reconstruction fail

Thank you ☺

Q & A