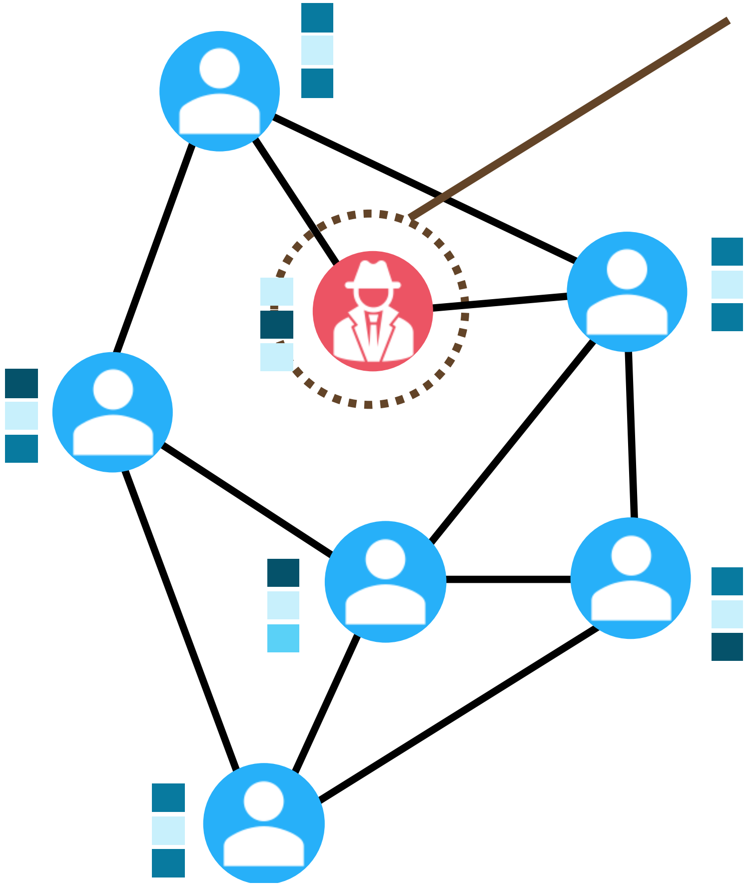
NEURAL INFORMATION PROCESSING SYSTEMS

ARC: A Generalist Graph Anomaly Detector with In-Context Learning

Presenter: Yixin Liu

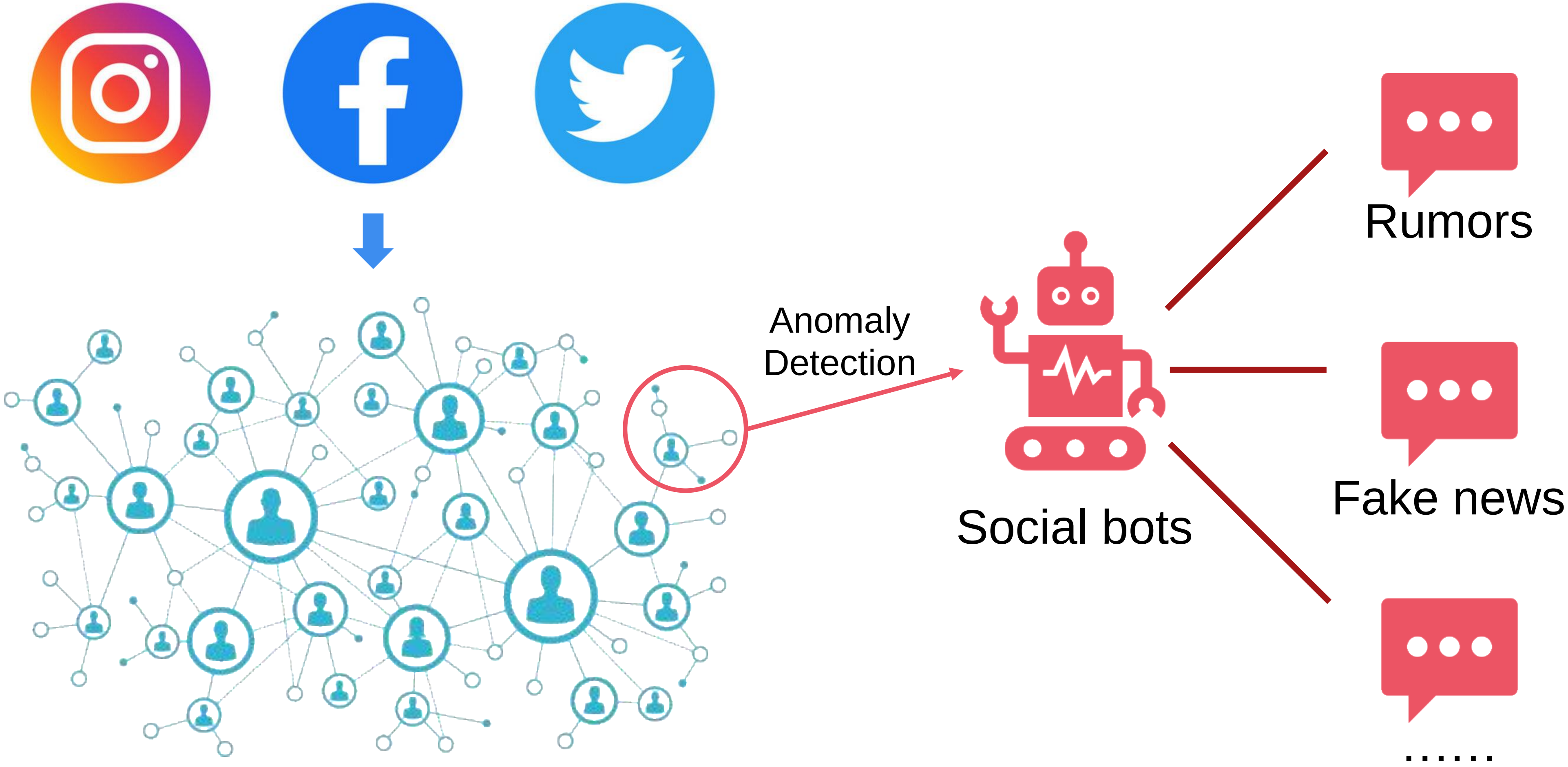
Griffith UNIVERSITY
Queensland, Australia

Graph Anomaly Detection (GAD)



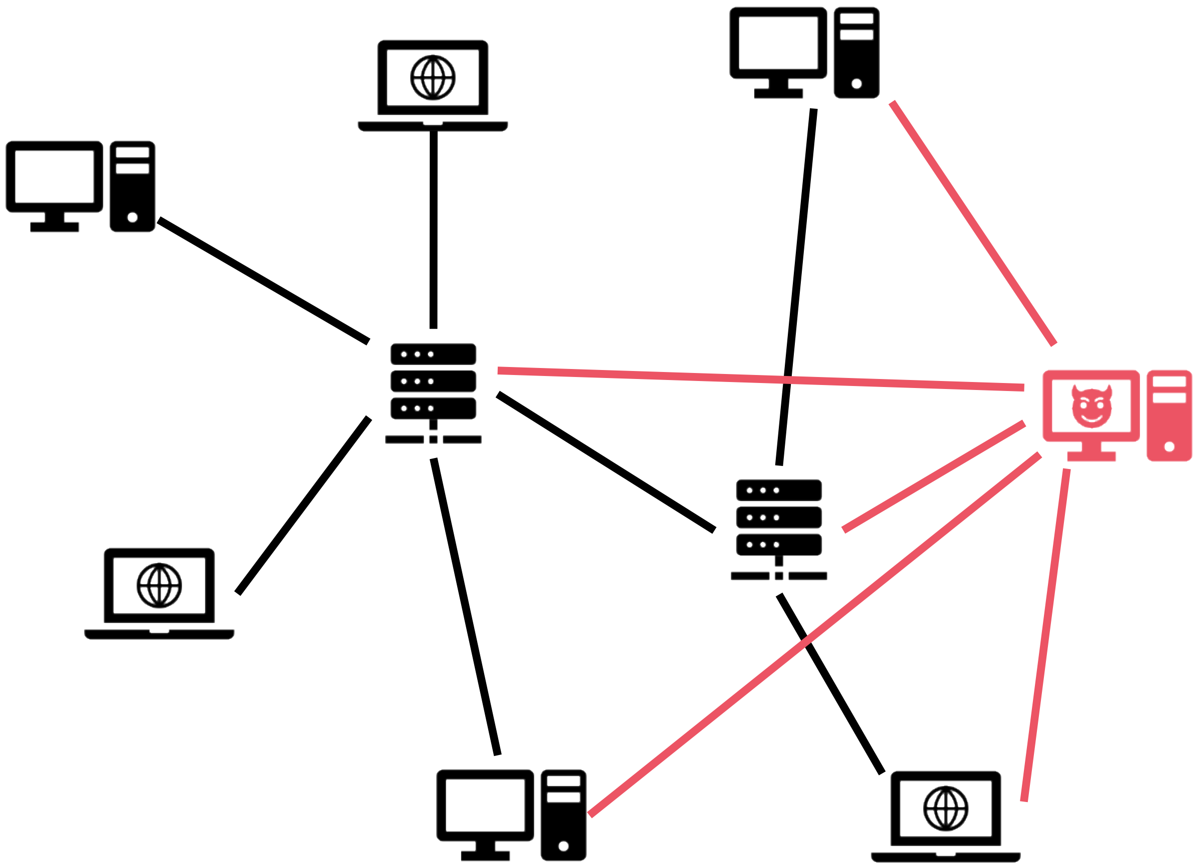
To detect the abnormal nodes that are different from the majority.

GAD’s Application: Social Networks



Social Network Graph

GAD’s Application: Cybersecurity

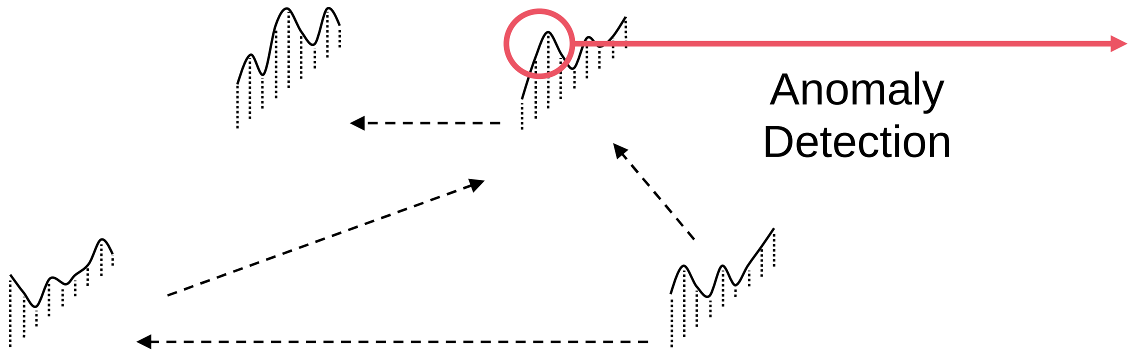


Anomaly Detection

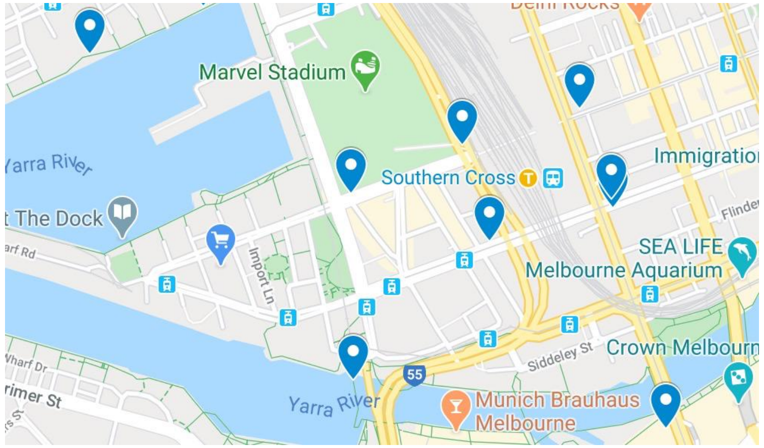
Hackers

Cyber Attacks

GAD’s Application: Traffic Networks



Spatial-temporal Graph



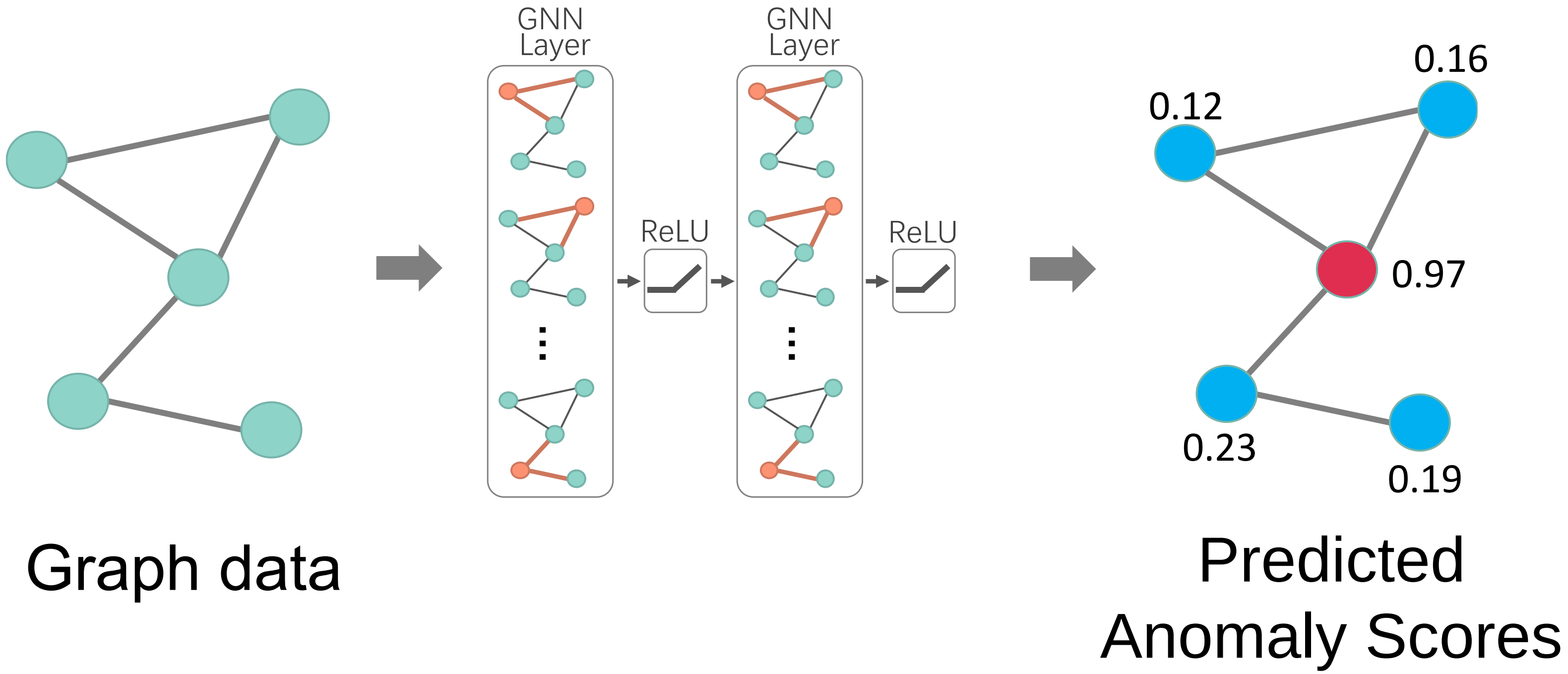
Traffic sensors displayed on GoogleMaps

Accident?

Congestion?

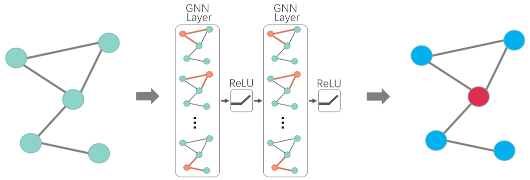
GAD: Existing Solutions

Mainstream solution: Graph neural networks (GNNs)



GAD: Existing Solutions

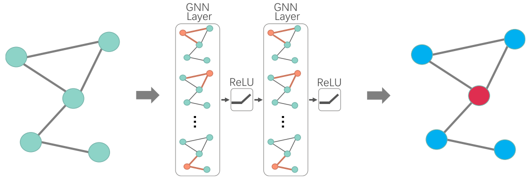
Graph neural networks (GNNs) based methods



- Supervised GAD methods
- Unsupervised GAD methods

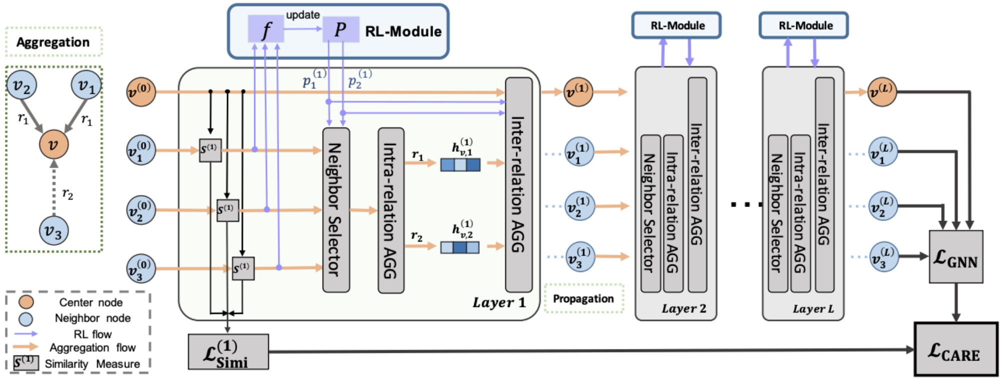
GAD: Existing Solutions

Graph neural networks (GNNs) based methods

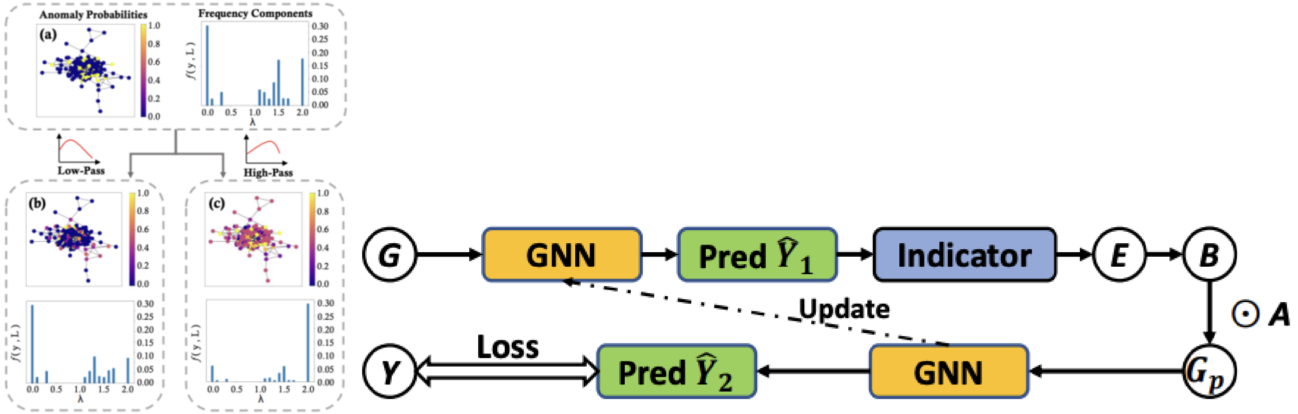


Supervised GAD methods:
training GAD model with labels (normal/anomaly)

Unsupervised GAD methods



CARE-GNN[1]

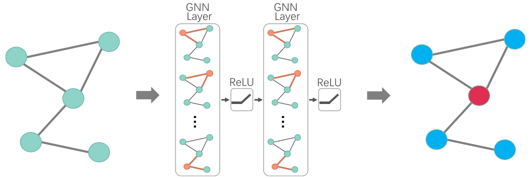


GHRN[2]

[1] Dou, Yingtong, et al. "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters." *Proceedings of the 29th ACM international conference on information & knowledge management*. 2020.
[2] Gao, Yuan, et al. "Addressing heterophily in graph anomaly detection: A perspective of graph spectrum." *Proceedings of the ACM Web Conference 2023*. 2023.

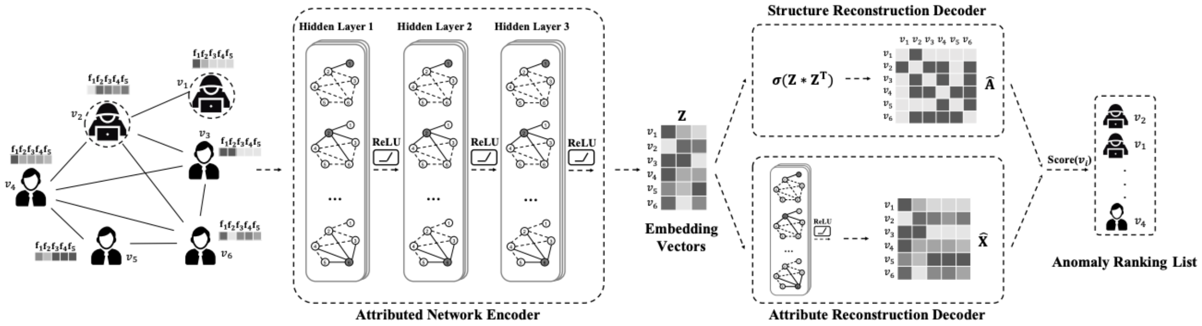
GAD: Existing Solutions

Graph neural networks (GNNs) based methods

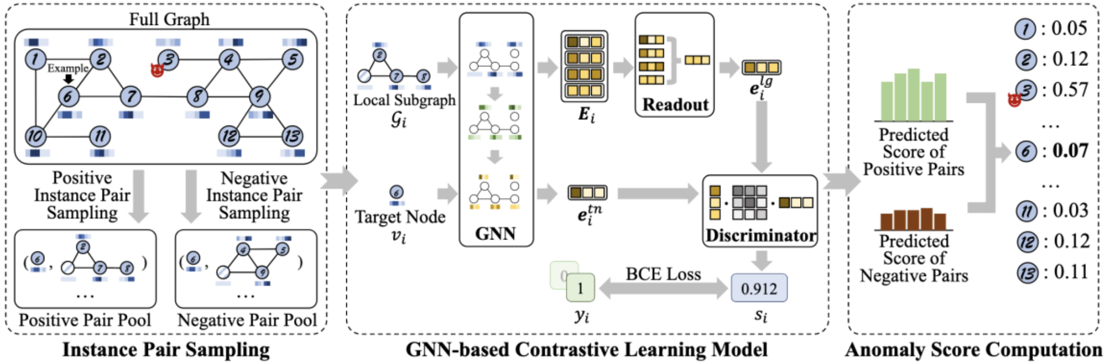


Supervised GAD methods:
training GAD model with labels (normal/anomaly)

Unsupervised GAD methods:
training GAD model without labels



DOMINANT[3]

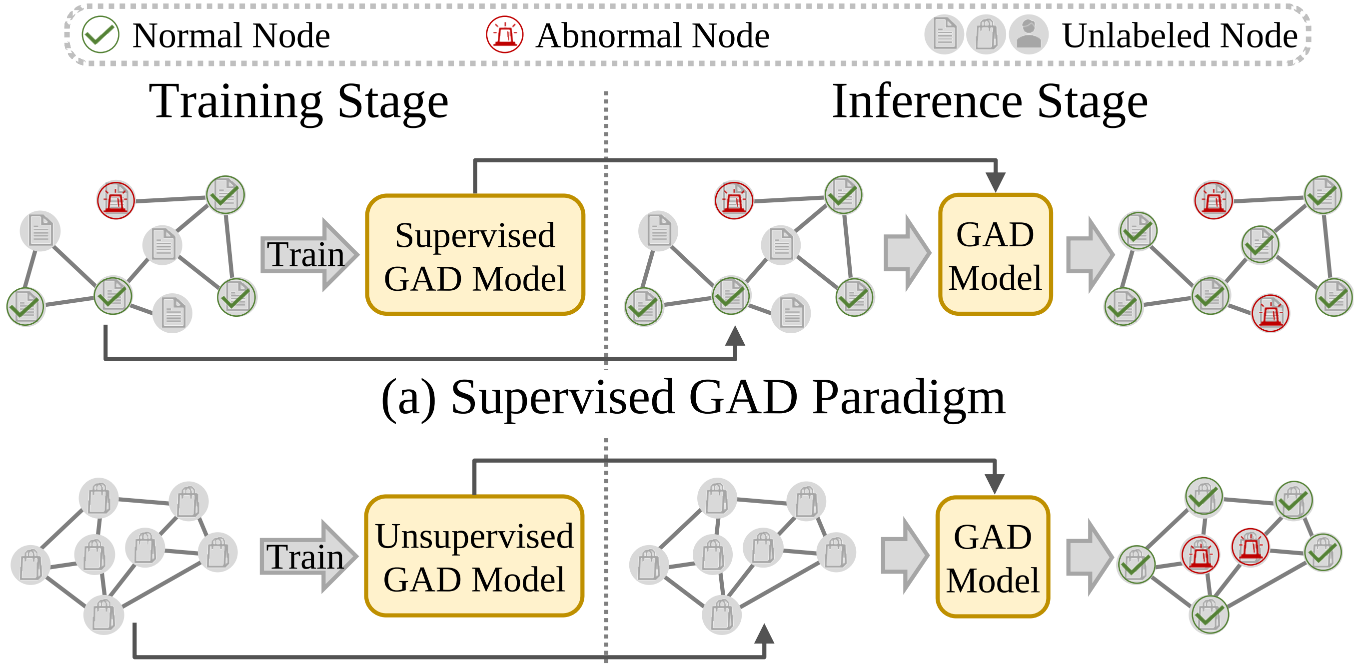


CoLA[4]

[3] Ding, Kaize, et al. "Deep anomaly detection on attributed networks." Proceedings of the 2019 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2019.
[4] Liu, Yixin, et al. "Anomaly detection on attributed networks via contrastive self-supervised learning." IEEE transactions on neural networks and learning systems 33.6 (2021): 2378-2392.

GAD: Existing Solutions

Graph neural networks (GNNs) based methods
- **Supervised GAD methods**
- **Unsupervised GAD methods**



(a) Supervised GAD Paradigm

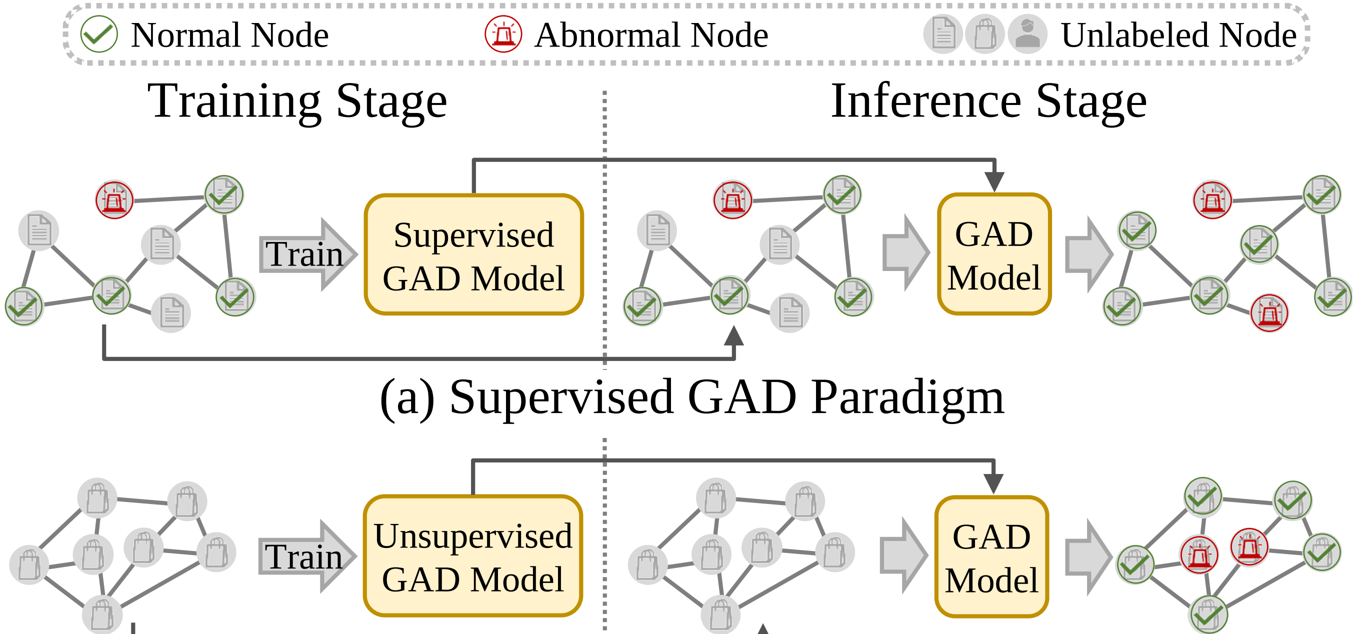
(b) Unsupervised GAD Paradigm

Learning paradigm:

one model for one dataset

GAD: Existing Solutions

Graph neural networks (GNNs) based methods
- **Supervised GAD methods**
- **Unsupervised GAD methods**

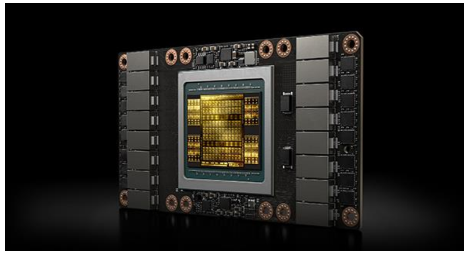


(a) Supervised GAD Paradigm

(b) Unsupervised GAD Paradigm

Learning paradigm:
one model for one dataset

✕ Expensive training cost

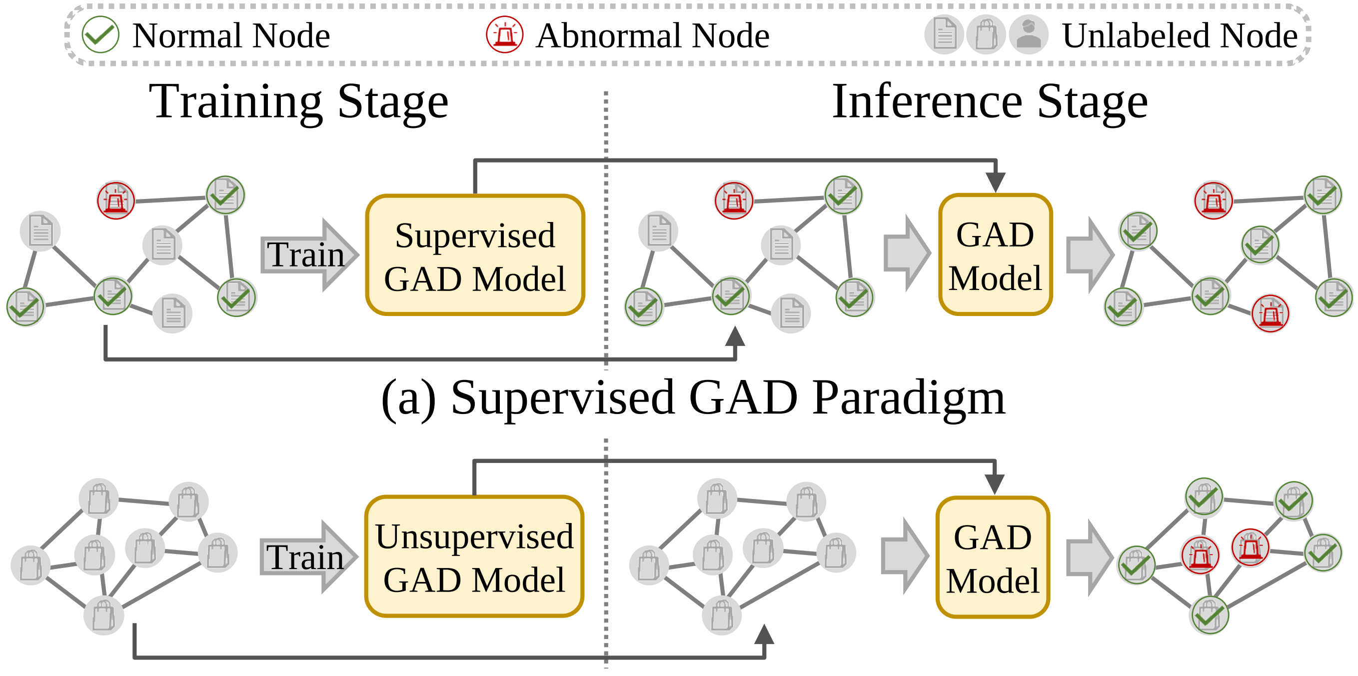


Devices

Time

GAD: Existing Solutions

Graph neural networks (GNNs) based methods
- **Supervised GAD methods**
- **Unsupervised GAD methods**



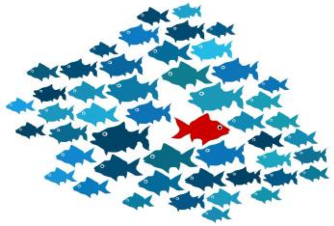
(a) Supervised GAD Paradigm

(b) Unsupervised GAD Paradigm

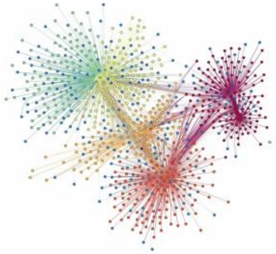
Learning paradigm:
one model for one dataset

✗ Expensive training cost

✗ Data requirements



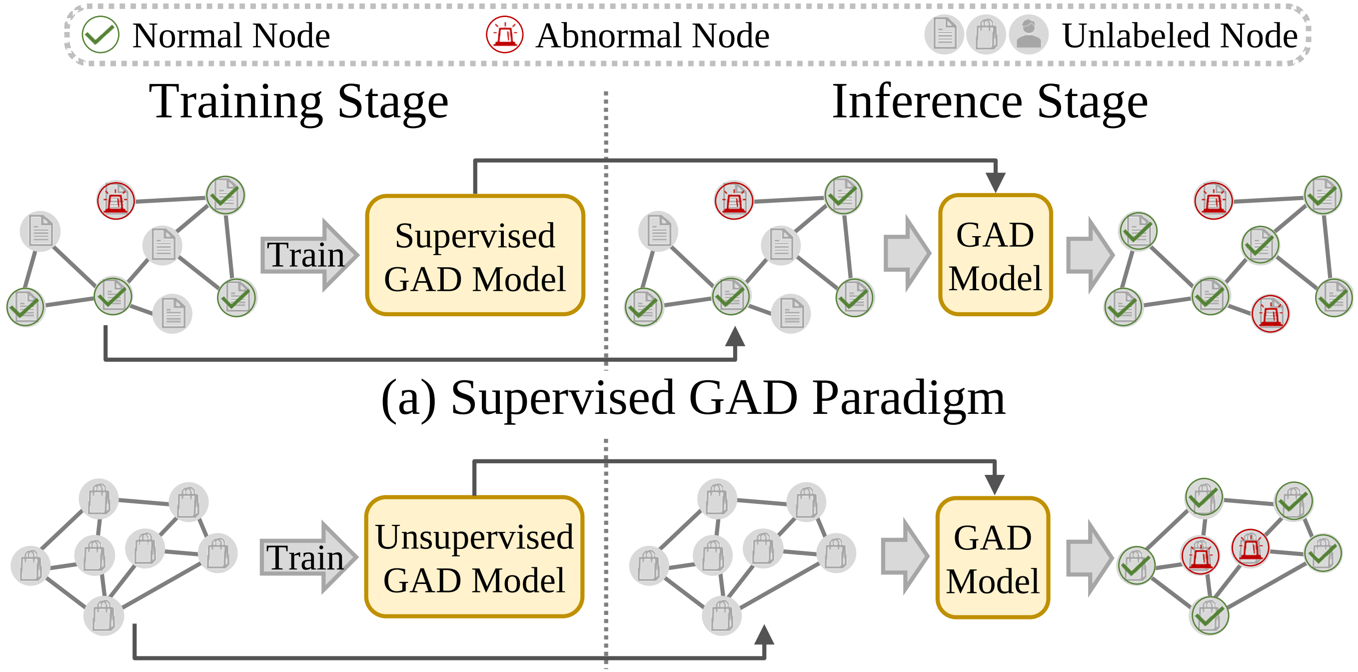
Labels



Full Data

GAD: Existing Solutions

Graph neural networks (GNNs) based methods
- **Supervised GAD methods**
- **Unsupervised GAD methods**



(a) Supervised GAD Paradigm

(b) Unsupervised GAD Paradigm

Learning paradigm:

one model for one dataset

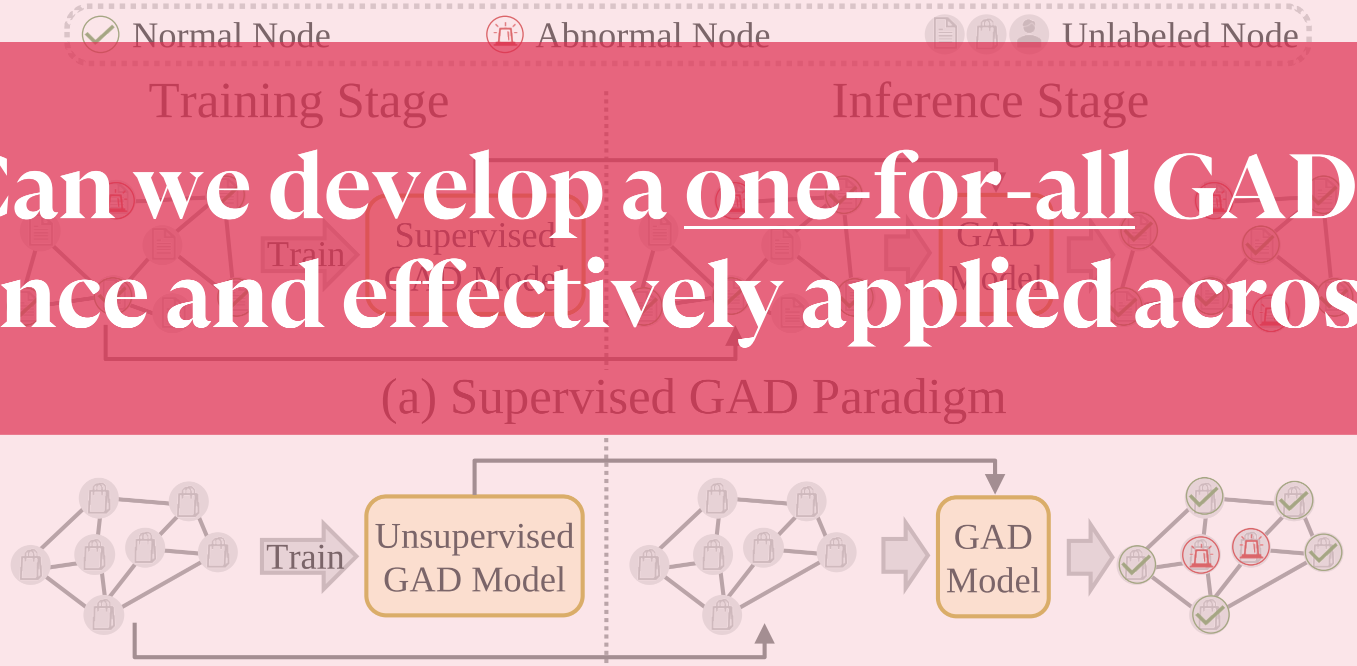
- ✗ Expensive training cost
- ✗ Data requirements
- ✗ Poor generalizability

Can't be transferred to new datasets!

GAD: Existing Solutions

Graph neural networks (GNNs) based methods

- **Supervised GAD methods**
- **Unsupervised GAD methods**



(a) Supervised GAD Paradigm

(b) Unsupervised GAD Paradigm

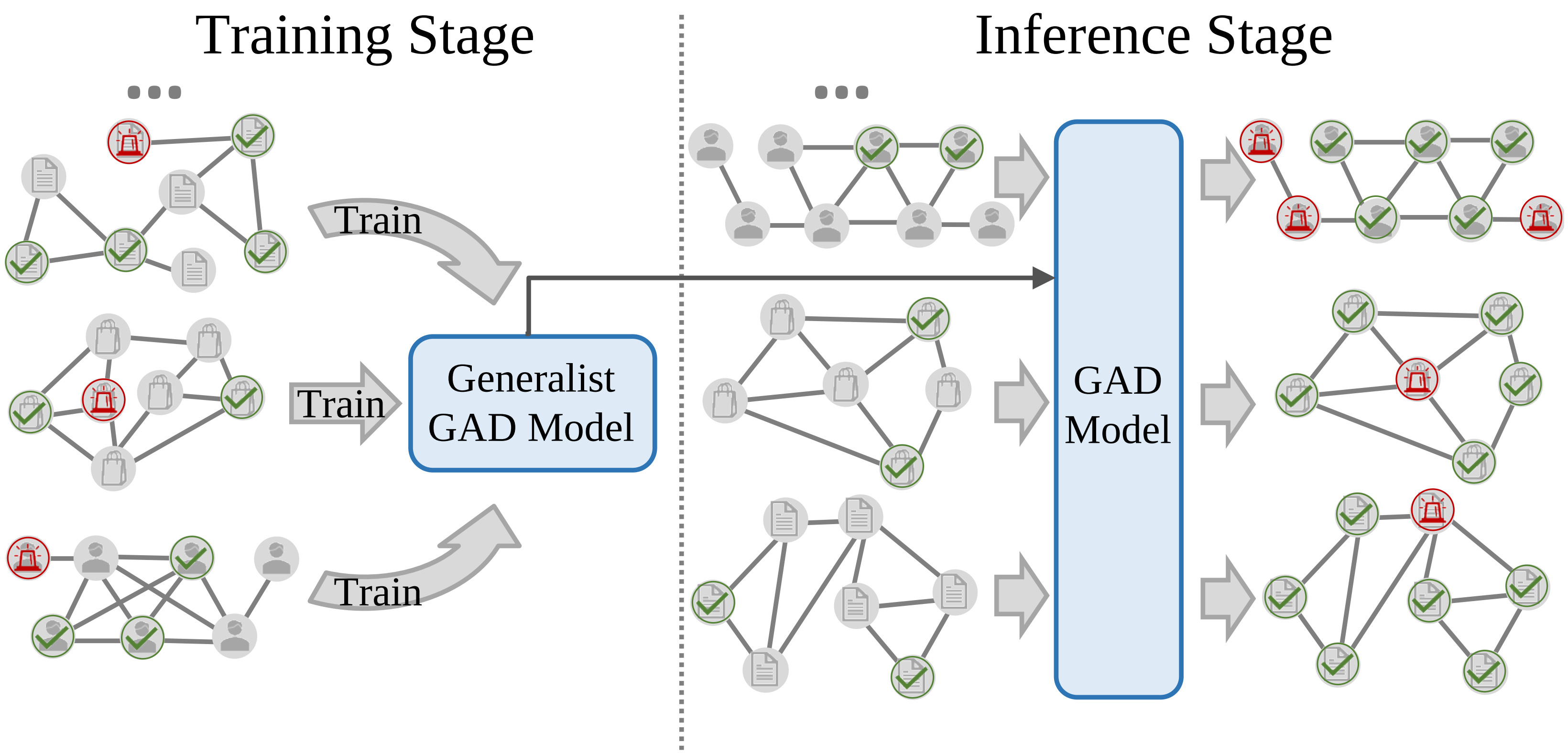
Learning paradigm:
one model for one dataset

- ✗ Expensive training cost
- ✗ Data requirements
- ✗ Poor generalizability

Can't be transferred to new datasets!

Can we develop a one-for-all GAD model that can be trained once and effectively applied across various datasets?

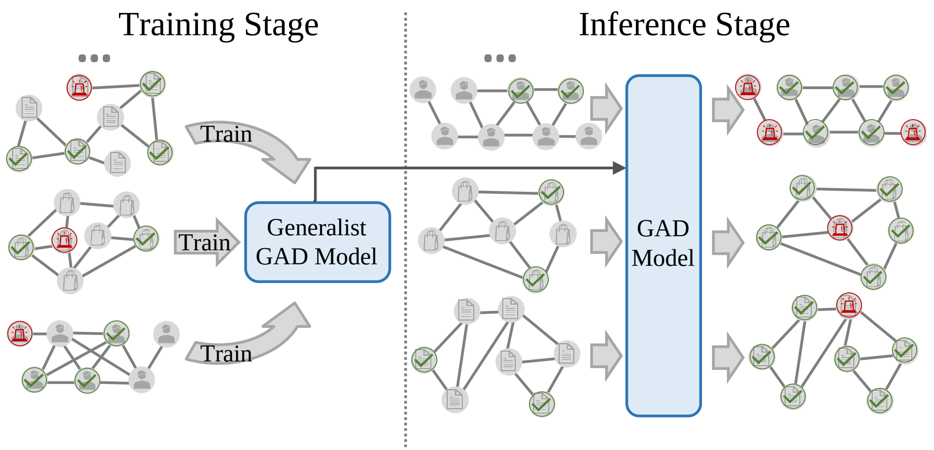
Generalist GAD: a New Paradigm



Training on multiple datasets → Directly inference on various datasets without re-training or fine-tuning

<u>The "foundation model" of GAD!</u>

Generalist GAD: a New Paradigm



Ours “generalist GAD” paradigm

Training on multiple datasets

Directly inference on various datasets

☑ No fine-tuning

→ low application costs

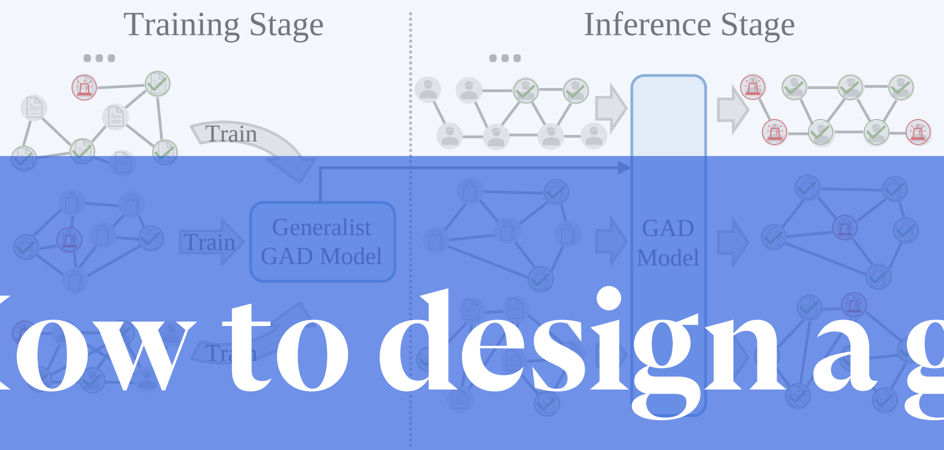
☑ Only need few-shot normal

→ low data requirement

☑ Great generalizability

→ one-for-all model

Generalist GAD: a New Paradigm



Ours "generalist GAD" paradigm

Training on multiple datasets

Directly inference on various datasets

☑ No fine-tuning

→ low application costs

☑ Only need few-shot normal

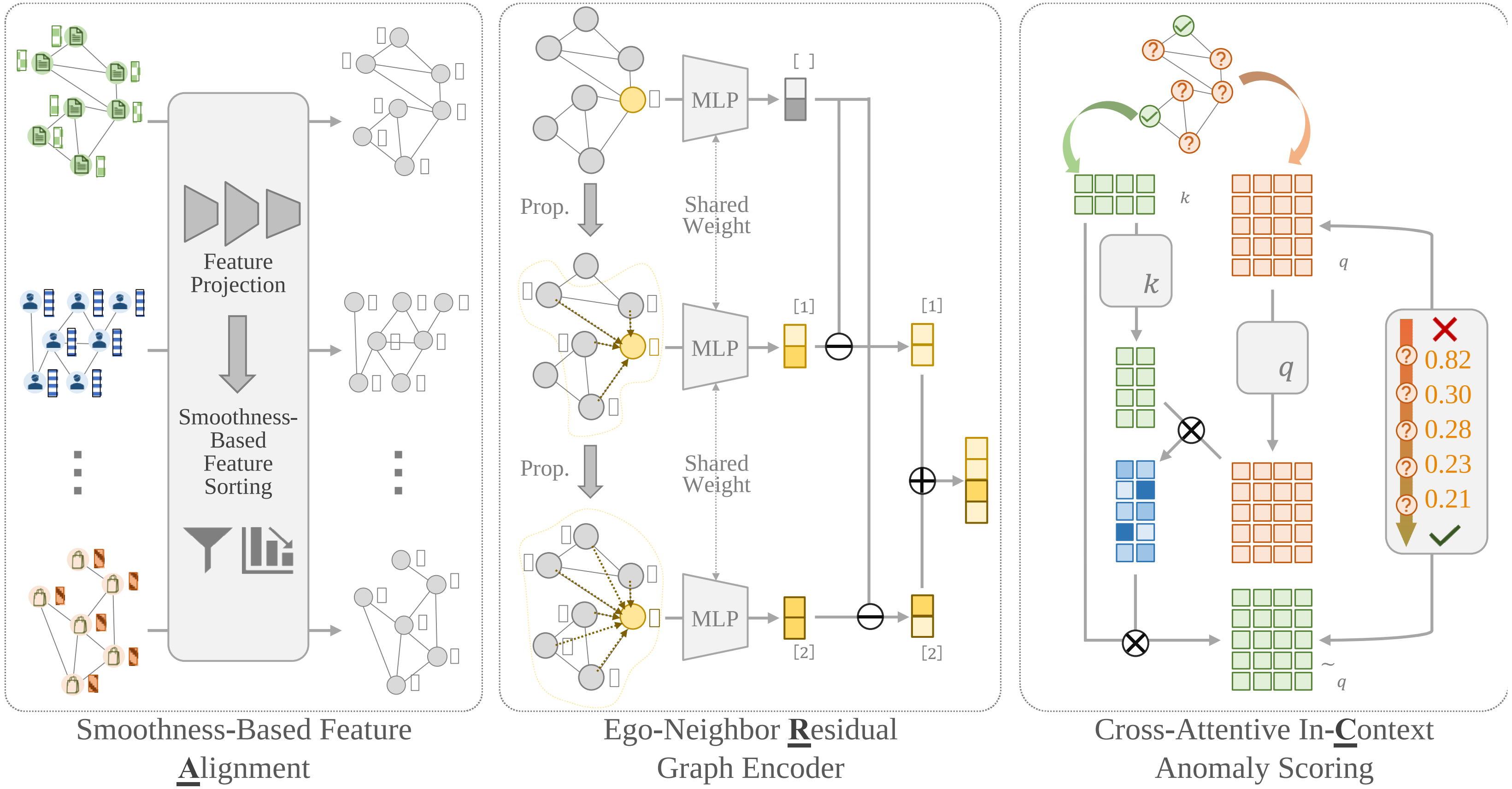
→ low data requirement

☑ Great generalizability

→ one-for-all model

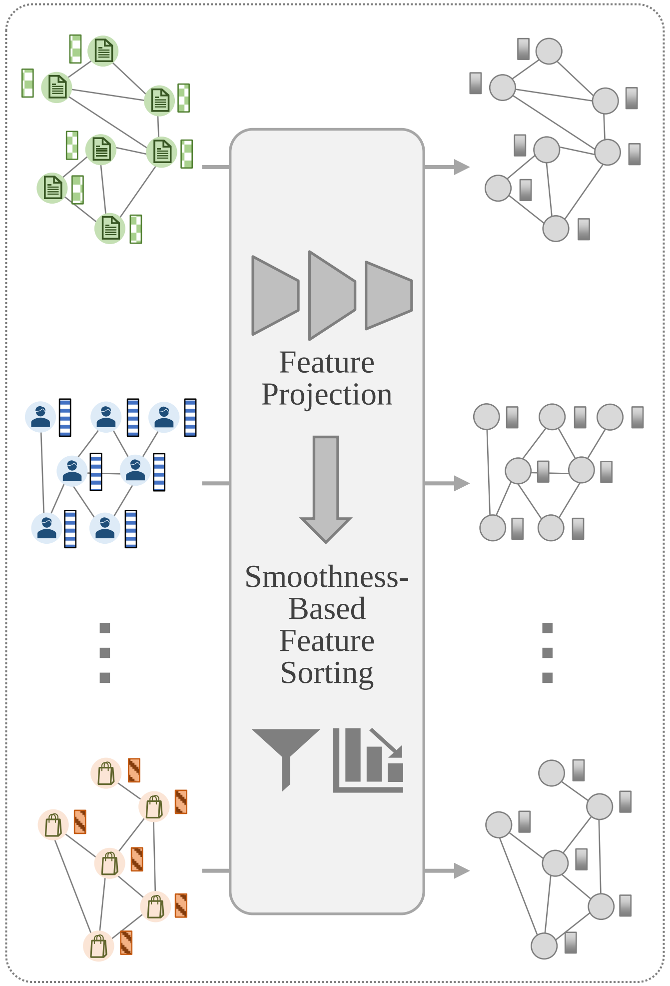
How to design a generalist GAD model?

The proposed generalist GAD method - ARC



Alignment → Encoding → Scoring

The proposed generalist GAD method - ARC



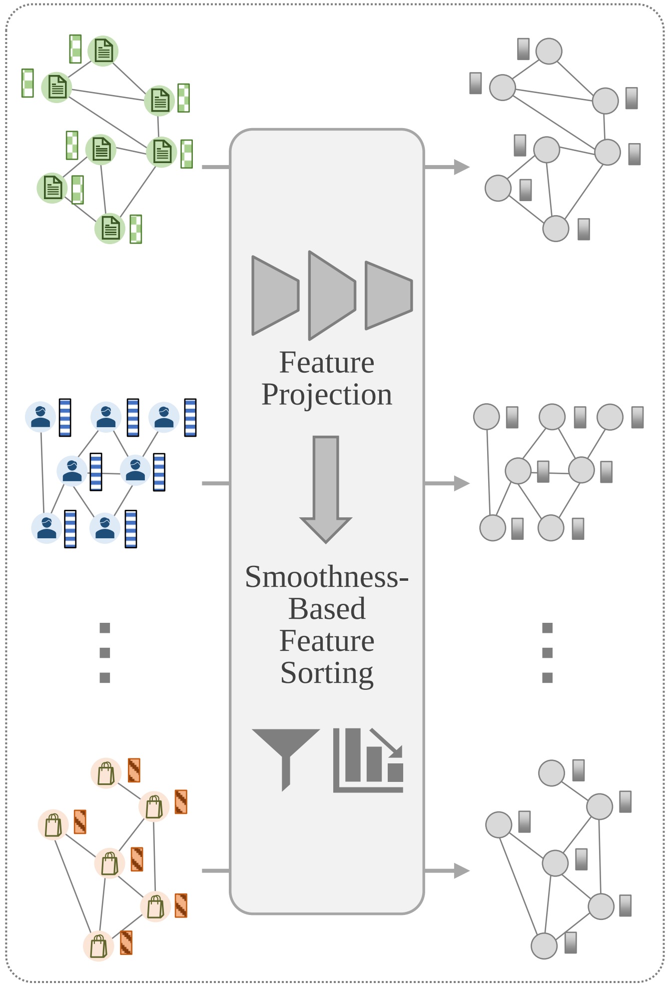
Step 1: Smoothness-Based Feature **A**lignment

- **Feature projection**

$$\tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^{n^{(i)} \times d_u} = \operatorname{Proj}\left(\mathbf{X}^{(i)}\right) = \mathbf{X}^{(i)}\mathbf{W}^{(i)},$$

Linear projection – PCA

The proposed generalist GAD method - ARC



Step 1: Smoothness-Based Feature **A**lignment

- **Feature projection**

$$\tilde{\mathbf{X}}^{(i)} \in \mathbb{R}^{n^{(i)} \times d_u} = \mathrm{Proj}\left(\mathbf{X}^{(i)}\right) = \mathbf{X}^{(i)}\mathbf{W}^{(i)},$$

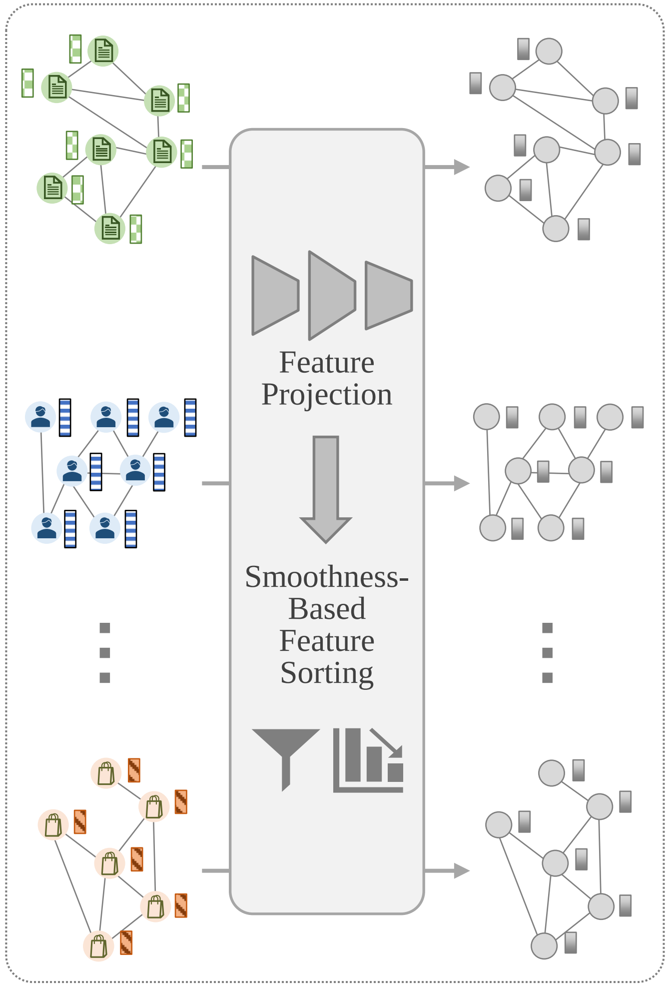
Linear projection – PCA

- **Smoothness-based feature sorting**

$$s_k(\mathbf{X}) = -\frac{1}{|\mathcal{E}|} \sum_{(v_i,v_j)\in\mathcal{E}} \left(\mathbf{X}_{ik} - \mathbf{X}_{jk}\right)^2$$

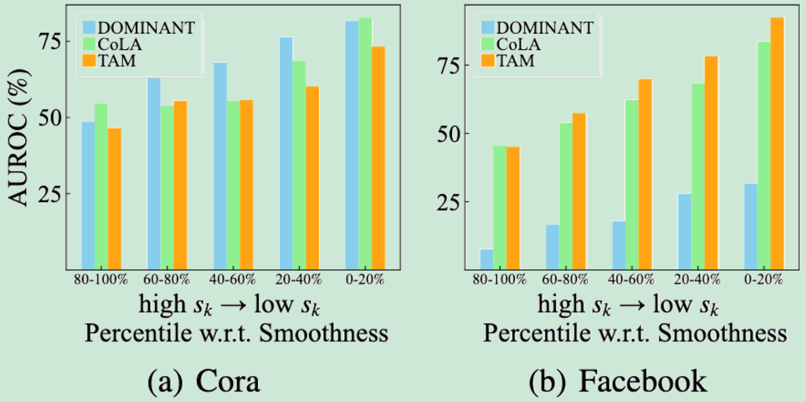
Reorder the projected features according to s

The proposed generalist GAD method - ARC



Motivation:

The contributions of features with low/high smoothness are similar across datasets!

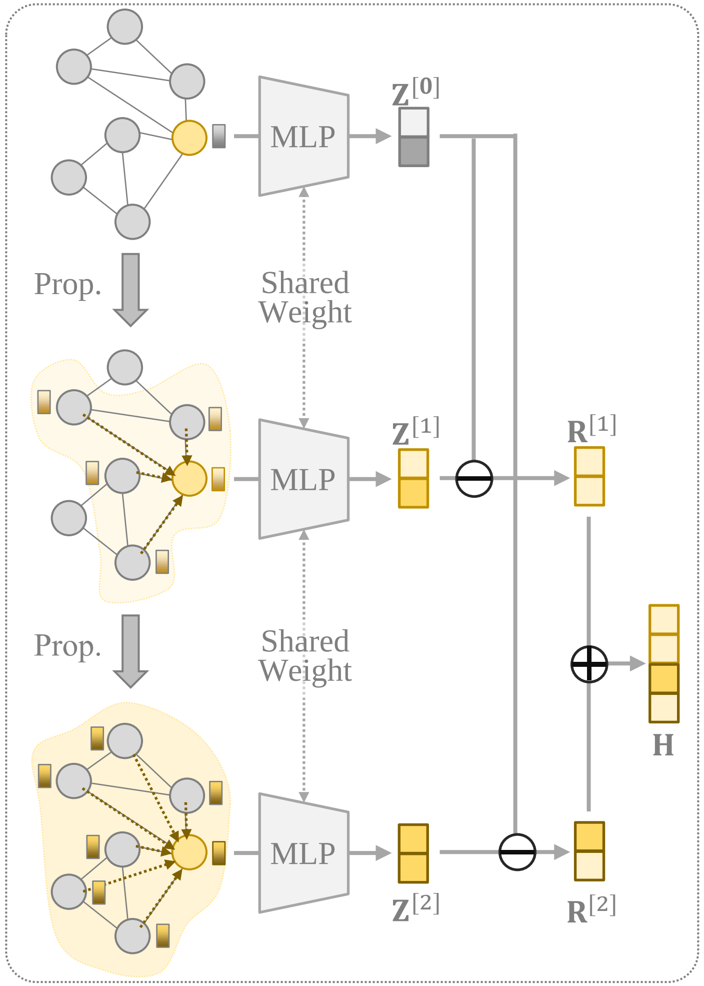


- **Smoothness-based feature sorting**

$$s_k(\mathbf{X}) = -\frac{1}{|\mathcal{E}|} \sum_{(v_i, v_j) \in \mathcal{E}} (\mathbf{X}_{ik} - \mathbf{X}_{jk})^2$$

Reorder the projected features according to s

The proposed generalist GAD method - ARC



Step 2: Ego-Neighbor **R**esidual Graph Encoder

- **Propagation**

$$\mathbf{X}^{[l]} = \tilde{\mathbf{A}}\mathbf{X}^{[l-1]}$$

- **Transformation**

$$\mathbf{Z}^{[l]} = \mathrm{MLP}\left(\mathbf{X}^{[l]}\right)$$

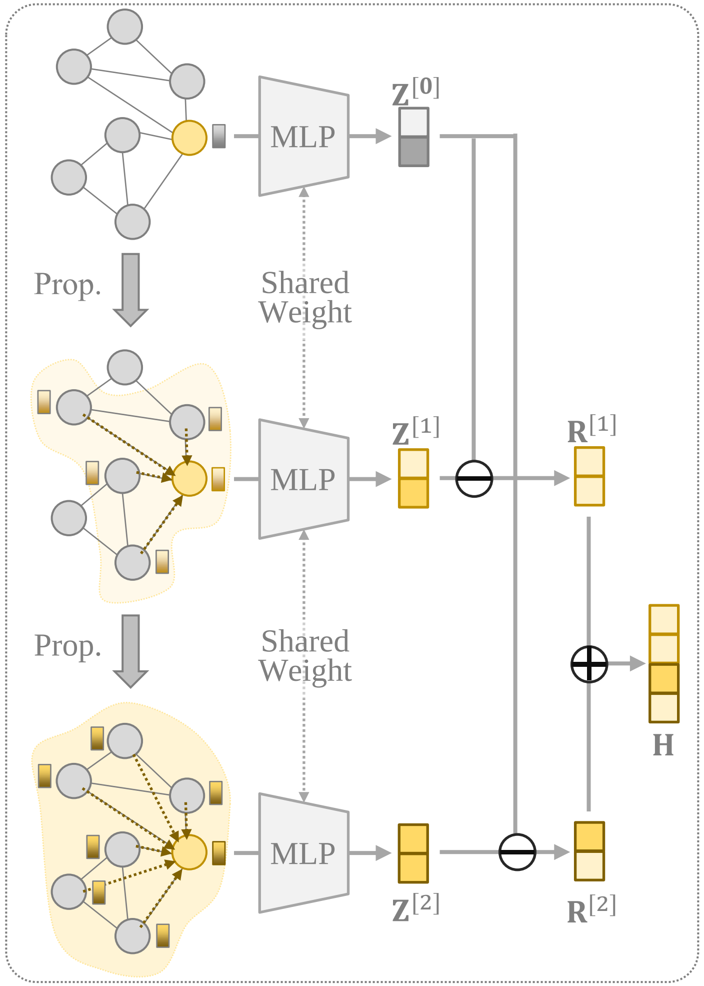
- **Residual operation**

$$\mathbf{R}^{[l]} = \mathbf{Z}^{[l]} - \mathbf{Z}^{[0]}$$

- **Concatenation**

$$\mathbf{H} = [\mathbf{R}^{[1]}||\cdots||\mathbf{R}^{[L]}]$$

The proposed generalist GAD method - ARC



Step 2: Ego-Neighbor **R**esidual Graph Encoder

- **Propagation**

$$\mathbf{X}^{[l]} = \tilde{\mathbf{A}}\mathbf{X}^{[l-1]}$$

- **Transformation**

$$\mathbf{Z}^{[l]} = \mathrm{MLP}\left(\mathbf{X}^{[l]}\right)$$

- **Residual operation**

$$\mathbf{R}^{[l]} = \mathbf{Z}^{[l]} - \mathbf{Z}^{[0]}$$

- **Concatenation**

$$\mathbf{H} = [\mathbf{R}^{[1]} || \cdots || \mathbf{R}^{[L]}]$$

Motivation:

- Residual → Local Affinity[5]

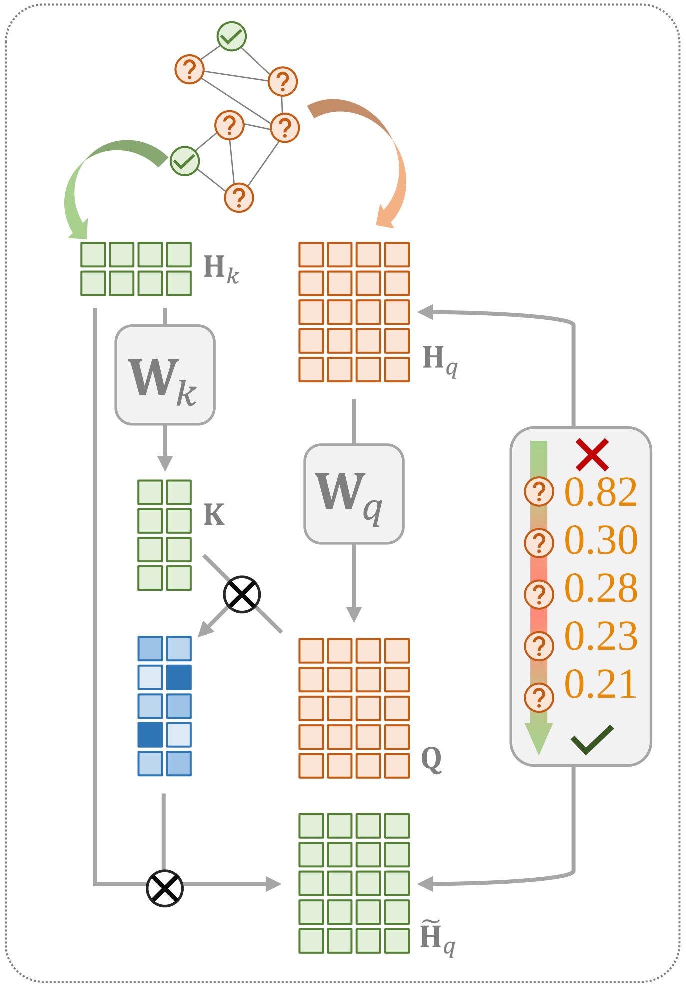
$$h(v_i) = \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i)} \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)$$

- Residual → Heterophily and High-Frequency Signals

$$\mathbf{R}^{[1]} = \mathbf{Z}^{[1]} - \mathbf{Z}^{[0]} = \tilde{\mathbf{A}}\mathbf{X}\mathbf{W} - \mathbf{X}\mathbf{W} = -\mathbf{L}\mathbf{X}\mathbf{W}$$

[5] Qiao, Hezhe, and Guansong Pang. "Truncated affinity maximization: One-class homophily modeling for graph anomaly detection." Advances in Neural Information Processing Systems 36 (2023).

The proposed generalist GAD method - ARC



Step 3: Cross-Attentive In-**C**ontext Anomaly Scoring

- **Cross-attention**

Key: labelled normal nodes $\mathbf{H}_k$

Query: unlabelled nodes $\mathbf{H}_q$

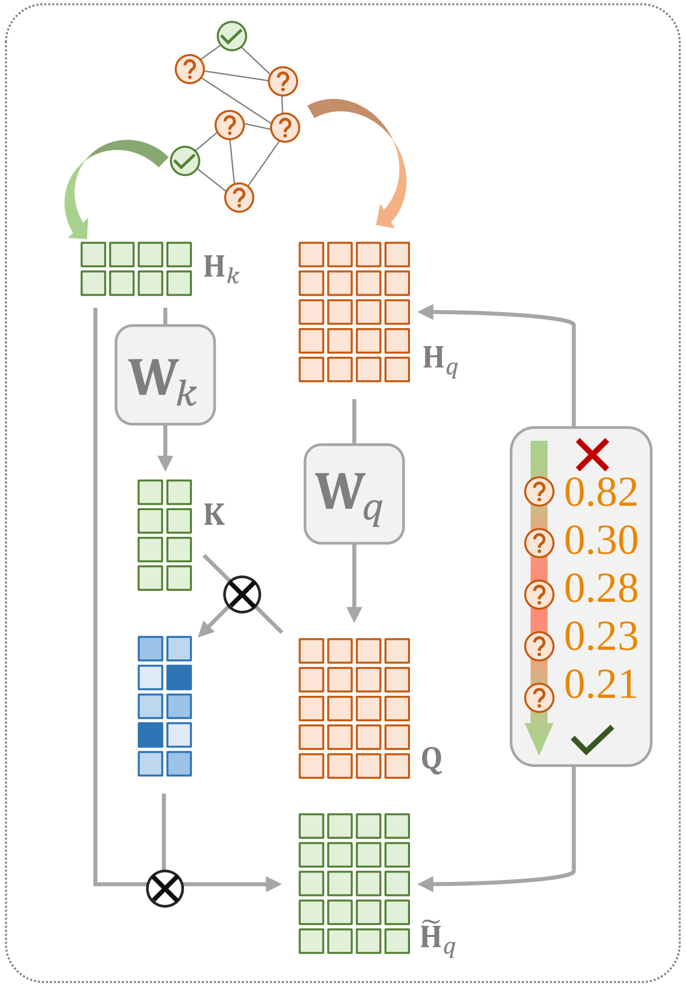
$$\mathbf{Q} = \mathbf{H}_q \mathbf{W}_q$$
$$\mathbf{K} = \mathbf{H}_k \mathbf{W}_k$$

$$\tilde{\mathbf{H}}_q = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_e}}\right)\mathbf{H}_k$$

Training objective: Reconstruct $\mathbf{H}_q$ with $\mathbf{H}_k$

$$\mathcal{L} = \begin{cases} 1 - \cos\left(\mathbf{H}q_i, \tilde{\mathbf{H}}q_i\right), & \text{if } \mathbf{y}_i = 0 \\ \max\left(0, \cos\left(\mathbf{H}q_i, \tilde{\mathbf{H}}q_i\right) - \epsilon\right), & \text{if } \mathbf{y}_i = 1 \end{cases}$$

The proposed generalist GAD method - ARC



Step 3: Cross-Attentive In-**C**ontext Anomaly Scoring

- **Cross-attention**

Key: labelled normal nodes $\mathbf{H}_k$
Query: unlabelled nodes $\mathbf{H}_q$

$$\mathbf{Q} = \mathbf{H}_q\mathbf{W}_q$$
$$\mathbf{K} = \mathbf{H}_k\mathbf{W}_k$$

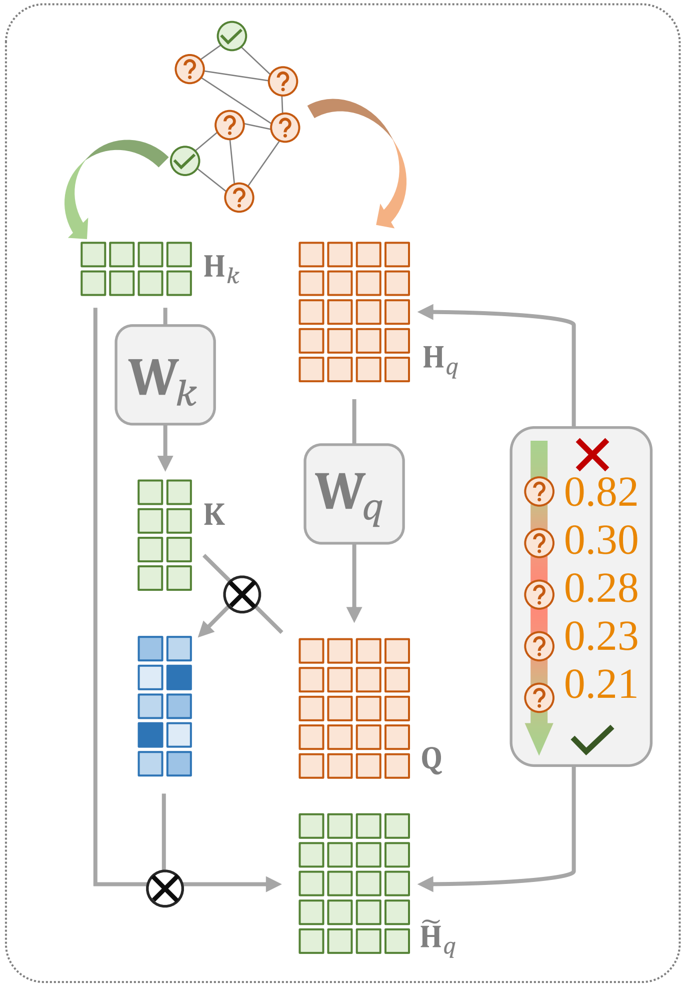
$$\tilde{\mathbf{H}}_q = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_e}}\right)\mathbf{H}_k$$

- **Anomaly scoring**

Reconstruction errors as anomaly scores

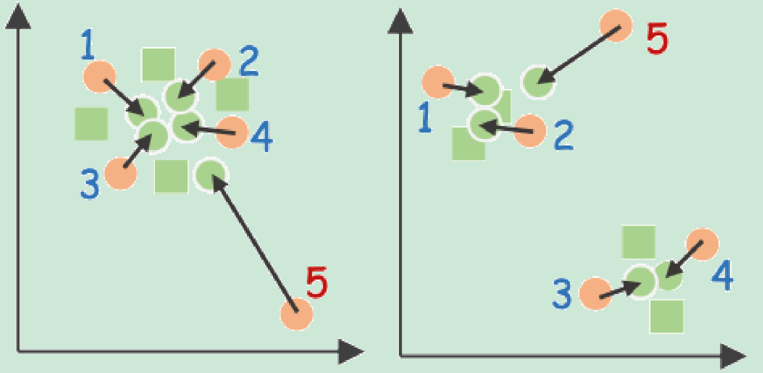
$$f(v_i) = d(\mathbf{H}q_i, \tilde{\mathbf{H}}q_i) = \sqrt{\sum_{j=1}^{d_e}\left(\mathbf{H}q_{ij} - \tilde{\mathbf{H}}q_{ij}\right)^2}$$

The proposed generalist GAD method - ARC



<u>**Step 3**</u>: Cross-Attentive In-<u>**C**</u>ontext Anomaly Scoring

Motivation:
normal query nodes can be easily reconstructed by the key nodes (other normal nodes)



(a) Case I (b) Case II

- **Anomaly scoring**

Reconstruction errors as anomaly scores

$$f(v_i) = d(\mathbf{H}q_i, \tilde{\mathbf{H}}q_i) = \sqrt{\sum_{j=1}^{d_e} \left(\mathbf{H}q_{ij} - \tilde{\mathbf{H}}q_{ij}\right)^2}$$

Experiments: Settings

- 4 groups of datasets
- the largest dataset → training datasets; the rest → testing datasets

Dataset	Train	Test	#Nodes	#Edges	#Features	Avg. Degree	#Anomaly	%Anomaly
Citation network with injected anomalies								
Cora	-	✓	2,708	5,429	1,433	3.90	150	5.53
CiteSeer	-	✓	3,327	4,732	3,703	2.77	150	4.50
ACM	-	✓	16,484	71,980	8,337	8.73	597	3.62
PubMed	✓	-	19,717	44,338	500	4.50	600	3.04
Social network with injected anomalies								
BlogCatalog	-	✓	5,196	171,743	8,189	66.11	300	5.77
Flickr	✓	-	7,575	239,738	12,047	63.30	450	5.94
Social network with real anomalies								
Facebook	-	✓	1,081	55,104	576	50.97	25	2.31
Weibo	-	✓	8,405	407,963	400	48.53	868	10.30
Reddit	-	✓	10,984	168,016	64	15.30	366	3.33
Questions	✓	-	48,921	153,540	301	3.13	1,460	2.98
Co-review network with real anomalies								
Amazon	-	✓	10,244	175,608	25	17.18	693	6.76
YelpChi	✓	-	23,831	49,315	32	2.07	1,217	5.10

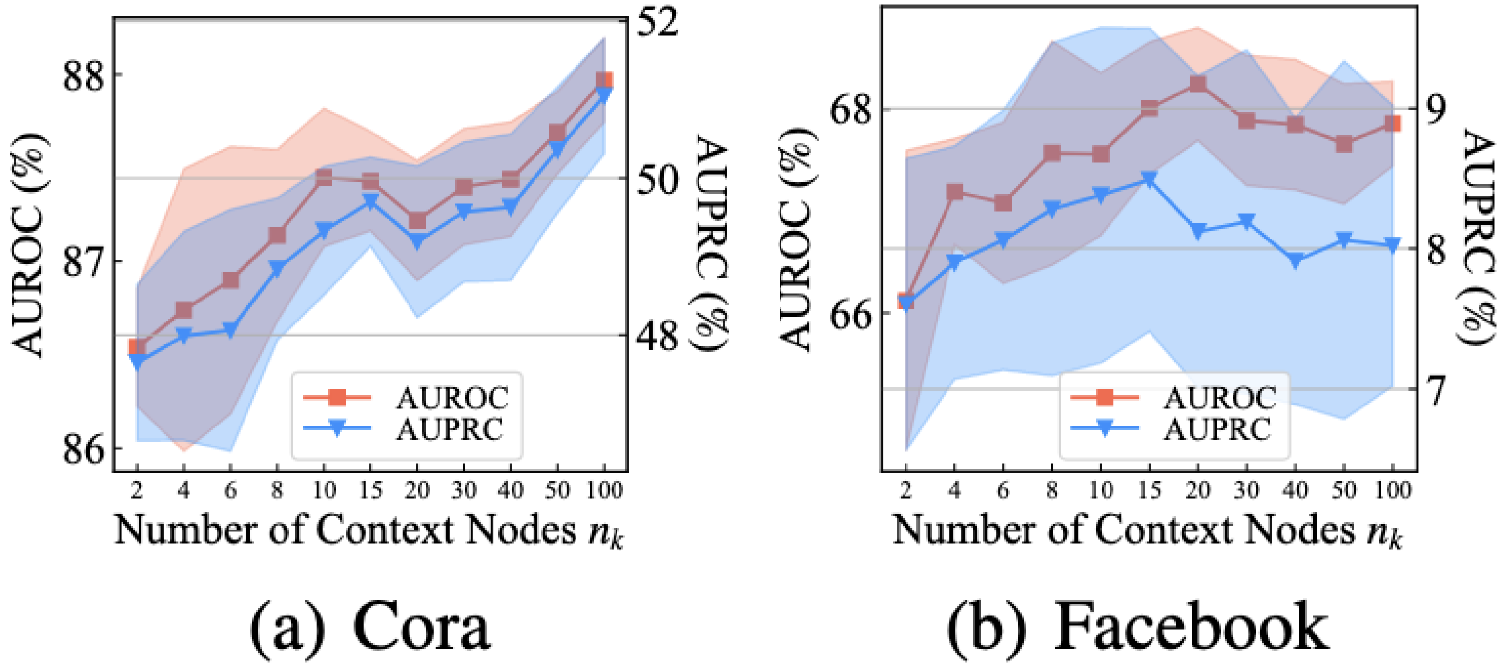
Experiments: Main Results

Method	Cora	CiteSeer	ACM	BlogCatalog	Facebook	Weibo	Reddit	Amazon	Rank
Supervised - Pre-Train Only									
GCN	59.64±8.30	60.27±8.11	60.49±9.65	56.19±6.39	29.51±4.86	76.64±17.69	50.43±4.41	46.63±3.47	8.9
GAT	50.06±2.65	51.59±3.49	48.79±2.73	50.40±2.80	51.88±2.16	53.06±7.48	51.78±4.04	50.52±17.22	10.0
BGNN	42.45±11.57	42.32±11.82	44.00±13.69	47.67±8.52	54.74±25.29	32.75±35.35	50.27±3.84	52.26±3.31	11.1
BWGNN	54.06±3.27	52.61±2.88	67.59±0.70	56.34±1.21	45.84±4.97	53.38±1.61	48.97±5.74	55.26±16.95	9.0
GHRN	59.89±6.57	56.04±9.19	55.65±6.37	57.64±3.48	44.81±8.06	51.87±14.18	46.22±2.33	49.48±17.13	9.8
Unsupervised - Pre-Train Only									
DOMINANT	66.53±1.15	69.47±2.02	70.08±2.34	74.25±0.65	51.01±0.78	**92.88±0.32**	50.05±4.92	48.94±2.69	5.8
CoLA	63.29±8.88	62.84±9.52	66.85±4.43	50.04±3.25	12.99±11.68	16.27±5.64	52.81±6.69	47.40±7.97	9.5
HCM-A	54.28±4.73	48.12±6.80	53.70±4.64	55.31±0.57	35.44±13.97	65.52±12.58	48.79±2.75	43.99±0.72	11.4
TAM	62.02±2.39	72.27±0.83	74.43±1.59	49.86±0.73	65.88±6.66	71.54±0.18	55.43±0.33	56.06±2.19	5.6
Unsupervised - Pre-Train & Fine-Tune									
DOMINANT	72.23±0.34	74.69±0.32	74.34±0.12	74.61±0.04	49.92±0.55	92.21±0.10	52.14±5.06	59.06±2.80	3.6
CoLA	67.62±4.26	70.75±3.42	69.11±0.67	62.49±3.38	64.70±18.86	31.55±6.02	58.12±0.67	52.51±6.66	5.4
HCM-A	56.45±4.93	55.54±4.07	57.69±3.59	55.10±0.29	36.57±10.72	71.89±2.79	49.15±2.72	42.20±0.55	10.1
TAM	62.56±2.10	76.54±1.33	**86.29±1.57**	57.69±0.88	**76.26±3.70**	71.73±0.16	56.62±0.49	57.13±1.59	3.4
Ours									
ARC	**87.45±0.74**	**90.95±0.59**	79.88±0.28	**74.76±0.06**	67.56±1.60	88.85±0.14	**60.04±0.69**	**80.67±1.81**	1.5

🎯 Strong detection capability without fine-tuning

🌍 Generalizability in different datasets/domains

Experiments: Sensitivity In Terms of #shots



✨ Works well with extremely few shots

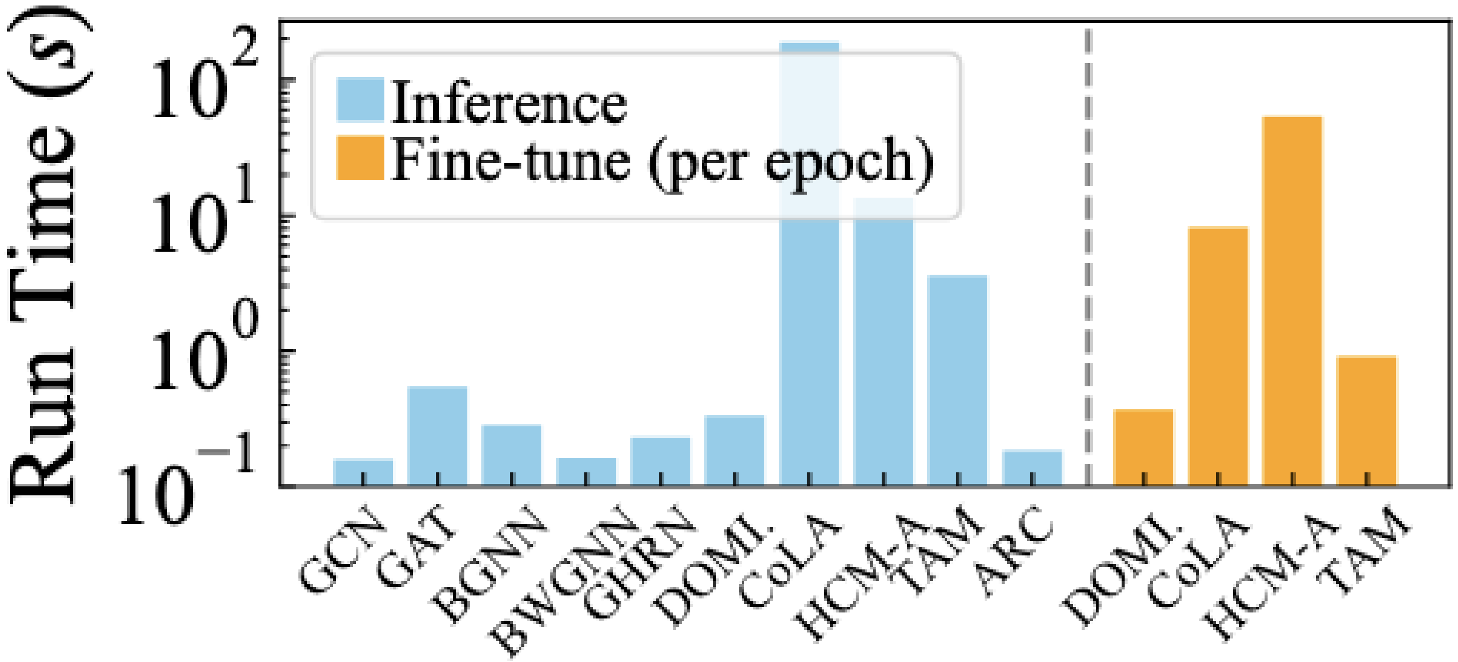
📈 More labelled normal samples bring better performance

Experiments: Ablation Study

Variant	Cora	CiteSeer	ACM	BlogCatalog	Facebook	Weibo	Reddit	Amazon
ARC w/o A	80.65±0.71	83.35±0.64	79.29±0.16	73.86±0.18	62.80±2.06	89.69±0.17	54.60±1.92	64.76±2.13
ARC w/o R	37.44±1.40	31.52±0.71	61.83±1.16	49.30±2.06	20.38±9.63	97.72±0.59	52.94±0.96	50.15±0.24
ARC w/o C	47.39±0.42	53.98±0.72	54.24±1.32	60.46±1.23	48.86±0.97	42.84±3.01	51.03±0.86	69.02±0.97
ARC	87.45±0.74	90.95±0.59	79.88±0.28	74.76±0.06	67.56±1.60	88.85±0.14	60.04±0.69	80.67±1.81

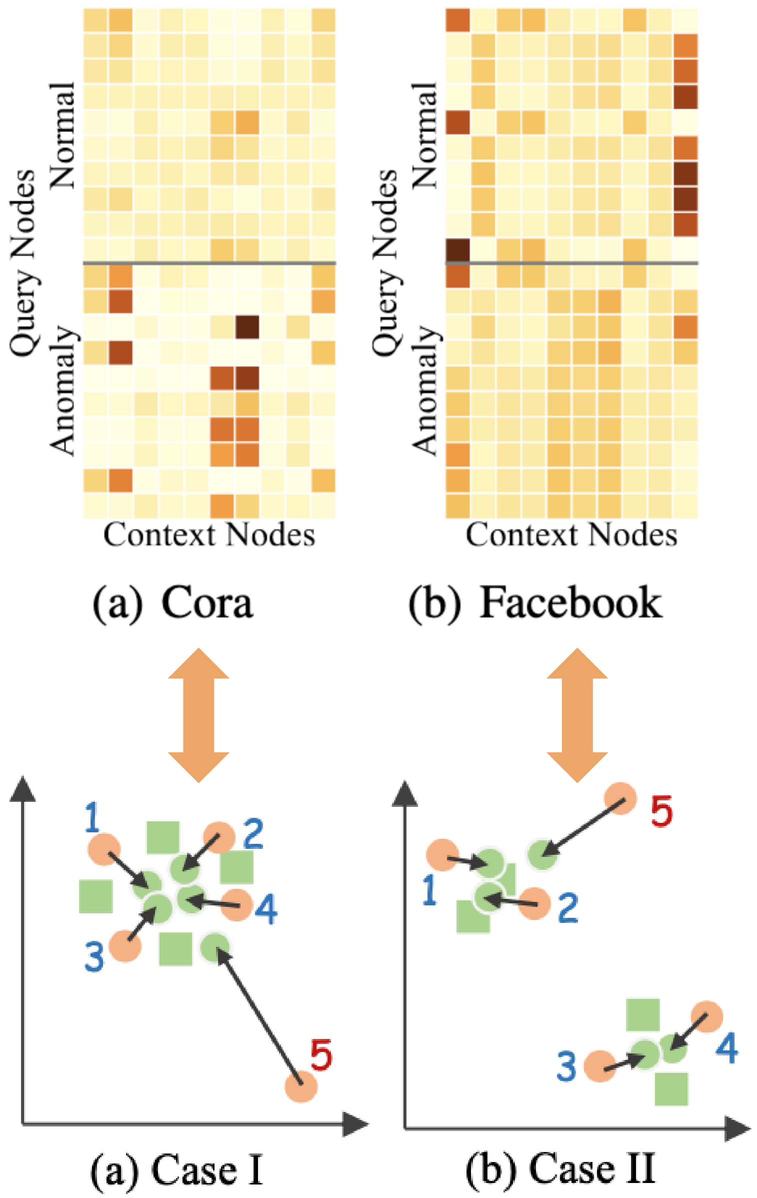
♻️ Each component has a significant contribution to the final performance

Experiments: Efficiency Analysis



High inference efficiency – comparable to GCN

Experiments: Visualization



🧐 Interpretability – attention score

Case 1: uniform attention weights
→ "Single-class normal ": Reconstructed embeddings that closely to the average embedding of the context nodes

Case 2: two fixed patterns for normal queries
→ "Multi-class normal": Two cluster centers

Summary

New paradigm: generalist GAD: one model for all datasets!

Effective solution: ARC – a simple yet effective methods

Extensive experiments: ARC enjoys superior performance, great generalizability, high running efficiency, and potential explainability

Full paper

GitHub

Thanks for your listening!