Recent advances in Spectral Perspectives for Anomaly Detection

스펙트럴 관점에서 이상탐지

Bonyou Koo(구본유)

Supervisor: Byungkook Oh

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

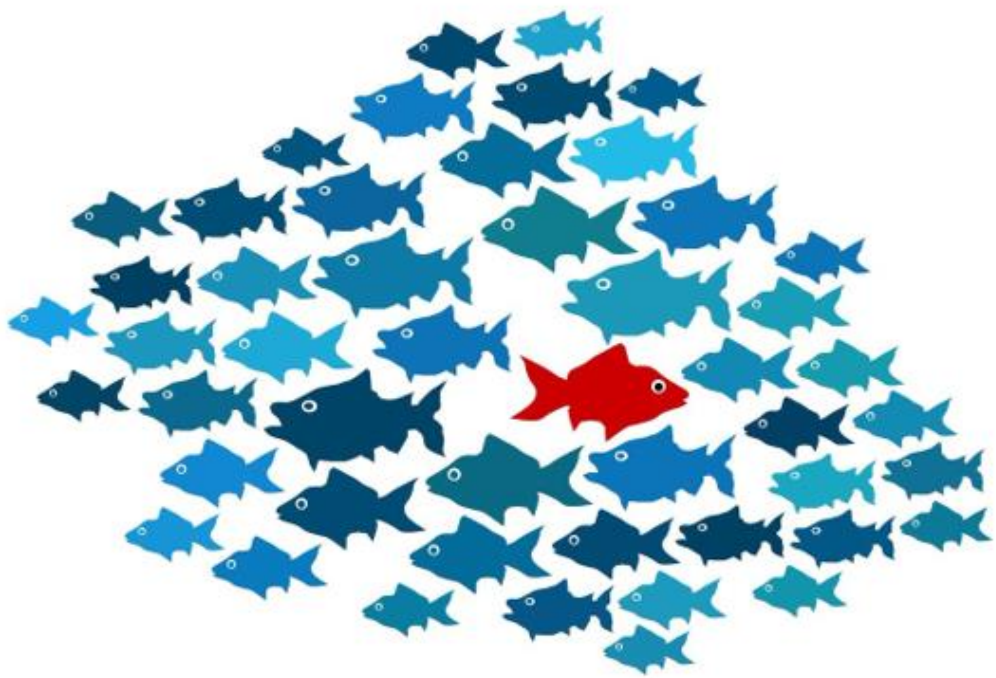
2026.02.19

CONTENTS

CONTENTS

Introduction

Node level anomaly Detection



- Anomalies
  - ✓ Not conform to a notion of normal behavior.

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

Background

Node level anomaly Detection

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

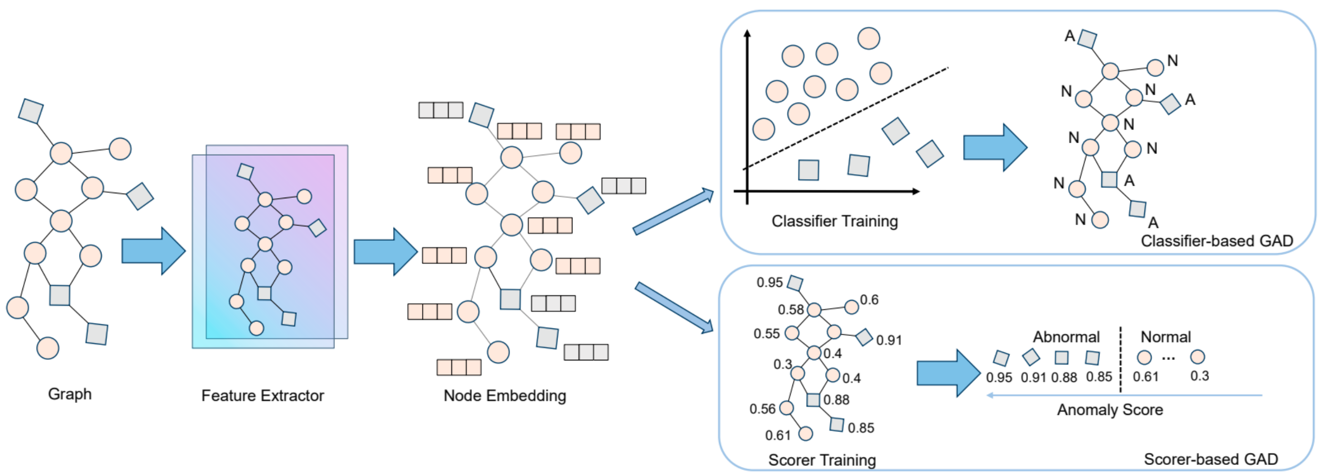


Fig. 1. Illustration of Classifier-based and Scorer-based GAD. After obtaining features of nodes, classifier-based GAD method will train a classifier to determine whether a node is normal (N) or abnormal (A). Differently, Scorer-based GAD will train a scorer and assign anomaly scores to nodes, set thresholds according to score sorting and filter out abnormal nodes.

Background

Node level GAD

- $f: \mathbb{R}^{N\times \mathrm{N}} \times \mathbb{R}^{N\times D} \rightarrow \mathbb{R}^{N}$
  - $s_i = f(A, X)_i \;\; i \in V$
  - $v_a \sim V_a \; v_n \sim V_n$
  - $\max P(s_a > s_n)$
- Higher score ≈ higher likelihood of being anomalous

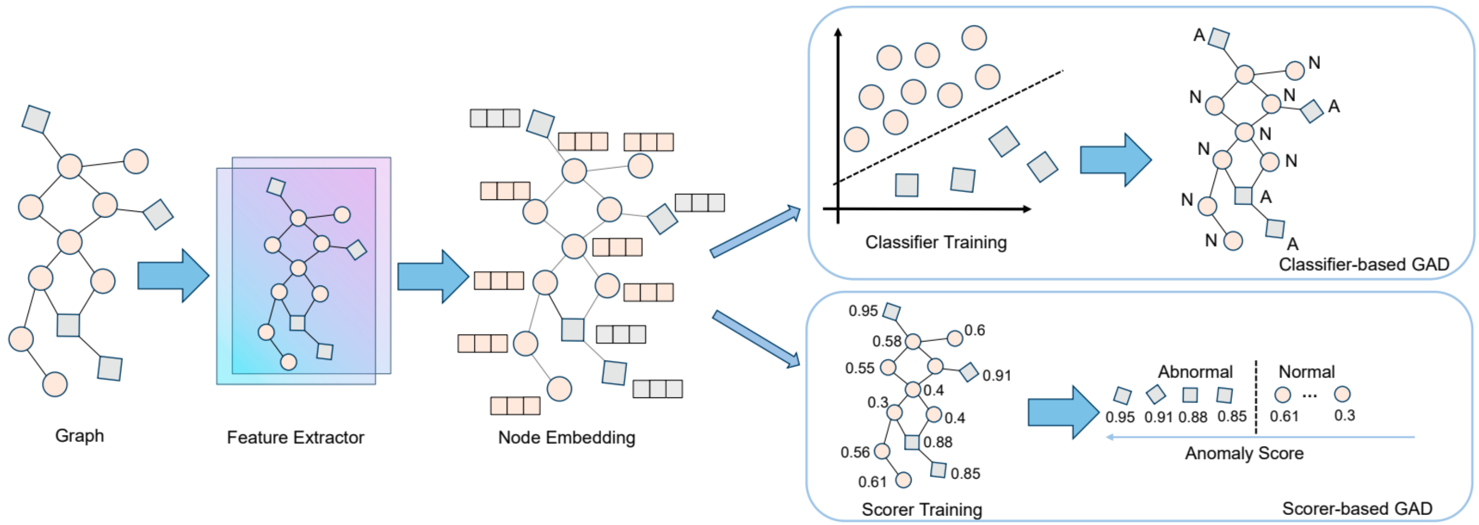


Fig. 1. Illustration of Classifier-based and Scorer-based GAD. After obtaining features of nodes, classifier-based GAD method will train a classifier to determine whether a node is normal (N) or abnormal (A). Differently, Scorer-based GAD will train a scorer and assign anomaly scores to nodes, set thresholds according to score sorting and filter out abnormal nodes.

A Brief Survey on Graph Anomaly Detection, Procedia Computer Science 2024

Background

Node level GAD

- Required to learn anomalous nodes and structures
  - ✓ Without explicit supervision.
- Due to the rarity of anomalies
  - ✓ node-level anomaly label are difficult to obtain

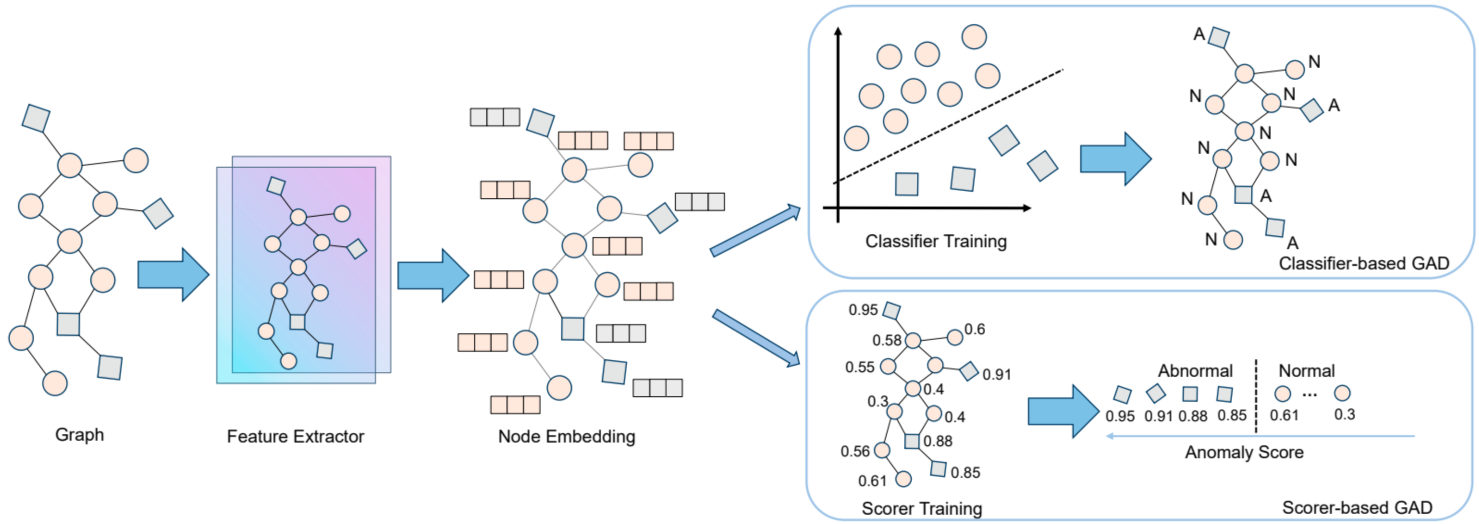
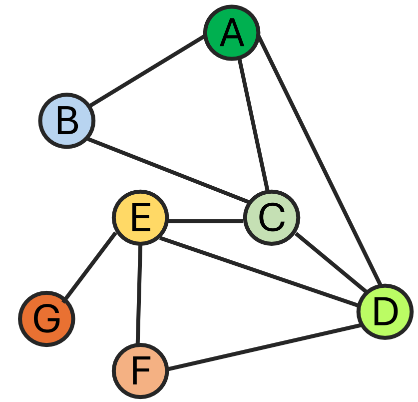


Fig. 1. Illustration of Classifier-based and Scorer-based GAD. After obtaining features of nodes, classifier-based GAD method will train a classifier to determine whether a node is normal (N) or abnormal (A). Differently, Scorer-based GAD will train a scorer and assign anomaly scores to nodes, set thresholds according to score sorting and filter out abnormal nodes.

A Brief Survey on Graph Anomaly Detection, Procedia Computer Science 2024

CONTENTS

Preliminary



$\mathcal{G} = (A, X)$

- ✓ undirected graph with N nodes, M edges

- $A \in \{0,1\}^{N \times N}$ with $N = |V|$
  - ✓ $A_{ij} = 1 \; iff$ exist an edge between nodes i and j
    - If with self loops
    - $\tilde{A} = A + I_n$
      - $I_n \in \mathbb{R}^{N \times N}$

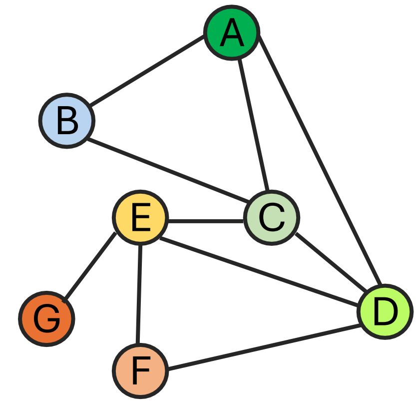
- X
  - ✓ Features $\in \mathbb{R}^{N \times d}$
  - ✓ Graph signal $(x_1, x_2, \cdots x_n)^T \in \mathbb{R}^N$

Preliminary

- Continual space
  - ✓ 2d spatial <-> frequency
    - ▪ Axis : x,y <-> Fourier basis: $e^{i\omega t}$

- Graph domain
  - ✓ spatial <-> frequency
    - ▪ Axis : graph structure(nodes edges) <-> Fourier basis: Laplacian's eigenvectors
  - ✓ Spectral = frequency view
    - ▪ Low frequency vs high frequency
      - – Normal vs abnormal
        - » Dirichlet energy, Rayleigh Quotient : measure the high frequency
        - » smoothGNN, RQGNN, APF : modeling the high frequency

Preliminary

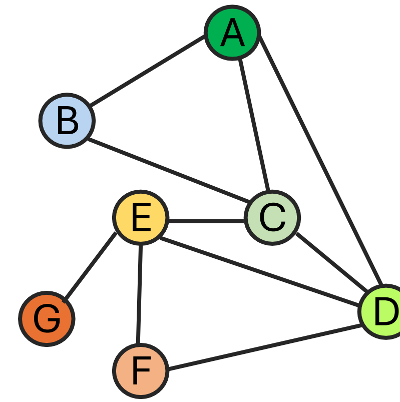


Laplacian matrix

- L = D – A
  - ✓ D : Diagonal degree matrix

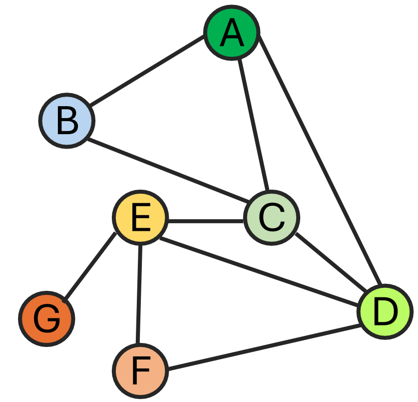
- L = $I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$
  - ✓ S = $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$
    - $||S|| \leq 1, |\lambda_i(S)| \leq 1$
  - ✓ L ∈ $[0,2]$
    - L $\succcurlyeq 0$

Preliminary



- $L = I - S = U\Lambda U^T$
  - ✓ $U = (u_1, u_2, \dots u_n)$
    - Orthonormal eigenvectors
    - iff L is symmetric and normalized
  - ✓ $\Lambda = (\lambda_1, \lambda_2, \dots \lambda_n)$
    - corresponding eigen value
      - If sorted in ascending order
        - » $0 \leq \lambda_1 \leq \dots \leq \lambda_n$
        - » $0 \preccurlyeq S$
        - » $0 = \lambda_1$

Preliminary



- L = $I - S = U\Lambda U^T$
  - ✓ $U = (u_1, u_2, \ldots u_n)$
    - Orthonormal eigenvectors
    - iff L is symmetric and normalized
  - ✓ Orthonormal eigenvector
    - $U^T U = I$
  - ✓ Parseval's theorem
    - $||x||^2 = ||U^T x||^2 = \sum_{i=1}^{n} \hat{x}^2$
    - $U^T x$: graph Fourier transform on x

Preliminary

- Spectral energy $e_k = \frac{\hat{x}_k^2}{\sum_{i=1}^{N} \hat{x}_i^2}$
  - Fourier coefficient
    - the proportion of total spectral energy at frequency k
    - $E_{total} = ||x||_2^2 = x^T x = (Ux)^T(Ux) = \hat{x}U^T U\hat{x} = \sum_i \hat{x}_i^2$
      - $\hat{x} = Ux$
        - coefficient after graph Fourier transform

- Accumulated Spectral energy $\frac{\sum_{j=1}^{k} \hat{x}_j^2}{\sum_{i=1}^{n} \hat{x}_i^2}$
  - the cumulative proportion of spectral energy From e_1 *to* e_k
  - Capture low pass energy distribution

Preliminary

- Dirichlet energy
  - ✓ $x^T L x = \frac{1}{2} \sum_{ij} A_{ij} \left(x_i - x_j \right)^2$
    - $L = D - A \in \mathbb{R}^{n \times n}$
      - – $D = \sum_j A_{ij} \in \mathbb{R}^{n \times n}$
        - » Degree diagonal matrix
      - – A $\in \mathbb{R}^{n \times n}$
        - » Adjacency matrix
    - $X \in \mathbb{R}^n$ or $\in \mathbb{R}^{n \times d}$
    - Measure the high frequency energy
  - ✓ $x^T L x = x^T \mathrm{U} \Lambda U^T x = \hat{x}^T \Lambda \hat{x} = \sum_i \lambda_k \hat{x}_k^2$
    - $\hat{x}_k^2$: quantity of energy
    - λ_k : frequency
    - λ-weighted spectral energy decomposition

Preliminary

- Rayleigh Quotient
  - ✓ $\frac{x^T L x}{x^T x} \in \mathbb{R}^{1\times 1}\ or\ \mathbb{R}^{d\times d}$
  - ✓ $\frac{x^T L x}{x^T x} = \Lambda$
    - $\frac{x^T \mathrm{U} \Lambda U^T x}{\hat{x} U^T U \hat{x}} = \Lambda \frac{\hat{x}^T \mathrm{U} U^T \hat{x}}{\hat{x} U^T U \hat{x}}$

Right-shift phenomenon

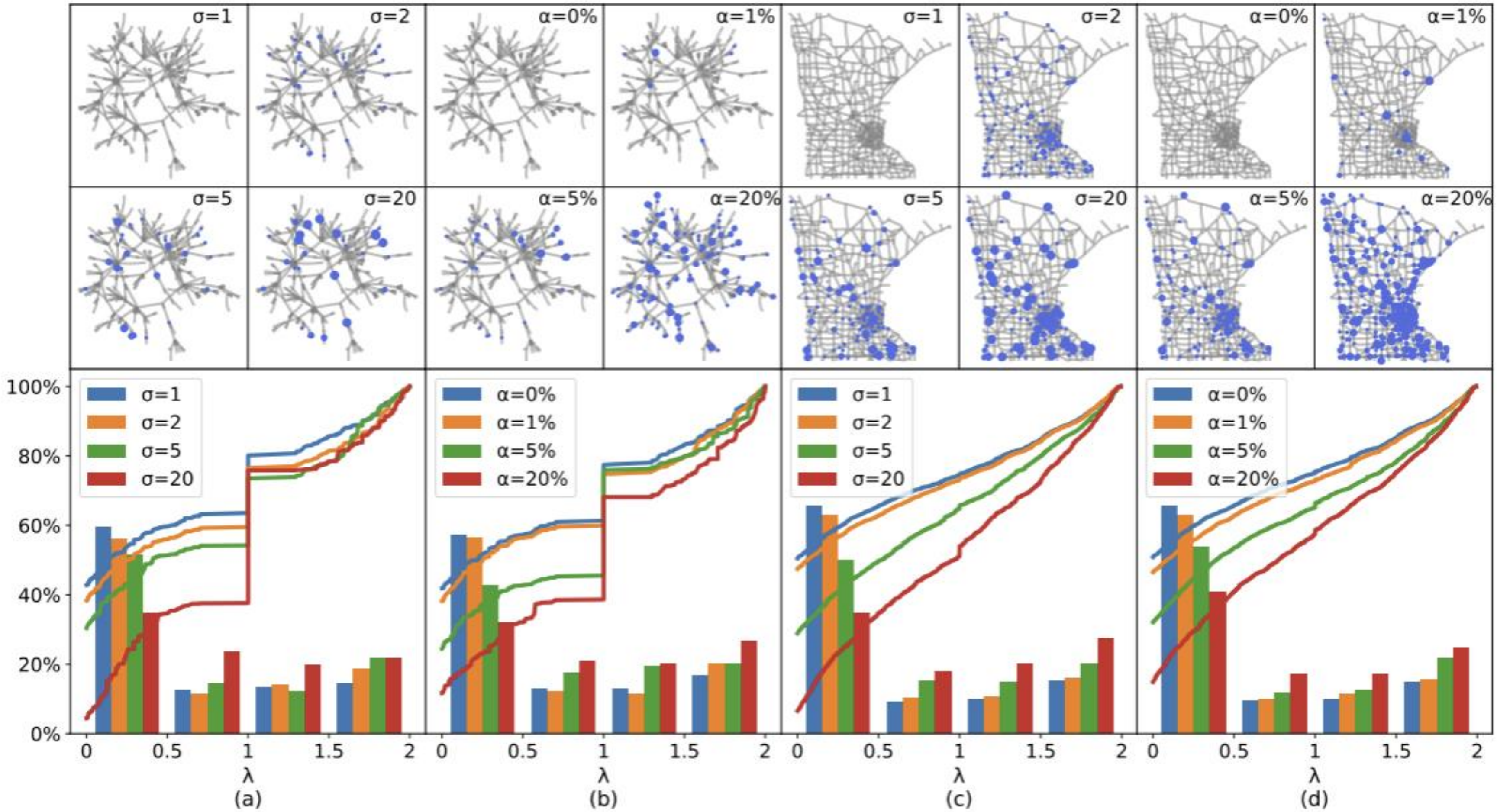


Figure 1. The effect of graph anomalies in the spatial domain (top) and spectral domain (bottom) with different anomaly degrees. The cases (a,c) are related to different standard deviation of anomalies (type (i), $\sigma = 1, 2, 5, 20$) while the cases (b,d) are about different fraction of anomalies (type (ii), $\alpha = 0\%, 1\%, 5\%, 20\%$).

Rethinking graph neural networks for anomaly detection, ICML,2022.

Preliminary

- Graph convolution operator
  - ✓ $g_\theta * (L)x = U^T g_\theta(\Lambda)Ux$
    - $\theta \in \mathbb{R}^n$: parameter
  - ✓ $U^T g_\theta(\Lambda)Ux \approx U(\sum_{t=0}^{T} \theta_t \Lambda^T)U^T x = (\sum_{t=0}^{T} \theta_t L^T)x$
    - $\theta \in \mathbb{R}^{T+1}$
      - Polynomial coefficient

CONTENTS

SmoothGNN: Smoothing-aware GNN for Unsupervised Node Anomaly Detection

Proceedings of the ACM on Web Conference 2025

Xiangyu Dong[1)], Xingyi Zhang[1)], Yanni Sun[1)], Lei Chen[2)]
Mingxuan Yuan[2)], Sibo Wang[1)]

1) The Chinese University of Hong Kong
2) Huawei Noah's Ark Lab

SmoothGNN

Smoothing issue

- Lead to indistinguishable node representation
- the over-smoothing problem in node anomaly detection

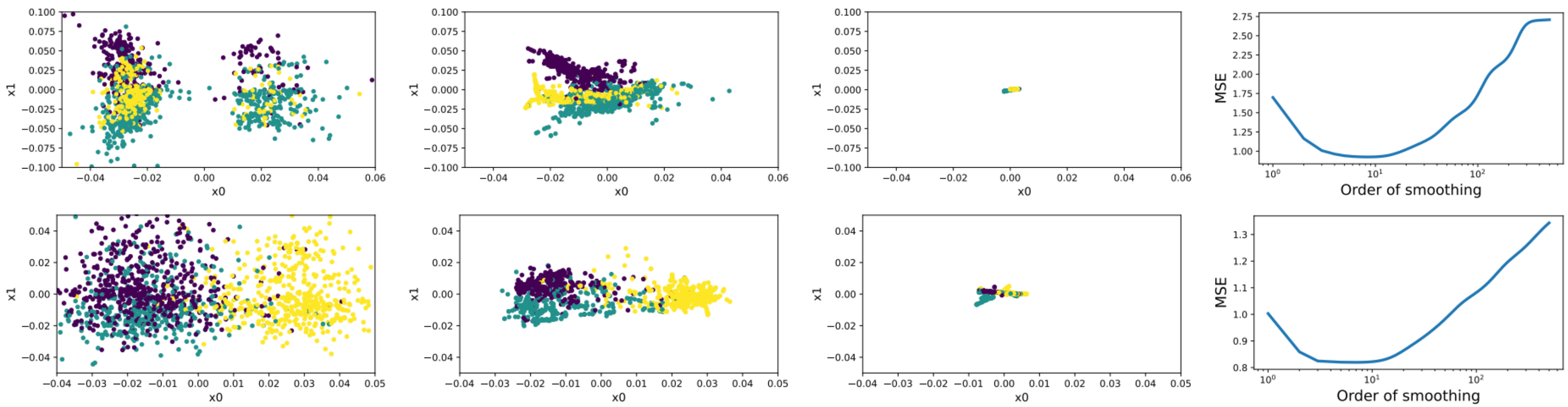


Figure 1: Illustration of both beneficial smoothing and oversmoothing on Cora [32] (top) and Citeseer [14] (bottom). **From left to right:** node features after performing respectively $k = 0, 10$, and 500 steps of mean aggregation, along the first two principal-components (of the original unsmoothed features), for three classes of nodes for better visibility. **Figure on the right:** Mean Square Error of Linear Ridge Regression (LRR) on the smoothed features with respect to the order of smoothing k. We observe that smoothing first gather same-labels nodes and improves learning, before they eventually collapses to a single point (note that here we show LRR for consistency with the analysis presented in this paper, even though these are node classification tasks).

Not too little, not too much: a theoretical analysis of graph (over)smoothing , NeurIPS 2022

SmoothGNN

- Smoothing issue
  - Lead to indistinguishable node representation
  - the over-smoothing problem in node anomaly detection

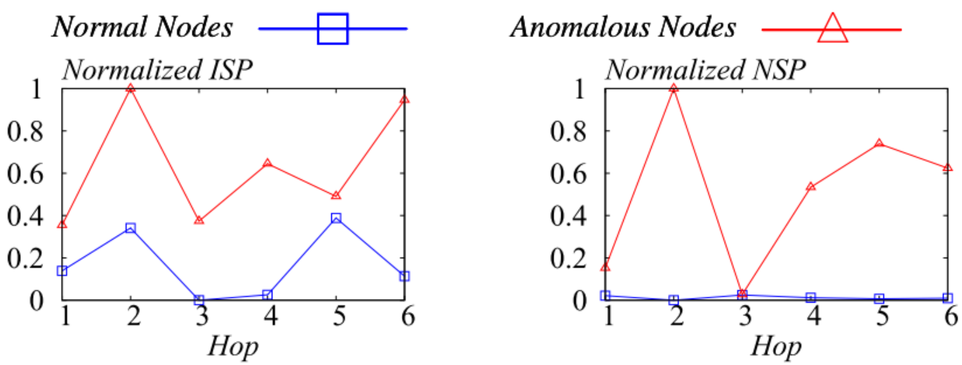
- GCN
  - ✓ X = $(\sum_{t=0}^{T} \theta_t L^T)x$
  - ✓ X = SX
    - S = $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ = 1 − L
    - T =1
    - $\theta_1 = 1$
      - $\theta_{\mathrm{T}>1} = 0$: 나머지는 버림

- $S^k X = (1-L)^k X = (UIU^T - ULU^T)^k X = U(1-\lambda)^k U^T X$
  - $if\ \lambda \approx 1, (1-\lambda)^k \approx 0$

SmoothGNN

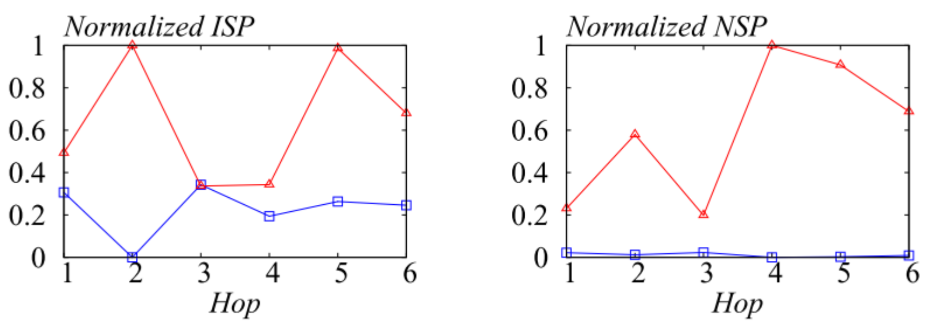
During propagation

- anomalous nodes exceed ISP, NSP of normal nodes at most hops.
- Anomalous nodes are harder to smooth than normal ones



(a) ISP with varying hops (b) NSP with varying hops

Figure 1: Smoothing Patterns of Amazon.



(a) ISP with varying hops (b) NSP with varying hops

Figure 2: Smoothing Patterns of T-Finance.

SmoothGNN

- Smoothing issue
  - Lead to indistinguishable node representation
  - the over-smoothing problem in node anomaly detection

  - However previous research has missed.

- Smoothing issue can provide potential advantage for detecting anomalies
  - ✓ Unsupervised method face effectiveness and efficiency issue
    - Shallow model limited expressiveness
    - Recon models and self-sup model are unlikely to be used in real application
      - Due to high computation complexity

SmoothGNN

- Individual Smoothing pattern (ISP)
  - $I(x) = ||(P^t - P^\infty)x||_2^2$
    - P^t : propagation matix after t hop
    - P^∞ : converged state
      - Node representation finally converge to a stable state
    - I(x) : T hop propagation - Converged state

- Neighborhood Smoothing Patterns(NSP)
  - ✓ $N(x^t) = \sum_{i,j=1}^{n} a_{i,j} \left\|\frac{x_i^t}{\sqrt{d_i+1}} - \frac{x_j^t}{\sqrt{d_j+1}}\right\|_2^2$
    - $a_{i,j}$: (I,j)-th entry of the adjacency matrix
    - d_i : the degree of node I
    - $x^t = P^t x$
    - NSP measures the similarities between neighboring nodes

SmoothGNN

Algorithm 2: SmoothGNN

Input: $X, T, n, [P^0, \cdots, P^T], [B^0, \cdots, B^T]$

Output: H^{SCSLC}, H^{SCGNN}

for $t = 0$ *to* T **do**

  $\tilde{X}_t \leftarrow \sigma(\text{MLP}(X))$;

$\alpha \leftarrow \sigma(\text{MLP}(\text{CAT}(SC(P^0\tilde{X}_0), \cdots, SC(P^T\tilde{X}_T))))$;

for $i = 0$ *to* n **do**

  $h_i^{SLC} \leftarrow \text{MLP}(\text{CAT}((B^0\tilde{X}_0)_i, \cdots, (B^T\tilde{X}_T))_i)$;

  $h_i^{GNN} \leftarrow \text{MLP}(\text{CAT}((g(\tilde{X}_0)_0)_i, \cdots, (g(\tilde{X}_T)_T)_i)$;

  $h_i^{SCSLC} \leftarrow h_i^{SLC} * \alpha$;

  $h_i^{SCGNN} \leftarrow h_i^{GNN} * \alpha$;

$H^{SCSLC} \leftarrow [h_1^{SCSLC}, \cdots, h_n^{SCSLC}]$;

$H^{SCGNN} \leftarrow [h_1^{SCGNN}, \cdots, h_n^{SCGNN}]$;

Return H^{SCSLC}, H^{SCGNN};

$SC = \text{NSP}$

α = MLP(SC(ISP$_{\text{matrix}}$))

$\alpha\, H_{\text{ISP}}^{\text{matix}}$,$\alpha\, H_{\text{ISP}}^{\text{GNN}}$

$\alpha\, H_{ISP}^{GNN} \rightarrow$ feature reconstruction

$\alpha\, H_{ISP}^{matix} \rightarrow$ regularization

$$L = L_{con} + L_{smooth}$$

SmoothGNN

4.2 Smoothing aware Learning Component

- h_i^{SLC} =MLP(CONCAT($(B^0\,\tilde{X}_0)_{,i}$ …, $(B^T\,\tilde{X}_T)_i$)

  - ✓ $B^t = P^t - P^\infty$.
  - ✓ $\tilde{X}_T = after\, t_{th}$ feature transform with MLP
  - ✓ $B^0\,\tilde{X}_0 = ||X^t - P^\infty X||$

- h_i^{SLC}
  - ✓ i-th node in SLC
  - ✓ 모든 hop에서 abnormal 정보(high frequency 정보)를 CONCAT
  - ✓ ISP 값을 explicitly 학습

SmoothGNN

4.3 Smoothing-aware Spectral GNN

- $h_i^{GNN} = \mathrm{MLP}\left(\mathrm{Concat}\left((g(\tilde{X}_0)_0)_i, \ldots, (g(\tilde{X}_T)_T)_i\right)\right)$
  - ✓ $g(X)_T = \sum_{t=0}^{T} \theta_t L^t X$
    - Theorem2 graph spectral space can capture node properties for NAD.
    - 모든 hop에서 ISP값을 concat
      - To fuse the spectral node representation obtained from each propagation hop
    - Implicty 하게 ISP 값을 학습

- h_i^{SLC}
  - ✓ i-th node in SLC
  - ✓ 모든 hop에서 abnormal 정보(high frequency 정보)를 CONCAT
  - ✓ ISP 값을 explicitly 학습

SmoothGNN

4.4 Smoothing-aware Coefficient

- Coefficient for node representations
- $\alpha = \sigma\left(\mathrm{MLP}\left(\mathrm{Concat}\left(\left(\mathrm{SC}(\mathrm{P}^0\tilde{X}_0), \ldots, \mathrm{SC}(\mathrm{P}^T\,\tilde{X}_T)\right)\right)\right)\right) \in \mathbb{R}^D$
  - ✓ $\mathrm{P}^{\mathrm{k}}\tilde{X}_k \in \mathbb{R}^D$
  - ✓ $SC(X) = diag\left(\frac{x^T Lx}{x^T x}\right) \in \mathbb{R}^D$
  - ✓ $\mathrm{Concat}(\cdot) \in \mathbb{R}^{D(T+1)}$
  - ✓ $\alpha \in \mathbb{R}^d$
- $h_i^{SCSLC} = h_i^{SLC} * \alpha\,, h_i^{SCGNN} = h_i^{GNN} * \alpha$
  - ✓ Final representation generated by SLC and SSGNN with SC

4.5 Smoothing-aware Measure

- $L = L_{con} + L_{smooth}$
  - ✓ $L_{smooth} = \frac{1}{n}\sum_{i=1}^{n} f_{smooth}\left(h_i^{SCSLC}\right) \in \mathbb{R}^1$
    - $f_{smooth}\left(h_i^{SCSLC}\right) = \sigma(AVG\left(h_i^{SCSLC}\right) \in \mathbb{R}^1$
      - Regularize Smoothness
  - ✓ $L_{con} = \frac{1}{n}\left|\left|h_i^{SCGNN} - x_i\right|\right|$
    - x_i : i_{th} row of feature matrix
    - Feature reconstruction effective measure for node anomaly detection

SmoothGNN

5. Results

Table 2: AUC and Precision (%) on 9 datasets, where "-" represents failed experiments due to memory constraint. The best result on each dataset is highlighted in boldface.

		Shallow		Reconstruction		Self-supervised					Special			
Datasets	Metrics	RADAR	ANOMALOUS	CLAD	GADNR	NLGAD	GRADATE	PREM	ARISE	TAM	RAND	VGOD	REC	SmoothGNN
Reddit	AUC	0.4372	0.4481	0.5784	0.5532	0.5380	0.5261	0.5518	0.5273	0.5729	0.5417	0.4931	0.5510	**0.5946**
	AP	0.0273	0.0309	**0.0502**	0.0373	0.0415	0.0393	0.0413	0.0402	0.0425	0.0356	0.0324	0.0421	0.0438
Tolokers	AUC	0.3625	0.3706	0.4061	0.5768	0.4825	0.5373	0.5654	0.5514	0.4699	0.4377	0.4988	0.4314	**0.6870**
	AP	0.1713	0.1731	0.1921	0.2991	0.2025	0.2364	0.2590	0.2505	0.1963	0.1939	0.2212	0.1946	**0.3517**
Amazon	AUC	0.2318	0.2318	0.2026	0.2608	0.5425	0.4781	0.2782	0.4782	0.8028	0.3585	0.5182	0.5869	**0.8408**
	AP	0.0439	0.0439	0.0401	0.0424	0.0991	0.0634	0.0744	0.0677	0.3322	0.0492	0.0779	0.1349	**0.3953**
T-Finance	AUC	0.2824	0.2824	0.1385	0.5798	0.5231	0.4063	0.4484	0.4667	0.6901	0.4380	0.4814	0.5239	**0.7556**
	AP	0.0295	0.0295	0.0247	0.0542	0.0726	0.0376	0.0391	0.0393	0.1284	0.0403	0.0454	0.0454	**0.1408**
YelpChi	AUC	0.5261	0.5272	0.4755	0.4704	0.4981	0.4920	0.4900	0.4834	0.5487	0.5052	0.4878	0.5134	**0.5758**
	AP	0.1822	0.1700	0.1284	0.1395	0.1469	0.1447	0.1378	0.1415	0.1733	0.1470	0.1345	0.1623	**0.1823**
Questions	AUC	0.4963	0.4965	0.6207	0.5875	0.5428	0.5539	0.6033	0.6241	0.5042	0.6164	0.5075	0.4988	**0.6444**
	AP	0.0279	0.0279	0.0512	0.0577	0.0348	0.0350	0.0430	**0.0619**	0.0395	0.0442	0.0299	0.0279	0.0592
Elliptic	AUC	-	-	0.4192	0.4001	0.4977	-	0.4978	-	-	-	0.5723	**0.5848**	0.5729
	AP	-	-	0.0807	0.0778	0.1009	-	0.0905	-	-	-	0.1256	**0.1337**	0.1161
DGraph-Fin	AUC	-	-	-	-	-	-	-	-	-	-	0.5456	0.4710	**0.6499**
	AP	-	-	-	-	-	-	-	-	-	-	0.0148	0.0112	**0.0199**
T-Social	AUC	-	-	-	-	-	-	-	-	-	-	0.5999	0.0793	**0.7034**
	AP	-	-	-	-	-	-	-	-	-	-	0.0351	0.0157	**0.0631**

Table 4: Ablation study.

Datasets	Reddit		Tolokers		Amazon		T-Finance		YelpChi		Questions		Elliptic		DGraph-Fin		T-Social	
Metrics	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
SmoothGNN	0.5946	0.0438	0.6870	0.3517	0.8408	0.3953	0.7556	0.1408	0.5758	0.1823	0.6444	0.0592	0.5729	0.1161	0.6499	0.0199	0.7034	0.0631
w/o SC	0.5437	0.0356	0.6115	0.2967	0.5131	0.0645	0.2869	0.0292	0.5715	0.1770	0.6260	0.0630	0.5596	0.1076	0.5868	0.0161	0.6639	0.0622
w/o L_{con}	0.5801	0.0494	0.6645	0.3168	0.8106	0.3031	0.7311	0.0858	0.5608	0.1719	0.6335	0.0506	0.5655	0.1145	0.6189	0.0181	0.6715	0.0514

SmoothGNN

4.1 theoretical Analysis of Smoothing pattern

THEOREM 1. *Let* $P = \frac{I_n + A}{2}$ *denote the propagation matrix given the adjacency matrix* $\tilde{\mathbf{A}}$. *For an augmented propagation matrix* $\mathbf{B}^t = (P - P^{\infty})^t$, *where* P^{∞} *represents the converged status of* $\mathbf{P}$, *we can derive* $\mathbf{B}^t = P^t - P^{\infty}$ *with* (i, j)*-th entry*

$$\mathbf{B}_{i,j} = \frac{(2m+n)(\mathbb{I}[i=j]\sqrt{d_i+1} + 2a_{ij}) - 2(d_i+1)\sqrt{d_j+1}}{2\sqrt{d_i+1}(2m+n)},$$

where $\mathbb{I}[\cdot]$ *is the indicator function,* $a_{i,j}$ *is the* (i, j)*-th entry of the adjacency matrix,* d_i *is the degree of node i, and m, n represent the number of edges and nodes, respectively.*

- Graph signal x propagates on propagated matrix B
  - Node representation becomes aware of node features and local information
  - **Edge connection** and **degree of neighbors**

SmoothGNN

4.1 theoretical Analysis of Smoothing pattern

THEOREM 2. *The augmented propagation matrix* $\mathbf{B}$ *after* t *hops of propagation can be expressed by* $\boldsymbol{b}^t = \sum_{k=0}^{t} \tilde{\boldsymbol{\theta}}_k \mathbf{L}^k \boldsymbol{u}\boldsymbol{v}$, *where* $\boldsymbol{b}^t$ *is a column vector of* $\mathbf{B}^t$, $\tilde{\boldsymbol{\theta}}_k \in \mathbb{R}^n$ *is the spectral filter coefficients, and* $\boldsymbol{u}, \boldsymbol{v}$ *represent the linear combinations of the eigenvectors of* $\tilde{\mathbf{A}}$ *and* $\mathbf{L}$, *respectively.*

$$\boldsymbol{b}^t = \sum_{t=0}^{T} \tilde{\boldsymbol{\theta}}_t \mathbf{L}^t \boldsymbol{u}\boldsymbol{v},$$

Propagation matrix -> adjacency -> Laplacian -> convolution 으로 학습 가능

SmoothGNN

4.1 theoretical Analysis of Smoothing pattern

DEFINITION 2 ([27]). *For any GNN, we call it suffers from ϵ-smoothing if and only if after T hops of propagation, the resulting feature matrix $\mathbf{H}^t$ at hop $t \geq T$ has a distance no larger than ϵ with respect to a subspace S, namely, $d_S(\mathbf{H}^t) \leq \epsilon, \forall t \geq T$, where $d_S(\mathbf{H}^t) := min_{\mathbf{M} \in S} ||\mathbf{H}^t - \mathbf{M}||_F$ represents the Frobenius norm from $\mathbf{H}^t$ to the subspace S.*

THEOREM 4. *Given the subspace S with threshold ϵ, a GNN model will suffer from ϵ-smoothing issue when the propagation hop $t = \left\lceil \frac{\log(\epsilon / d_S(\mathbf{X}))}{\log(\tau\lambda)} \right\rceil$, where τ is the largest singular value of the graph filters over all layers, λ is the second largest eigenvalue of the propagation matrix, and $\mathbf{X}$ is the feature matrix of graph G.*

CONTENTS

Raleigh Quotient Graph Neural Networks for graph-level anomaly detection

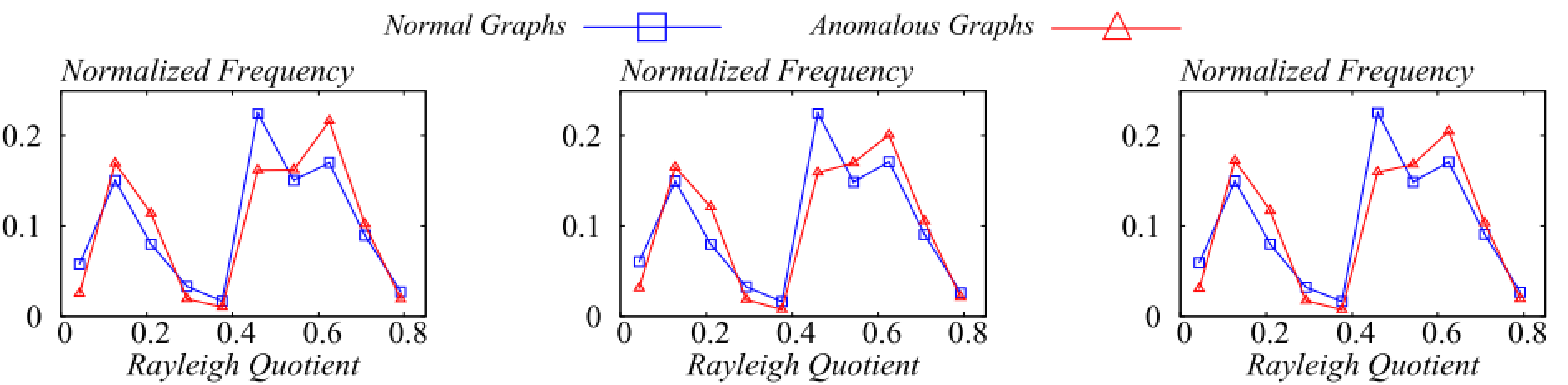
International Conference on Learning Representations 2024

Xiangyu Dong[1)], Xingyi Zhang[1)], Sibo Wang[1)]

1) The Chinese University of Hong Kong

RQGNN

- However existing methods fail to capture the spectral properties of graph anomalies
- Re-investigate the spectral difference between anomalous and normal graphs
  - ✓ Shows disparity in accumulated spectral energy between two classes.
  - ✓ Accumulated spectral energy can be represented by its Rayleigh Quotient
  - ✓ Rayleigh Quotient is a driving factor behind the anomalous properties of graphs

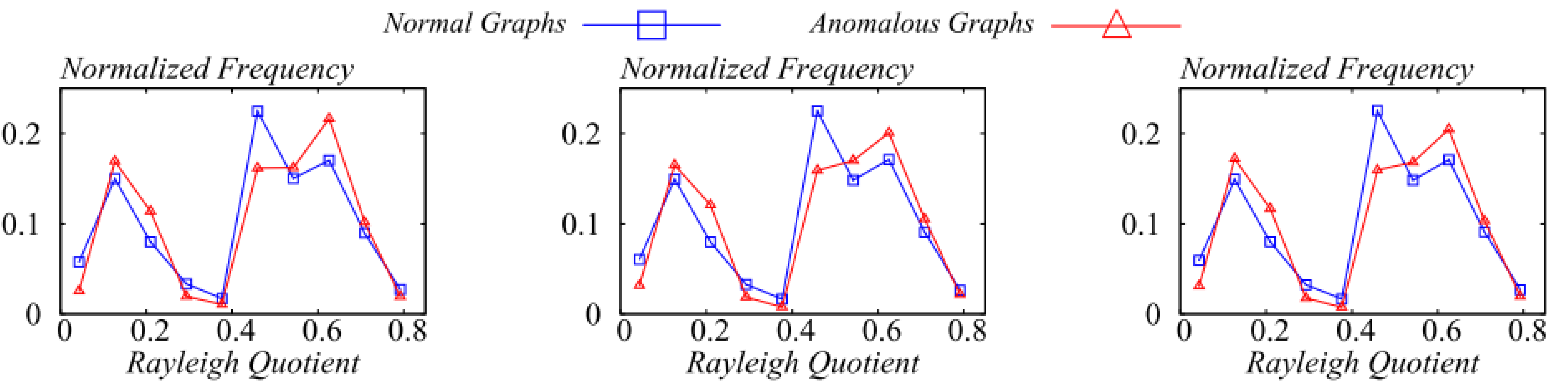


(a) $n_a = 488, n_n = 9512$ (b) $n_a = 977, n_n = 19024$ (c) $n_a = 1955, n_n = 38049$

Figure 1: Normalized Rayleigh Quotient distribution on SN12C.

RQGNN

- Regardless of the variation in the sample size
- RQ distribution of each class exhibits a consistent pattern across different sample size,
  - ✓ Can reveal the underlying difference between normal and anomalous graphs
  - ✓ RQ should be encoded and explored when identifying anomalous graphs
  - ✓ Although RQ is defined at the graph level, it can be used to measure spectral imbalance.



(a) $n_a = 488, n_n = 9512$ (b) $n_a = 977, n_n = 19024$ (c) $n_a = 1955, n_n = 38049$

Figure 1: Normalized Rayleigh Quotient distribution on SN12C.

RQGNN

3.2 RAYLEIGH QUOTIENT LEARNING COMPONENT

- $RQ(X, L) = diag\left(\frac{\tilde{x}^T L \tilde{x}}{\tilde{x}^T \tilde{x}}\right) \in \mathbb{R}^D$
  - ✓ $\tilde{x}$: node features after the feature transformation

- $H_{RQ}^G = \mathrm{MLP}\left(\mathrm{RQ(X, L)}\right) \in \mathbb{R}^D$
  - ✓ Two-layer MLP to get RQ representation

3.3 CHEBYSHEV WAVELET GNN with RQ-POOling

- Graph wavelet requires decomposing the graph Laplacian
- Employ Chebyshev polynomial to solve the problem.

✓ Lemma1. *There always a convergent Cheyshev series for any function* $f(t)$:

$$f(t) = \frac{1}{2}c_0 + \sum_{k}^{\infty} c_k T_{k(t)}$$

- *where* $c_k = \frac{2}{\pi}\int_0^{\pi} \cos(k\theta) f(\cos(\theta)) d\theta$,
- k : *order of chebyshev polynomials*
- $T_k(t) = 2tT_{k-1} - T_{k-2}(t)$,
- $T_0(t) = 1, T_1(t) = t$

RQGNN

3.3 CHEBYSHEV WAVELET GNN with RQ-POOling

- $h_j = \mathrm{CONCAT}\Big((f_1(L)\tilde{X}), (f_2(L)\tilde{X}), \ldots (f_q(L)\tilde{X}), \Big)$
  - ✓ the result of q graph wavelet
  - ✓ CWGNN
  - ψ : graph wavelet
    - $W = (W_{\psi 1}, W_{\psi 2}, \ldots, W_{\psi q})$
    - $W_{\psi i} = U g_i(\Lambda) U^T$
      - $g_i(\cdot)$: kernel function on $[0, \lambda_n]$
      - $Wx = [W_{\psi 1}, W_{\psi 2}, \ldots, W_{\psi q}]x = [U g_1(\Lambda) U^T x, U g_2(\Lambda) U^T x, \ldots, U g_q(\Lambda) U^T x]$
- RQ-pooling
  - ✓ $h_{Att}^G = \sigma(\sum_{j \in V} a_j h_j)$
    - $a_j = RQ(X, L) h_j$
- $h^G = MLP(CONCAT(h_{ATT}^G, h_{RQ}^G))$
  - ✓ Final representation

RQGNN

3.3 CHEBYSHEV WAVELET GNN with RQ-POOling

- $f_i(L) = \frac{1}{2}\bar{c}_{i,0}I_n + \sum_{k=1}^{iK} \bar{c}_{i,k}\bar{T}_k(L)$

  - Eigen value domain
    - $\bar{T}_k = \frac{4}{\lambda_n}(\boldsymbol{L} - \boldsymbol{I})\bar{T}_{k-1}(L) - \bar{T}_{k-2}(L)$
      - $t = \tilde{\mathrm{L}} = \frac{2}{\lambda_n} L - I$

        $\because$ *Laplacian eigne value* $\in [0, \lambda_n] \rightarrow [-1,1]$
    - $\bar{T}_0(L) = I_n, \bar{T}_0(L) = tI_n$
  - Wavelets filter domain
    - K: hop information
    - $\bar{c}_{i,k} = \frac{2}{\pi}\int_0^{\pi}\cos(k\theta)\, f\left(s_i\left(\frac{\lambda_n(\cos(\theta)+1)}{2}\right)\right) d\theta \; with\; 1 \le i \le q$
      - $\because$ $\cos(\theta) \in [-1,1] \Rightarrow \lambda_n \frac{\cos(\theta)+1}{2} \in [0, \lambda_n]$,s_i :wavelet filter

3.4 CLASS-BALACNED FOCAL LOSS

- $L_{CB_{focal}} = \frac{L_{focal}}{\eta(n_y)} = \frac{1-\beta}{1-\beta^{n_Y}} \sum_i^C (1-p_i)^\gamma \log(p_i)$
  - ✓ To tackle the imbalanced nature, introduce a re-weighting based on focal loss
  - ✓ Proposition 2. The expected number $\eta(n_t) = \frac{1-\beta^{n_t}}{1-\beta}, where\ \beta = \frac{N-1}{N}$ with N = # data point in class t

5. Results

Table 1: AUC and Macro-F1 scores (%) on ten datasets with random split.

		Spectral GNN		Graph Classification			Graph-level Anomaly Detection							
Datasets	Metrics	ChebyNet	BernNet	GMT	Gmixup	TVGNN	OCGIN	OCGTL	GLocalKD	HimNet	iGAD	RQGNN-1	RQGNN-2	RQGNN
MCF-7	AUC	0.6612	0.6172	0.7706	0.6954	0.7180	0.5348	0.5866	0.6363	0.6369	0.8146	0.8094	0.8346	**0.8354**
	F1	0.4780	0.4784	0.4784	0.4779	0.5594	-	-	-	-	0.6468	0.6626	0.7205	**0.7394**
MOLT-4	AUC	0.6647	0.6144	0.7606	0.6232	0.7159	0.5299	0.6191	0.6631	0.6633	0.8086	0.8246	0.8196	**0.8316**
	F1	0.4854	0.4794	0.4814	0.4789	0.4916	-	-	-	-	0.6617	0.7119	0.7113	**0.7240**
PC-3	AUC	0.6051	0.6094	0.7896	0.6908	0.7974	0.5810	0.6349	0.6727	0.6703	0.8723	0.8553	0.8671	**0.8782**
	F1	0.4853	0.4853	0.4853	0.4852	0.6206	-	-	-	-	0.6697	0.7003	**0.7241**	0.7184
SW-620	AUC	0.6759	0.6072	0.7467	0.6479	0.7326	0.4955	0.6398	0.6542	0.6544	0.8512	0.8401	0.8427	**0.8550**
	F1	0.4898	0.4847	0.4874	0.4844	0.5365	-	-	-	-	0.6627	0.6941	0.7209	**0.7335**
NCI-H23	AUC	0.6728	0.6114	0.8030	0.7324	0.7782	0.4948	0.6122	0.6837	0.6814	0.8297	0.8413	0.8554	**0.8680**
	F1	0.4930	0.4869	0.4869	0.4869	0.5520	-	-	-	-	0.6646	0.6735	**0.7349**	0.7214
OVCAR-8	AUC	0.6303	0.5850	0.7692	0.5663	0.7653	0.5298	0.6007	0.6750	0.6757	0.8691	0.8549	0.8650	**0.8799**
	F1	0.4900	0.4868	0.4868	0.4869	0.5406	-	-	-	-	0.6638	0.6876	0.7077	**0.7215**
P388	AUC	0.7266	0.6707	0.8495	0.6516	0.7957	0.5252	0.6501	0.6445	0.6667	0.8995	0.8911	0.8904	**0.9023**
	F1	0.5635	0.5001	0.6583	0.4856	0.5557	-	-	-	-	0.7437	0.7552	0.7738	**0.7963**
SF-295	AUC	0.6650	0.6353	0.7992	0.6471	0.7346	0.4774	0.6440	0.7069	0.7073	0.8770	0.8691	0.8781	**0.8825**
	F1	0.4871	0.4871	0.4871	0.4866	0.4935	-	-	-	-	0.6919	0.7120	0.7335	**0.7416**
SN12C	AUC	0.6598	0.6014	0.7919	0.7211	0.7441	0.5004	0.5617	0.6880	0.6916	0.8747	0.8851	**0.8904**	0.8861
	F1	0.4972	0.4874	0.4874	0.4871	0.5437	-	-	-	-	0.6714	0.7204	0.7549	**0.7660**
UACC257	AUC	0.6584	0.6115	0.7735	0.6564	0.7410	0.5411	0.6148	0.6647	0.6659	0.8512	0.8447	0.8596	**0.8724**
	F1	0.4894	0.4895	0.4894	0.4904	0.5373	-	-	-	-	0.6483	0.6889	0.7087	**0.7362**

RQGNN -2 : w/o RQL

RQGNN -1 : replace RQ-pooling with average pooling

CONTENTS

Towards Anomaly-Aware Pre-Training and Fine-Tuning for Graph Anomaly Detection

International Conference on Learning Representations 2026

Yunhui Liu[1], Jiashun Cheng[2], Yiqing Lin[3], Qizhuo Xie[1], Jia Li[2], Fugee Tsung[2], Hongzhi Yin[4], Tao Zheng[1], Tao Zheng Zjao[1], Tieke He[1]

1) Nanjing University
2) HKUST HKUST(GZ)
3) Tsinghua University
4) The University of Queensland

APF

- Label scarcity
- Homophily disparity at node and class level

- Utilizing RQ sampler and spectral filter for pretraining stage
    - ✓ Capture for anomaly signals without labels

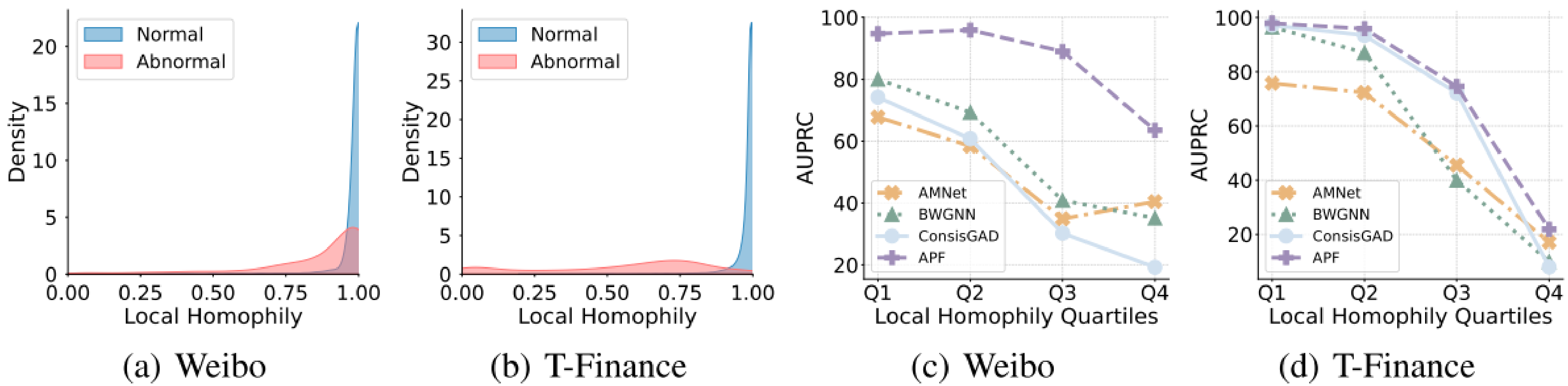


Figure 1: (a), (b): Distribution of local homophily for Weibo and T-Finance. (c), (d): Performance across local homophily quartiles (Q1 = top 25%, Q4 = bottom 25%) on Weibo and T-Finance.

3.1 RAYLEIGH QUOTIENT LEARNING COMPONENT

- Capturing high-frequency component is essential for heterophilic pattern
- We complement the conventional low-pass encoder
  - ✓ to better capture anomaly cues

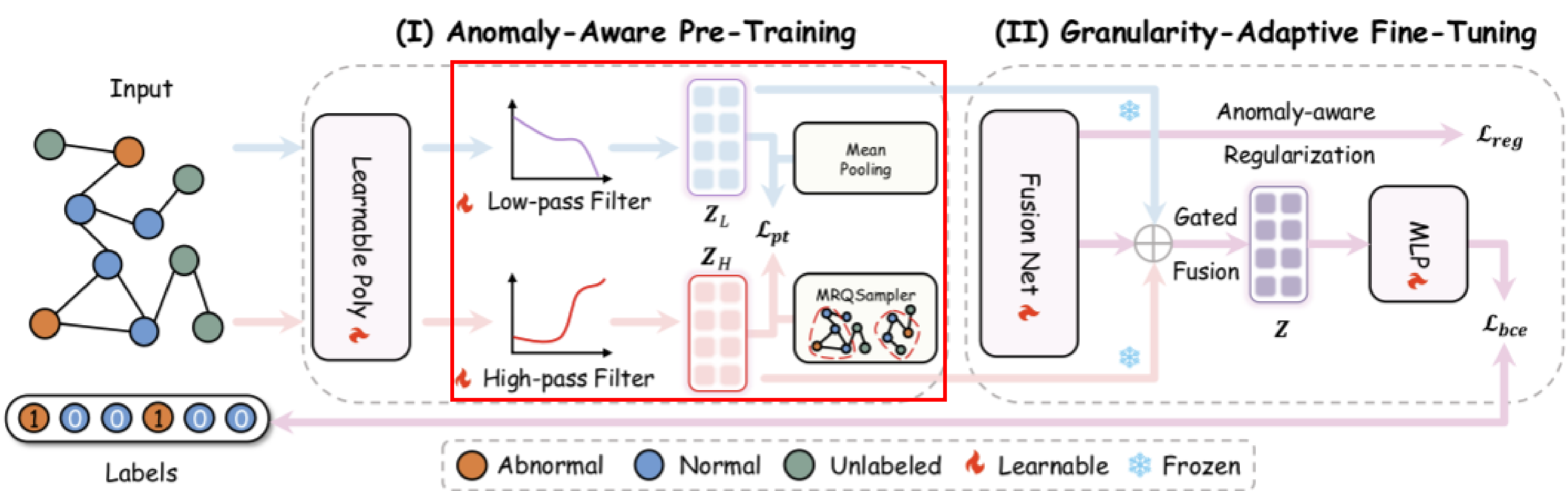


Figure 2: Overview of our proposed APF.

APF

3.1 RAYLEIGH QUOTIENT LEARNING COMPONENT

- Capturing high-frequency component is essential for heterophilic pattern
- We complement the conventional low-pass encoder
  - ✓ to better capture anomaly cues

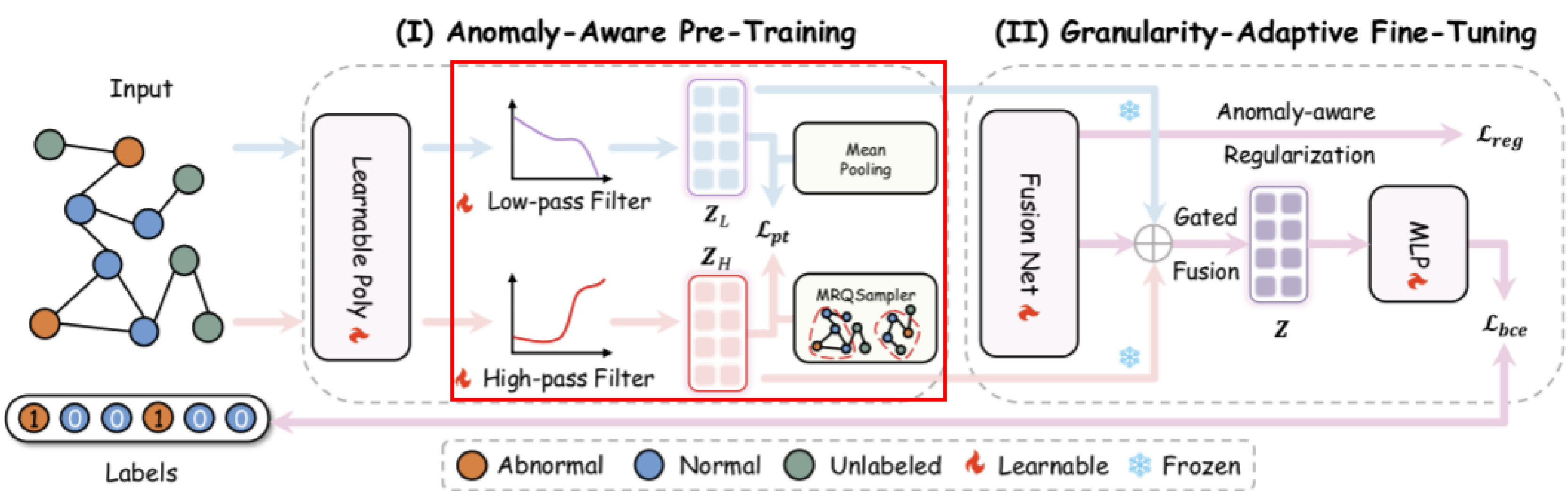


Figure 2: Overview of our proposed APF.

3.1 RAYLEIGH QUOTIENT LEARNING COMPONENT

- Facilitate flexible spectral encoding K-order Chebyshev polynomial
  - $g_L(\hat{L}) = \sum_{k=0}^{K} w_k^L T_k(\hat{L}),\ g_H(\hat{L}) = \sum_{k=0}^{K} w_k^H T_k(\hat{L}),$
  - $T_0(x) = 1, T_1(x) = x, T_k(x) = 2tT_{k-1} - T_{k-2}(x),$
  - $t_i = \cos\left(\frac{\frac{i+1}{2}}{K+1}\pi\right), i = 0, \ldots, K$

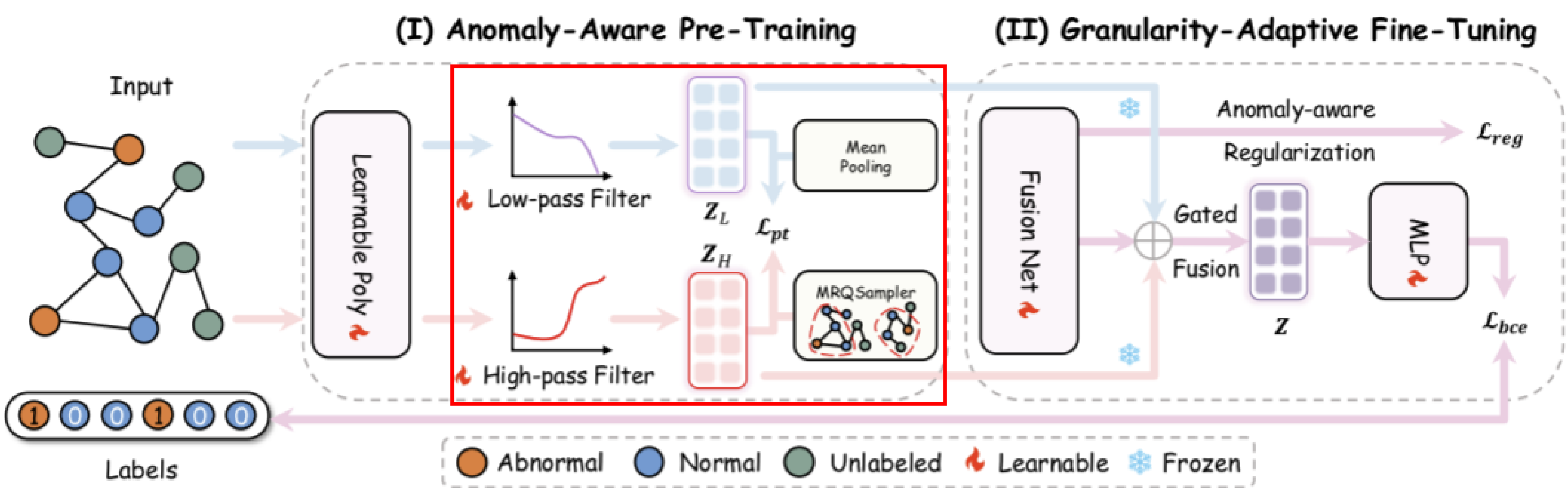
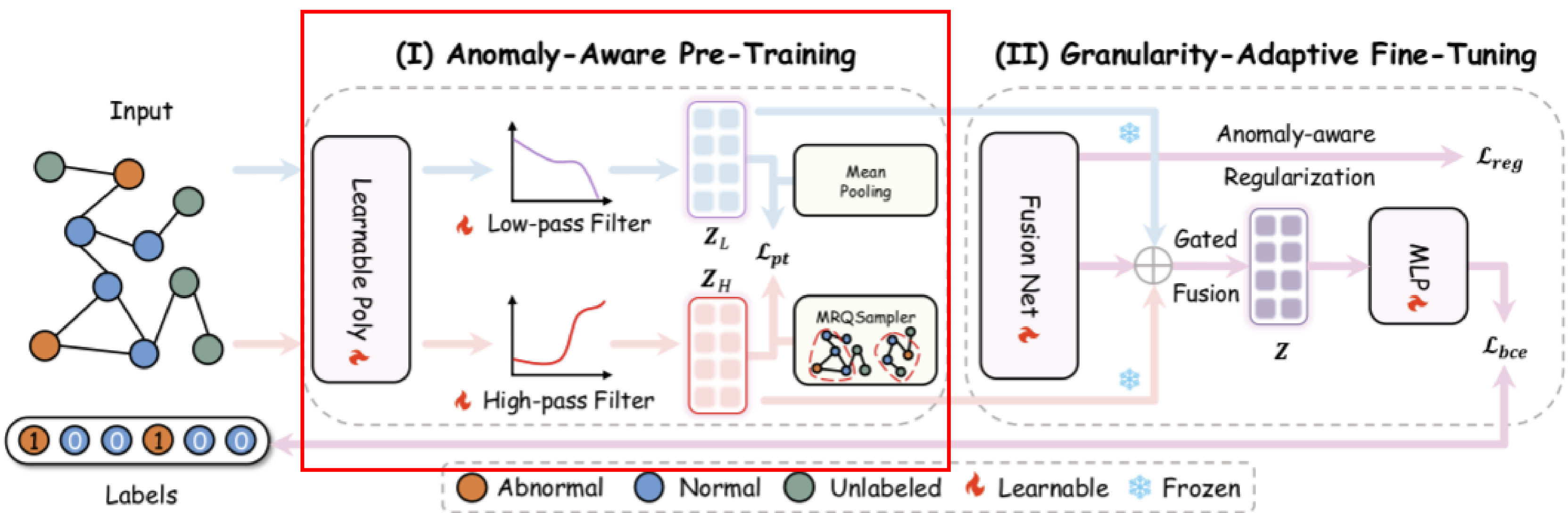


Figure 2: Overview of our proposed APF.

APF

3.1 RAYLEIGH QUOTIENT LEARNING COMPONENT

- $Z_L = f_{\theta_L}(g_L(\hat{L})X), Z_L = f_{\theta_H}(g_H(\hat{L})X)$
- $\gamma_k^L = \gamma_0 - \sum_j^k \gamma_j, \gamma_k^H = \sum_j^k \gamma_j$
  - $\gamma = (\gamma_0, \dots \gamma_M)$: shared learnable parameter
  - $\gamma_0 = \gamma_0^H = \gamma_0^L$
    - $\gamma_i^H \leq \gamma_{i+1}^H, \gamma_i^L \leq \gamma_{i+1}^L$
    - Guarantee the high/low pass property for encoders



3.1 RAYLEIGH QUOTIENT LEARNING COMPONENT

- ✓ To capture subtle anomaly cues
  - Maximizing mutual information
    - Between node and Rayleigh Quotient subtree computed from high-pass encoders
    - $L_{pt} = -\frac{1}{n}\sum_i^n \left(\log\left(Z_i^L, S^L\right) + \log\left(1 - D\left(\tilde{Z}_i^L, s^L\right)\right)\right)$
    - $-\frac{1}{n}\sum_i^n \left(\log\left(Z_i^H, s_i^H\right) + \log\left(1 - D\left(\tilde{Z}_i^H, s_i^H\right)\right)\right)$
      - » $\tilde{Z}$: *negative samples generated from randomly shuffled inputs*.
      - » S^L : *mean global summary of nodes in graph*
      - » S_i^H : *mean summary of nodes in subtree*
        - • Using MRQ sampler : heterophilic pattern subtree
      - » $D(z, s) = \sigma(z^T W s)$

3.2 Granularity-Adaptive Fine-Tuning

- Node-adaptive fusion
  - ✓ Different feature dimension contribute unequally to downstream
  - ✓ $Z = C \odot Z_L + (1 - C) \odot Z_H$
    - ▪ Z_L : Taks-agnostic sematic knowledge representation
    - ▪ Z_H : Node-specific structural disparities representation
    - ▪ $C \in [0,1]^{n \times e}$
      - – Free parameters
      - – $C_{i,j} \approx 1 : Z = Z_L$
    - ▪ But to excessive overhead ($O\ (n \times e\ )$)
      - – Inefficient learning under sparse supervision
  - ✓ $C = \sigma(XW_c + b_C)$
    - ▪ $W_C \in \mathbb{R}^{d \times e}$: learnable matrix
    - ▪ b_C : bias term

3.2 Granularity-Adaptive Fine-Tuning

- Anomaly-aware Regularization Loss
  - ✓ To indicate Class-level local homophily
    - Abnormal nodes tend to camouflage by connecting to normal nodes
  - ✓ $L_{reg} = -\frac{1}{|\mathcal{V}^L|}\sum_{v\in\mathcal{V}^L, y_i=1} p^a log c_i + (1-p^a) log(1-c_i))$
    $-\frac{1}{|\mathcal{V}^L|}\sum_{v\in\mathcal{V}^L, y_i=0} p^a log c_i + (1-p^n) log(1-c_i))$
    - $c_i = \frac{1}{e}\sum_{j=1}^{e} C_{ij}$: average fusion weight of node v_i **towards generic knowledge(low pass)**
    - $p^a, p^n \in [0,1]$, with $p^a \le p^n$

    - Normal nodes : $c_i \approx p_n$: low pass representation
    - Abnormal nodes : $c_i \approx p_a$: high pass representation
      - **reflect class level disparity to induce the class label**

APF

3.2 Granularity-Adaptive Fine-Tuning

- Overall finetuning
  - ✓ $L_{ft} = L_{bce} + L_{reg}$
  - ✓ $L_{bce} = -\frac{1}{|\mathcal{V}^L|}(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$

Results

Table 1: Comparison of AUPRC for each model. "-" denotes "out of memory". The best and runner-up models are **bolded** and underlined.

Model	Reddit	Weibo	Amazon	Yelp.	T-Fin.	Ellip.	Tolo.	Quest.	DGraph.	T-Social	Avg.
GCN	4.2±0.8	86.0±6.7	32.8±1.2	16.4±2.6	60.5±10.8	43.1±4.6	33.0±3.6	6.1±0.9	2.3±0.2	8.4±3.8	29.3
GIN	4.3±0.6	67.6±7.4	75.4±4.3	23.7±5.4	44.8±7.1	40.1±3.2	31.8±3.2	6.7±1.1	2.0±0.1	6.2±1.7	30.3
GAT	4.7±0.7	73.3±7.3	81.6±1.7	25.0±2.9	28.9±8.6	44.2±6.6	33.0±2.0	7.3±1.2	2.2±0.2	9.2±2.0	30.9
ACM	4.4±0.7	66.0±8.7	54.0±19.0	21.4±2.7	29.2±16.8	63.1±4.8	34.4±3.5	7.2±1.9	2.2±0.4	6.0±1.6	28.8
FAGCN	4.7±0.7	70.1±10.6	77.0±2.3	22.5±2.6	39.8±27.2	43.6±10.6	35.0±4.3	7.3±1.4	2.0±0.3	-	-
AdaGNN	4.9±0.8	28.3±2.8	75.7±6.3	22.7±2.1	23.3±7.6	39.2±7.9	32.2±3.9	5.3±0.9	2.1±0.3	4.8±1.1	23.9
BernNet	4.9±0.3	66.6±5.5	81.2±2.4	23.9±2.7	51.8±12.4	40.0±4.1	28.9±3.5	6.7±2.1	2.5±0.2	4.2±1.2	31.1
GAS	4.7±0.7	65.7±8.4	80.7±1.7	21.7±3.3	45.7±13.4	46.0±4.9	31.7±3.0	6.3±2.0	2.5±0.2	8.6±2.4	31.4
DCI	4.3±0.4	76.2±4.3	72.5±7.9	24.0±4.8	51.0±7.2	43.4±4.9	32.1±4.2	6.1±1.3	2.0±0.2	7.4±2.5	31.9
PCGNN	3.4±0.5	69.3±9.7	81.9±1.9	25.0±3.5	58.1±11.3	40.3±6.6	33.9±1.7	6.4±1.8	2.4±0.4	8.0±1.6	32.9
AMNet	4.9±0.4	67.1±5.1	82.4±2.2	23.9±3.5	60.2±8.2	33.3±4.8	28.6±1.5	7.4±1.4	2.2±0.3	3.1±0.3	31.3
BWGNN	4.2±0.7	80.6±4.7	81.7±2.2	23.7±2.9	60.9±13.8	43.4±5.5	35.3±2.2	6.5±1.7	2.1±0.3	15.9±6.2	35.4
GHRN	4.2±0.6	77.0±6.2	80.7±1.7	23.8±2.8	63.4±10.4	44.2±5.7	35.9±2.0	6.5±1.7	2.3±0.3	16.2±4.6	35.4
ConsisGAD	4.5±0.5	64.6±5.5	78.7±5.7	25.9±2.9	79.7±4.7	47.8±8.2	33.7±2.7	7.9±2.4	2.0±0.2	41.3±5.0	38.6
SpaceGNN	4.6±0.5	79.2±2.8	81.1±2.3	25.7±2.4	81.0±3.5	44.1±3.5	33.8±2.5	7.4±1.6	2.0±0.3	59.0±5.7	41.8
XGBGraph	4.1±0.5	75.9±6.2	**84.4±1.1**	24.8±3.1	78.3±3.1	**77.2±3.2**	34.1±2.8	7.7±2.1	1.9±0.2	40.6±7.6	42.9
DGI	4.8±0.6	90.8±2.5	46.5±3.7	17.0±1.2	75.0±4.9	45.9±2.5	39.7±0.8	6.4±1.2	2.1±0.2	37.8±6.1	36.6
GRACE	4.7±0.3	90.8±1.8	51.3±4.3	18.2±1.6	79.3±0.7	48.1±3.6	37.4±2.8	8.9±1.7	-	-	-
G-BT	5.0±0.7	87.5±3.9	38.7±2.2	18.8±1.6	76.8±1.8	45.2±4.4	37.9±2.8	9.1±1.9	2.6±0.3	42.2±7.4	36.4
GraphMAE	4.3±0.1	91.4±2.6	39.4±0.3	17.3±0.1	70.8±4.7	32.7±3.8	36.0±2.1	5.9±0.5	2.1±0.1	42.6±10.5	34.2
BGRL	5.3±0.3	93.6±1.9	43.9±4.6	19.2±1.6	61.7±5.1	47.4±6.0	38.3±3.3	8.4±2.0	2.0±0.2	46.7±8.4	36.6
SSGE	4.8±0.9	87.7±2.6	39.1±2.4	18.8±1.6	77.6±0.9	47.4±3.0	38.2±2.8	8.1±1.1	2.5±0.3	46.4±3.9	37.1
PolyGCL	5.2±0.8	87.3±2.1	79.7±6.6	24.3±2.5	43.3±6.4	50.0±5.2	33.0±1.8	5.7±0.8	2.2±0.3	40.6±7.0	37.1
BWDGI	4.5±0.6	72.5±2.7	79.4±5.7	26.8±2.7	80.0±2.1	44.9±6.5	38.5±3.1	5.0±0.7	2.4±0.2	38.0±5.2	39.2
APF (*w/o* $\mathcal{L}_{pt}$)	5.2±0.6	85.8±7.9	82.7±3.0	24.1±2.2	79.4±3.4	55.5±4.9	37.4±1.2	9.4±1.5	2.3±0.2	64.8±10.5	44.7
APF	**5.9±0.9**	**93.9±1.1**	83.8±2.9	**28.4±1.4**	**82.5±2.6**	67.7±3.4	**40.5±2.0**	**12.3±1.6**	**2.9±0.2**	**77.8±5.6**	**49.6**

Results

Table 2: Ablation study on each component of our APF.

Variants					YelpChi			Questions			DGraph-Fin		
g_L	g_H	$\mathcal{G}^{RQ}$	$\mathcal{L}_{reg}$	Fusion	AUPRC	AUROC	Rec@K	AUPRC	AUROC	Rec@K	AUPRC	AUROC	Rec@K
✓	✓	✓	✓	NDApt	**28.4±1.4**	**68.2±2.3**	**31.4±1.6**	**12.3±1.6**	**71.9±2.1**	**16.5±0.9**	**2.9±0.2**	**72.4±1.3**	**4.2±0.7**
✓	✓	✗	✓	NDApt	27.1±1.6	67.6±1.4	30.2±1.6	10.9±1.8	71.2±2.5	16.0±1.4	2.6±0.1	71.1±0.5	3.3±0.5
✓	✗	✗	✗	-	18.7±1.2	56.5±1.3	20.6±1.7	11.8±1.2	70.6±1.2	16.1±0.9	2.5±0.1	71.0±1.0	3.2±0.1
✗	✓	✓	✗	-	24.4±2.4	64.0±2.5	27.2±2.5	6.3±0.7	66.1±3.2	9.8±1.9	2.6±0.1	71.1±0.5	3.1±0.5
✓	✓	✓	✗	NDApt	27.5±1.6	67.4±2.3	30.6±1.9	11.1±1.6	71.2±2.7	15.9±0.9	2.8±0.1	71.7±0.4	3.7±0.2
✓	✓	✓	✗	Mean	27.6±1.6	67.2±2.1	30.5±1.8	10.8±1.7	71.0±2.4	15.7±1.2	2.8±0.1	71.6±0.4	3.8±0.2
✓	✓	✓	✗	Concat	27.3±1.9	67.3±2.2	30.4±2.1	11.1±1.5	68.5±3.2	15.2±1.7	2.8±0.0	71.5±0.2	3.6±0.3
✓	✓	✓	✗	Atten.	26.2±1.6	66.7±1.7	29.5±1.7	8.3±2.2	67.5±3.5	13.6±2.6	2.8±0.0	71.7±0.3	3.8±0.2

Thank you ☺

Q & A