

# 图神经网络导论

## 图异常检测

周晟

浙江大学 软件学院

2023.12



# 课程内容

- ① 异常检测任务
- ② 基于分类的异常检测
- ③ 基于自编码器的异常检测
- ④ 基于对比学习的图异常检测
- ⑤ 其他异常检测方法



# 数据中的异常

## 异常的定义

An **outlier** is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.

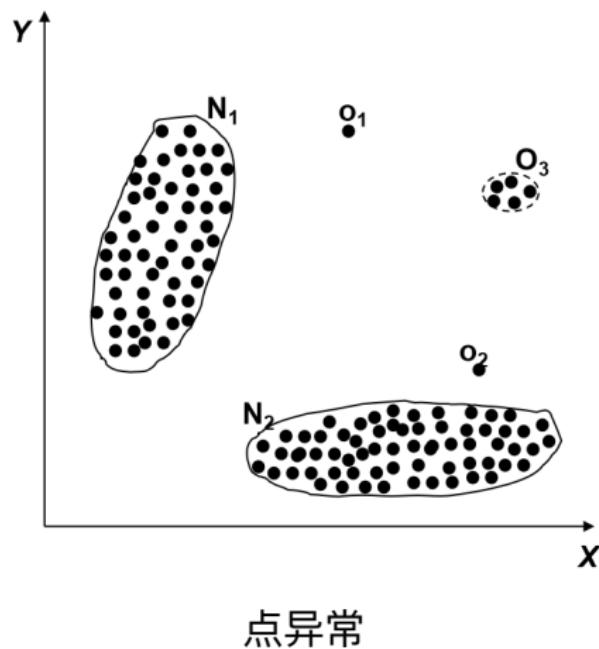
数据集中与大部分样本不一样的样本称为异常样本 (anomaly sample, abnormalities, deviants, outliers)。

## 异常检测的定义

离群点检测 (又称为异常检测) 是找出行为不同于预期的异常的过程。

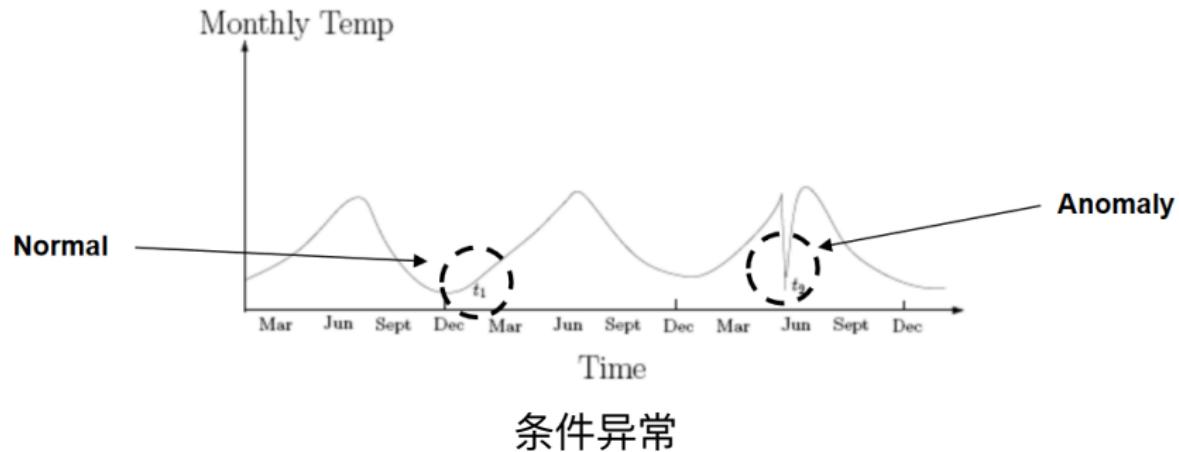
# 点异常

- 单个数据实例是异常的



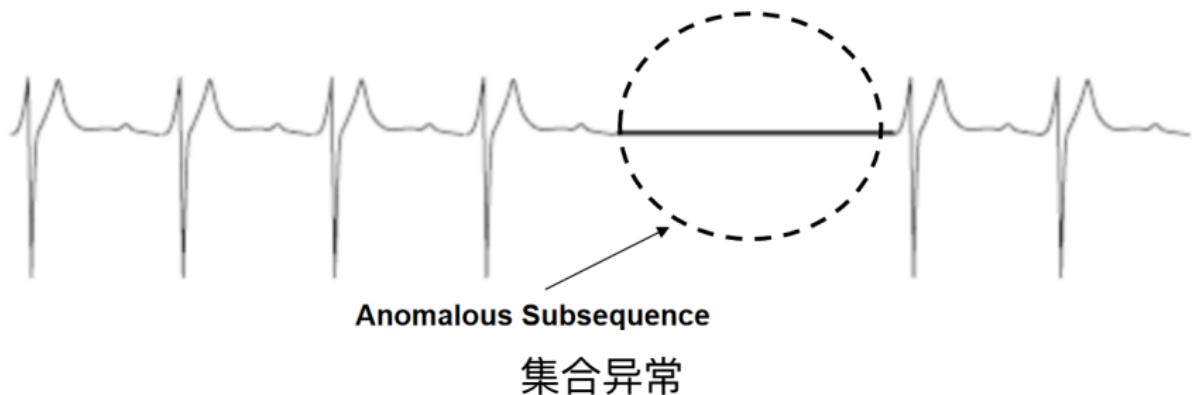
# 条件异常

- 单个数据实例在某个“条件下”是异常的
- 也称为“上下文异常 (Contextual Anomalies) ”

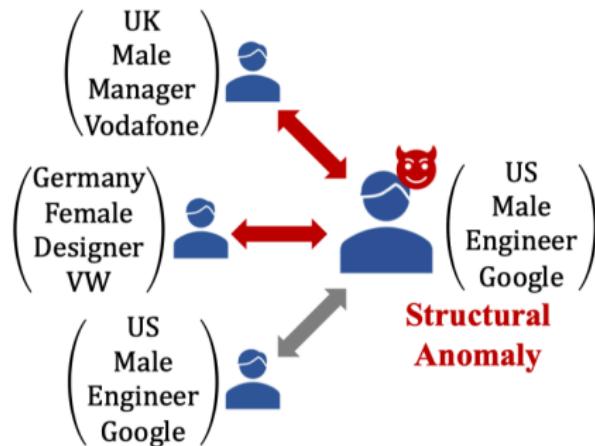


# 集合异常

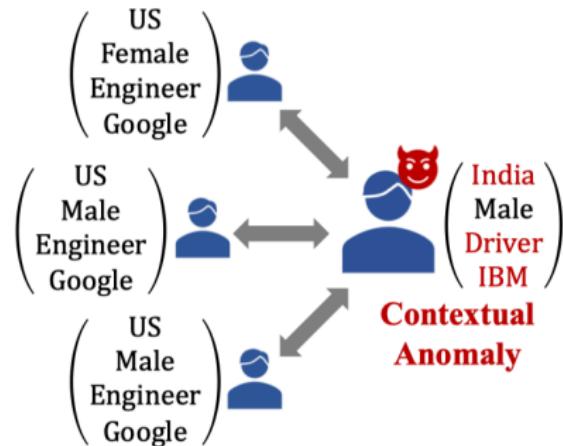
- 一个由相关数据实例构成的集合是异常的
- 数据实例间存在某种关系
  - 连续数据
  - 空间数据
  - 图数据
- 集合中的单个数据实例本身并不是异常



# 图上的异常



(a) Structural Anomaly



(b) Contextual Anomaly

## 图上的异常类型

# 不同异常类型的案例

- ① 超大金额的单笔交易（点异常）
- ② 发生在国外的小额交易（条件异常）
- ③ 发生在凌晨的小额交易（条件异常）
- ④ 连续十天购买同一件商品（集合异常）



真实大规模场景中往往同时包含多种类型的异常！



# 异常检测范式

## ① 有监督异常检测

- ① 标签获得困难
- ② 类别不平衡问题

## ② 半监督异常检测

- ① 数据分布偏移 (Distribution Shift)

## ③ 弱监督异常检测

- ① 异常标签噪声大
- ② 训练数据分布偏移 (Distribution Shift)

## ④ 无监督异常检测

- ① 缺少监督信号和数据分布信息
- ② 训练目标设计困难

Out-of-distribution (OOD) Detection



① 异常检测任务

② 基于分类的异常检测

③ 基于自编码器的异常检测

④ 基于对比学习的图异常检测

⑤ 其他异常检测方法



# 单分类问题

常见的分类方法：

- ① 单分类（只能定义正样本不能定义负样本）
- ② 二分类（邮件分类）
- ③ 多分类（图像分类）

## 单分类与二分类的区别

单分类问题中的训练样本只有一类，因此训练出的分类器将不属于该类的所有其他样本判别为“不是”即可，而不是由于属于另一类才返回“不是”的结果。

## 模型假设

寻找一个超平面将样本中的正例圈出来，预测就是用这个超平面做决策，在圈内的样本就认为是正样本。

- 无监督学习的方法，不需要训练集的标签。
- 如何在无监督场景下寻找划分的超平面以及寻找支持向量？
- SVDD(support vector domain description) 支持向量域描述

# One Class SVM

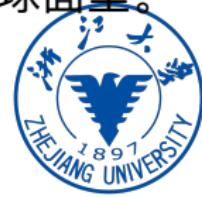
SVDD 的优化目标就是，求一个中心为  $a$ , 半径为  $R$  的最小球面：

$$F(R, a, \xi_i) = R^2 + C \sum_i \xi_i$$

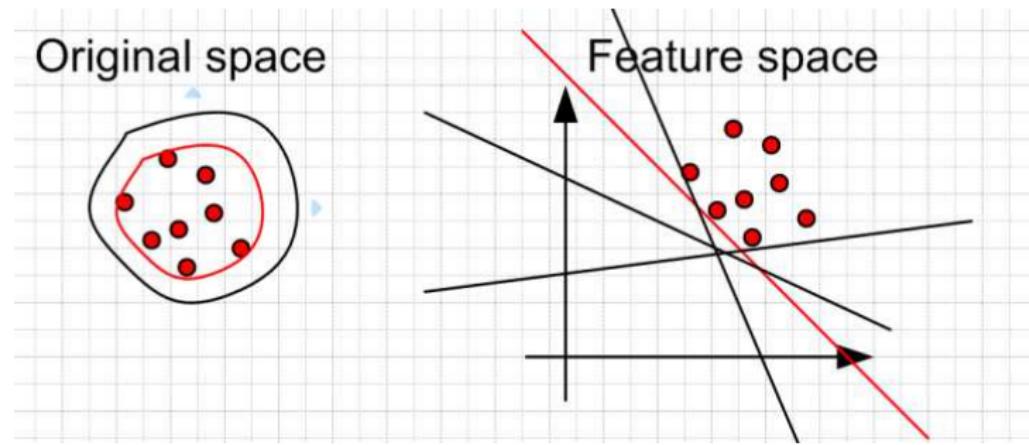
$\xi_i$  是球体的松弛变量，使得这个球面满足：

$$(x_i - a)^T (x_i - a) \leq R^2 + \xi_i \quad \forall i, \xi_i \geq 0$$

满足这个条件就是说要把 training set 中的数据点都包在球面里。



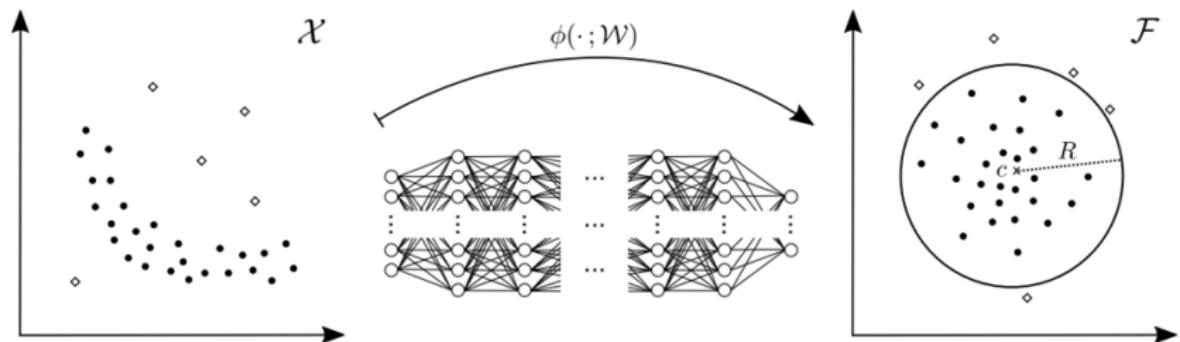
# One Class SVM



寻找超平面与特征空间中的零点距离最大，并且将零点与所有的数据点分隔开。

# Deep One-Class Classification(DeepSVDD)

采用深度学习的方法来实现传统的 One-Class SVM 算法



DeepSVDD[Ruff et al., 2018]<sup>1</sup>



<sup>1</sup>Deep One-Class Classification(ICML2018)

# DeepSVDD

将样本学习到表征空间，并且学习出一个尽可能小的超平面来包裹住所有的训练样本

$$\min_{R, \mathcal{W}} R^2 + \frac{1}{\nu n} \sum_{i=1}^n \max \left\{ 0, \|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|^2 - R^2 \right\} + \frac{\lambda}{2} \sum_{\ell=1}^L \|\mathbf{W}^\ell\|_F^2$$

- ① 第一项为超平面的半径
- ② 第二项对超平面外的点进行惩罚
- ③ 第三项利用正则化防止模型坍塌

这里允许训练样本中包含少量的异常样本， $\nu$  控制异常比例

# DeepSVDD

在实际场景中，训练集可以只包含正常样本，因此直接优化所有点到中心的距离

$$\min_{\mathcal{W}} \frac{1}{n} \sum_{i=1}^n \|\phi(\mathbf{x}_i; \mathcal{W}) - \mathbf{c}\|^2 + \frac{\lambda}{2} \sum_{\ell=1}^L \|\mathbf{W}^\ell\|_F^2$$

中心  $c$  的选择通常采用经验方法，对表征学习器随机初始化参数，取所有样本输出的均值作为中心，中心固定不变

异常分数：

$$s(\mathbf{x}) = \|\phi(\mathbf{x}; \mathcal{W}^*) - \mathbf{c}\|^2$$

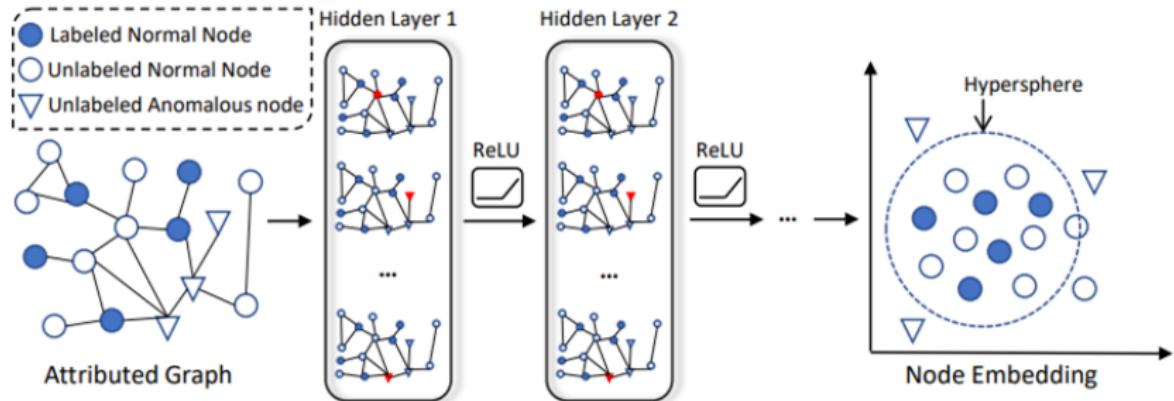


# DeepSVDD

Table 1. Average AUCs in % with StdDevs (over 10 seeds) per method and one-class experiment on MNIST and CIFAR-10.

NORMAL CLASS	OC-SVM/ SVDD	KDE	IF	DCAE	ANOGAN	SOFT-BOUND. DEEP SVDD	ONE-CLASS DEEP SVDD
0	<b>98.6</b> $\pm$ 0.0	97.1 $\pm$ 0.0	98.0 $\pm$ 0.3	97.6 $\pm$ 0.7	96.6 $\pm$ 1.3	97.8 $\pm$ 0.7	98.0 $\pm$ 0.7
1	99.5 $\pm$ 0.0	98.9 $\pm$ 0.0	97.3 $\pm$ 0.4	98.3 $\pm$ 0.6	99.2 $\pm$ 0.6	99.6 $\pm$ 0.1	<b>99.7</b> $\pm$ 0.1
2	82.5 $\pm$ 0.1	79.0 $\pm$ 0.0	88.6 $\pm$ 0.5	85.4 $\pm$ 2.4	85.0 $\pm$ 2.9	89.5 $\pm$ 1.2	<b>91.7</b> $\pm$ 0.8
3	88.1 $\pm$ 0.0	86.2 $\pm$ 0.0	89.9 $\pm$ 0.4	86.7 $\pm$ 0.9	88.7 $\pm$ 2.1	90.3 $\pm$ 2.1	<b>91.9</b> $\pm$ 1.5
4	<b>94.9</b> $\pm$ 0.0	87.9 $\pm$ 0.0	92.7 $\pm$ 0.6	86.5 $\pm$ 2.0	89.4 $\pm$ 1.3	93.8 $\pm$ 1.5	<b>94.9</b> $\pm$ 0.8
5	77.1 $\pm$ 0.0	73.8 $\pm$ 0.0	85.5 $\pm$ 0.8	78.2 $\pm$ 2.7	88.3 $\pm$ 2.9	85.8 $\pm$ 2.5	<b>88.5</b> $\pm$ 0.9
6	96.5 $\pm$ 0.0	87.6 $\pm$ 0.0	95.6 $\pm$ 0.3	94.6 $\pm$ 0.5	94.7 $\pm$ 2.7	98.0 $\pm$ 0.4	<b>98.3</b> $\pm$ 0.5
7	93.7 $\pm$ 0.0	91.4 $\pm$ 0.0	92.0 $\pm$ 0.4	92.3 $\pm$ 1.0	93.5 $\pm$ 1.8	92.7 $\pm$ 1.4	<b>94.6</b> $\pm$ 0.9
8	88.9 $\pm$ 0.0	79.2 $\pm$ 0.0	89.9 $\pm$ 0.4	86.5 $\pm$ 1.6	84.9 $\pm$ 2.1	92.9 $\pm$ 1.4	<b>93.9</b> $\pm$ 1.6
9	93.1 $\pm$ 0.0	88.2 $\pm$ 0.0	93.5 $\pm$ 0.3	90.4 $\pm$ 1.8	92.4 $\pm$ 1.1	94.9 $\pm$ 0.6	<b>96.5</b> $\pm$ 0.3
AIRPLANE	61.6 $\pm$ 0.9	61.2 $\pm$ 0.0	60.1 $\pm$ 0.7	59.1 $\pm$ 5.1	<b>67.1</b> $\pm$ 2.5	61.7 $\pm$ 4.2	61.7 $\pm$ 4.1
AUTOMOBILE	63.8 $\pm$ 0.6	64.0 $\pm$ 0.0	50.8 $\pm$ 0.6	57.4 $\pm$ 2.9	54.7 $\pm$ 3.4	64.8 $\pm$ 1.4	<b>65.9</b> $\pm$ 2.1
BIRD	50.0 $\pm$ 0.5	50.1 $\pm$ 0.0	49.2 $\pm$ 0.4	48.9 $\pm$ 2.4	<b>52.9</b> $\pm$ 3.0	49.5 $\pm$ 1.4	50.8 $\pm$ 0.8
CAT	55.9 $\pm$ 1.3	56.4 $\pm$ 0.0	55.1 $\pm$ 0.4	58.4 $\pm$ 1.2	54.5 $\pm$ 1.9	56.0 $\pm$ 1.1	<b>59.1</b> $\pm$ 1.4
DEER	66.0 $\pm$ 0.7	<b>66.2</b> $\pm$ 0.0	49.8 $\pm$ 0.4	54.0 $\pm$ 1.3	65.1 $\pm$ 3.2	59.1 $\pm$ 1.1	60.9 $\pm$ 1.1
DOG	62.4 $\pm$ 0.8	62.4 $\pm$ 0.0	58.5 $\pm$ 0.4	62.2 $\pm$ 1.8	60.3 $\pm$ 2.6	62.1 $\pm$ 2.4	<b>65.7</b> $\pm$ 2.5
FROG	74.7 $\pm$ 0.3	<b>74.9</b> $\pm$ 0.0	42.9 $\pm$ 0.6	51.2 $\pm$ 5.2	58.5 $\pm$ 1.4	67.8 $\pm$ 2.4	67.7 $\pm$ 2.6
HORSE	62.6 $\pm$ 0.6	62.6 $\pm$ 0.0	55.1 $\pm$ 0.7	58.6 $\pm$ 2.9	62.5 $\pm$ 0.8	65.2 $\pm$ 1.0	<b>67.3</b> $\pm$ 0.9
SHIP	74.9 $\pm$ 0.4	75.1 $\pm$ 0.0	74.2 $\pm$ 0.6	<b>76.8</b> $\pm$ 1.4	75.8 $\pm$ 4.1	75.6 $\pm$ 1.7	75.9 $\pm$ 1.2
TRUCK	75.9 $\pm$ 0.3	<b>76.0</b> $\pm$ 0.0	58.9 $\pm$ 0.7	67.3 $\pm$ 3.0	66.5 $\pm$ 2.8	71.0 $\pm$ 1.1	73.1 $\pm$ 1.2

# One-Class Graph Neural Networks for Anomaly Detection in Attributed Networks(OCGNN)



OCGNN<sup>2</sup>直接将 DeepSVDD 的表征学习器替换为 GNN

<sup>2</sup>One-Class Graph Neural Networks for Anomaly Detection in Attributed Networks(Neural Comput & Applic 2021)

# OCGNN

	Method	Cora	Citeseer	Pubmed
Raw Features	IForest	53.09 ± 0.03	46.33 ± 0.03	65.57 ± 0.02
	OCSVM	54.35 ± 0.02	57.05 ± 0.03	45.50 ± 0.01
	PCA	62.17 ± 0.01	58.10 ± 0.03	71.06 ± 0.01
	AE	62.17 ± 0.01	58.11 ± 0.03	71.05 ± 0.01
DeepWalk	IForest	57.87 ± 0.02	51.00 ± 0.03	60.73 ± 0.01
	OCSVM	52.10 ± 0.03	43.13 ± 0.02	60.22 ± 0.01
	PCA	55.90 ± 0.03	46.65 ± 0.02	61.66 ± 0.01
	AE	55.91 ± 0.03	46.42 ± 0.02	61.66 ± 0.01
DeepWalk+Raw Feat.	IForest	53.56 ± 0.04	45.55 ± 0.06	65.60 ± 0.02
	OCSVM	51.59 ± 0.03	42.95 ± 0.02	60.10 ± 0.01
	PCA	62.38 ± 0.02	57.96 ± 0.03	72.04 ± 0.01
	AE	62.39 ± 0.02	57.96 ± 0.03	71.91 ± 0.01
GAE based	GCN-AE	80.53 ± 0.05	59.52 ± 0.09	58.26 ± 0.02
	GAE [11]	60.15 ± 0.08	51.80 ± 0.03	54.27 ± 0.02
	Dom [7]	67.50 ± 0.25	62.44 ± 0.15	53.92 ± 0.04
Our OCGNNs	<b>OC-GCN</b>	73.25 ± 0.02	62.81 ± 0.01	54.53 ± 0.01
	<b>OC-GAT</b>	<b>88.19</b> ± 0.02	79.06 ± 0.03	60.98 ± 0.01
	<b>OC-SAGE</b>	86.97 ± 0.04	<b>85.62</b> ± 0.01	<b>74.72</b> ± 0.03

## OCGNN 实验效果



# Subtractive Aggregation for Attributed Network Anomaly Detection(CIKM2021)

## 研究动机

图上的异常节点往往呈现出与邻居不同的模式，这种局部的便宜可以被用于发现异常

使用节点的表征与邻居表征的差值作为节点的特征

$$\begin{aligned} z_i &= \phi(\mathbf{x}_i; \mathbf{W}), \\ \mathbf{h}_i &= \sigma(z_i - \text{AGGREGATE}(z_j, \forall j \in \mathcal{N}_i^k)) \end{aligned}$$

进而使用 DeepSVDD 学习超平面：

$$\min_{\Theta} \frac{1}{n} \sum_{i=1}^n \|\mathbf{h}_i - \mathbf{c}\|_2^2 + \frac{\lambda}{2} \|\Theta\|_F^2$$



两种具体的聚合邻居方式

Mean aggregator:

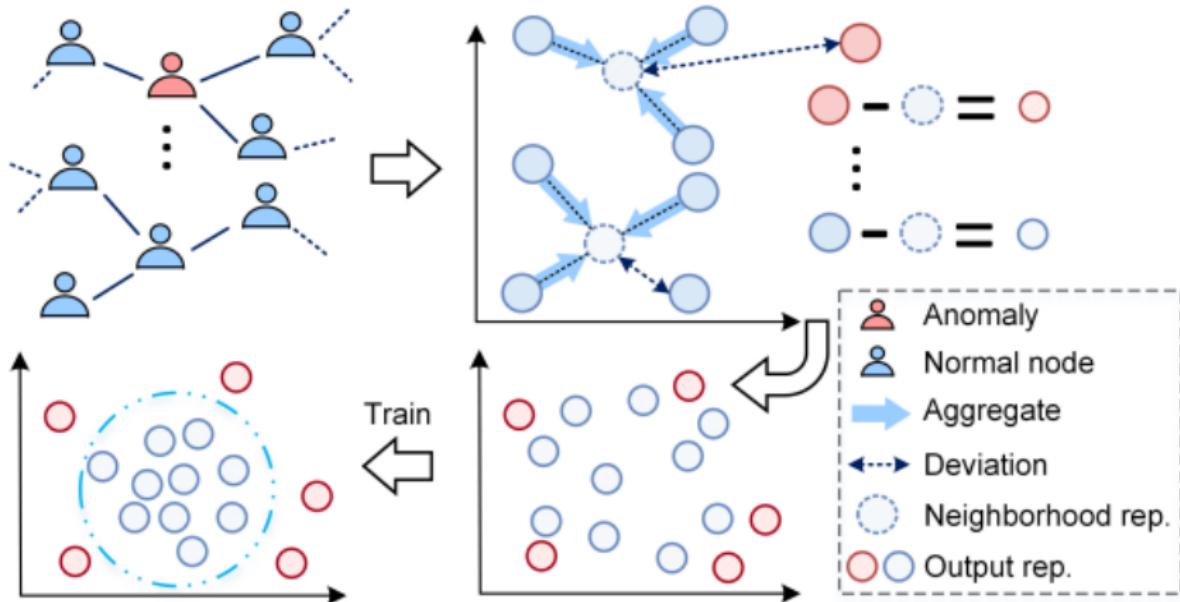
$$\mathbf{h}_i = \sigma \left( \mathbf{z}_i - \frac{1}{|\mathcal{N}_i^k|} \sum_{j \in \mathcal{N}_i^k} \mathbf{z}_j \right)$$

Attention aggregator:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{z}_i \oplus \mathbf{z}_j]))}{\sum_{j \in \mathcal{N}_i^k} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{z}_i \oplus \mathbf{z}_j]))}$$
$$\mathbf{h}_i = \sigma \left( \mathbf{z}_i - \sum_{j \in \mathcal{N}_i^k} \alpha_{ij} \mathbf{z}_j \right)$$



# AAGNN



- 1 异常检测任务
- 2 基于分类的异常检测
- 3 基于自编码器的异常检测
- 4 基于对比学习的图异常检测
- 5 其他异常检测方法



# 基于自编码器的异常检测

## 基本假设

自编码器的目标是最小化所有数据的重构损失，按照少数服从多数的原则，正常样本可以被更好地重构而异常样本则难以被完美重构。

## 模型结构

$$\begin{aligned}\mathbf{z} &= \phi_e(\mathbf{x}; \Theta_e), \hat{\mathbf{x}} = \phi_d(\mathbf{z}; \Theta_d) \\ \{\Theta_e^*, \Theta_d^*\} &= \arg \min_{\Theta_e, \Theta_d} \sum_{\mathbf{x} \in X} \|\mathbf{x} - \phi_d(\phi_e(\mathbf{x}; \Theta_e); \Theta_d)\|^2 \\ s_{\mathbf{x}} &= \|\mathbf{x} - \phi_d(\phi_e(\mathbf{x}; \Theta_e^*); \Theta_d^*)\|^2\end{aligned}$$



# 基于自编码器的异常检测

常用的 AutoEncoder 结构：

- ① Denoising AutoEncoder
- ② Sparse AutoEncoder
- ③ Contractive AutoEncoder
- ④ Variational AutoEncoder
- ⑤ Robust AutoEncoder
- ⑥ Masked AutoEncoder



# Outlier Detection with Robust Deep AutoEncoders(KDD 2017)

## 研究动机

- ① 自编码器在训练过程中容易受到异常样本的影响
- ② Robust Principal Component Analysis (RPCA) 也是一种降维方法，但是为异常样本做了专门的优化

RPCA 将数据矩阵  $X$  拆分为低秩的矩阵  $L$  和一个稀疏矩阵  $S$ :

$$X = L + S$$

矩阵分解的过程可以理解为如下的优化目标:

$$\begin{aligned} & \min_{L,S} \|L\|_* + \lambda \|S\|_1 \\ \text{s.t. } & \|X - L - S\|_F^2 = 0 \end{aligned}$$

# Robust Deep AutoEncoders(RDA)

$$X = L_D + S$$

$L_D$  是指能被 AutoEncoder 重构的特征， $S$  包含了难以被 AutoEncoder 重构的噪声和异常。

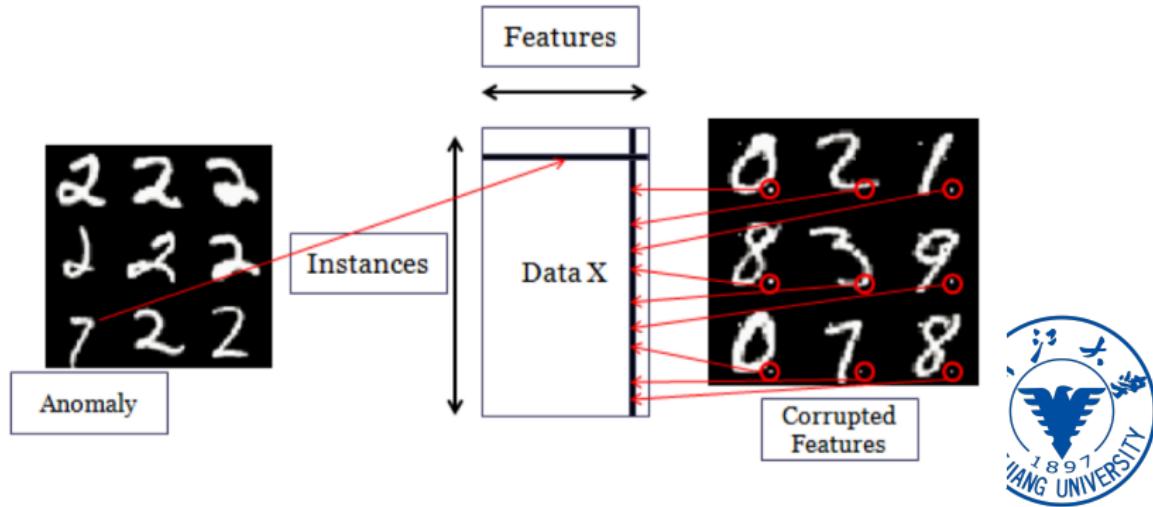
$$\begin{aligned} \min_{\theta} & \|L_D - D_{\theta}(E_{\theta}(L_D))\|_2 + \lambda \|S\|_1 \\ \text{s.t. } & X = L_D + S = 0 \end{aligned}$$



# Robust Deep AutoEncoders(RDA)

## Group Anomalies

- ① 许多样本共享一个相同的特征维度（系统噪声而不是异常）
- ② 一个样本中异常的特征应当相对确定



# Robust Deep AutoEncoders(RDA)

$\mathcal{L}_{2,1}$  norm

$\mathcal{L}_2$  norm 作用于所有特征维度,  $\mathcal{L}_1$  norm 作用于所有的样本

$$\|X\|_{2,1} = \sum_{j=1}^n \|x_j\|_2 = \sum_{j=1}^n \left( \sum_{i=1}^m |x_{ij}|^2 \right)^{1/2}$$

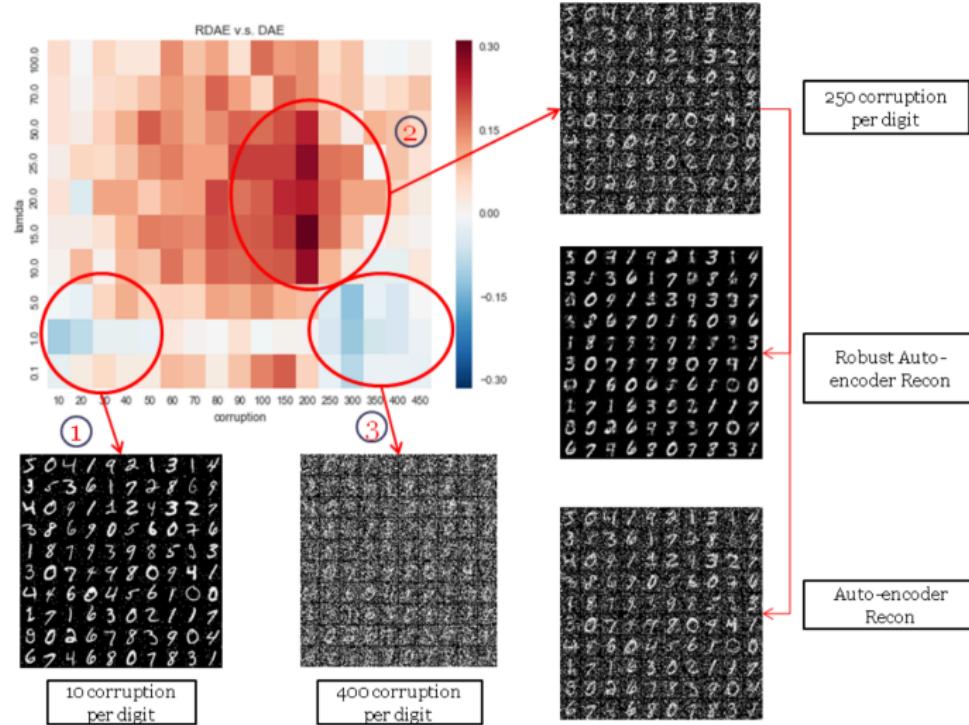
$$\min_{\theta, S} \|L_D - D_\theta(E_\theta(L_D))\|_2 + \lambda \|S\|_{2,1}$$

$$\min_{\theta, S} \|L_D - D_\theta(E_\theta(L_D))\|_2 + \lambda \|S^T\|_{2,1}$$

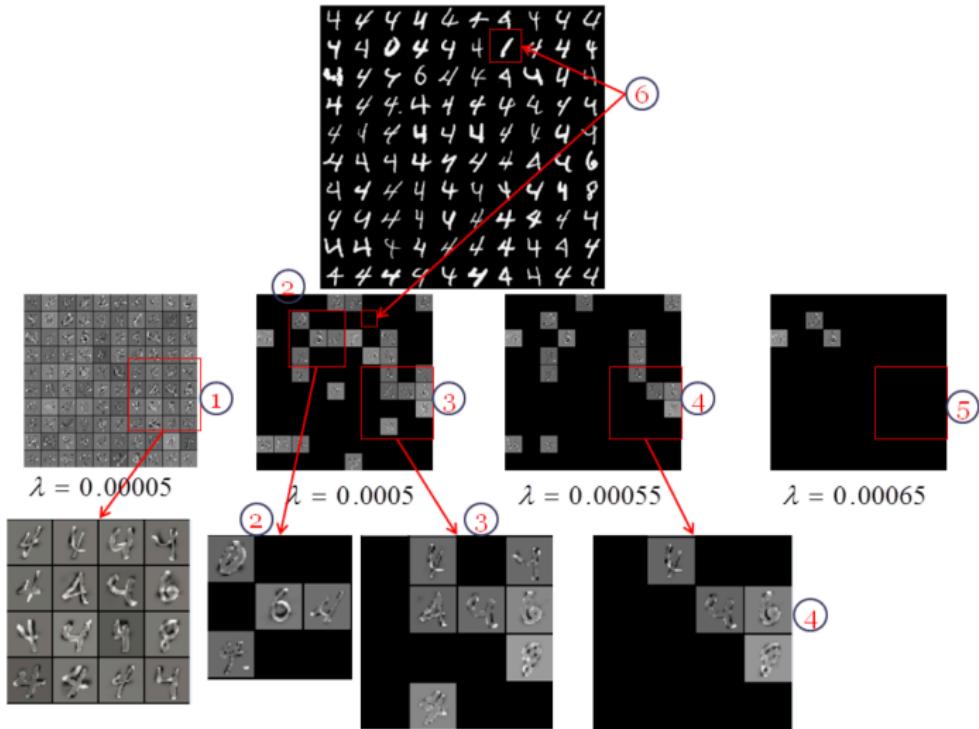
$$\text{s.t. } X - L_D - S = 0$$



# Robust Deep AutoEncoders(RDA)



# Robust Deep AutoEncoders(RDA)



# Deep Anomaly Detection on Attributed Networks(Dominant)

## 研究动机

使用图神经网络对结构和属性进行统一编码，进而提升异常捕获的能力。使用重构损失作为异常分值。

核心模块<sup>3</sup>:

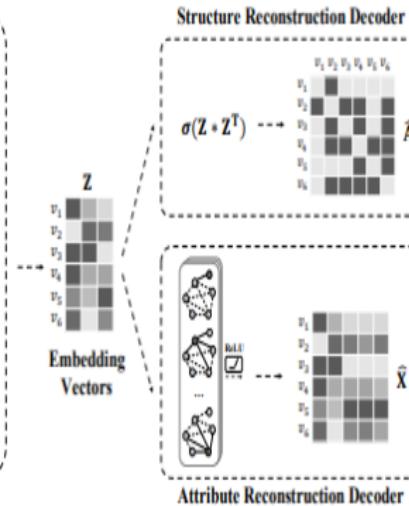
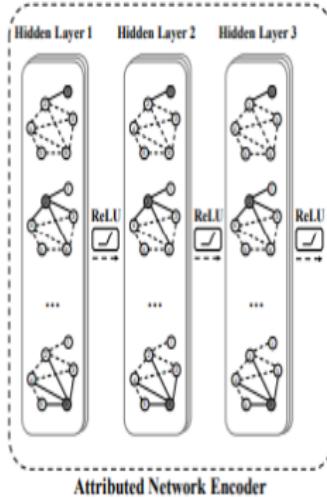
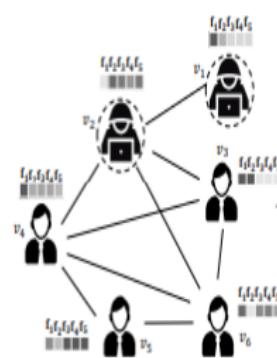
$$\hat{\mathbf{A}} = \text{sigmoid}(\mathbf{Z}\mathbf{Z}^T)$$

$$\hat{\mathbf{X}} = f_{\text{Relu}}(\mathbf{Z}, \mathbf{A} \mid \mathbf{W}^{(3)})$$

$$\begin{aligned}\mathcal{L} &= (1 - \alpha)\mathbf{R}_S + \alpha\mathbf{R}_A \\ &= (1 - \alpha)\|\mathbf{A} - \hat{\mathbf{A}}\|_F^2 + \alpha\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2,\end{aligned}$$

<sup>3</sup>Deep Anomaly Detection on Attributed Networks(SDM 2019)

# Dominant



Dominant 模型架构

# DOMINANT

Precision@K												
	BlogCatalog				Flickr				ACM			
$K$	50	100	200	300	50	100	200	300	50	100	200	300
LOF	0.300	0.220	0.180	0.183	0.420	0.380	0.270	0.237	0.060	0.060	0.045	0.037
Radar	0.660	0.670	0.550	0.416	0.740	0.700	0.635	0.503	0.560	0.580	0.520	0.430
ANOMALOUS	0.640	0.650	0.515	0.417	<b>0.790</b>	0.710	0.650	0.510	0.600	0.570	0.510	0.410
DOMINANT	<b>0.760</b>	<b>0.710</b>	<b>0.590</b>	<b>0.470</b>	0.770	<b>0.730</b>	<b>0.685</b>	<b>0.593</b>	<b>0.620</b>	<b>0.590</b>	<b>0.540</b>	<b>0.497</b>
Recall@K												
	BlogCatalog				Flickr				ACM			
$K$	50	100	200	300	50	100	200	300	50	100	200	300
LOF	0.050	0.073	0.120	0.183	0.047	0.084	0.120	0.158	0.005	0.010	0.015	0.018
Radar	0.110	0.223	0.367	0.416	0.082	0.156	0.282	0.336	0.047	0.097	0.173	0.215
ANOMALOUS	0.107	0.217	0.343	0.417	<b>0.087</b>	0.158	0.289	0.340	0.050	0.095	0.170	0.205
DOMINANT	<b>0.127</b>	<b>0.237</b>	<b>0.393</b>	<b>0.470</b>	0.084	<b>0.162</b>	<b>0.304</b>	<b>0.396</b>	<b>0.052</b>	<b>0.098</b>	<b>0.180</b>	<b>0.248</b>

Table 2: Performance of different anomaly detection methods w.r.t. precision@ $K$  and recall@ $K$ .

## Dominant 实验结果

# AnomalyDAE: Dual autoencoder for anomaly detection on attributed networks

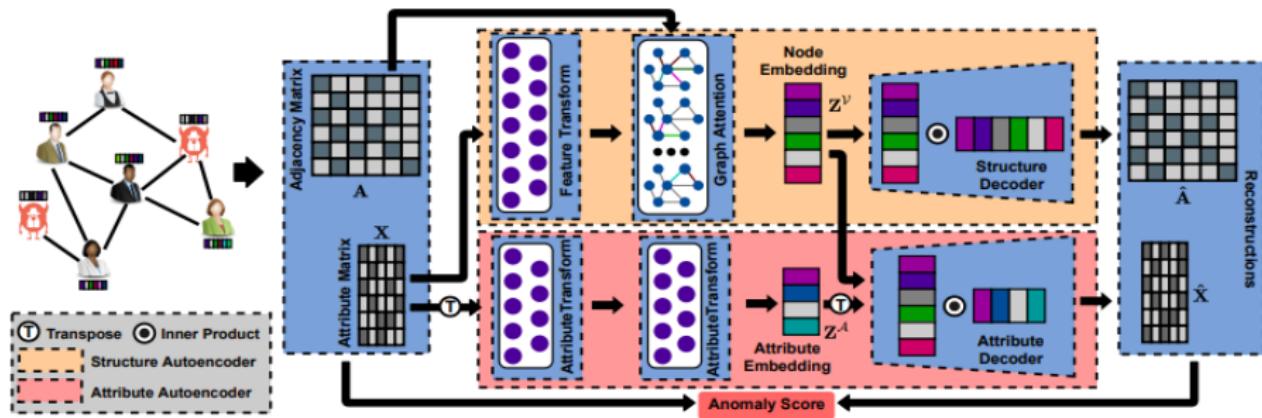


Fig. 1. The framework of the proposed AnomalyDAE.

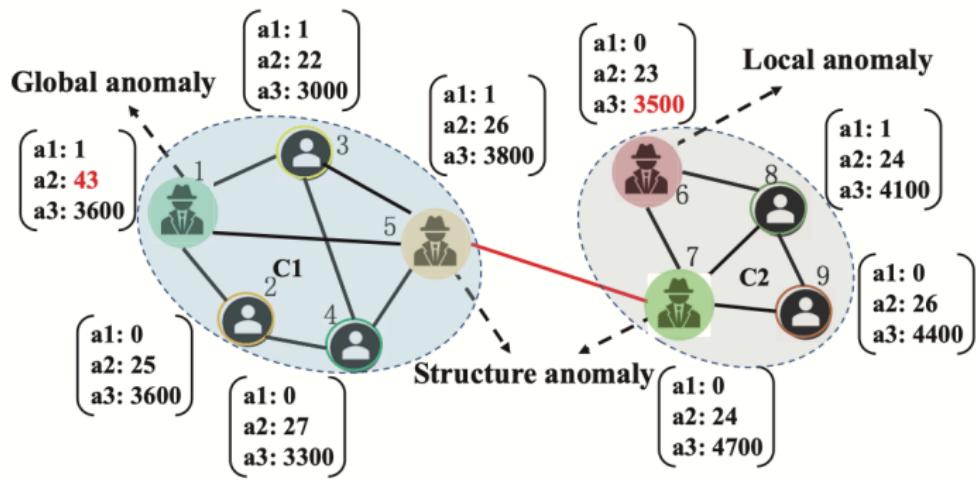
AnomalyDAE<sup>4</sup>与 Dominant 的区别在于使用了两个编码器

<sup>4</sup>AnomalyDAE: Dual autoencoder for anomaly detection on attributed networks(ICASSP)

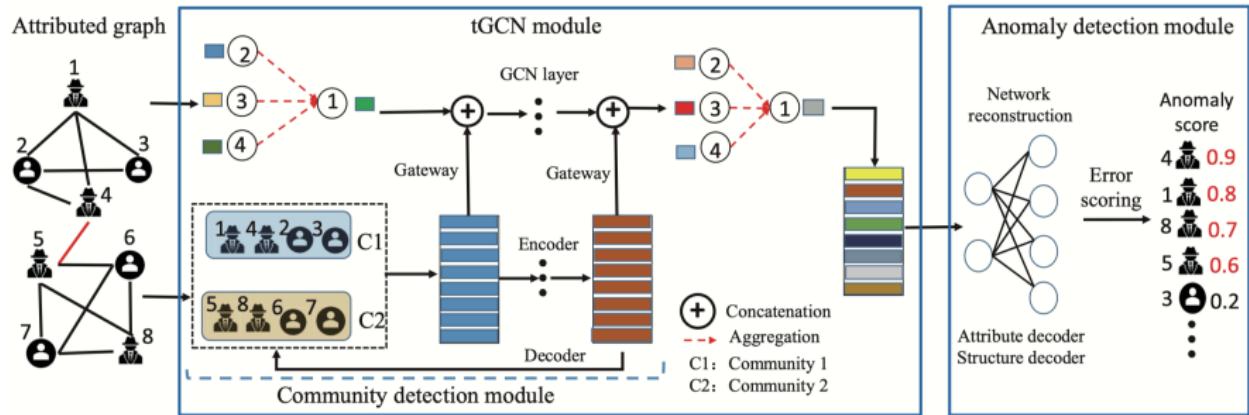
# ComGA: Community-Aware Attributed Graph Anomaly Detection(WSDM 2022)

## 研究动机

真实网络中的异常往往包含多种粒度的异常，而不同异常的表现形式往往不同。



# ComGA



社区发现：基于自编码器的模块度分解

$$L_{res} = \|\mathbf{B} - \hat{\mathbf{B}}\|_F^2 = \sum_{i=1}^n \left\| \mathbf{b}_i - \hat{\mathbf{b}}_i \right\|_2^2$$

社区信息与局部信息的融合：

$$\tilde{\mathbf{Z}}_{l-1} = \mathbf{Z}_{l-1} + \mathbf{H}_{l-1}$$

社区信息与局部信息的约束：

$$L_{gui} = KL(\mathbf{Z} || \mathbf{H})$$

整体优化目标：

$$L_{rec} = (1 - \alpha) \|\mathbf{A} - \hat{\mathbf{A}}\|_F^2 + \alpha \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2.$$

$$L = L_{res} + L_{gui} + L_{rec}$$

# SL-GAD: Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection

## Challenge

现有的数据挖掘和机器学习方法要么是浅层方法，无法有效捕捉图数据的复杂相互依赖性；要么是基于图的自编码器方法，未能充分利用上下文信息作为有效的异常检测监督信号。

## TL;DR

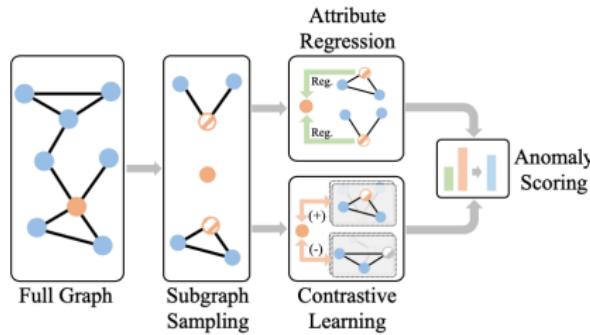
该论文通过结合生成属性重构和多视图对比学习，有效地在属性和结构空间中检测异常，其算法在六个数据集上的表现显著优于现有最先进的算法。

# SL-GAD

## Solution

该方法基于目标节点构建不同的上下文子图（视图），并采用两个模块：生成属性回归和多视图对比学习进行异常检测。

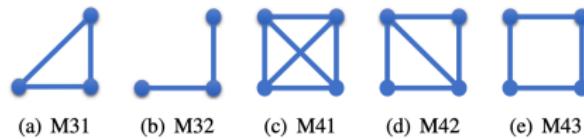
- 属性重构利用生成学习模式，识别在属性空间中与邻居不同的节点。
- 而多视图对比学习模块能够从多个子图中利用更丰富的结构信息，从而能够捕捉结构空间中的异常，以及结构和属性信息的混合异常。



# GUIDE: Higher-order Structure Based Anomaly Detection on Attributed Networks

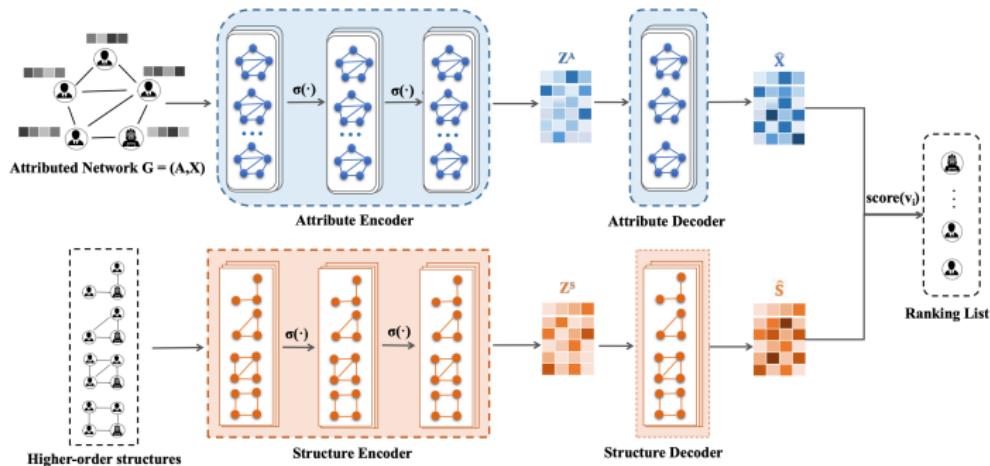
网络基序 (Network motifs) 是指在网络中频繁出现的特殊子图结构，有助于理解网络的关键结构。在学术网络中，例如三阶三角形基序可以描述三位学者的合作关系，展现多实体间复杂的互动模式。

GUIDE 主要分析三至四节点的网络基序，并运用节点基序度量方法来表示节点在高阶结构中的地位。



## Solution

利用属性自编码器和结构自编码器来重构节点属性和高阶结构。



此外，作者设计了一个图注意力层，通过评估节点与其邻居之间的高阶结构差异来确定邻居的重要性。

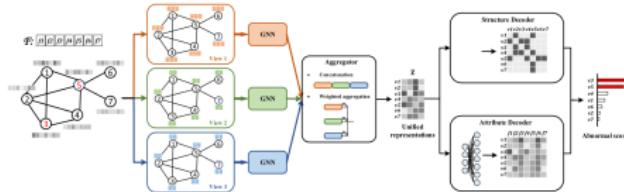
$$h_i^{(l+1)} = \sigma \left( W_1 h_i^{(l)} + \sum_{j \in N(i) \cup \{i\}} \alpha_{ij} W_2 h_j^{(l)} \right)$$

$$\alpha_{ij} = \frac{\exp \left( a^T W_2 (h_i^{(l)} - h_j^{(l)}) \right)}{\sum_{k \in N(i) \cup \{i\}} \exp \left( a^T W_2 (h_i^{(l)} - h_k^{(l)}) \right)}$$

# ALARM: A Deep Multi-View Framework for Anomaly Detection on Attributed Networks

Challenge:

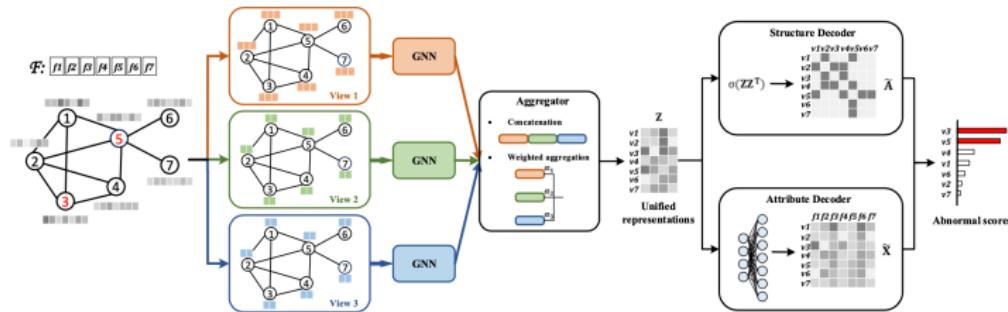
与普通网络相比，具有属性的网络能够包含更丰富的语义信息。因此，如何进一步提高异常检测的性能，以及如何利用丰富的语义来指导检测过程。此外，论文还指出，大多数现有的方法未能适应人们的需求，因为它们没有考虑到用户偏好的特殊性，这也是一个需要解决的挑战。



# ALARM

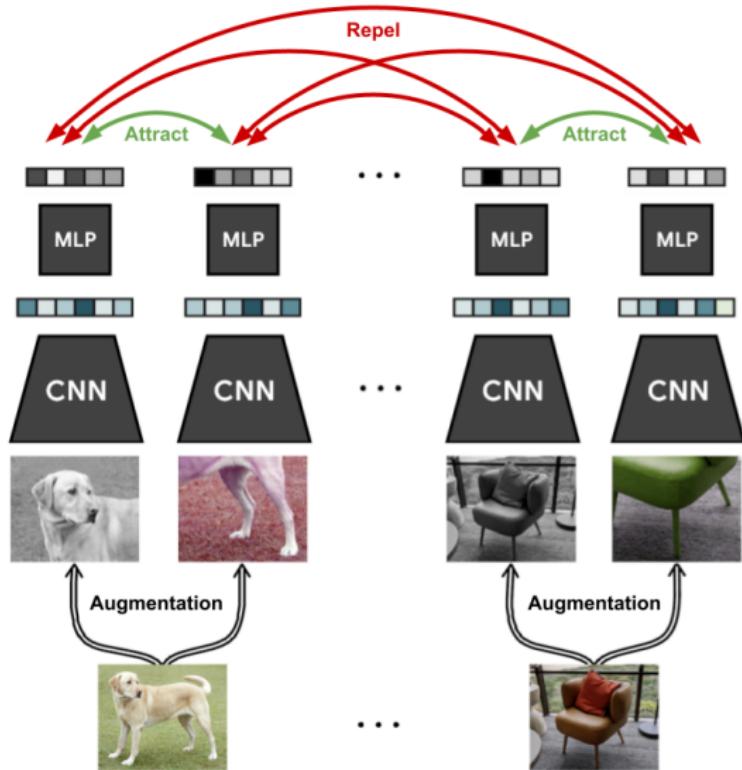
Solution:

ALARM 框架的关键思想是分别对给定的具有属性的网络在不同视图下进行编码，然后使用自学习参数或用户偏好将基于视图的嵌入聚合到一个新的统一潜在空间中。这个框架支持自学习和用户引导学习两种机制。



- 1 异常检测任务
- 2 基于分类的异常检测
- 3 基于自编码器的异常检测
- 4 基于对比学习的图异常检测
- 5 其他异常检测方法

# 对比学习



# 基于对比学习的异常检测

## 基本假设

对比学习是一个利用分类模型捕获样本的变换不变性的表征学习方法。

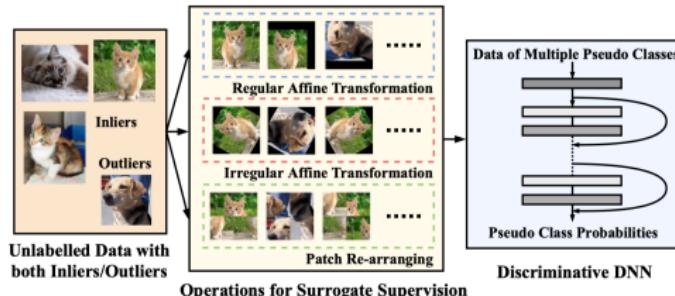
基本思路：

- ① 分类模型的置信度（正常样本可以被准确分类，异常样本无法准确分类）
- ② 变换不变的能力（正常样本多种变换间差异不大，异常样本多种变换间差异较大）

# Effective End-to-end Unsupervised Outlier Detection via Inlier Priority of Discriminative Network (NIPS'19)

## 数据增强

- ① 旋转
- ② 翻折
- ③ 平移
- ④ 区块打乱



# 表征学习对正常样本的偏好

## Motivation

- ① 类别不平衡状态下，有监督训练会在训练过程中偏向于捕获规模较大的类的信息
- ② 正常样本在训练过程中会提供更强的梯度方向指引（模型优化方向），且与异常样本有显著差异

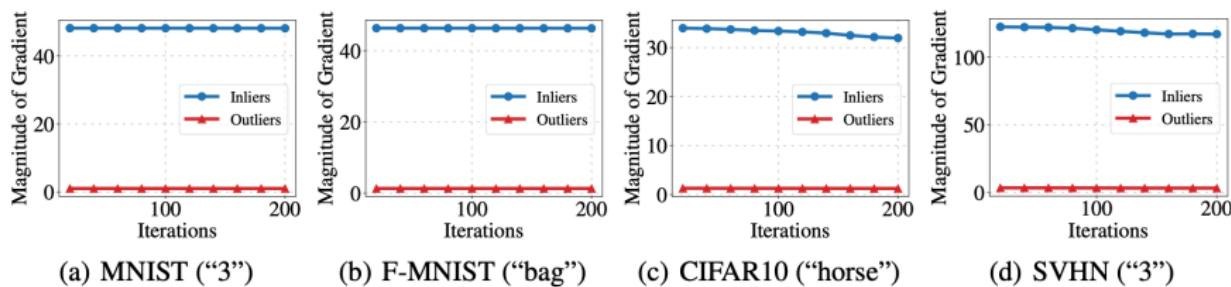


Figure 2: Inliers and outliers' gradient magnitude on example cases of benchmark datasets during SSD training. The class used as inliers is in brackets.

## 训练过程中的梯度对比

# 异常分值

Pseudo Label based Score (PL):

$$S_{pl}(\mathbf{x}) = \frac{1}{K} \sum_{y=1}^K P^{(y)} (\mathbf{x}^{(y)} | \boldsymbol{\theta})$$

Maximum Probability based Score (MP):

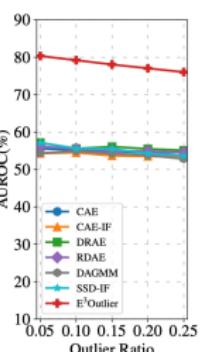
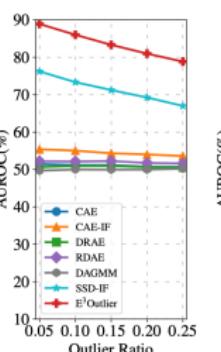
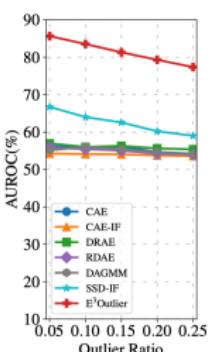
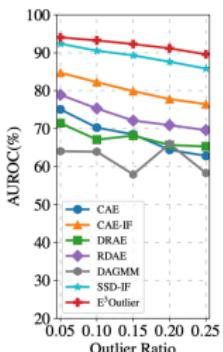
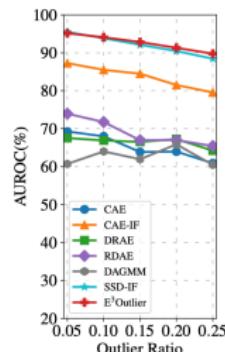
$$S_{mp}(\mathbf{x}) = \frac{1}{K} \sum_{y=1}^K \max_t P^{(t)} (\mathbf{x}^{(y)} | \boldsymbol{\theta})$$

Negative Entropy based Score (NE)

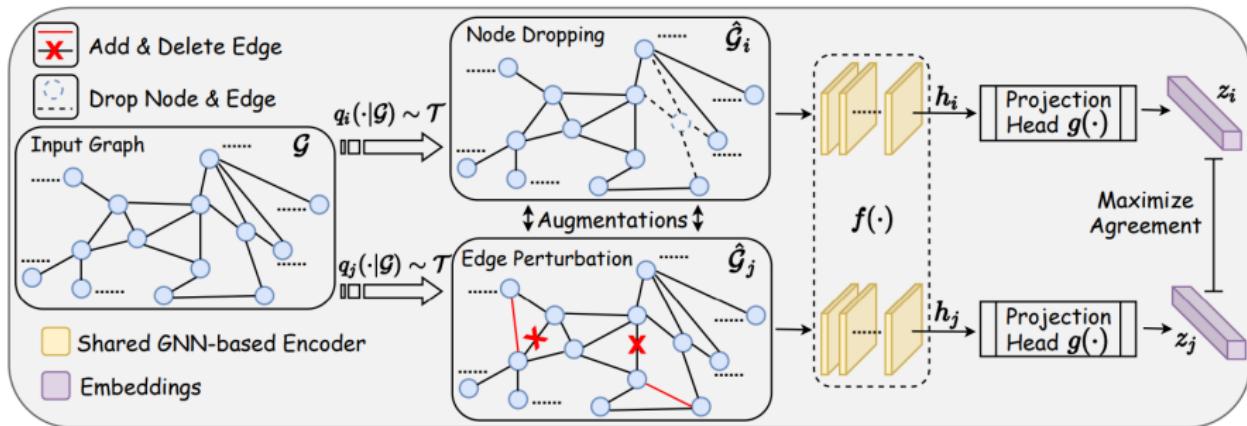
$$S_{ne}(\mathbf{x}) = \frac{1}{K} \sum_{y=1}^K \sum_{t=1}^K P^{(t)} (\mathbf{x}^{(y)} | \boldsymbol{\theta}) \log (P^{(t)} (\mathbf{x}^{(y)} | \boldsymbol{\theta}))$$

# $E^3$ Outlier: a Self-Supervised Framework for Unsupervised Deep Outlier Detection (TPAMI'22)

Dataset	$\rho$	CAE	CAE-IF	DRAE	RDAE	DAGMM	SSD-IF	$E^3$ Outlier
MNIST	10%	68.0/92.0/32.9	85.5/97.8/49.0	66.9/93.0/30.5	71.8/93.1/35.8	64.0/92.9/26.6	93.8/99.2/68.7	<b>94.1/99.3/67.5</b>
	20%	64.0/82.7/40.7	81.5/93.6/57.2	67.2/86.6/42.5	67.0/84.2/43.2	65.9/86.4/41.3	90.5/97.3/71.0	<b>91.3/97.6/72.3</b>
F-MNIST	10%	70.3/94.3/29.3	82.3/97.2/40.3	67.1/93.9/25.5	75.3/95.8/31.7	64.0/92.7/30.3	90.6/98.5/68.6	<b>93.3/99.0/75.9</b>
	20%	64.4/85.3/36.8	77.8/92.2/49.0	65.7/86.9/36.6	70.9/89.2/41.4	66.0/86.7/43.5	87.6/95.6/71.4	<b>91.2/97.1/78.9</b>
CIFAR10	10%	55.9/91.0/14.4	54.1/90.2/13.7	56.0/90.7/14.7	55.4/90.7/14.0	56.1/91.3/15.6	64.0/93.5/18.3	<b>83.5/97.5/43.4</b>
	20%	54.7/81.6/25.5	53.8/80.7/25.3	55.6/81.7/26.8	54.2/81.0/25.7	54.7/81.8/26.3	60.2/85.0/28.3	<b>79.3/93.1/52.7</b>
SVHN	10%	51.2/90.3/10.6	55.0/91.4/11.9	51.0/90.3/10.5	52.1/90.6/10.8	50.0/90.0/19.3	73.4/95.9/22.0	<b>86.0/98.0/36.7</b>
	20%	50.7/80.2/20.7	54.0/82.0/22.4	50.6/80.4/20.5	51.8/80.9/21.1	50.0/79.9/29.6	69.2/89.5/33.7	<b>81.0/93.4/47.0</b>
CIFAR100	10%	55.2/91.0/14.5	54.5/90.7/13.8	55.6/90.9/15.0	55.8/90.9/15.0	54.9/91.1/14.2	55.6/91.5/13.0	<b>79.2/96.8/33.3</b>
	20%	54.4/81.7/25.6	53.5/80.9/25.1	55.5/81.8/27.0	54.9/81.5/26.5	53.8/81.5/24.7	54.3/82.1/23.4	<b>77.0/92.4/46.5</b>



# 图对比学习



图对比学习沿用经典对比学习架构，  
并在数据增强和表征学习器上进行针对性设计。

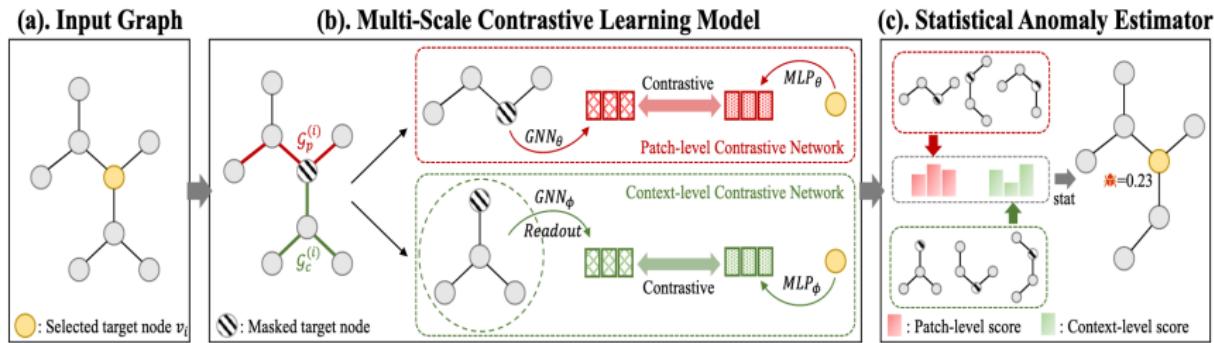
核心思路：

- ① 正负样本对对比作为异常分数
- ② 节点与邻居的对比作为异常分数

# ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning (CIKM'21)

## 研究动机

现有的图异常检测方法通常只考虑单个尺度的视图对比，因此会丢失不同尺度之间的信息，进而限制异常检测的效果。



## Multi-scale Contrastive Learning

对比“节点-节点”和“节点-子图”两个尺度，其中子图通过随机游走采样，并掩码中心节点的属性，避免任务设计上存在漏洞，模型学到捷径解。

- patch-level (i.e., node versus node) agreement

$$\mathcal{L}_p = -\frac{1}{2n} \sum_{i=1}^n \left( \log(s_p^{(i)}) + \log(1 - \tilde{s}_p^{(i)}) \right).$$

- context-level (i.e., node versus ego-net) agreement

$$\mathcal{L}_c = -\frac{1}{2n} \sum_{i=1}^n \left( \log(s_c^{(i)}) + \log(1 - \tilde{s}_c^{(i)}) \right).$$

statistical anomaly estimator: 推断异常分数，考虑到对比学习的正负样本和数据增强的随机性，设计了启发式的异常分数。

- 基础分数通过采样正负样本，计算对比分数的差值

$$b_{view,j}^{(i)} = \tilde{s}_{view,j}^{(i)} - s_{view,j}^{(i)},$$

- 提点的 trick：在基础分数之上计算均值和方差

$$\bar{b}_{view}^{(i)} = \sum_{j=1}^R b_{view,j}^{(i)} / R,$$

$$y_{view}^{(i)} = \bar{b}_{view}^{(i)} + \sqrt{\sum_{j=1}^R \left( b_{view,j}^{(i)} - \bar{b}_{view}^{(i)} \right)^2 / R},$$

# Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning (CoLA, TNNLS'21)

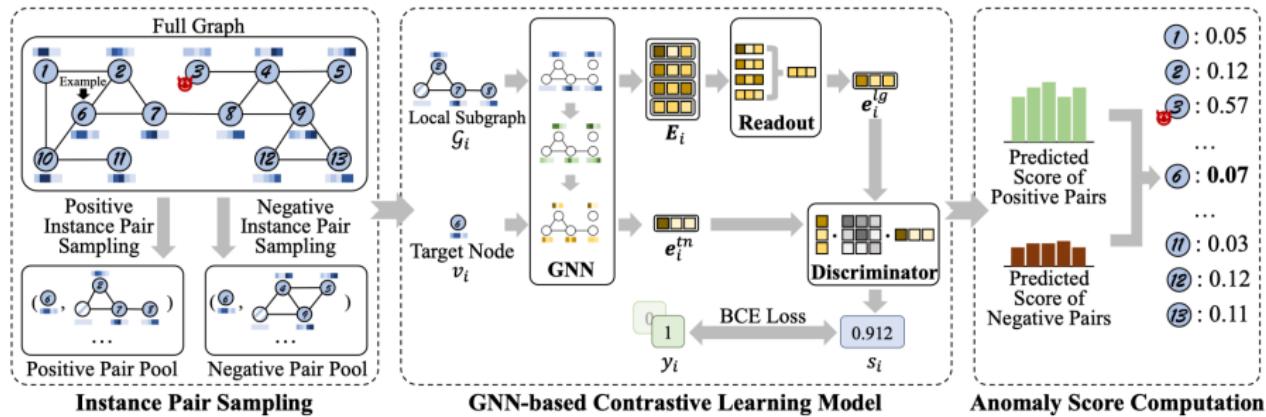
## 研究动机

由于同质性 (Homophily) 的存在，节点和其周围节点会表现出相似的属性或标签，而与周围节点模式不一致的节点则为异常节点。

核心思路：Node-Subgraph 对比学习框架，通过节点与邻居子图的一致性作为异常的指标。

- ① Random Walk 邻居子图生成
- ② Node VS Random Walk 邻居子图对比学习

# CoLA



异常分值：负例对比分数和正例对比分数之间的差值

$$f(v_i) = s_i^{(-)} - s_i^{(+)}$$

# Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection (SL-GAD, TKDE'21)

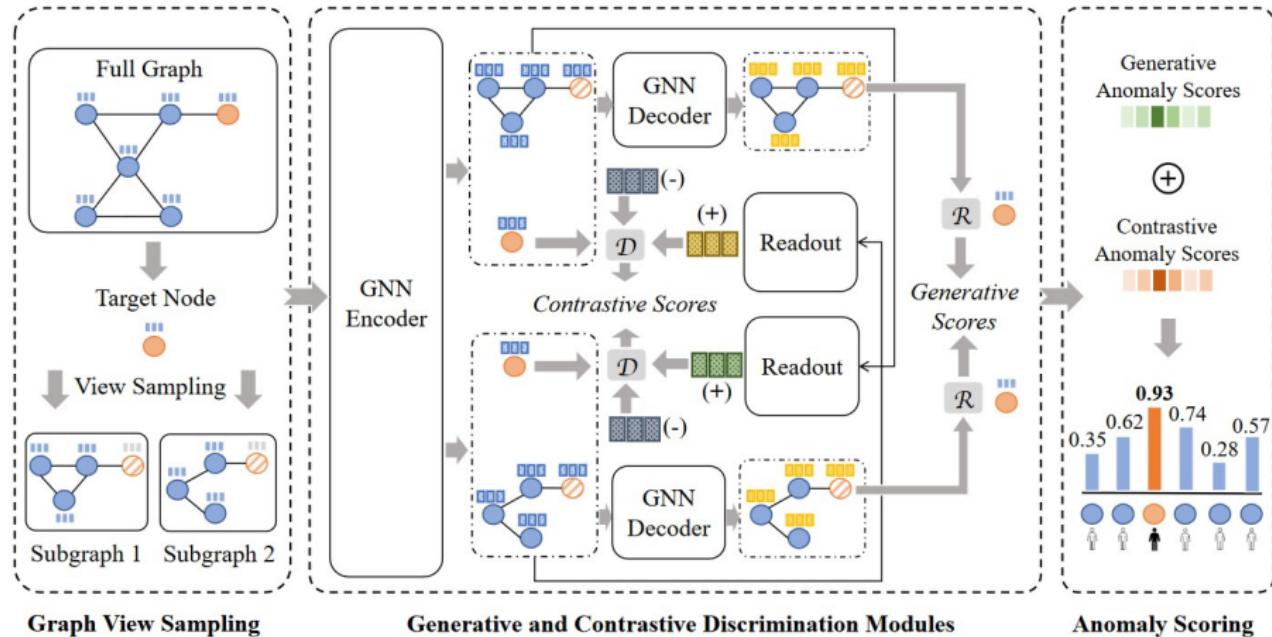
## 研究动机

现有的数据挖掘和机器学习方法要么是浅层方法，无法有效捕捉图数据的复杂相互依赖关系，要么是图自编码器方法，无法充分利用上下文信息作为有效的异常检测监督信号。

核心思路：基于目标节点构建不同的上下文子图（视图），生成和对比双视角进行异常检测。

- ① Generative attribute regression：从生成视角捕获属性空间中的异常
- ② Multi-view contrastive learning：从多个子图中获取更丰富的结构信息，捕获结构空间中的异常

# SL-GAD



# Reconstruction Enhanced Multi-View Contrastive Learning for Anomaly Detection on Attributed Networks (Sub-CR, IJCAI'22)

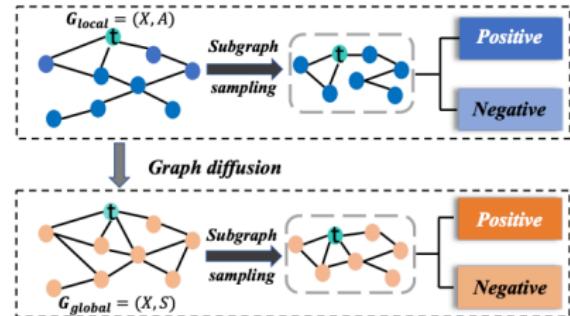
## 研究动机

检测带属性网络中的异常节点在许多实际应用中具有重要意义。然而由于异常节点与其他对应节点之间的复杂相互作用以及它们在属性上的不一致性，这个任务具有挑战性。

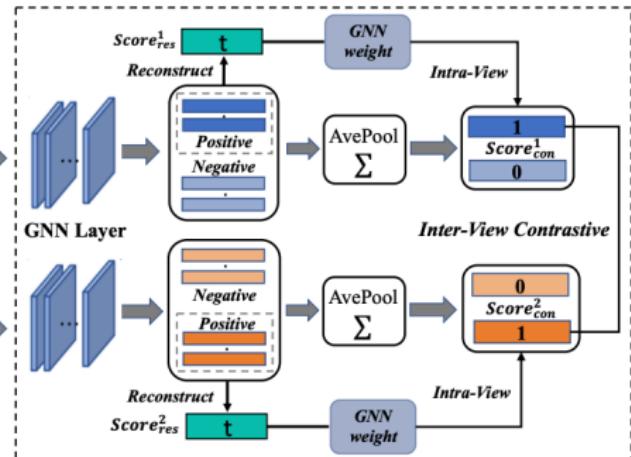
核心思路：提出一个自监督学习框架，通过同时优化基于多视图对比学习的模块和基于属性重构的模块，以更准确地检测带属性网络上的异常。

- ① 建立了两个对比学习视图，使模型能够更好地编码与异常相关的丰富的局部和全局信息
- ② 受相邻节点之间属性一致性原则的启发，引入基于掩码自编码器的重构模块，用于识别具有较大重构误差的节点

# Sub-CR



(a) Graph diffusion & Subgraph sampling



(b) Contrastive learning & Attribute reconstruction

(1) 对比视角生成与各图视角中的子图采样

Graph diffusion (生成对比视角图) :

$$\mathbf{S} = \sum_{k=0}^{\infty} \theta_k \mathbf{T}^k \in \mathbb{R}^{N \times N} \quad (1)$$

Unfolding:

$$\mathbf{S} = \alpha(\mathbf{I} - (1 - \alpha)\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2})^{-1} \quad (2)$$

Subgraph sampling:

Random walk with restart (RWR) from each central node. 在每个图视角中, 对于各节点  $v_i$ , 它自身的子图被视为正样本, 其他节点的子图被视为负样本。

# Sub-CR

## (2) Local & Global Graph Contrastive Learning

### Intra-View Contrastive Learning:

GCN 进行节点表征编码 (整个子图) :

$$\mathbf{H}_i^l = \sigma(\hat{\mathbf{D}}_i^{-1/2} \mathbf{A}_i \hat{\mathbf{D}}_i^{-1/2} \mathbf{H}_i^{(l-1)} \mathbf{W}^{(l-1)}) \quad (3)$$

节点表征编码 (各中心节点) :

$$\mathbf{h}_i^l = \sigma(\mathbf{h}_i^{l-1} \mathbf{W}^{l-1}) \quad (4)$$

子图 readout 并计算相应的 discriminative score:

$$\mathbf{e}_i = Readout(\mathbf{H}_i^l) = \sum_{k=1}^{n_i} \frac{(\mathbf{H}_i)_k}{n_i} \quad (5)$$

$$s_i = \sigma(\mathbf{h}_i \mathbf{W}_s \mathbf{e}_i^T) \quad (6)$$

# Sub-CR

对两个图视角都进行对比学习，以 local-view 为例：

$$\mathcal{L}_{intra}^1(v_i) = -\frac{1}{2}(\log(s_i) + \log(1 - \tilde{s}_i)) \quad (7)$$

最终 intra-view CL 的优化目标为：

$$\mathcal{L}_{intra} = \frac{1}{2N} \sum_{i=1}^N (\mathcal{L}_{intra}^1(v_i) + \mathcal{L}_{intra}^2(v_i)) \quad (8)$$

**Inter-View Contrastive Learning:** 计算两个图视角之间的  
discriminative scores 差异，希望越小越好

$$\mathcal{L}_{inter} = (\|\mathbf{s}_1 - \mathbf{s}_2\|_F^2) \quad (9)$$

### (3) Attribute Reconstruction Based on Neighbors

对两个视角分别用 MLP 进行属性重构，得到各自损失并联合：

$$\mathcal{L}_{res}^1(v_i) = \|g(\mathbf{Z}_i) - x_i\|^2 \quad (10)$$

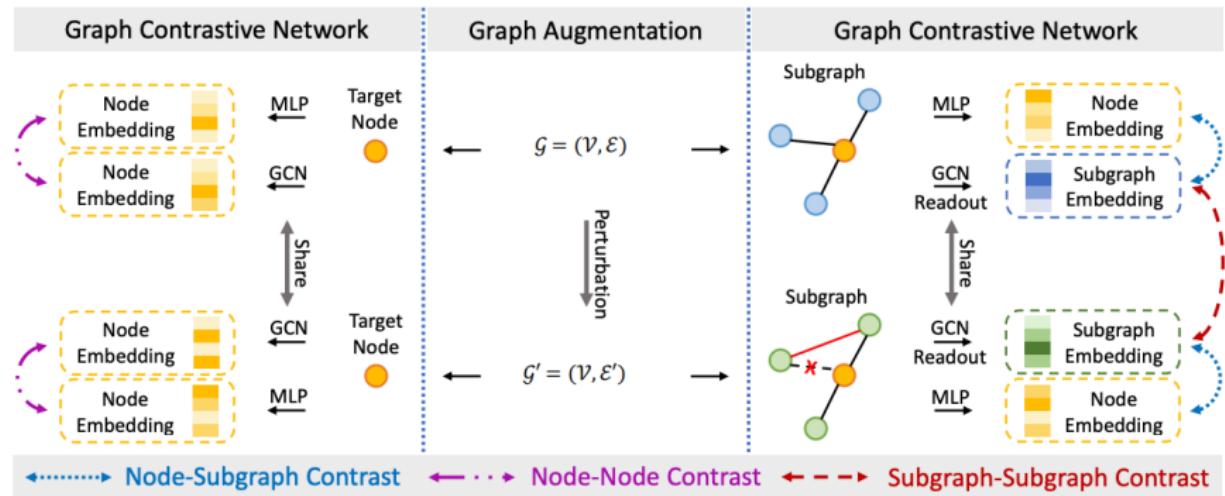
$$\mathcal{L}_{res} = \frac{1}{2N} \sum_{i=1}^N (\mathcal{L}_{res}^1(v_i) + \mathcal{L}_{res}^2(v_i)) \quad (11)$$

# Graph Anomaly Detection via Multi-Scale Contrastive Learning Networks with Augmented View (GRADATE, AAAI'23)

## 研究动机

最近的方法关注了各种尺度的对比策略，即节点-子图和节点-节点对比。然而，它们忽视了子图-子图比较信息，即在 GAD 中，正常和异常子图对在嵌入和结构方面表现不同。

核心思路：提出多视图多尺度对比学习框架中进行子图-子图对比，并与广泛采用的节点-子图和节点-节点对比相应策略相互配合，共同提升 GAD 性能。



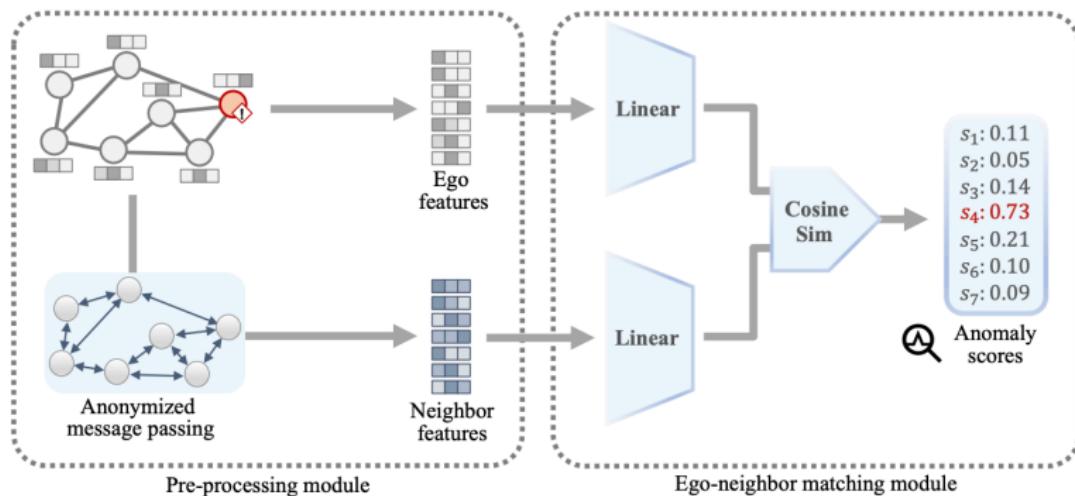
将原始输入图作为第一个视图，并通过边修改进行图扩充来生成第二个视图。在最大化子图对的相似性的指导下，提出子图-子图对比，联合节点-子图和节点-节点对比进行自监督学习。

# PREM: A Simple Yet Effective Approach for Node-Level Graph Anomaly Detection (ICDM'23)

## 研究动机

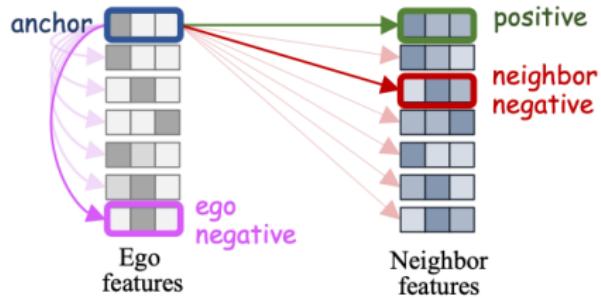
现有的方法，如基于重构和对比学习的方法，虽然有效，但往往由于复杂的目标和精细的模块而存在效率问题。

核心思路：为了提高 GAD 的效率，简化了 GAD，在训练过程中消除了消息传递传播的必要性，并采用简单的对比损失，从而显著降低了训练时间和存储空间使用。



PREM的整体流程由预处理模块和 ego-neighbor 匹配模块组成。在第一个模块中，对给定的图进行匿名消息传递，即不包括自环，以获取邻居特征（离线计算得到）。然后，在 ego-neighbor 匹配模块中（需在线计算的部分），将中心节点原始特征和预聚合得到邻居特征输入到鉴别器中，最终输出获得异常分数。

## Ego-neighbor matching contrastive learning:



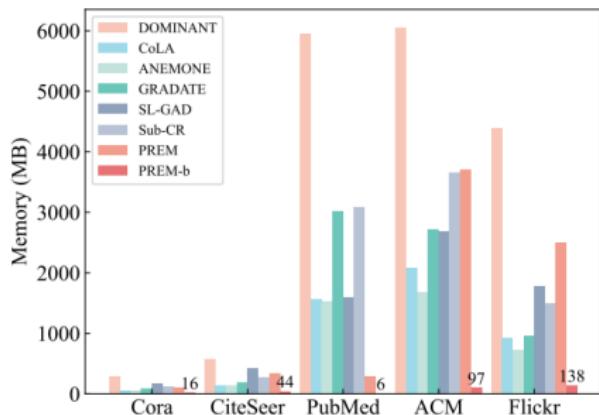
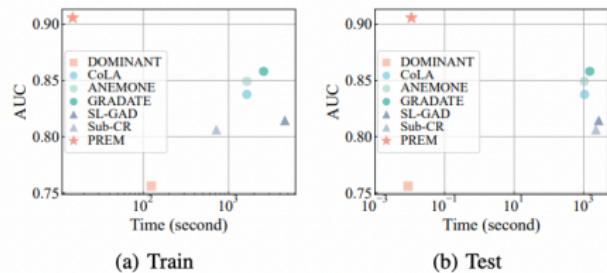
$$c_i^{pos} = \text{cosine}(\mathbf{h}_i^e, \mathbf{h}_i^n) \quad (12)$$

$$c_i^{negnbr} = \text{cosine}(\mathbf{h}_i^e, \mathbf{h}_j^n) \quad (13)$$

$$c_i^{negaego} = \text{cosine}(\mathbf{h}_i^e, \mathbf{h}_k^e) \quad (14)$$

$$\mathcal{L} = - \sum_{t=1}^N (\log(c_i^{pos}) + \alpha \cdot \log(1 - c_i^{negnbr}) + \gamma \cdot \log(1 - c_i^{negaego})) \quad (15)$$

## 同时平衡模型表现、计算时间和运行存储消耗：

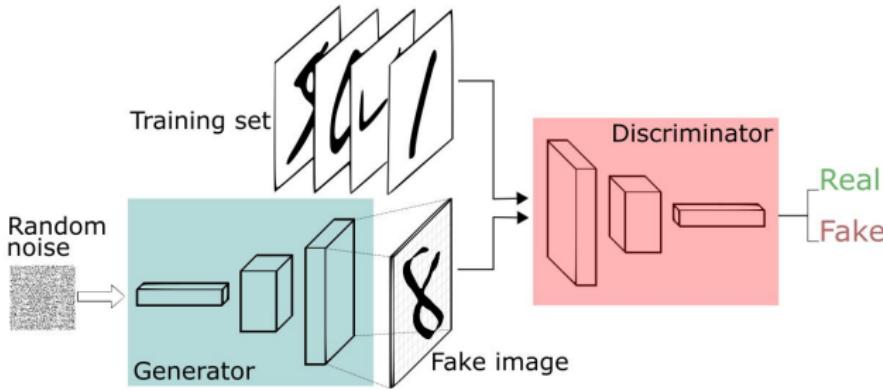


# 基于生成对抗网络的异常检测算法

## 生成对抗网络

生成对抗网络 Generative Adversarial Network 由生成器和判别器组成，两个网络相互对抗、不断调整参数，最终目的是使判别网络无法判断生成网络的输出结果是否真实。

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_X} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_Z} [\log(1 - D(G(\mathbf{z})))]$$



# GANomaly

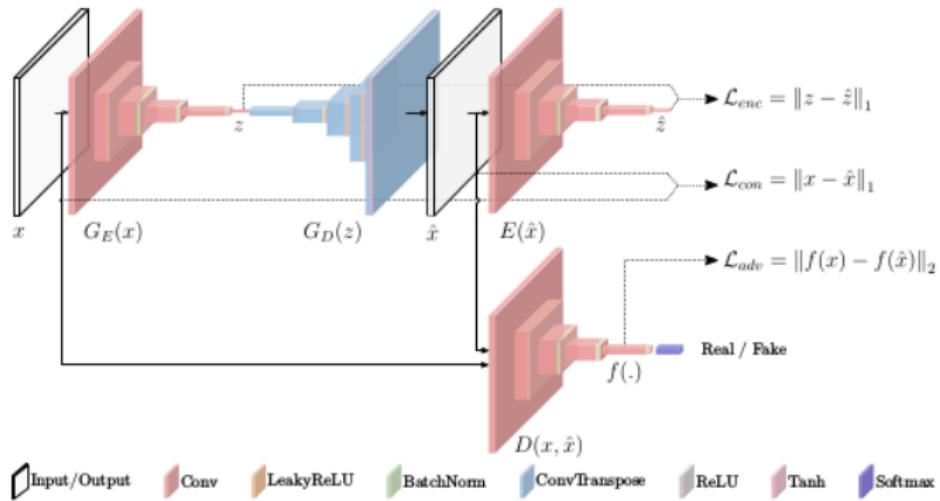


Figure 2: Pipeline of the proposed approach for anomaly detection.

GANomaly 网络结构

- 在训练阶段，整个模型均是通过正常样本做训练。也就是编码器  $GE(x)$ ，解码器  $GD(z)$  和重构编码器  $E(\hat{x})$ ，都是适用于正常样本的
- 当模型在测试阶段接受到一个异常样本，理论上此时模型的编码器，解码器将不适用于异常样本，此时得到的编码后潜在变量  $z$  和重构编码器得到的潜在变量  $\hat{z}$  的差距是大的。这个差距记为：

$$\mathcal{A}(\mathcal{X}) = \|G_E(x) - E(G(x))\|_1$$

通过设定阈值  $\phi$ ，一旦  $A(x) > \phi$  模型就认定送入的样本  $x$  是异常数据。

- 在训练阶段，整个模型均是通过正常样本做训练。也就是编码器  $GE(x)$ ，解码器  $GD(z)$  和重构编码器  $E(\hat{x})$ ，都是适用于正常样本的
- 当模型在测试阶段接受到一个异常样本，理论上此时模型的编码器，解码器将不适用于异常样本，此时得到的编码后潜在变量  $z$  和重构编码器得到的潜在变量  $\hat{z}$  的差距是大的。这个差距记为：

$$\mathcal{A}(\mathcal{X}) = \|G_E(x) - E(G(x))\|_1$$

通过设定阈值  $\phi$ ，一旦  $A(x) > \phi$  模型就认定送入的样本  $x$  是异常数据。

# 基于生成对抗网络的异常检测算法

## 优点

- ① GAN 作为最经典的深度生成模型之一，可以广泛用于生成与真实数据相似的样本。而难以从潜在空间生成的样本可能是异常样本。
- ② GAN 经过多年发展，已有大量成熟的模型可用于异常检测。

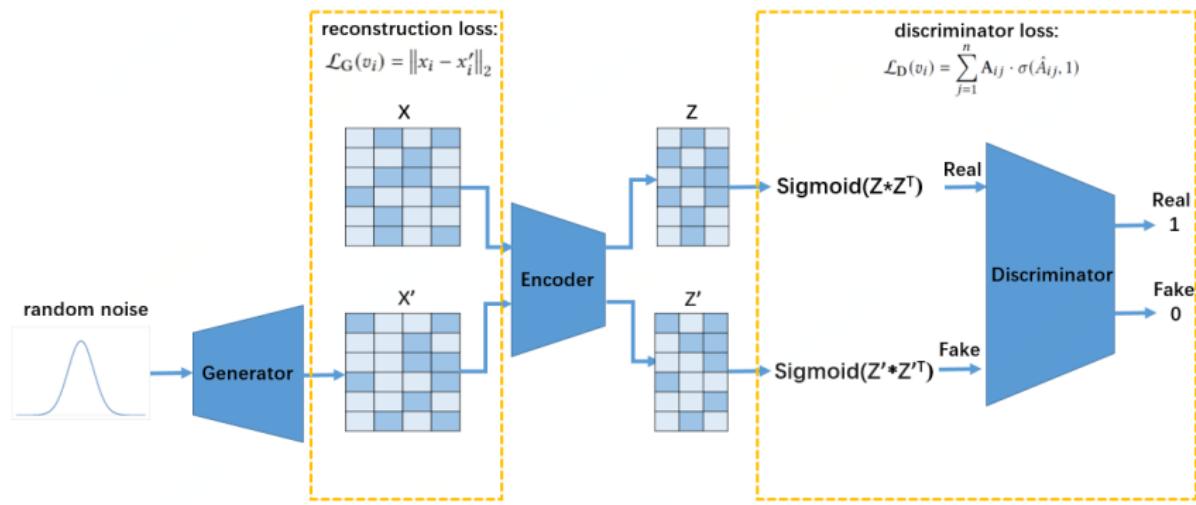
## 缺点

- ① GAN 模型的训练相对困难，容易出现模型坍塌等问题
- ② 当待检测数据较为复杂时，GAN 很容易生成与大部分样本不同的样本。异常数据集容易进一步加剧 GAN 模型的训练。
- ③ 基于 GAN 的异常检测模型本质上还是训练 GAN，而不是异常检测。

# Generative Adversarial Attributed Network Anomaly Detection(CIKM20)

## 研究动机

在异常检测场景中往往缺乏丰富的异常样本，GAN 能够生成大量异常样本从而解决异常样本不足的问题



# Generative Adversarial Attributed Network Anomaly Detection(CIKM20)

Generator: 使用多层 MLP 作为生成器，近似原始样本的属性分布

$$\mathbf{H}_G^{(l+1)} = f \left( \mathbf{W}_G^{(l)} \mathbf{H}_G^{(l)} + \mathbf{b}_G^{(l)} \right)$$

Encoder: 将节点原始属性和生成属性编码到低维空间

$$\mathbf{H}_E^{(l+1)} = f \left( \mathbf{W}_E^{(l)} \mathbf{H}_E^{(l)} + \mathbf{b}_E^{(l)} \right)$$

为了捕获图结构，使用 Encoder 生成的图表征重建图结构

$$\hat{\mathcal{A}} = \text{Sigmoid}(ZZ^T)$$

Discriminator: 判别器的目的是对于图上的边，判定原始属性重建的为真，生成属性重建的为假，使用交叉熵优化

# Generative Adversarial Attributed Network Anomaly Detection(CIKM20)

Optimization:

$$\begin{aligned}\mathcal{L}(D, E, G) = & \mathbb{E}_{x \sim p_X} [\mathbb{E}_{z \sim p_E(\cdot|x)} [\log D(Z)]] \\ & + \mathbb{E}_{x' \sim p_G} [\mathbb{E}_{z' \sim p_E(\cdot|x')} [1 - \log D(G(Z'))]]\end{aligned}$$

Anomaly Detection: 将生成器对属性的重建损失和判别器对结构的重建损失作为异常分数

$$\text{score}(v_i) = \alpha \mathcal{L}_G(v_i) + (1 - \alpha) \mathcal{L}_D(v_i)$$

$$\mathcal{L}_G(v_i) = \|x_i - x'_i\|_2$$

$$\mathcal{L}_D(v_i) = \sum_{j=1}^n \mathbf{A}_{ij} \cdot \sigma(\hat{A}_{ij}, 1) / \sum_{j=1}^n \mathbf{A}_{ij}$$

# 基于聚类的方法

## 研究动机

聚类和异常检测是最具代表性的两个无监督任务，两者假设类似且彼此依赖。

### 基于聚类的异常检测算法的假设

- ① 正常的数据会呈现一定的聚类分布，而异常数据不会属于任何类。
- ② 正常数据和他们最近的聚类中心比较接近，异常数据与聚类中心距离较远。
- ③ 正常数据属于大且稠密的类，异常数据不属于任何类或者属于小的稀疏的类。

# 基于聚类的异常检测算法

1. 对数据进行聚类，并把不在类中的数据作为异常数据。

- ① DBSCAN
- ② ROCK
- ③ SNN clustering

依赖于聚类的质量，效果难以保证

2. 对于每个数据，计算它到最近的聚类中心的距离，并作为异常分值

Two-step method

3. Cluster-Based Local Outlier Factor (CBLOF)  
聚类大小 + 到聚类中心的距离

# 基于概率分布的异常检测模型

## 先验假设

许多真实数据是由概率分布生成的，而异常数据则是与整体数据的概率分布不同。

## 模型假设

正常样本出现在一个分布的高概率密度区域，而异常样本则出现在低概率密度区域。

通过学习数据集的概率密度参数，估计出样本所处的概率密度空间，并估计置信度。

# 基于概率分布的异常检测模型

## 基本流程

- ① 为数据选择一个概率模型
- ② 根据概率模型选择一个概率阈值
- ③ 计算观测到每个样本的概率
- ④ 将低于阈值的样本作为异常样本

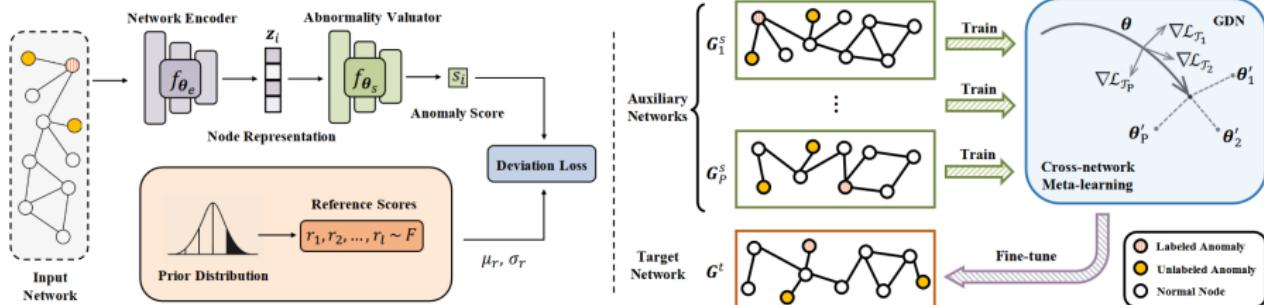
常见的概率模型：高斯分布，泊松分布（Poisson Distribution）

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

# Few-shot Network Anomaly Detection via Cross-network Meta-learning(WWW21)

## 研究动机

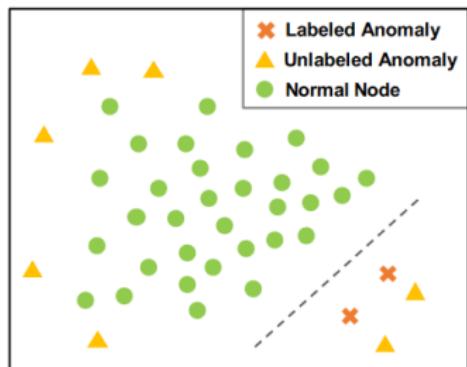
无监督方法缺乏对异常的先验知识，可能识别出噪声样本，真实场景往往能够获得少量异常标签



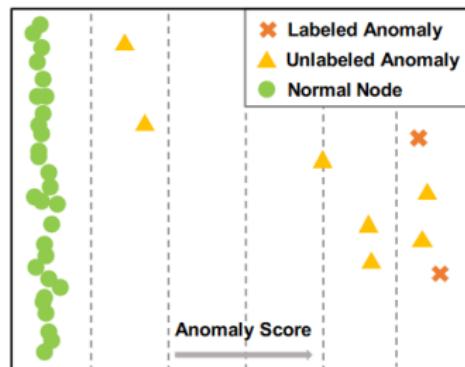
# Few-shot Network Anomaly Detection via Cross-network Meta-learning(WWW21)

## 研究动机

只有少量异常标记的情况下很难将无标记的异常检测从表征空间分离，增强正常和异常节点异常分数的偏差能够使他们在异常分數空间显著分离



(a) Latent Representation Space



(b) Anomaly Score Space

# Few-shot Network Anomaly Detection via Cross-network Meta-learning(WWW21)

Graph Deviation Networks: 假设正常节点的异常分数满足高斯分布，建模正常节点异常分数的概率密度分布

$$\mathcal{R} = \{r1, r2, \dots, rk\} \sim \mathcal{N}(\mu, \sigma^2)$$

定义异常分数偏差为

$$dev(v_i) = \frac{s_i - \mu}{\sigma}$$

引导正常节点异常分数偏差降低，增大异常节点异常分数偏差

$$\mathcal{L} = (1 - y_i) \cdot |dev(v_i)| + y_i \cdot \max(0, m - dev(v_i))$$

# Few-shot Network Anomaly Detection via Cross-network Meta-learning(WWW21)

## meta-learning

---

**Algorithm 1** The learning algorithm of Meta-GDN

---

**Input:** (1)  $P$  auxiliary networks, i.e.,  $\mathcal{G}^s = \{G_1^s = (A_1^s, X_1^s), G_2^s = (A_2^s, X_2^s), \dots, G_P^s = (A_P^s, X_P^s)\}$ ; (2) a target network  $G^t = (A^t, X^t)$ ; (3) sets of few-shot labeled anomalies and unlabeled nodes for each network (i.e.,  $\{\mathcal{V}_1^L, \mathcal{V}_1^U\}, \dots, \{\mathcal{V}_P^L, \mathcal{V}_P^U\}$  and  $\{\mathcal{V}_t^L, \mathcal{V}_t^U\}$ ); (4) training epochs  $E$ , batch size  $b$ , and meta-learning hyper-parameters  $\alpha, \beta$ .

**Output:** Anomaly scores of nodes in  $\mathcal{V}_t^U$ .

- 1: Initialize parameters  $\theta$ ;
- 2: **while**  $e < E$  **do**
- 3:   **for** each network  $G_i^s$  (task  $\mathcal{T}_i$ ) **do**
- 4:     Randomly sample  $\frac{b}{2}$  nodes from  $\mathcal{V}_i^L$  and  $\frac{b}{2}$  from  $\mathcal{V}_i^U$  to comprise the batch  $B_i$ ;
- 5:     Evaluate  $\nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$  using  $B_i$  and  $\mathcal{L}(\cdot)$  in Eq. (7);
- 6:     Compute adapted parameters  $\theta'$  with gradient descent using Eq. (8),  $\theta'_i \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(f_{\theta})$ ;
- 7:     Sample a new batch  $B'_i$  for the meta-update;
- 8:   **end for**
- 9:   Update  $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{i=1}^P \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$  using  $\{B'_i\}$  and  $\mathcal{L}(\cdot)$  according to Eq. (7);
- 10: **end while**
- 11: Fine-tune  $\theta$  on target network  $G^t$  with  $\{\mathcal{V}_t^L, \mathcal{V}_t^U\}$ ;
- 12: Compute anomaly scores for nodes in  $\mathcal{V}_t^U$ ;

---

# 基于距离的异常方法

## 基本假设

异常数据通常距离正常的数据较远。

异常的分值可以用样本到它的邻居的距离来定义。

## 优点

- ① 简单直接
- ② 无监督，数据驱动

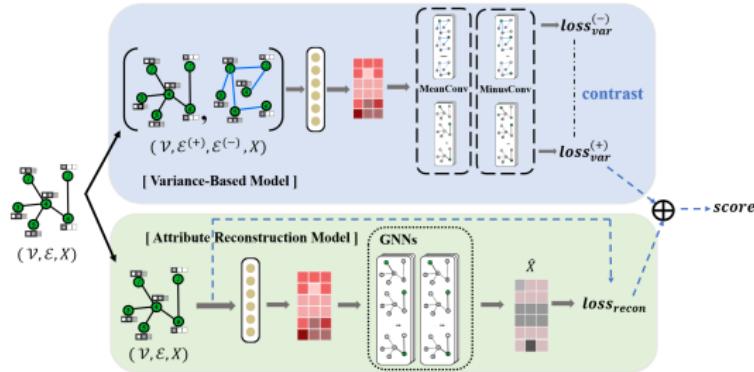
## 缺点

- ① 依赖特征
- ② 难以处理高维数据
- ③ 结果难以解释
- ④ 对数据的密度敏感

# Unsupervised Graph Outlier Detection: Problem Revisit, New Insight, and Superior Method(ICDE23)

## 研究动机

现有方法检测结构异常时对 degree 大的节点具有较大 bias, 采用方差来衡量节点和邻居节点的一致性能够克服这一 bias



# Unsupervised Graph Outlier Detection: Problem Revisit, New Insight, and Superior Method(ICDE23)

Neighbor Variance:

$$\bar{\mathbf{h}}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{h}_j$$

$$var(v_i) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\mathbf{h}_j - \bar{\mathbf{h}}_i)^2$$

$$o_i^{str} = loss_{var}(v_i) = \|var(v_i)\|_1$$

优化目标：最小化领域方差的同时放大非领域方差，使得正常节点具有低方差，而异常节点具有高方差

$$\min_{\theta} \mathbb{E}_{v_i \sim \mathcal{V}} [loss_{var}(v_i) - \frac{1}{|\mathcal{V} - \mathcal{N}_i|} \sum_{j \notin \mathcal{N}_i} (h_j - \frac{1}{|\mathcal{V} - \mathcal{N}_i|} \sum_{u \notin \mathcal{N}_i} h_u)^2]$$

# Unsupervised Graph Outlier Detection: Problem Revisit, New Insight, and Superior Method(ICDE23)

负图采样：由于每个节点不相邻的节点数量过大，对每个节点采样部分不相邻的节点构成负图  $\mathcal{G}^-$

$$loss_{var}^{(+)}(v_i) = \|var(v_i, \mathcal{G})\|_1$$

$$loss_{var}^{(-)}(v_i) = \|var(v_i, \mathcal{G}^-)\|_1$$

$$loss^{str}(v_i) = loss_{var}^{(+)}(v_i) - loss_{var}^{(-)}(v_i)$$

属性异常检测：通过重建属性实现

$$o_i^{attr} = loss^{recon}(v_i) = \|\hat{x}_i - x_i\|^2$$

# Unsupervised Graph Outlier Detection: Problem Revisit, New Insight, and Superior Method(ICDE23)

联合异常分数：

$$\hat{o}_i^{str} = \frac{o_i^{str} - \mu(\mathcal{O}^{str})}{std(\mathcal{O}^{str})}$$

$$\hat{o}_i^{attr} = \frac{o_i^{attr} - \mu(\mathcal{O}^{attr})}{std(\mathcal{O}^{attr})}$$

$$o_i = \hat{o}_i^{str} + \hat{o}_i^{attr}$$

# 总结与展望

当前图异常检测面临的主要困难：

- ① 异常假设的一致性
- ② 异常模式的动态性
- ③ 图神经网络的有效性
- ④ 异常结果的可解释性

# References I

-  Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. (2018). Deep one-class classification.  
In *International conference on machine learning*, pages 4393–4402. PMLR.