

# 基于增强视图的多尺度对比学习网络的图异常检测

## 摘要

图异常检测 (GAD) 是基于图的机器学习中的一项重要任务, 在许多实际应用中得到了广泛的应用。GAD 的主要目标是从图数据集中捕获明显偏离大多数节点的异常节点。最近的方法关注了 GAD 的各种尺度的对比策略, 即节点-子图和节点-节点对比。然而, 它们忽略了 GAD 中正常子图对和异常子图对在嵌入和结构上表现不同的子图-子图比较信息, 从而导致次优的任务性能。在本文中, 我们首次在提出的具有子图-子图对比度的多视图多尺度对比学习框架中实现了上述思想。具体来说, 我们将原始输入图视为第一个视图, 并通过边修改的图增强生成第二个视图。在最大化子图对相似性的指导下, 所提出的子图-子图对比度有助于在不考虑结构变化的情况下获得更鲁棒的子图嵌入。此外, 引入的子图-子图对比与被广泛采用的节点-子图和节点-节点对比对等体很好地配合, 实现了 GAD 性能的相互提升。此外, 我们还进行了充分的实验来研究不同的图增强方法对检测性能的影响。综合的实验结果很好地证明了我们的方法与最先进的方法相比的优越性以及多视图子图对对比策略在 GAD 任务中的有效性。然而, 上述的方法忽略了集体拓扑异常的情况。为此, 我们再提出了一种基于子结构感知的拓扑异常检测模块, 具体来说, 我们建立了一个区域建议模块来发现网络中的高密度子结构作为可疑区域, 平均节点对相似度可视为子结构内节点的拓扑异常程度。一般情况下, 相似度越低, 内部节点出现拓扑异常的概率越高。该方法可以同时检测拓扑和属性异常, 能够获得不错的 AUC 增益效果。

**关键词:** 属性网络; 异常检测; 对比学习; 图增强; 网络子结构

## 1 引言

属性网络具有强大的建模和分析现实世界中众多复杂场景的能力。其中带有属性的节点表示现实世界的实体, 链接表示实体之间的关系, 在各种场景中无处不在, 包括金融 (trading networks)、社交媒体 (social networks)、和电子商务 (itemuser networks)。在过去的几年中, 基于图的机器学习引起了极大的关注 [1]。作为图学习中的代表性任务, 图异常检测旨在从大多数节点中识别异常, 正成为研究人员的一个越来越有魅力的应用, 在金融欺诈检测、社交垃圾邮件检测、网络入侵检测、信息系统错误检测等诸多领域都有广泛的应用。本研究的成果适用于多个领域, 特别是在利用属性图存储数据的应用中, 去检测数据中的异常行为或构成系统安全威胁的数据, 在社交网络分析、金融欺诈监测和网络安全防护等领域具备广泛的应用可能。以社交网络分析为例, 由于存在账户身份验证不完备等问题, 社交网络中存在大量虚假账号, 这些虚假账号可能被用于恶意攻击正常用户, 或者被滥用以增加虚拟热度, 扰乱网络社区的秩序与安全。可以将各个账号视为节点, 彼此之间的互动看作边连接, 每个节

点带有特定的属性，构成属性图，通过异常检测技术，可以发现虚假账号的恶意行为、违规关注等问题，从而提升社交网络的品质 and 安全性水平。属性图异常检测技术能为社交网络平台提供了一种有效的方法来发现不良用户和清理，并维护社交网络中的秩序和保护用户的安全。

属性网络异常检测（ANAD）旨在识别偏离频繁模式的异常节点。与其他领域广泛应用的原始数据不同，属性网络既包含网络拓扑信息，也包含节点属性信息。根据资料可知，上述两类信息不匹配会导致属性网络出现两种典型的异常，即特征异常和结构异常 [2]。前者指的是与邻居节点在特征上存在差异的节点，后者指的是一组相异但联系紧密的节点。如图 1 所示，拓扑异常一目了然，属性值正常，但与不相关节点紧密相连；相反，对应属性具有正常的邻域连通性，但其属性值可能受到噪声的污染，与邻域不相似。进一步分析表明，拓扑异常模式是一组异常节点的集体行为，这些节点在网络中形成了异常密集的子结构。

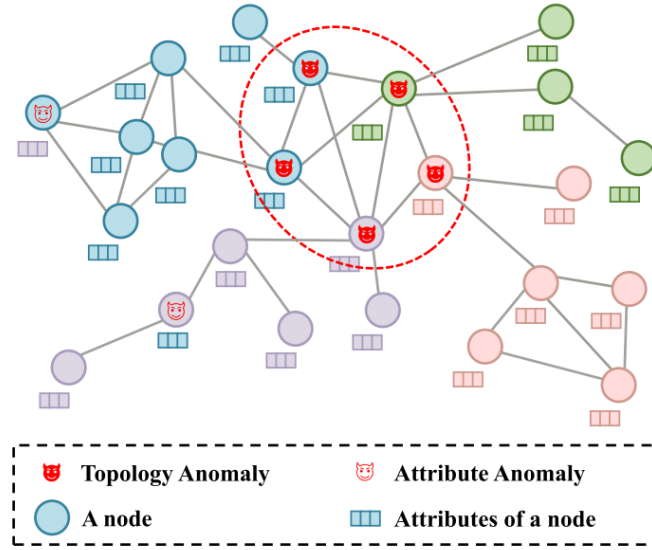


图 1. 特征异常和结构异常图例

## 2 相关工作

### 2.1 图异常检测

早期的工作 [3–6] 通常采用非深度范式，从节点特征和网络结构中检测异常信息。为了检测这两类异常，许多先前的方法已经做出了巨大的努力，并取得了令人印象深刻的结果。LOF 通过将节点与其上下文节点在特征上进行比较来获取节点的异常信息。SCAN 从网络结构上完成 GAD 任务。利用这两类信息，ANOMALOUS 基于 CUR 分解和残差分析进行异常检测。以及进行特征子空间选择，并在子空间中寻找异常节点。上述方法依赖于特定的领域知识，无法挖掘图数据集中深层次的非线性信息，这使得它们很难进一步提高检测性能。

然而，如果不挖掘更深层次的信息，他们就无法持续改进自己的业绩。近年来，神经网络的兴起 [7–9] 增强了模型挖掘深层非线性信息的能力。基于重构的方法 DOMINANT [10] 通过计算 GCN 后特征矩阵和结构矩阵的变化来获得节点异常得分。虽然它表现良好且实现简单，但是一些异常信息会被忽略。GCN 通过聚合来自邻域的信息来生成节点表示，这将使异常更加难以区分。AAGNN [11] 将单类 SVM 应用于图异常检测。HCM [12] 将节点及其一阶

邻居的跳数估计值作为其异常得分。CoLA [2] 首先引入对比学习范式来检测图中的节点异常。该方法挖掘了节点周围的局部特征和结构信息。后来的方法 [13-16] 在 CoLA 的基础上做了进一步的改进。

## 2.2 图对比学习

对比学习是无监督学习中最重要范式之一。图对比学习在没有昂贵标签的情况下为下游任务挖掘监督信息，并取得了巨大的成就 [17]。基于负样本使用策略，现有工作可以分为基于负样本和无负样本两大类。对于第一类，DGI 通过最大化节点和图之间的互信息来获得有用的监督信息。SUBG-CON 和 GraphCL 形成节点-子图对比，学习更好的节点表示。对于第二种类型，BGRL 使用孪生网络从两个视图中提取丰富的信息。[18] 利用了 Barlow Twins 的优点，它设计了一个特殊的损失函数来避免表示崩溃。然而，现有工作忽略了对子图信息的进一步挖掘，也没有直接优化其嵌入来进行图异常检测。已有证明子图表示学习有利于基于图的机器学习任务。这将极大地促进挖掘各自子图的局部特征和结构信息。对于 GAD，更具代表性和内在的子图嵌入可以帮助计算节点与其邻域之间更可靠的关系，这是对比策略的关键步骤。

## 2.3 图增强

图增强产生了图数据集 [19] 的可能变化。在不需要昂贵标签的情况下，扩充了数据集，提高了模型的泛化能力。大多数方法集中于对图中节点或边的操作。[20,21] 注意修改节点特征。RoSA [22] 使用带重启的随机游走作为图增强来学习节点的鲁棒表示。[23] 通过增加或删除边来调整邻接矩阵。

# 3 本文方法

## 3.1 本文方法概述

在这一部分，我们介绍了我们提出的框架。它由两个主要模块组成。在图增强模块中，我们将原始图作为第一个视图，并通过边修改产生第二个视图。在图对比网络模块中，我们首先从节点和子图的特征对比中获取异常信息，将其视为节点-子图对比。对于每个视图，子图通过随机游走进行采样，并与目标节点形成对。然后我们构建节点-节点对比来捕获节点级别的异常。新增加的子图-子图对比度直接优化了两个视图之间用于 GAD 的子图嵌入。在这个过程中，节点-子图对比的性能会得到提升。之后，我们采用一个集成的损失函数来训练这三种对比度。最后，我们综合各种异常信息，计算每个节点的异常得分。后续的优化改进为再提出了一种基于子结构感知的属性网络图异常检测模块。我们专注于图中的子结构来识别异常，具体来说，我们建立了一个区域建议模块来发现网络中的高密度子结构作为可疑区域，平均节点对相似度可视为子结构内节点的拓扑异常程度。一般情况下，相似度越低，内部节点出现拓扑异常的概率越高。该方法结合两个模块可以同时检测拓扑和属性异常。原论文方法框架图如 2。

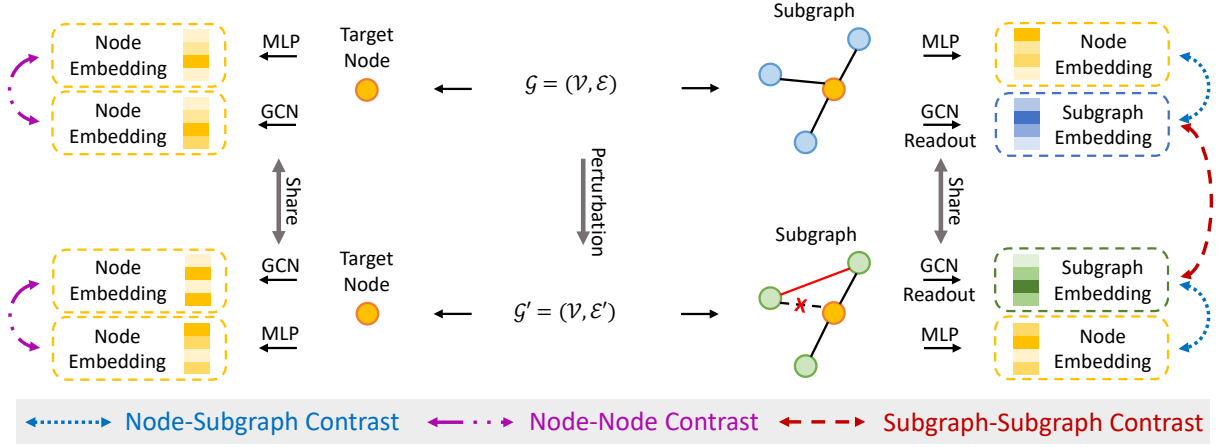


图 2. 原论文框架图

### 3.2 图增强模块

图增强对于自监督学习范式至关重要。可以帮助模型挖掘图的更深层次的语义信息。在本文中，我们利用边修改来创建第二个视图。然后我们对子图进行随机游走采样。节点和子图将构成图对比网络的输入。

#### 3.2.1 边缘修改

边缘修改 ( Edge Modification, EM ) 通过对图的边进行扰动来建立第二个视图。受 ( Jin 等 2021b) 的启发，我们不仅删除邻接矩阵中的边，而且同时添加相同数量的边。在实际应用中，我们首先设定一个固定的比例  $P$ ，从邻接矩阵中均匀随机地丢弃  $\frac{PM}{2}$  边。然后将  $\frac{PM}{2}$  条边均匀随机地添加到矩阵中。通过这种方式，我们试图在不破坏图的性质的前提下，学习子图的鲁棒表示。消融研究部分将进一步讨论生成第二个视图的图增强方法。

#### 3.2.2 随机游走

对于一个目标节点，一种有效的异常检测方法是测量它与其邻域 [2] 之间的特征距离。因此，我们使用带重启的随机游走 ( RWR ) [24] 来采样节点周围的子图。较低的特征相似度表明目标节点的异常度较高。

### 3.3 图对比学习模块

图对比网络对比学习范式已被证明对 GAD [2] 是有效的。我们构建了一个多视图图对比网络，它包括三个部分，即节点子图，节点-节点和子图-子图对比。前两个对比度定义在每个视图中，并将通过两个视图的信息融合来加强。节点-子图对比主要用于捕获来自节点邻域的异常信息。第二种对比度可以更好地检测节点级异常。同时，子图-子图对比工作在两个视图之间。它将直接优化 GAD 的子图嵌入，这大大有利于节点-子图的对比。

#### 3.3.1 节点-子图对比

目标节点  $v_i$  与其所在的子图形成正对，与另一个节点  $v_j$  所在的随机子图形成负对。我们首先采用一个 GCN 层，将子图中节点的特征映射到嵌入空间。值得注意的是，子图中目标

节点的特征被掩盖，即设置为 0。子图隐层表示可以定义为：

$$\mathbf{H}_i^{(\ell+1)} = \sigma \left( \widetilde{\mathbf{D}}_i^{-\frac{1}{2}} \widetilde{\mathbf{A}}_i \widetilde{\mathbf{D}}_i^{-\frac{1}{2}} \mathbf{H}_i^{(\ell)} \mathbf{W}^{(\ell)} \right), \quad (1)$$

其中  $\mathbf{H}_i^{(\ell+1)}$  和  $\mathbf{H}_i^{(\ell)}$  表示  $(\ell+1)$ -th 和  $\ell$ -th 层隐藏层表示， $\widetilde{\mathbf{D}}_i^{-\frac{1}{2}} \widetilde{\mathbf{A}}_i \widetilde{\mathbf{D}}_i^{-\frac{1}{2}}$  是邻接矩阵的标准化矩阵， $\mathbf{W}^{(\ell)}$  表示网络的参数。

然后，子图的最后表示  $\mathbf{z}_i$  通过 *Readout* 函数输出。具体来说，我们使用平均函数来实现 *Readout*：

$$\mathbf{z}_i = \text{Readout}(\mathbf{Z}_i) = \sum_{j=1}^{n_i} \frac{(\mathbf{Z}_i)_j}{n_i}. \quad (2)$$

相应地，我们利用 MLP 将目标节点特征转换到与子图相同的嵌入空间。节点隐层表示为：

$$\mathbf{h}_i^{(\ell+1)} = \sigma \left( \mathbf{h}_i^{(\ell)} \mathbf{W}^{(\ell)} \right), \quad (3)$$

其中  $\mathbf{W}^{(\ell)}$  与上方 GCN 共享参数。  $\mathbf{e}_i$  为节点的最后表示。

在每个视图中，目标节点的异常程度与子图的相似度  $s_i^1$  和节点嵌入有关。我们采用双线性模型来衡量这种关系：

$$s_i^1 = \text{Bilinear}(\mathbf{z}_i, \mathbf{e}_i) = \text{sigmoid}(\mathbf{z}_i \mathbf{W} \mathbf{e}_i^\top). \quad (4)$$

一般情况下，目标节点和子图表示在正对中趋于相似，即  $s_i^1 = 1$ 。相反，它们在负对中可能是不相似的，即  $s_i^1 = 0$ 。因此，我们采用二进制叉熵（BCE）损失来训练对比度：

$$\mathcal{L}_{NS}^1 = - \sum_{i=1}^N (y_i \log(s_i^1) + (1 - y_i) \log(1 - s_i^1)), \quad (5)$$

其中  $y_i$  正对为 1，负对为 0。

我们可以从另一个角度得到相似度  $s_i^2$  和 BCE 损失  $\mathcal{L}_{NS}^2$ 。值得一提的是，两个视图下的两个网络使用相同的架构和共享参数。因此，最终的节点-子图对比损失为：

$$\mathcal{L}_{NS} = \alpha \mathcal{L}_{NS}^1 + (1 - \alpha) \mathcal{L}_{NS}^2, \quad (6)$$

其中  $\alpha \in (0, 1)$  一个权衡参数，用于平衡两个视图之间的重要性。

### 3.3.2 节点-节点对比

节点-节点对比可以有效发现节点层面的异常。同样，目标节点特征也会被掩盖。并且其表示是从子图中的另一个节点聚合而来。在每个视图中，MLP 后与同一个节点形成正对，MLP 后与另一个节点形成负对。我们利用一个新的 GCN 来获得子图的表示：

$$\mathbf{H}_i'^{(\ell+1)} = \sigma \left( \widetilde{\mathbf{D}}_i'^{-\frac{1}{2}} \widetilde{\mathbf{A}}_i' \widetilde{\mathbf{D}}_i'^{-\frac{1}{2}} \mathbf{H}_i'^{(\ell)} \mathbf{W}'^{(\ell)} \right), \quad (7)$$

其中  $\mathbf{W}'^{(\ell)}$  不同于节点-子图对比中使用的参数矩阵。 $\mathbf{h}_i'^{(\ell+1)} = \mathbf{H}_i'^{(\ell+1)} [1, :]$  是  $(\ell+1)$ -th 层隐藏层节点  $v_i$  的表示。 $\mathbf{u}_i$  是节点的最终表示。



同时，我们使用 MLP 将节点特征映射到相同的隐藏空间：

$$\hat{\mathbf{h}}_i^{(\ell+1)} = \sigma \left( \hat{\mathbf{h}}_i^{(\ell)} \mathbf{W}^{(\ell)} \right), \quad (8)$$

其中  $\mathbf{W}^{(\ell)}$  与 Eq. (7) 共享参数。在 MLP 后,  $\hat{\mathbf{e}}_i$  是目标节点的最终嵌入。

与节点-子图对比相似，我们采用 *Bilinear* 模型来评估  $\hat{s}_i^1$  为  $\mathbf{u}_i$  与  $\hat{\mathbf{e}}_i$ 。然后节点与节点的损失函数表示为：

$$\mathcal{L}_{NN}^1 = - \sum_{i=1}^N \left( \hat{y}_i \log(\hat{s}_i^1) + (1 - \hat{y}_i) \log(1 - \hat{s}_i^1) \right). \quad (9)$$

同理，也可以从另一个角度计算相似度  $\hat{s}_i^2$  和损失  $\mathcal{L}_{NN}^2$ 。因此最终的节点-节点对比度损失函数为：

$$\mathcal{L}_{NN} = \alpha \mathcal{L}_{NN}^1 + (1 - \alpha) \mathcal{L}_{NN}^2, \quad (10)$$

其中视图平衡参数  $\alpha$  与公式 Eq. (6) 中的节点子图对比度损失共享。

### 3.3.3 子图-子图对比

子图-子图对比度定义在两个视图之间。它旨在为 GAD 学习更具代表性和内在的子图嵌入，这将有助于节点-子图对比判别节点及其邻域的关系。在实际应用中，我们直接在节点-子图对比度联合损失下优化子图表示。

一个子图与扰动子图形成一个正对，其中同一个目标节点  $v_i$  位于另一个视图中。与常见的图对比方法不同，它与另一个节点  $v_j$  位于两个视图中的两个子图形成负对。节点  $v_j$  是子图在节点-子图对比中与  $v_i$  形成负对的节点。受 (Oord, Li 和 Vinyals 2018) 的启发，我们使用了一个损失函数来优化对比度：

$$\mathcal{L}_{SS} = - \sum_{i=1}^n \log \frac{\exp(\mathbf{z}_i^1 \cdot \mathbf{z}_i^2)}{\exp(\mathbf{z}_i^1 \cdot \mathbf{z}_j^1) + \exp(\mathbf{z}_i^1 \cdot \mathbf{z}_j^2)}, \quad (11)$$

其中  $\mathbf{z}_i^1$  and  $\mathbf{z}_i^2$  表示目标节点  $v_i$  在两个视图中的子图嵌入。 $\mathbf{z}_j^1$  和  $\mathbf{z}_j^2$  分别是另一个节点  $v_j$  在两个视图中的子图嵌入。

### 3.3.4 损失函数

为了综合三种对比的优点，我们对联合损失函数进行优化：

$$\mathcal{L} = \beta \mathcal{L}_{NS} + (1 - \beta) \mathcal{L}_{NN} + \gamma \mathcal{L}_{SS}, \quad (12)$$

其中  $\beta \in (0, 1)$  是子图级和节点级异常信息的平衡参数。 $\gamma \in (0, 1)$  是  $\mathcal{L}_{SS}$  一个权衡参数。

## 3.4 异常评分估计

在节点-子图和节点-节点对比中，正常节点与其正对中的子图或节点相似，与其负对中的子图或节点不同。相反，异常节点在正负对中都不同于节点或子图。自然地，我们定义目标节点的异常分数如下：

$$s_i = s_i^n - s_i^p, \quad (13)$$

其中  $s_i^p$  和  $s_i^n$  分别表示正负对的相似度。

然后我们从两个视图和三个对比度来综合融合异常信息。异常分数可以进一步表示为：

$$\begin{aligned} s_i &= \alpha s_i^1 + (1 - \alpha) s_i^2, \\ \hat{s}_i &= \alpha \hat{s}_i^1 + (1 - \alpha) \hat{s}_i^2, \\ S_i &= \beta s_i + (1 - \beta) \hat{s}_i, \end{aligned} \quad (14)$$

其中  $\alpha$  和  $\beta$  与公式 Eq. (6), (10) 和 (12) 共享参数。

仅使用一次随机游走进行一次检测无法捕获足够的语义信息。为了计算每个节点的异常得分，需要进行多轮检测。受启发，我们计算最终的异常根据多轮检测结果的平均值和标准差进行评分：

$$\begin{aligned} \bar{S}_i &= \frac{1}{R} \sum_{r=1}^R S_i^{(r)}, \\ S_i &= \bar{S}_i + \sqrt{\frac{1}{R} \sum_{r=1}^R \left( S_i^{(r)} - \bar{S}_i \right)^2}, \end{aligned} \quad (15)$$

其中  $R$  是异常检测的轮数。

### 3.5 优化方案

#### 3.5.1 拓扑异常检测模块

通过随机游走在对比网络中构成节点-子图对。然而，随机游走引起的不确定性并不总是有助于学习节点局部拓扑信息的。在极端情况下，异常节点周围的采样子图可能在其邻域中包含几个相似的节点，这将不利于检测异常节点。另一方面，这种节点-子图对比模式忽略了网络拓扑上拓扑异常的集体异常行为。因此，我们尝试从子结构的角度来利用网络中的局部拓扑信息。我们设计了一个区域建议模块来提取可疑子结构，克服了上述不足。然后，我们根据平均节点对相似度来区分子结构中的异常节点。方法框架如 3。

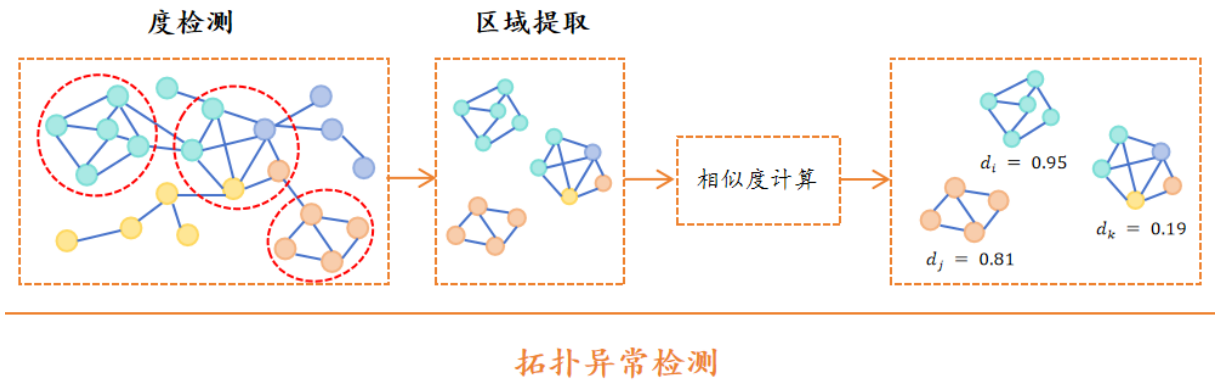


图 3. 改进方法框架图

**区域建议模块。** 一组拓扑异常将在网络中形成异常密集的子结构。为此，我们采用基于密度的算法来发现可能包含拓扑异常的高密度子结构。在实践中，我们首先使用  $k$  核方法来

查找  $k$  核图。然后，根据节点的连通性将它们划分为子结构。最终，这些子结构  $\{C_0, \dots, C_b\}$  中可能存在异常节点的区域可视为可疑区域。

**拓扑异常估计.** 在现实中，正常样本之间存在紧密联系的情况，例如经理和秘书之间的电子邮件通信频繁。相反，流水线工人和秘书之间密集的电子邮件交流可以被视为异常。因此，一些可疑区域只有正常节点，而另一些则包含正常节点和拓扑异常。鉴于此，我们通过子结构中平均节点对相似度来衡量内部节点的异常程度。需要指出的是，属性差异越显著的节点异常可能性越高。为了获得更准确的相似度，我们利用节点属性的嵌入来代替原始属性值。具体来说，我们选择余弦距离作为节点对相似度度量。

$$d_{k,q} = \text{Similarity}(\mathbf{z}_k, \mathbf{z}_q) = \cos(\mathbf{z}_k, \mathbf{z}_q), \quad (16)$$

其中  $\mathbf{z}_k$  和  $\mathbf{z}_q$  表示节点  $v_k$  and  $v_q$  的嵌入。

然后，我们利用一个平均函数来计算每个子结构：

$$d_j = \frac{1}{\hat{n}_j} \sum_{l=1}^{\hat{n}_j} d_{k,q}^{(l)}, \quad (17)$$

其中  $\hat{n}_j$  是子结构  $C_j$  的节点对个数。节点  $v_k$  和  $v_q$  属于  $C_j$ 。

Eq.(16)中的相似度度量采用属性异常检测模块学习的节点嵌入。

**拓扑异常得分.** 由于节点对平均相似度与子结构中节点的异常程度呈负相关，因此自然使用节点对平均相似度的倒数来表示其异常程度。然后，我们将倒数值作为子结构内每个内部节点的拓扑异常估计。我们计算子结构中每个节点的拓扑异常估计：

$$t_i = \frac{1}{d_j}, \quad (18)$$

其中  $d_j$  为子结构  $C_j$  的节点内相似度。 $t_i$  表示目标节点  $v_i$  的拓扑异常评分，该节点属于  $C_j$ 。

由于目前尚不清楚拓扑异常属于哪个子结构。我们以逐渐增加的  $k$  进行多轮检测。为了避免不必要的计算，我们以等于平均节点度  $\delta$  的特定  $k$  开始检测。返回 NULL 时，子结构发现算法将停止。在每次检测中，我们通过 Eq. (16), (17), 和 (18) 计算在任何子结构中检测到的节点的拓扑异常估计。在任何子结构中未检测到的其他节点将被分配拓扑异常评分为 0。根据 ANAD 的特点，结合节点对平均相似度和子结构大小，设计了拓扑异常评分方法。总的来说，我们采用了一种新的多尺度子结构评分策略来进行拓扑异常检测。在实践中，我们采用子结构中的节点数来衡量这种影响。因此，我们最终对多轮检测的异常得分进行平均，以确定每个节点的拓扑异常得分。因此，目标节点的最终拓扑异常评分可定义为：

$$\text{score\_t}(v_i) = \frac{1}{R^t} \sum_{r=1}^{R^t} |C_j| t_i^{(r)}, \quad (19)$$

其中  $|C_j|$  表示  $v_i$  所在子结构  $C_j$  中的节点对数。 $R^t$  为拓扑异常检测轮数。 $\text{score\_t}(v_i)$  为目标节点  $v_i$  的最终拓扑异常评分。

### 3.5.2 异常得分融合

在确定每个节点的拓扑异常评分后，我们使用聚合器将其与原论文所得的异常评分  $\text{score\_a}(v_i)$  进行融合。首先，对每一种分数进行归一化，统一其大小。然后，聚合器根据权重对两个分数



求和，得到每个节点得分的最终异常得分：

$$score(v_i) = (1 - \alpha) \cdot score\_t(v_i) + \alpha \cdot score\_a(v_i), \quad (20)$$

其中  $\alpha \in [0, 1]$  是平衡两个分数之间重要性的权衡参数。

## 4 复现细节

### 4.1 与已有开源代码对比

论文的开源代码地址：<https://github.com/FelixDJC/GRADATE>。本人复现了原文大部分实验，并参考其他论文引入新的拓扑异常检测模块，对两部分的最终异常得分进行权重求和，提高异常检测对两种异常模式的检测性能。

### 4.2 实验环境搭建

使用 anaconda3 管理，以下是 python 和一些必要库的版本：

```
python==3.8  
torch==2.0.1+cu117  
dgl==0.4.3post2
```

### 4.3 创新点

- 首次将子图-子图对比引入到 GAD 中，优化节点与子图对比的学习嵌入，提出一个具有增强视图的多尺度对比学习网络框架，并取得了不错的 AUC 增益。
- 研究了不同的图增强对任务的子图表示学习的影响。
- 重新设计了一个新的原理来检测拓扑异常。具体来说，它从网络子结构的角度来感知他们的集体异常行为。与原论文框架结合，使得框架能够同时检测拓扑异常和属性异常。

## 5 实验结果分析

### 5.1 结果与分析

我们通过将 GRADATE 与八种基线方法进行比较来评估其异常检测性能。同时，AUC 值对比结果如表1所示。对于结果，我们得出以下结论：

- 我们可以直观地发现，在这六个数据集上，GRADATE 优于其竞争对手。其中，GRADATE 分别在 EAT、WebKB、UAT、Cora、UAI2010 和 Citation 上实现了显著的 AUC 增益 **2.54%**、**0.62%**、**3.64%**、**0.45%**、**5.31%**和 **0.43%**。
- 我们观察到大多数基于神经网络的方法优于浅层方法、LOF 和 ANOMALOUS。浅方法在处理图数据的高维特征时存在固有的局限性。

表 1. AUC 的性能比较。粗体和下划线分别表示最佳和第二名的结果。

Methods	EAT	WebKB	UAT	Cora	UAI2010	Citation
LOF	0.5255	0.2903	0.4906	0.3538	0.7052	0.3059
ANOMALOUS	0.4109	0.3417	0.3356	0.3198	0.5026	0.5656
DOMINANT	0.6023	0.7787	0.6503	0.8929	0.7698	0.7748
CoLA	0.6762	0.8175	0.6538	0.8847	0.7949	0.7296
ANEMONE	<u>0.7726</u>	0.8208	<u>0.8087</u>	0.9122	<u>0.8731</u>	0.8028
SL-GAD	0.6974	<u>0.8678</u>	0.6851	<u>0.9192</u>	0.8454	<u>0.8095</u>
HCM	0.4536	0.5064	0.3262	0.6276	0.5210	0.5414
Sub-CR	0.6672	0.8423	0.6788	0.9133	0.8571	0.7903
<b>GRADATE</b>	<b>0.7980</b>	<b>0.8740</b>	<b>0.8451</b>	<b>0.9237</b>	<b>0.9262</b>	<b>0.8138</b>

- 深层方法中,基于对比学习的方法、CoLA、ANEMONE、SL-GAD、Sub-CR 和 GRADATE 效果较好。结果表明,基于对比学习的模式通过从图中挖掘特征和结构信息,可以有效地检测异常。通过新增加的子图-子图对比和多视图学习策略, GRADATE 达到了最佳性能。

## 5.2 消融验证

### 5.2.1 各种对比策略

为了验证所提出的子图-子图对比的有效性,我们进行了消融研究实验。为方便起见, NS、NS + SS、NS + NN 和 NS + NN + SS 表示仅使用节点-子图对比 (CoLA)、使用节点-子图和子图-子图对比、使用节点-子图和节点-节点对比 (ANEMONE), 以及分别使用以上三种对比 (GRADATE)。如表2所示, 添加子图-子图对比度可以通过提高节点-子图对比度来提高检测性能。使用所有三种对比将导致最佳性能。

表 2. 各种对比策略的消融研究

	EAT	WebKB	UAT	Cora	UAI2010	Citation
NS	0.6762	0.7949	0.6538	0.8847	0.8175	0.7296
NS+SS	0.6800	0.8310	0.6603	0.8956	0.9055	0.6978
NS+NN	0.7726	0.8208	0.8087	0.9122	0.8731	0.8028
<b>NS+NN+SS</b>	<b>0.7980</b>	<b>0.8740</b>	<b>0.8451</b>	<b>0.9237</b>	<b>0.9262</b>	<b>0.8138</b>

### 5.2.2 图增强策略

我们采用了四种不同的图增广, 形成了第二种视图, 并探讨了它们对性能的影响。高斯噪声特征 (GNF) 是指节点特征被高斯噪声随机扰动。FM (Feature Mask) 是指对节点特征的随机部分进行屏蔽。图扩散 (GD) 是通过扩散模型生成图扩散矩阵。GNF 和 FM 是节点特征的扰动。GD 和边缘修正 (Edge Modification, EM) 是目前广泛应用的图边增强方法。如表3所示, EM 在所有数据集上的性能最好。进一步分析, 节点特征的扰动可能会破坏正常节点的特征。它会损害节点与其邻域之间的比较, 而这是广泛性遗传的对比学习的基础。

础。这可能会导致一些正常节点被错误地分类为异常节点，从而导致性能下降。此外，GD 是一种基于结构的图增广方法。然而，它的主要目的是捕获全局信息。因此，EM 比 GD 更兼容子图-子图对比，可以提高节点-子图对比，挖掘节点的局部邻域信息。

表 3. 图增强的消融研究

	EAT	WebKB	UAT	Cora	UAI2010	Citation
GNF	0.7548	0.8183	0.8327	0.9031	0.9193	0.7902
FM	0.7782	0.8148	0.8256	0.8924	0.9171	0.8034
GD	0.7618	0.8062	0.8143	0.9026	0.9161	0.8030
<b>EM</b>	<b>0.7980</b>	<b>0.8740</b>	<b>0.8451</b>	<b>0.9237</b>	<b>0.9262</b>	<b>0.8138</b>

### 5.3 超参数分析

#### 5.3.1 平衡参数 $\alpha$ , $\beta$ 与 $\gamma$

讨论了损失函数中三个重要的平衡参数。如图4所示，超参数  $\alpha$  和  $\beta$  对于提高 EAT 和 UAI2010 的检测性能是有效的。在其他数据集上也可以观察到类似的现象。在实践中，我们将 EAT、WebKB、UAT、Cora、UAI2010 和 Citation 上的  $\alpha$  设置为 0.9、0.1、0.7、0.9、0.7 和 0.5。同时，我们将  $\beta$  设为 0.3、0.7、0.1、0.3、0.5 和 0.5。

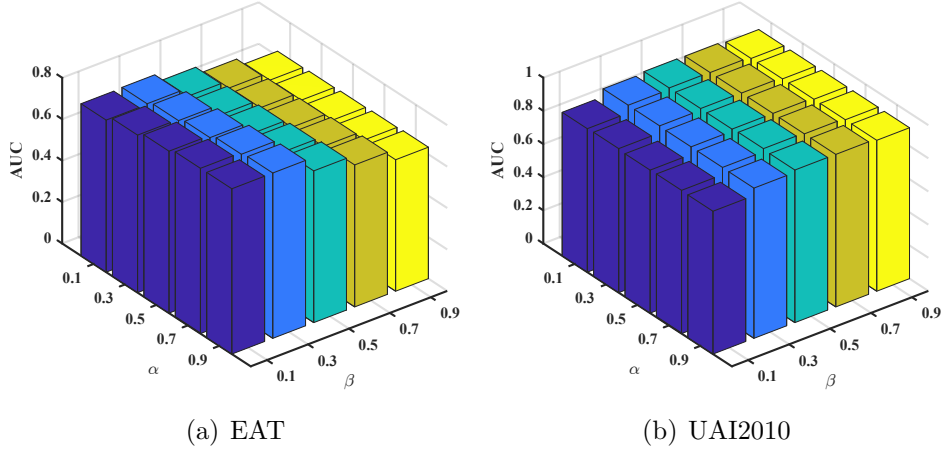


图 4. 平衡参数  $\alpha$  和  $\beta$  对 EAT 和 UAI2010 的敏感性分析

图5显示了  $\gamma$  在 0.1 到 0.9 之间变化时 GRADATE 的性能变化。从图中我们观察到，通过将  $\gamma$  设置为 0.1，GRADATE 倾向于在所有基准测试中表现良好。

#### 5.3.2 边缘修正比例 $P$

还研究了不同参数化边缘修正的影响。从图6可以看出，在 UAT、UAI2010 和 Citation 上，通过修改比例  $P$ ，检测性能波动较小。综合而言，我们在所有数据集上固定设置  $P = 0.2$ 。

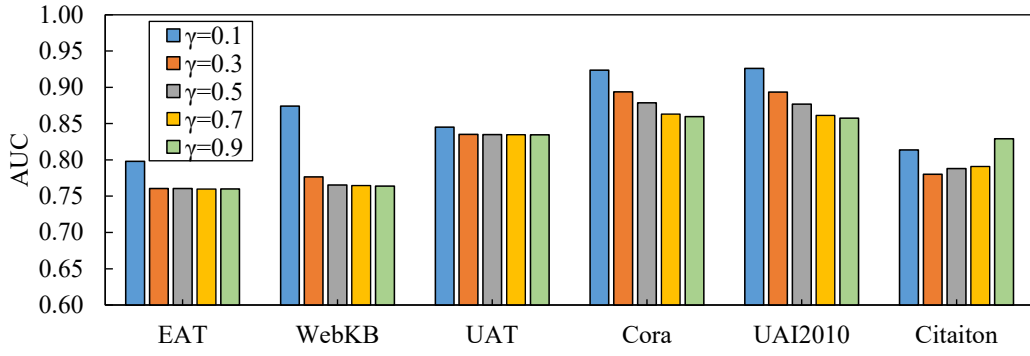


图 5. 平衡参数  $\gamma$  分析

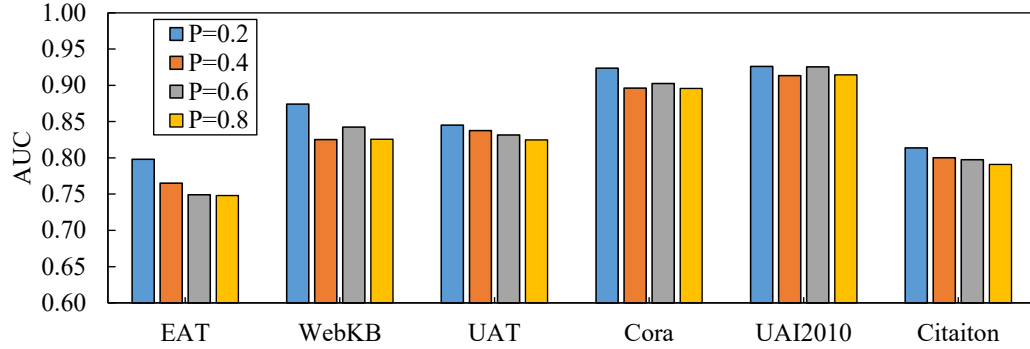


图 6. 边缘修正比例  $P$  分析

#### 5.4 优化结果

优化方法针对 Cora 和 Citation 两个数据集进行测试，如表4所示。可以看到实验跑的结果与原论文差异不大，代表复现效果还是可以的。差异主要是因为数据集异常注入的随机性，但是整体的异常检测性能差异不大。其次优化的模块给原框架也带来了少许的 AUC 增益，但提升效果一般，分析可能是因为注入的异常方式并不是真实场景下的数据集，所以该拓扑异常检测方式并不是很有效。

表 4. 实验结果与优化结果

	Cora	Citation
GRADATE(论文)	0.9237	0.8138
GRADATE(实验)	0.9117	0.7888
<b>IMPROVE</b>	<b>0.9279</b>	<b>0.8076</b>

## 6 总结与展望

在本文中，提出了一种新的基于增强视图的多尺度对比学习网络的图异常检测框架。我们将子图-子图对比引入到 GAD，并研究了不同的图增强技术对任务的影响。在六个基准数据集上的大量实验证明，GRADATE 优于竞争对手。在未来，我们将继续探索任务的对比学习模式，以及如何更好地聚合节点的属性和拓扑异常得分，以避免信息丢失，能够更好的同

时检测属性和结构两种异常模式。

## 参考文献

- [1] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.
- [2] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6):2378–2392, 2021.
- [3] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. Focused clustering and outlier detection in large attributed graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1346–1355, 2014.
- [4] Bryan Perozzi and Leman Akoglu. Scalable anomaly ranking of attributed neighborhoods. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 207–215. SIAM, 2016.
- [5] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. Radar: Residual analysis for anomaly detection in attributed networks. In *IJCAI*, volume 17, pages 2152–2158, 2017.
- [6] Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, Qinghua Zheng, et al. Anomalous: A joint modeling approach for anomaly detection on attributed networks. In *IJCAI*, volume 18, pages 3513–3519, 2018.
- [7] Wenxuan Tu, Sihang Zhou, Xinwang Liu, Yue Liu, Zhiping Cai, En Zhu, Changwang Zhang, and Jieren Cheng. Initializing then refining: A simple graph attribute imputation network. In *IJCAI*, pages 3494–3500, 2022.
- [8] Ke Liang, Yue Liu, Sihang Zhou, Xinwang Liu, and Wenxuan Tu. Relational symmetry based knowledge graph contrastive learning. *arXiv preprint arXiv:2211.10738*, 2022.
- [9] Yue Liu, Jun Xia, Sihang Zhou, Xihong Yang, Ke Liang, Chenchen Fan, Yan Zhuang, Stan Z Li, Xinwang Liu, and Kunlun He. A survey of deep graph clustering: Taxonomy, challenge, application, and open resource. *arXiv preprint arXiv:2211.12875*, 2022.
- [10] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM international conference on data mining*, pages 594–602. SIAM, 2019.
- [11] Shuang Zhou, Qiaoyu Tan, Zhiming Xu, Xiao Huang, and Fu-lai Chung. Subtractive aggregation for attributed network anomaly detection. In *Proceedings of the 30th ACM*



- International Conference on Information & Knowledge Management*, pages 3672–3676, 2021.
- [12] Tianjin Huang, Yulong Pei, Vlado Menkovski, and Mykola Pechenizkiy. Hop-count based self-supervised anomaly detection on attributed networks. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 225–241. Springer, 2022.
  - [13] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 3122–3126, 2021.
  - [14] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa T Phan, and Yi-Ping Phoebe Chen. Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12220–12233, 2021.
  - [15] Jiaqiang Zhang, Senzhang Wang, and Songcan Chen. Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks. *arXiv preprint arXiv:2205.04816*, 2022.
  - [16] Jingcan Duan, Siwei Wang, Xinwang Liu, Haifang Zhou, Jingtao Hu, and Hu Jin. Gadmsl: Graph anomaly detection on attributed networks via multi-scale substructure learning. *CoRR*, 2022.
  - [17] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. Graph self-supervised learning: A survey. *IEEE transactions on knowledge and data engineering*, 35(6):5879–5900, 2022.
  - [18] Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7603–7611, 2022.
  - [19] Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter*, 24(2):61–77, 2022.
  - [20] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, pages 3663–3674, 2021.
  - [21] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33:22092–22103, 2020.
  - [22] Yun Zhu, Jianhao Guo, Fei Wu, and Siliang Tang. Rosa: A robust self-aligned framework for node-node graph contrastive learning. *arXiv preprint arXiv:2204.13846*, 2022.

- [23] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.
- [24] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1150–1160, 2020.