基于对比自监督学习的属性网络异常检测

摘要

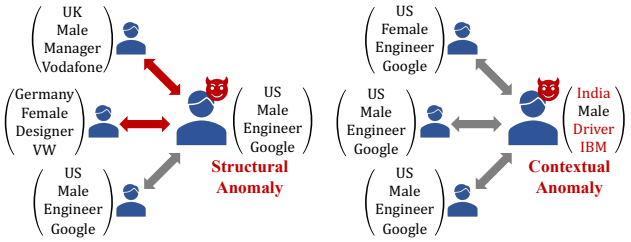
由于属性网络在建模各种复杂系统中的广泛应用，属性网络上的异常检测引起了广泛的研究关注。近年来，基于深度学习的异常检测方法在高维属性和复杂结构的网络上相较于浅层方法表现出了更好的结果。但是现有的方法通常将图自编码器作为其骨干网络，并未充分利用网络的丰富信息，导致性能次优。此外，这些方法并未在学习目标中直接针对异常检测进行优化，并且由于全图训练机制，无法扩展到大规模网络。为了解决这些问题，本文提出了一种新颖的对比自监督学习框架，用于属性网络上的异常检测（简称 CoLA）。该框架通过采样一种新型的对比实例对，充分利用网络数据中的局部信息，这些实例对能够无监督地捕捉每个节点与其邻接子结构之间的关系。本文还提出了一种基于图神经网络的对比学习模型，利用高维属性和局部结构学习信息丰富的嵌入，并衡量每个实例对与其输出得分的一致性。通过对对比学习模型的多轮预测得分进行统计估计，进一步评估每个节点的异常性。通过这种方式，学习模型通过特定的异常检测目标进行训练。此外，由于图神经网络模块的输入是实例对的批次，而非全图，所以该框架能够灵活地适应大规模网络。实验结果表明，所提出的框架在所有七个基准数据集上均优于最先进的基线方法。

关键词：无监督学习；图神经网络；对比自监督学习；异常检测；属性网络

1 引言

属性网络（也称为属性图），其中节点的属性表示真实世界中的实体，而边表示实体之间的关系，在多个场景中普遍存在，包括金融（交易网络）、社交媒体（社交网络）、以及电子商务（商品-用户网络）。为了利用属性网络数据解决实际问题，近年来吸引了大量研究兴趣的图分析任务包括节点分类、图分类、和链接预测。在这些任务中，属性网络上的异常检测是一个至关重要的研究问题。异常检测旨在检测那些明显偏离大多数实例的情况（在属性网络中，数据实例通常是节点）。异常检测在许多与安全相关的应用中具有重要意义，例如欺诈检测和社交垃圾信息检测。

然而，在属性网络上有效地检测异常并非易事，因为异常的种类繁多且缺乏监督。由于属性网络同时包含属性信息和结构信息，它们通常包含不同类型的异常。图 1 提供了一个示例，说明了两种基本类型的异常：结构异常和上下文异常。结构异常的属性信息通常是正常的，但它们与其他节点之间有几个异常的链接。与此不同的是，上下文异常具有自然的邻接结构，但它们的属性被破坏（噪声或与所有邻居完全不同）。这种多样性使得直接将针对仅属性数据（如 OC-SVM）或普通网络（如 LOF）的异常检测方法应用于属性网络变得困难。因此，一个有效的异常检测方法应考虑多种异常模式。此外，由于获取异常的真实标签的成本



(a) 结构异常　　(b) 上下文异常

图 1. 属性网络中不同类型的异常

过高，属性网络上的异常检测主要以无监督方式进行 [1, 2]。也就是说，算法必须在没有监督的情况下从损坏的网络中推断出正常的数据模式。因此，关键是充分而合理地利用来自属性网络数据的现有信息。

最近，针对属性网络的异常检测任务，提出了多种方法。浅层方法包括 AMEN [2]、Radar [3] 和 ANOMALOUS [4]，它们利用浅层学习机制（如自我网络分析、残差分析或 CUR 分解）来检测异常。不幸的是，由于浅层机制的局限性，这些模型无法充分应对属性网络中的计算挑战，并且未能捕捉到不同信息模态之间的复杂交互，尤其是在特征高维时。随着深度学习在异常检测领域的飞速发展，研究人员也提出了基于深度神经网络的方法来解决属性网络上的异常检测问题。DOMINANT [1] 是其中的代表性方法之一。它构建了一个图自编码器，同时重建属性信息和结构信息，通过重建误差评估异常。SpecAE [5] 也利用图自编码器提取低维嵌入，并通过密度估计进行异常检测。

尽管现有的基于深度学习的方法在图形上的异常检测取得了相当的性能，但它们仍然存在几个缺点，这些缺点主要归因于其架构中的自编码器主干。首先，自编码器旨在通过重建原始数据来学习潜在表示，而不是直接检测异常。尽管可以根据重建误差计算异常得分，但由于这些方法并不直接针对异常检测目标，因此只能实现次优性能。其次，基于自编码器的方法可能无法充分利用属性图的丰富信息进行有效的图表示学习。具体来说，自编码器仅仅是重建原始数据，并没有对数据进行任何优化。然而，最近的研究 [6] 显示，如果我们根据增强数据精心设计某些前置任务，便可以在无监督的方式下挖掘出更多有用的信息。第三，图自编码器是执行大规模网络异常检测的瓶颈。通常，图自编码器中的图卷积操作需要输入并重建整个网络数据，而当网络规模较大时，由于巨大的内存需求，这种方法是不可行的。

作为一种替代的无监督学习技术，自监督对比学习是解决上述限制的有前景的解决方案。通过学习对比精细的实例对，模型可以在没有人工标签的情况下获取有价值的信息。对比自监督学习对于异常检测任务具有良好的特性。首先，对比学习主要研究实例对的匹配，这为异常检测提供了有用的信息。对于图中的正常实例，每个节点与其邻居之间存在潜在的匹配模式，例如同质假设。相反，异常通常表现为属性与结构之间的不一致/不匹配，这违反了网络的原始匹配模式。此外，不同类型的异常有不同的匹配失效方式：在图 1 中，结构异常具有与不相关节点的异常边，这是部分不一致；而上下文异常则与所有邻居的属性不匹配。自然地，对比学习能够通过其内在的判别机制学习匹配模式并捕捉各种不匹配模式。第二，对比学习模型提供了一个特定的预测得分，用于衡量每个实例对中元素之间的一致性，这个得分与实例的异常性高度相关。由于异常检测方法通常输出一系列得分或排名来表示每个节点的异常性，因此对比学习模型的预测得分可以直接用于异常检测。通过这种方式，我们可以

通过与异常检测高度相关的目标来训练模型。

在本文中，我们提出了一种新颖的自监督对比学习框架用于属性网络上的异常检测（简称 **CoLA**）。通过从完整网络中采样精心设计的实例对，并利用它们训练对比学习模型，网络信息得到了更好的利用。具体而言，我们的框架专注于建模每个节点与其部分邻近子结构之间的关系，这有助于揭示网络中的各种异常类型。同时，我们的 CoLA 框架具有直接的训练目标，以辅助异常检测任务。我们将模型的学习目标设置为区分实例对内元素之间的一致性，结果可以进一步用于评估节点的异常性。此外，通过将网络分割成独立的轻量级实例对，我们的异常检测框架与大规模网络兼容。具体来说，我们的框架无需在完整网络上运行图卷积，从而成功避免了内存爆炸问题。总结起来，本文的主要贡献如下：

- 我们提出了一种基于对比自监督学习的框架 CoLA，用于属性网络上的异常检测问题。据我们所知，这是首个用于图异常检测的对比自监督学习方法。
- 我们提出了一种新型的对比实例对“目标节点 vs. 局部子图”，旨在将其应用于属性网络中的异常检测任务，该方法有效地捕捉节点及其邻接子结构的局部信息。
- 我们设计了一个对比学习模型，从节点-子图实例对中学习表征信息，并提供用于异常性排序的判别分数。所提出的学习模型对大规模网络数据友好。
- 我们在多个数据集上进行广泛实验，展示了 CoLA 的有效性，并与一系列基准方法进行了比较，证明了其优越性。

本文其余部分的组织结构如下：在第2节，我们首先回顾相关工作；然后，在第3节介绍实现方法；第4节详细阐述复现细节；接下来，在第5节中，我们分析了实验结果；最后，在第6节中对我们的工作进行了总结。

2 相关工作

在本节中，我们介绍了最相关的工作：网络嵌入与图神经网络、属性网络上的异常检测以及对比学习。

2.1 网络嵌入与图神经网络

网络嵌入旨在将节点嵌入到潜在的向量空间中，在该空间中图的固有属性得以保留。对于属性网络，学习到的嵌入应该同时包含结构信息和语义信息。例如，SNE [7] 使用神经网络建模结构和属性之间的相互关系。TriDNR [8] 通过包括节点的结构、属性和标签在内的三方信息源联合学习节点嵌入。NETTENTION [9] 利用对抗训练机制和自注意力模块学习有用的节点嵌入。

图神经网络（GNNs）是一类深度神经网络 [10]，用于建模非欧几里得网络/图数据的内在关系 [11]。GNN 的概念最早在 [12] 中提出。此后，提出了一系列基于谱的 GNN，这些方法从图信号处理的角度使用滤波器。GCN [13] 执行谱图卷积的局部一阶近似，以高效地学习节点表示。GAT [14] 引入了注意力机制 [15]，通过自适应权重聚合邻居的信息。一些最近的工作尝试从不同方向改进 GNN，例如简化计算复杂度，使用对抗训练方案，应用于大规模图，

以及引入新颖的算子。目前，GNN 已被应用于多个研究领域，如时间序列预测，高光谱图像分类和知识图谱。

在我们提出的 CoLA 中，GNN 是对比学习模型的重要组成部分。我们选择 GCN 作为 GNN 模块的骨干。灵活地，我们框架中的 GNN 模块可以设置为上述任意类型的 GNN。

2.2 属性网络上的异常检测

近年来，由于属性网络在建模复杂系统中的广泛应用，属性网络上的异常检测吸引了大量的研究兴趣。AMEN [2] 通过利用每个节点的自我网络信息来检测属性网络上的异常。Radar [3] 特征化属性信息的残差及其与网络信息的一致性来进行异常检测。进一步，ANOMALOUS [4] 结合 CUR 分解和残差分析来进行属性网络上的异常检测。Zhu 等人 [16] 提出了一个联合学习模型，通过核心初始化和扩展来检测混合异常。尽管这些方法在低维属性网络数据上取得了一定的成功，但由于其浅层机制的局限性，当网络结构复杂且属性维度较高时，这些方法表现不佳。

随着深度学习技术的迅猛发展 [10]，出现了几种深度方法来解决属性网络上的异常检测问题。DOMINANT [1] 构建了一个包含 GCN 层的自编码器，以重建属性矩阵和邻接矩阵。它将节点的异常得分定义为其属性和结构的重建误差的加权和。SpecAE [5] 利用谱图自编码器提取每个节点的潜在嵌入，并使用高斯混合模型（GMM）进行检测。对于动态网络，NetWalk [17] 通过随机游走采样和自编码器模型动态学习网络表示，并通过基于聚类的检测器进行异常检测。

上述深度方法通过引入深度神经网络，相较于浅层方法，取得了优越的性能，但也存在一些由于其自编码器重建机制导致的缺点。首先，重建是一种简单的无监督学习方法，未能充分利用数据。相反，CoLA 以自监督的方式更好地利用了属性和结构信息。其次，它们的重建优化目标与异常检测无关。与之不同，我们的学习目标是区分节点与子图之间的一致性，这可以直接指示节点的异常性。第三，这些方法需要完整的邻接矩阵和属性矩阵作为模型的输入，由于爆炸性的内存需求，这使得这些算法无法在大规模网络数据上运行。相比之下，我们的框架通过学习采样的实例对而非完整网络，从而使其能够灵活适应大规模网络。

2.3 对比自监督学习

对比自监督学习是自监督学习的一个重要分支。通过精心设计的对比前任务，这些方法通过对比正实例对与负实例对来学习表示 [18]。Deep InfoMax [19] 通过最大化局部补丁与其全局上下文之间的互信息（MI）来学习图像的嵌入。作为继承者，CPC [6] 通过最大化音频片段与其上下文音频之间的关联，将对比学习应用于语音识别。MoCo [20] 构建了一个动量编码器，通过动量更新的编码器生成对比嵌入。SimCLR [18] 利用不同的增强方法组合来构建配对样本。最近，BYOL [21] 提出了对比在线网络与目标网络的表示，其中两个网络相互学习。Simsiam [22] 采用了 Siamese 网络，通过对比方式学习视觉表示。

一些最近的研究也将对比方法应用于图学习。DGI [23] 将节点的表示和通过读取函数获得的图级摘要向量视为一个对比实例对，并通过图的腐蚀生成负样本。在 DGI 的基础上，Hassani 等人 [24] 提出了一个多视角对比学习框架，将原始图结构和图扩散视为两种不同的视角。GCC 通过为每个节点采样两个子图作为正实例对，使用 InfoNCE 损失 [6] 来进行预训

练，适用于宇宙图数据。GMI 考虑最大化节点嵌入与其邻居原始特征之间的相似度，以及两个相邻节点的嵌入之间的相似度。

然而，大多数现有的研究旨在学习数据表示，而非检测异常。为了适应异常检测，我们提出的 CoLA 框架在动机和实现上与上述方法存在本质的区别。从动机的角度来看，这些方法仅将对比模型的嵌入模块作为编码器，而在测试时，判别模块变得无用。相比之下，我们提出的 CoLA 框架利用整个对比模型来计算每个节点的异常得分。从实现的角度来看，由于现有的实例对定义无法有效捕捉节点的异常性，我们设计了一种新型的图对比学习实例对，它更加关注每个节点的局部信息，而不是全局特性。

3 本文方法

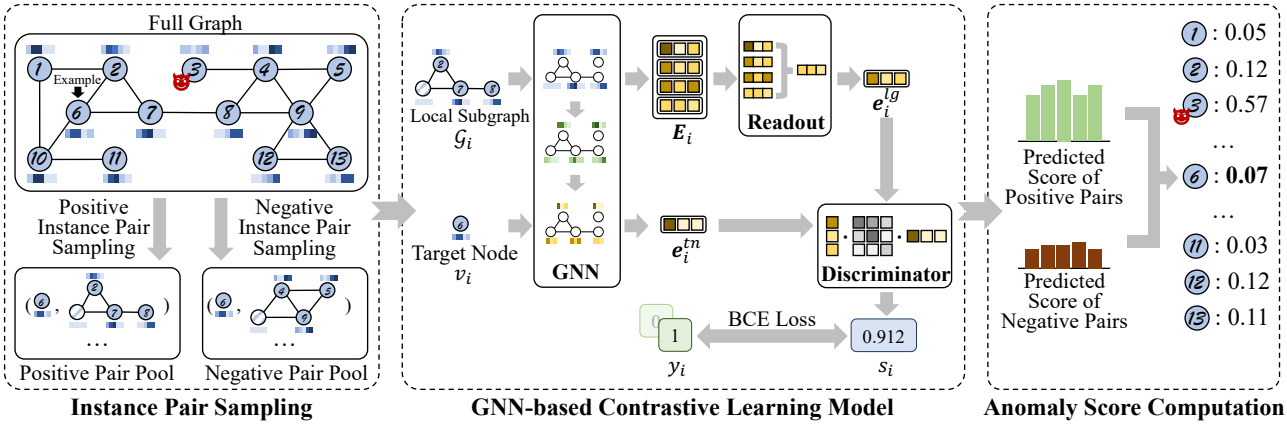


图 2. CoLA 总体框架

在本节中，我们描述了我们提出的 CoLA 框架，如图2所示。CoLA 在最高层次上由三个组件组成，分别是实例对采样、基于 GNN 的对比学习模型和异常得分计算。首先，为了生成用于对比自监督学习的基本样本，我们执行实例对采样，通过基于局部子结构的采样策略来采样精心设计的“目标节点 vs. 局部子图”实例对。我们设计的实例对能够充分利用原始数据，重点关注每个节点及其局部邻居。随后，基于 GNN 的对比学习模型使用 GNN 和读取模块提取目标节点和局部子图的低维嵌入，并通过判别器为每个实例对计算出一个区分性得分。评估目标节点与子图之间一致性的预测得分可以进一步指示相应目标节点的异常性。因此，模型的训练是通过与异常检测任务相关的目标来指导的。最后一步，异常得分计算是通过异常得分来衡量所有节点的异常性，因此我们可以通过对得分进行排名来识别异常。具体来说，每个节点的异常得分是通过多轮采样获得的正/负对的预测得分来计算的。检测结果可以看作是对每个节点与其局部子结构兼容性的多次观察的期望。在本节的其余部分，我们将详细介绍框架的三个主要组件（子节3.1到3.3）。然后，在子节3.4中，我们描述 CoLA 的整体流程和算法。在子节3.5中，我们分析了框架的时间复杂度。

3.1 对比实例对定义

对比学习框架的成功在很大程度上依赖于对比实例对的定义。与计算机视觉或自然语言处理任务中，实例可以简单地定义为图像或句子不同，图中实例的定义并不那么明确。一些

先前的工作已经在图中定义了不同类型的实例对，例如“完整图 v.s. 节点”、“大子图 v.s. 节点”和“子图 v.s. 子图”。然而，这些定义都没有针对异常检测任务进行设计或优化。由于节点的异常性通常与节点自身及其邻域结构之间的关系相关，因此我们应该设计一种新的对比实例对类型，以捕捉这种局部特性。

为了建模网络中节点的局部分布模式，我们对比实例对的定义侧重于节点与其包围子结构之间的关系。具体来说，我们为属性网络上的异常检测设计了“目标节点 v.s. 局部子图”实例对。实例对的第一个元素是一个单一节点，在我们的框架中称为“目标节点”。目标节点可以是网络中的任何一个节点。实例对的第二个元素是从初始节点采样的局部子图。对于正例实例对，初始节点设置为目标节点，然后采样的子图由目标节点的邻居节点组成。对于负例实例对，初始节点从除目标节点之外的所有节点中随机选择。因此，负例对中的目标节点与局部子图之间存在不匹配。

这种设计的主要动机是，属性网络中的异常通常反映在节点与其局部邻居之间的不一致性上，而全局信息通常与异常无关。如图 1 所示，两种类型的异常节点与其邻近邻居存在不匹配。我们的设计有意聚焦于通过学习“节点-局部子图”匹配模式来挑选出这种不匹配。与此不同，现有的工作（例如 DGI）主要考虑节点的全局属性，这对于网络嵌入有帮助，但对检测异常的贡献较小。在第 5.2 小节的对比实验中，展示了我们设计的实例对在捕捉异常方面的重要性。

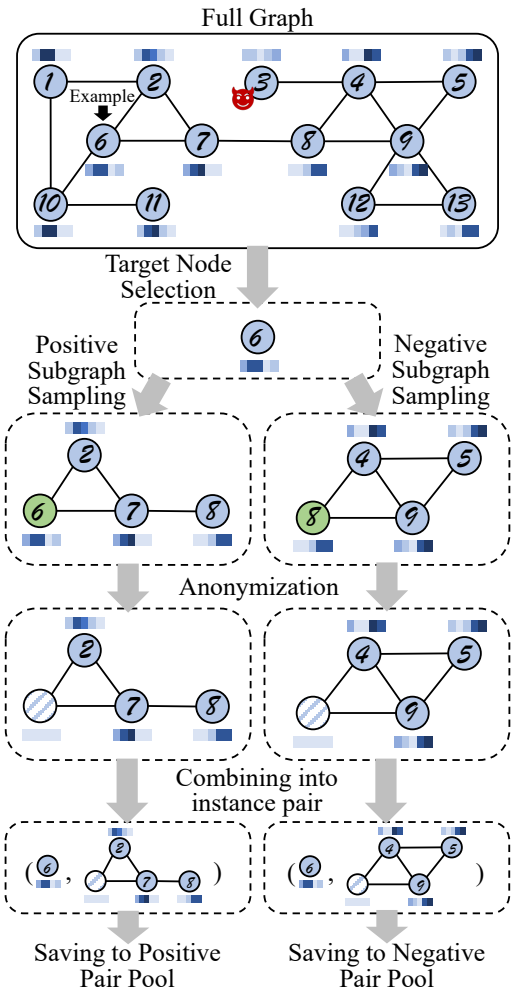


图 3. 对比实例对的采样过程

如图 3 所示，简单的示意图展示了我们提出的对比实例对的采样过程。采样过程包括四个步骤：目标节点选择、子图采样、匿名化和组合成实例对。

(1) **目标节点选择**。首先需要指定一个目标节点。在实际操作中，我们在每个训练周期内按随机顺序遍历图中的每个节点作为目标节点。因此，目标节点选择可以视为一个不重复的随机采样过程。

(2) **子图采样**。本地子图被定义为与初始节点相邻的子结构。如上所述，我们通过将初始节点设置为目标节点和随机采样节点来分别为正实例对和负实例对采样子图。受到启发，我们采用了带重启的随机游走（RWR）作为本地子图采样策略，因为它具有良好的实用性和效率。我们框架中也支持其他图采样算法，如森林火灾采样。

(3) **匿名化**。匿名化的目的是防止对比学习模型轻易识别本地子图中目标节点的存在。具体来说，我们将初始节点的属性向量设置为零向量。这样，目标节点的信息就被隐藏了。

(4) **组合成实例对**。最后的步骤是将目标节点和相关的子图组合成实例对。组合完成后，正实例对和负实例对分别保存到相应的样本池中。

3.2 基于 GNN 的对比学习模型

采样得到的实例对用于训练基于 GNN 的对比学习模型。对于具有标签的实例对 P_i，其中包含的数据可以表示为：

$$\mathcal{G}_i = (\mathbf{A}_i, \mathbf{X}_i), \tag{1}$$

并且 y_i 是 P_i 的标签，可以表示为：

$$y_i = \begin{cases} 1, \ P_i \text{ 是一个正实例对} \\ 0, \ P_i \text{ 是一个负实例对} \end{cases}. \tag{2}$$

正如图 2 中部所示，我们提出的基于 GNN 的对比学习模型由三个主要组件组成：GNN 模块、读出模块和判别模块。

3.2.1 GNN 模块

GNN 模块的目标是聚合本地子图中节点之间的信息，并将高维属性转化为低维嵌入空间。本地子图 $\mathcal{G}_i$ 被输入到具有多个堆叠层的 GNN 中，其中单层可以表示为：

$$\mathbf{H}_i^{(\ell)} = GNN(\mathbf{A}_i, \mathbf{H}_i^{(\ell-1)}; W^{(l-1)}), \tag{3}$$

其中，$\mathbf{H}_i^{(\ell-1)}$ 和 $\mathbf{H}_i^{(\ell)}$ 分别是由第 $(\ell-1)$ 层和第 ℓ 层学习到的隐藏表示矩阵，$W^{(\ell-1)}$ 是第 $(\ell-1)$ 层的可学习参数集。对于一个 L 层的 GNN，输入表示 $\mathbf{H}_i^{(0)}$ 被定义为属性矩阵 $\mathbf{X}_i$，输出表示 $\mathbf{H}_i^{(L)}$ 是子图节点的嵌入表示，记作 $\mathbf{E}_i$。$GNN(\cdot)$ 可以设置为任何类型的主流 GNN，例如 GCN [13]、GAT [14] 或 GIN。在实际应用中，我们采用 GCN，因为它具有较高的效率。然后，方程 (3) 可以具体写为：

$$\mathbf{H}_i^{(\ell)} = \phi\left(\widetilde{\mathbf{D}}_i^{-\frac{1}{2}}\widetilde{\mathbf{A}}_i\widetilde{\mathbf{D}}_i^{-\frac{1}{2}}\mathbf{H}_i^{(\ell-1)}\mathbf{W}^{(\ell-1)}\right), \tag{4}$$

其中，$\widetilde{\mathbf{A}}_i = \mathbf{A}_i + \mathbf{I}$ 是带有自环的子图邻接矩阵，$\widetilde{\mathbf{D}}_i$ 是本地子图的度矩阵，$\mathbf{W}^{(\ell-1)} \in \mathbb{R}^{d^{(\ell-1)} \times d^{(\ell)}}$ 是第 $(\ell-1)$ 层的权重矩阵，$\phi(\cdot)$ 是激活函数，例如 ReLU。

为了将它们对比于相同的特征空间，除了本地子图中的节点外，目标节点也应该映射到相同的嵌入空间。由于单个节点没有结构信息，我们仅使用 GCN 的权重矩阵和相应的激活函数来转换目标节点的属性。具体来说，这种转换可以视为一个深度神经网络（DNN）：

$$\mathbf{z}_i^{(\ell)} = \phi\left(\mathbf{z}_i^{(\ell-1)}\mathbf{W}^{(\ell-1)}\right), \tag{5}$$

其中，$\mathbf{z}_i^{(\ell-1)}$ 和 $\mathbf{z}^{(\ell)}i$ 分别是通过 $(\ell-1)$ 层和 ℓ 层学习得到的目标节点的隐藏表示行向量，$\mathbf{W}^{(\ell-1)}$ 是与 GCN 共享的权重矩阵。输入 $\mathbf{z}^{(0)}i$ 定义为目标节点 $\mathbf{x}v_i$ 的属性行向量，输出则标记为目标节点嵌入 $\mathbf{e}^{tn}i$。

3.2.2 读取模块

我们读取模块的目标是将子图中节点的嵌入 $\mathbf{E}_i$ 转换为一个本地子图嵌入向量 $\mathbf{e}_i^{lg}$。为了简化，我们使用平均池化函数作为我们的读取函数，这在之前的研究中已被广泛使用。具体来说，读取函数可以写作如下：

$$\mathbf{e}_i^{lg} = Readout(\mathbf{E}_i) = \sum_{k=1}^{n_i} \frac{(\mathbf{E}_i)_k}{n_i}, \tag{6}$$

其中，$(\mathbf{E}i)\,k$ 是 $\mathbf{E}_i$ 的第 k 行，n_i 是本地子图 $\mathcal{G}_i$ 中节点的数量。

3.2.3 判别器模块

判别器模块是我们对比学习模型的核心组件。它对实例对中两个元素的嵌入进行对比，并输出最终的预测分数。在这里，我们采用一个简单的双线性评分函数，该函数也被 [6] 使用。预测分数可以通过以下公式计算：

$$s_i = Discriminator(\mathbf{e}_i^{lg}, \mathbf{e}_i^{tn}) = \sigma\left(\mathbf{e}_i^{lg}\mathbf{W}^{(d)}\mathbf{e}_i^{tn\top}\right), \tag{7}$$

其中，$\mathbf{W}^{(d)}$ 是判别器的权重矩阵，$\sigma(\cdot)$ 是逻辑 sigmoid 函数。

3.2.4 目标函数

通过整合上述三个组件，我们提出的基于 GNN 的对比学习模型可以被视为一个二分类模型，用于预测对比实例对的标签：

$$s_i = CLM(v_i, \mathcal{G}_i), \tag{8}$$

其中，$CLM(\cdot)$ 是对比学习模型。

在这里，我们的目标是使得预测的 s_i 和真实标签 y_i 尽可能接近。因此，我们采用标准的二元交叉熵（BCE）损失，作为我们的目标函数，这也是二分类问题中的常见选择。其有效性已

经在其他对比自监督学习工作中得到了验证 [19]。具体来说，对于一个批次的 $P_i = (v_i, \mathcal{G}_i, y_i)$，批次大小为 N，目标函数如下所示：

$$\begin{aligned}\mathcal{L} &= -\sum_{i=1}^{N} y_i \log(s_i) + (1-y_i)\log(1-s_i) \\ &= -\sum_{i=1}^{N} y_i \log(CLM(v_i, \mathcal{G}_i)) + \\ &\quad (1-y_i)\log(1-CLM(v_i, \mathcal{G}_i)).\end{aligned} \tag{9}$$

需要注意的是，与 [19] 中使用的 softplus 版本 BCE 中使用的标签平衡版本 BCE 不同，这里采用的是普通的 BCE。原因在于，我们在采样正负实例对时执行了平衡采样。为了适应未来研究中的更复杂采样策略，我们的框架中也可以替换为相应的目标函数。

3.3 异常分数计算

在对比学习模型训练完成后，我们得到一个分类器，用于判断子结构和节点之间的一致性。一个理想的 GNN 模型，如果参数数量合适，通常会学习到正常样本的匹配模式，因为正常样本在训练数据中占据了绝大多数。而对于异常样本，由于其不规则性和多样性，模型更难以拟合其模式。在理想情况下，对于一个正常节点，其正实例对的预测分数 $s^{(+)}$ 应该接近 1，而负实例对的预测分数 $s^{(-)}$ 应该接近 0。对于一个异常节点，其正负实例对的预测分数则会变得不那么具有区分度（接近 0.5），因为模型无法很好地区分其匹配模式。基于上述特性，对于每个节点 v_i，我们可以简单地将其异常分数定义为负分数和正分数之间的差值：

$$f(v_i) = s_i^{(-)} - s_i^{(+)}. \tag{10}$$

然而，采样得到的本地子图只能视为目标节点邻域结构的部分观测，这不能代表目标节点的整个邻居分布。不完整的观测会导致对异常的感知不完整，从而影响异常检测的性能。例如，一些结构异常节点可能与多个与其无关的异常节点存在异常连接，而它们的大多数邻居是正常的。那么，如果我们仅通过一次采样来估计异常，一旦正常邻居被采样到本地子图中，这种异常就会被忽视。

为了解决这个问题，我们提出使用多轮正负采样的预测得分来生成异常分数。具体而言，对于每个节点 v_i 在带属性网络中，我们通过在子节 3.1 中介绍的采样策略采样 R 个正实例对和 R 个负实例对，其中 R 是采样轮数。这些实例对分别表示为 $(P_{i,1}^{(+)}, \cdots, P_{i,R}^{(+)})$ 和 $(P_{i,1}^{(-)}, \cdots, P_{i,R}^{(-)})$。然后，这些实例对被输入到对比学习模型 $CLM(\cdot)$ 中，计算预测得分 $(s_{i,1}^{(+)}, \cdots, s_{i,R}^{(+)}, s_{i,1}^{(-)}, \cdots, s_{i,R}^{(-)})$，这些得分通过公式 (8) 计算。最后，通过计算负实例对和正实例对得分之间多轮差值的平均值，得到 v_i 的异常分数：

$$f(v_i) = \frac{\sum_{r=1}^{R}(s_{i,r}^{(-)} - s_{i,r}^{(+)})}{R}, \tag{11}$$

其中 $f(\cdot)$ 是异常分数映射函数，它是我们异常检测框架的最终目标。

从统计角度来看，执行 R 轮采样是为了估计节点邻近子结构与远程子结构之间的正常性差异。原则上，R 越大，估计越准确。在实际操作中，我们将 R 设为框架的超参数。

Algorithm 1: CoLA 的整体过程

Input: $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{X})$, 训练轮数 T, 批量大小 B, 采样轮数 R

Output: 异常分数映射函数 $f(\cdot)$

随机初始化对比学习模型的参数 $(\mathbf{W}^{(0)},\cdots,\mathbf{W}^{(L)},\mathbf{W}^{(d)})$;

训练阶段

for $t=1,2,\cdots,T$ **do**

  $\mathcal{B}\leftarrow$ 随机将 $\mathcal{V}$ 划分为大小为 B 的批次;

  for 批次 $b=(v_1',\cdots,v_B')\in\mathcal{B}$ **do**

    采样正实例对 $(P_1^{(+)},\cdots,P_B^{(+)})$，其中 $(v_1',\cdots,v_B')$ 是目标节点;

    采样负实例对 $(P_1^{(-)},\cdots,P_B^{(-)})$，其中 $(v_1',\cdots,v_B')$ 是目标节点;

    计算预测分数 $(s_1^{(+)},\cdots,s_B^{(+)},s_1^{(-)},\cdots,s_B^{(-)})$ 通过公式 (8);

    计算损失函数 $\mathcal{L}$ 通过公式 (9);

    反向传播并更新对比学习模型的参数 $(\mathbf{W}^{(0)},\cdots,\mathbf{W}^{(L)},\mathbf{W}^{(d)})$;

推理阶段

for $v_i\in\mathcal{V}$ **do**

  采样 R 轮正实例对 $(P_{i,1}^{(+)},\cdots,P_{i,R}^{(+)})$，其中 v_i 为目标节点;

  采样 R 轮负实例对 $(P_{i,1}^{(-)},\cdots,P_{i,R}^{(-)})$，其中 v_i 为目标节点;

  计算预测分数 $(s_{i,1}^{(+)},\cdots,s_{i,R}^{(+)},s_{i,1}^{(-)},\cdots,s_{i,R}^{(-)})$ 通过公式 (8);

  计算节点 v_i 的异常分数 $f(v_i)$ 通过公式 (11);

此外，计算均值是处理多轮结果的最简单方法。从理论上讲，还可以挖掘更多的统计属性，如方差、最小值/最大值以及分布特性。然而，在实践中，我们发现，与引入上述因素相比，计算平均值是最有效的解决方案。尽管如此，我们仍然认为，进一步挖掘多轮预测分数的统计特性是未来潜在的研究方向之一，因为不同类型的异常会展现出不同的分数分布特征。

3.4 CoLA：一种异常检测框架

在本小节中，我们介绍了所提 CoLA 框架的整体流程。该流程分为两个阶段：训练阶段和推理阶段。在训练阶段，使用采样的实例对以无监督方式训练对比学习模型。随后，在推理阶段，获得每个节点的异常分数。

我们 CoLA 框架的整体流程如算法1所示。在训练阶段的每个 epoch 中，首先将节点集合 $\mathcal{V}$ 划分为若干个小批次。然后，在每次迭代中，为当前小批次中的每个节点采样一个正实例对和一个负实例对。接着，计算这些实例对的对应预测分数，然后计算 BCE 损失。为了优化对比学习模型的参数，通过梯度下降算法执行反向传播。训练阶段结束后，进行多轮正负采样过程来生成异常分数。如小节3.3所述，对于每个节点 v_i，采样 R 个正实例对和 R 个负实例对，然后将它们输入训练好的模型以计算预测分数。最后，利用公式 (11) 获得异常分数。

关于大规模网络异常检测的讨论。如上所述，对比学习模型是在每次迭代中独立地通过一个小批次的实例对进行训练的。同时，异常分数的计算也是完全独立的。也就是说，CoLA 的空间复杂度与节点数 n 完全无关。这个优点使得我们提出的 CoLA 框架可以应用于大规模网络。当网络规模较大（即 n 很大）时，我们不需要将整个网络输入到 GCN 模型中，因为

这样会因空间复杂度的爆炸性增长而变得不可行。相反，在我们的框架中，整个网络被分解为实例对，我们所需要做的只是调整批次大小和子图大小，以满足内存限制。

3.5 复杂度分析

我们通过分别考虑三个主要组件来分析所提框架的时间复杂度。对于实例对采样，每个 RWR 子图采样的时间复杂度是 $\mathcal{O}(c\delta)$（其中 δ 是网络的平均度数）。在推理阶段，我们对每个节点进行 R 轮采样，因此总的时间复杂度为 $\mathcal{O}(cn\delta R)$。对于基于 GNN 的对比学习模型，时间复杂度主要由 GNN 模块产生，对于每对实例，其时间复杂度为 $\mathcal{O}(c^2)$，总的时间复杂度为 $\mathcal{O}(c^2nR)$。对于异常分数计算，其时间复杂度远低于上述两个阶段，因此在这里我们忽略这一项。综上所述，CoLA 的总体时间复杂度为 $\mathcal{O}(cnR(c+\delta))$。

4 复现细节

4.1 实验环境搭建

实验在虚拟平台 Anaconda 上进行，编程语言为 Python，所使用的相应库版本要求如下：

- Python>=3.7
- PyTorch>=1.8.1
- Numpy>=1.19.2
- Scipy>=1.6.2
- Scikit-learn>=0.24.1
- Networkx>=2.5.1
- Tqdm>=4.59.0
- DGL==0.4.0

4.2 代码对比

对论文源代码的核心代码部分做的改动如下所示：

在我们的实验过程中，最初我们采用了为每个目标节点采样单个子图的策略。然而，在分析了模型的表现后，我们发现采样多个子图显著提高了模型的准确性。具体来说，通过为每个目标节点采样两个子图，模型能够学习到更加鲁棒且具有代表性的特征，从而提升了性能。

此外，在原始方法中，计算损失函数时我们仅关注对比学习模块。这使得我们能够计算目标节点与正负样本之间的相似度。然而，这只是学习过程中的一个方面。我们提出通过引入生成模块来扩展损失函数。通过这一调整，整体损失函数将由两个部分组成：对比损失和生成损失。通过微调这两个部分的相对权重，我们能够更好地引导模型学习样本的潜在模式。这种双重损失方法不仅确保模型能够捕捉数据的判别性和生成性特征，还提高了其泛化能力，从而带来了显著的准确性提升。原代码和改进后的代码如下所示：

原代码

```
#子图采样
subgraphs = generate_rwr_subgraph(dgl_graph, subgraph_size)
#loss计算
loss_all = b_xent(logits, lbl)
loss = torch.mean(loss_all)
```

改进后代码

```
#子图采样
subgraphs_1 = generate_rwr_subgraph(dgl_graph, subgraph_size)
subgraphs_2 = generate_rwr_subgraph(dgl_graph, subgraph_size)
#loss计算
loss_all = b_xent(logits, lbl)
loss1 = torch.mean(loss_all)
loss2 = 0.5 * (mse_loss(f_1[:, -2, :], raw_bf1[:, -1, :]) + mse_loss(f_2[:, -2, :], raw_bf2[:, -1, :]))
loss = args.alpha * loss1 + args.beta * loss2
```

4.3 创新点

基于上述描述，本研究的创新点可以总结为以下几点：

(1) **多子图采样策略：**通过对目标节点采样多个子图（具体为两个子图），模型能够从更丰富的局部结构中学习特征，相比于单一子图采样，显著提升了模型的表现和准确性。这种策略能够有效增强模型对节点邻域特征的学习能力，提高其对复杂数据的适应性。

(2) **双重损失函数设计：**在传统的对比学习框架中，损失函数主要关注目标节点与正负样本之间的相似度。我们在此基础上引入了生成模块，提出了一种包含对比损失和生成损失的双重损失函数。通过控制这两部分损失的权重，能够更好地引导模型学习样本的潜在特征，提升其辨别和生成能力，从而在提高准确率的同时，增强了模型的泛化能力。

这两项创新不仅改善了模型的性能，还为未来的图学习和异常检测任务提供了新的思路。

5 实验结果分析

在本节中，我们通过实验展示 CoLA 框架的有效性。我们首先介绍用于实验的数据集。接着，我们展示实验结果。

5.1 数据集

我们在七个广泛使用的基准数据集上评估了所提出的框架，这些数据集用于属性网络中的异常检测。数据集包括两个社交网络数据集（BlogCatalog 和 Flickr）以及五个引文网络数据集（Cora、Citeseer、Pubmed、ACM 和 ogbn-arxiv）。这些数据集的统计信息如表1所示，具体描述如下：

表 1. 数据集的统计信息

Dataset	♯ nodes	♯ edges	♯ attributes	♯ anomalies
BlogCatalog	5,196	171,743	8,189	300
Flickr	7,575	239,738	12,407	450
ACM	16,484	71,980	8,337	600
Cora	2,708	5,429	1,433	150
Citeseer	3,327	4,732	3,703	150
Pubmed	19,717	44,338	500	600
ogbn-arxiv	169,343	1,166,243	128	6000

- **社交网络**.BlogCatalog 和 Flickr 是两个典型的社交网络数据集，分别来自博客分享网站 BlogCatalog 和图片托管与分享网站 Flickr。在这些数据集中，节点表示网站的用户，边表示用户之间的关注关系。在社交网络中，用户通常会生成个性化的内容，例如发布博客或分享带标签描述的照片，因此这些文本内容被视为节点的属性。

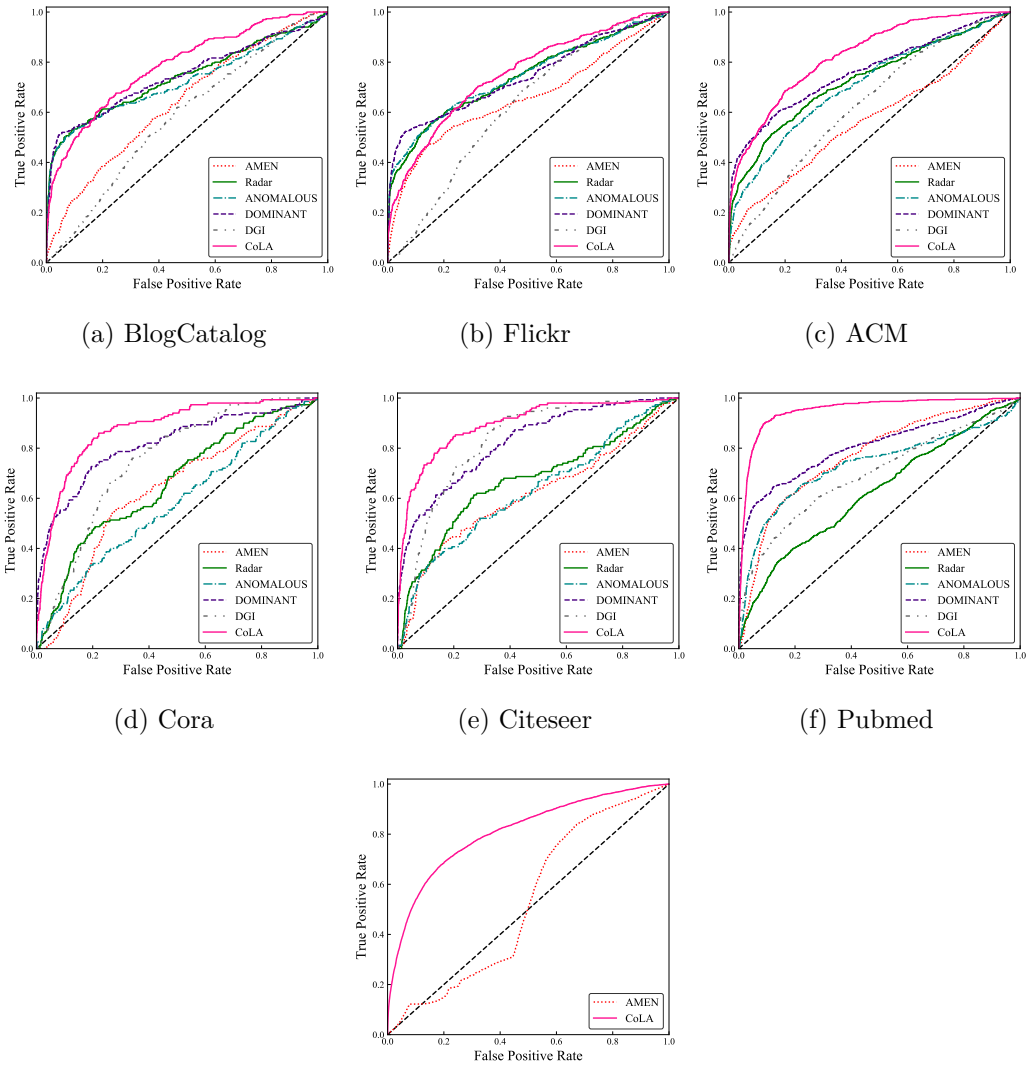
- **引文网络**. ora、Citeseer、Pubmed、ACM 和 ogbn-arxiv 是五个公开的可用数据集，这些数据集由科学出版物组成。在这些网络中，节点表示已发布的论文，而边表示论文之间的引用关系。对于前四个数据集，每个节点的属性向量是基于词袋模型的表示，其维度由字典大小决定。对于 ogbn-arxiv 数据集，每个节点有一个 128 维的属性向量，该向量是通过平均论文标题和摘要中的单词嵌入得到的。需要注意的是，ogbn-arxiv 数据集是来自 Open Graph Benchmark (OGB) 的一个大规模图数据集，包含超过 16.9 万个节点和 110 万个边。

5.2 异常检测结果

在本小节中，我们通过与基准方法的比较，评估了所提框架在异常检测上的表现。ROC 曲线的比较结果如图4所示。通过计算 ROC 曲线下的面积，七个基准数据集的 AUC 得分如表2所示，以便进行比较。根据结果，我们得到以下几点观察：

- 在所有七个数据集上，我们提出的 CoLA 框架都实现了最佳的异常检测性能。特别是，相较于基线方法的最佳结果，我们的框架在 AUC 上的平均提升达到了 6.44%。其主要原因在于，CoLA 通过实例对采样成功捕捉了每个节点与其局部子结构之间的关系，并利用基于 GNN 的对比学习模型从上下文和结构信息中提取了具有判别力的分数。

- 与基于自编码器的深度方法 DOMINANT 相比，CoLA 取得了显著的性能提升，尤其是在五个引用网络数据集上。主要原因有两点：（1）CoLA 通过构建实例对很好地利用了网络数据，而不是仅仅重构原始数据；（2）CoLA 的目标与异常检测目标相关，这使得学习模型能够生成具有区分度的分数，从而实现最终的异常排名。



(a) BlogCatalog (b) Flickr (c) ACM

(d) Cora (e) Citeseer (f) Pubmed

(g) ogbn-arxiv

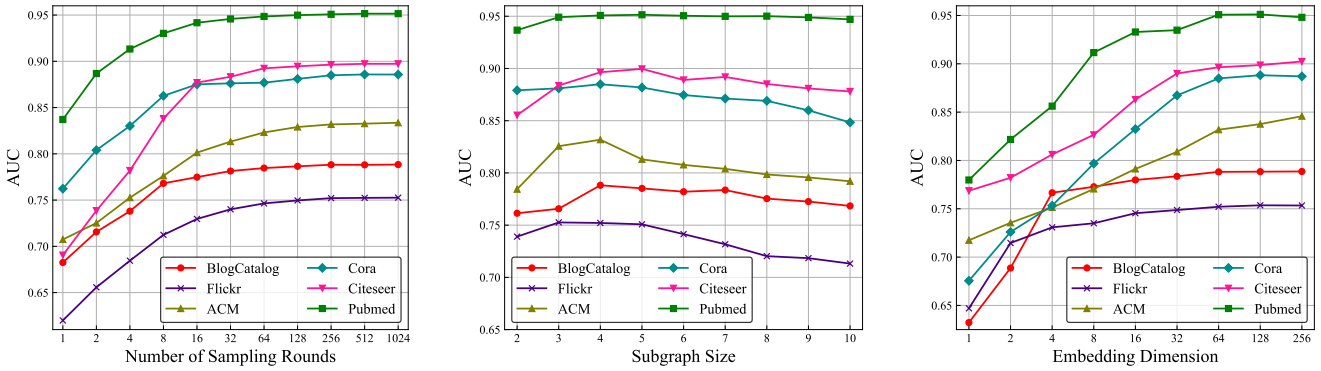
图 4. 在七个基准数据集上的 ROC 曲线比较

表 2. 在七个基准数据集上的 AUC 值比较

Methods	Blogcatalog	Flickr	ACM	Cora	Citeseer	Pubmed	ogbn-arxiv
AMEN	0.6392	0.6573	0.5626	0.6266	0.6154	0.7713	0.5279
Radar	0.7401	0.7399	0.7247	0.6587	0.6709	0.6233	OOM
ANOMALOUS	0.7237	0.7434	0.7038	0.5770	0.6307	0.7316	OOM
DOMINANT	0.7468	0.7442	0.7601	0.8155	0.8251	0.8081	OOM
DGI	0.5827	0.6237	0.6240	0.7511	0.8293	0.6962	OOM
CoLA	**0.7854**	**0.7513**	**0.8237**	**0.8779**	**0.8968**	**0.9512**	**0.8073**

5.3 参数研究

在本小节中，我们探讨了三个重要参数对所提框架性能的影响：采样轮次、子图大小和潜在嵌入维度。由于效率的限制，我们仅在六个小规模数据集上进行这些实验。



(a) 采样轮次与 AUC 值的关系　(b) 子图大小与 AUC 值的关系　(c) 嵌入维度与 AUC 值的关系

图 5. 参数研究的实验结果

5.3.1 采样轮次 R 的影响

在这个实验中，我们修改了 R 的值，以研究其对 AUC 的影响。性能变化结果如图 5a 所示。从图中可以看出，当检测结果仅通过一次采样计算时，检测性能较差。随着采样轮次的增加，各数据集的 AUC 在一定范围内有了显著提升。然而，当 R 大于 256 时，增大 R 所带来的性能提升变得很小。实验结果实证证明了我们在小节 3.3 中的分析：随着 R 的增大，每个节点的异常估计变得更加准确。基于此结果，我们在其他实验中将 R 设置为 256，以平衡性能和效率。

5.3.2 子图大小 c 的影响

我们进一步分析了子图大小 c 对不同数据集的重要性。图 5b 展示了不同子图大小选择下的 AUC 得分。如图所示，当 c 极小（$c = 2$）时，AUC 相对较低。可能的原因是，在这种情况下，子图仅包含目标节点本身及其一个邻居，除了这两个节点之间的连接外，没有考虑其他结构信息。缺乏足够的邻域结构信息导致了较差的性能。在一定的范围内，AUC 随着 c 的增大而增加。然后，当 $c > 5$ 时，随着 c 的增大，检测性能下降。一个合理的解释是：当子图大小较大时，离目标节点较远的节点会被采样进子图。然而，这些远离节点通常与异常无关，反而成为了我们检测任务中的“噪声信息”。在所有数据集中，当 c 的值约为 4 时，能够获得更好的检测性能。因此，为了保证运行效率和在所有数据集上的鲁棒性，我们将 c 的值固定为 4。

5.3.3 嵌入维度 d 的影响

我们探讨了嵌入维度 d 对 CoLA 框架的敏感性。我们改变了 d 的值，观察其对我们方法性能的影响。CoLA 的性能变化如图 5c 所示。对于每个数据集，随着嵌入维度的增大，AUC

值逐渐提高。当将嵌入维度从 1 维增加到 32 维时，异常检测性能稳步上升；但当我们进一步增大 d 时，性能提升变得较为轻微。我们观察到，对于大多数数据集，64 维的潜在嵌入可以为下游的对比学习和异常检测任务提供足够的信息。因此，为了考虑效率，我们将超参数 d 设置为 64。

6 总结与展望

本文首次尝试将对比自监督学习应用于带属性网络的异常检测问题。我们提出了一种新的异常检测框架——CoLA，该框架由三个主要组件组成：对比实例对采样、基于 GNN 的对比学习模型和基于多轮采样的异常分数计算。我们的模型成功捕捉了每个节点与其邻接结构之间的关系，并通过与异常相关的目标来训练对比学习模型。通过在七个基准数据集上的一系列实验，证明了所提出的框架在解决带属性网络异常检测问题上的有效性和优越性。

我们相信，所提出的框架为将自监督学习和对比学习扩展到日益增多的图异常检测应用开辟了新的机会。在未来的工作中，我们将把基于自监督对比学习的异常检测方法扩展到更复杂的网络/图数据，如异构图、时空图和动态图。

参考文献

[1] 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.

[2] 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.

[3] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu. Radar: Residual analysis for anomaly detection in attributed networks. In *IJCAI*, pages 2152–2158, 2017.

[4] Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, and Qinghua Zheng. Anomalous: A joint modeling approach for anomaly detection on attributed networks. In *IJCAI*, pages 3513–3519, 2018.

[5] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. Specae: Spectral autoencoder for anomaly detection in attributed networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2233–2236, 2019.

[6] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[7] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat Seng Chua. Attributed social network embedding. In *arXiv:1705.04969*, 2017.

[8] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. In *IJCAI*, pages 1895–1901, 2016.

[9] W. Yu, W. Cheng, C. Aggarwal, B. Zong, H. Chen, and W. Wang. Self-attentive attributed network embedding through adversarial learning. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 758–767, 2019.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[11] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21, 2020.

[12] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005.

[13] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, pages 1–14, 2017.

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, pages 1–12, 2018.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[16] Mengxiao Zhu and Haogang Zhu. Mixedad: A scalable algorithm for detecting mixed anomalies in attributed graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1274–1281, 2020.

[17] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2672–2681, 2018.

[18] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, pages 1–24, 2018.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, pages 21271–21284, 2020.

[22] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[23] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, pages 1–17, 2019.

[24] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of International Conference on Machine Learning*, pages 3451–3461. 2020.