软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007579] [CSTR: 32375.14.jos.007579]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于核心-边缘结构感知的图对比学习方法*

王业江[1,2], 赵宇海[1,2], 黄苗苗[1], 王梅霞[1], 李芳婷[1], 王兴伟[1]

[1](东北大学 计算机科学与工程学院, 辽宁 沈阳 110167)

[2](医学影像智能计算教育部重点实验室 (东北大学), 辽宁 沈阳 110167)

通信作者: 赵宇海, E-mail: zhaoyuhai@mail.neu.edu.cn

**摘　要**: 图自监督学习旨在无需人工标注的条件下学习有效的图结构表示. 尽管图对比学习 (graph contrastive learning, GCL) 通过构造标签保持不变的扰动视图并最大化其与原始视图的相似性来实现自监督训练, 但现有方法普遍采用全局均匀扰动策略, 未能考虑真实网络中节点角色的异质性——这种无差别的随机破坏会违反标签不变性假设. 实际观测表明, 真实网络普遍呈现核心-边缘的双层拓扑架构: 核心节点通过高度互连形成信息枢纽, 对其进行破坏性扰动将导致语义失真与标签偏移. 针对上述问题, 提出基于核心-边缘结构感知的图对比学习框架, 其创新性体现在: (i) 摒弃全局均匀扰动范式, 通过核心-边缘检测算法精准定位边缘节点, 实施局部扰动以保持核心拓扑完整性, 严格遵循标签不变性原则; (ii) 设计了边缘节点删除等增强操作, 模拟真实网络的动态演化过程, 强制模型捕获拓扑稳定性与噪声鲁棒性特征; (iii) 构建了核心-边缘对比损失函数, 对不同结构重要性的节点在损失计算中赋予差异化权重, 有效引导模型关注核心信息并抑制边缘潜在的负面干扰. 在多个基准数据集上的实验表明, 所提方法在多个任务中均显著优于现有最优模型.

**关键词**: 图对比学习; 自监督学习; 表示学习

**中图法分类号**: TP18

中文引用格式: 王业江, 赵宇海, 黄苗苗, 王梅霞, 李芳婷, 王兴伟. 基于核心-边缘结构感知的图对比学习方法. 软件学报. http://www.jos.org.cn/1000-9825/7579.htm

英文引用格式: Wang YJ, Zhao YH, Huang MM, Wang MX, Li FT, Wang XW. Graph Contrastive Learning Method Based on Core-periphery Structure awareness. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7579.htm

## Graph Contrastive Learning Method Based on Core-periphery Structure awareness

WANG Ye-Jiang[1,2], ZHAO Yu-Hai[1,2], HUANG Miao-Miao[1], WANG Mei-Xia[1], LI Fang-Ting[1], WANG Xing-Wei[1]

[1](School of Computer Science and Engineering, Northeastern University, Shenyang 110167, China)

[2](Key Laboratory of Intelligent Computing of Medical Images (Northeastern University), Ministry of Education, Shenyang 110167, China)

**Abstract**: Graph self-supervised learning aims to acquire effective graph structural representations without manual annotations. Although graph contrastive learning (GCL) enables self-supervised training by generating label-preserving perturbed views and maximizing their similarity to the original view, existing methods predominantly adopt a globally uniform perturbation strategy, which neglects the heterogeneity of node roles in real-world networks. Such indiscriminate random corruption violates the label-invariance assumption. Empirical observations indicate that real-world networks generally exhibit a core-periphery bi-layered topological architecture: core nodes form highly interconnected information hubs, and destructive perturbations applied to them are likely to cause semantic distortion and label shifts. To address these limitations, this study proposes a core-periphery structure-aware graph contrastive learning framework, with the following innovations: (i) instead of global uniform perturbation paradigm, peripheral nodes are accurately identified via a core-periphery detection algorithm, and localized perturbations are applied to preserve the integrity of the core topology, thereby strictly satisfying the principle of label invariance; (ii) augmentation operations such as peripheral node deletion are designed to simulate the dynamic evolution

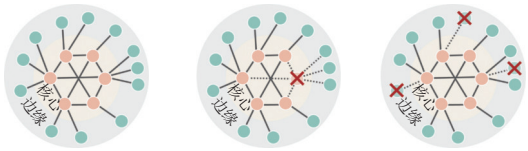
* 基金项目: 国家自然科学基金 (62432003, 92267206)
收稿时间: 2025-06-25; 修改时间: 2025-10-12; 采用时间: 2025-11-28; jos 在线出版时间: 2026-04-22

of real-world networks, encouraging the model to capture topological stability and noise robustness; (iii) a core-periphery contrastive loss function is constructed, in which differentiated weights are assigned to nodes with varying structural importance during loss computation, effectively guiding the model to emphasize core information while suppressing potential negative interference from peripheral nodes. Extensive experiments on multiple benchmark datasets demonstrate that the proposed method consistently outperforms state-of-the-art models across various tasks.

**Key words**: graph contrastive learning (GCL); self-supervised learning; representation learning

图自监督学习 (self-supervised graph learning) 旨在无需人工标注的情况下, 从真实网络/图数据中学习出有效的图结构表示. 近年来, 图对比学习 (graph contrastive learning, GCL)[1−3]成为图自监督学习领域中最成功的范式之一. GCL 首先通过图增强操作 (如随机节点删除) 对原始 (锚) 图生成扰动视图, 然后将锚图与其正样本视图的表示相似性最大化, 而与其他不同锚图视图的表示相似性最小化. 这一过程的核心思想在于假设增强操作所产生的扰动视图与原始图具有标签不变性, 即扰动不应改变图数据本质的语义信息. 这种标签不变性假设能否成立, 极大程度上取决于所使用的图增强方法及其对图结构所施加的扰动性质[4−6].

然而, 现有方法普遍采用全局均匀的随机扰动策略, 并未考虑真实网络中节点角色的异质性. 这种无差别的随机破坏往往会违反标签不变性假设, 进而损害模型的泛化性能[7]. 实际观测表明, 大量真实世界网络普遍呈现明显的核心-边缘 (core-periphery, CP) 的双层拓扑架构[8−10]: 核心节点通常高度互连、形成网络的信息枢纽, 而边缘节点则稀疏连接、承担次要角色[11]. 因此, 对核心节点的破坏性扰动往往会造成网络语义的严重失真和标签的偏移. 例如, 如图 1 所示, 图 1(a) 为具有明显核心-边缘结构的原始网络, 核心节点紧密连接形成信息中心. 当现有随机节点删除策略破坏核心节点时 (图 1(b)), 网络被严重割裂, 原本连通的图结构发生解体, 这种破坏的连锁反应导致了多个边的丢失和语义的扭曲, 违背了图对比学习所需的标签不变性原则. 这种现象说明, 忽略节点角色差异的全局均匀扰动策略可能会降低获得不变表示的有效性, 进而限制图对比学习的表现.




(a) 原始图的核心-边缘双层结构　(b) 全局均匀扰动造成核心节点破坏, 导致拓扑断裂与语义偏移　(c) 局部边缘扰动保持核心完整性

图 1　不同扰动策略对核心-边缘拓扑图结构的影响对比

为了解决上述问题, 本文提出了基于核心-边缘结构感知的图对比学习 (core-periphery structure-aware graph contrastive learning, CP-GCL) 框架, 具体来说, 该框架具有以下 3 个方面特点.

首先, 本文摒弃了现有普遍采用的全局均匀扰动范式[12], 通过引入能够量化节点“核心度”的识别算法, 得以精准识别网络中的核心与边缘区域, 并在此基础上实施针对性的局部扰动. 具体来说, 本文仅对边缘节点进行扰动操作 (例如随机删除或微调), 而严格保持核心节点及其关联拓扑结构的完整性, 以确保网络的整体语义信息不会发生剧烈变化, 从而更好地遵循图对比学习所依赖的标签不变性原则. 如图 1(c) 所展示的例子所示, 当有选择性地删除边缘节点时, 网络整体结构仍然能够保持连通状态, 尽管局部结构略有变动, 但核心结构依旧完整. 与图 1(b) 中全局随机删除核心节点导致网络结构解体的情形相比, 这种局部扰动策略显著降低了语义失真与标签偏移的风险, 保证了扰动样本与原始样本之间的语义一致性.

其次, 为了更好地模拟真实网络在实际场景中的动态演化过程[13], 进一步设计了一系列基于边缘节点的增强操作, 包括边缘节点的随机删除等. 这种设计旨在强制模型从动态演化的视角捕获网络拓扑结构的稳定性和对局部噪声的鲁棒特征. 具体而言, 真实网络往往表现出边缘节点的频繁波动 (例如社交网络中用户的退出或者通信网络中的临时连接等), 而核心节点则相对稳定. 因此, 通过模拟边缘节点的动态变化, 能够有效地激励模型聚焦于网络结构的本质特征, 忽略由边缘波动带来的局部噪声, 从而获得更具泛化能力的表征.

最后, 本文构建了核心-边缘对比损失. 该损失函数利用前述核心-边缘识别算法计算得到的节点核心度分数, 作为权重因子来调整每个节点在对比学习目标中的贡献. 具体地, 核心度高的节点 (即核心节点) 在损失计算中被赋予更大的权重, 而核心度低的节点 (即边缘节点) 则被赋予较小的权重. 这种设计使得模型在优化过程中更加关注核心节点的表示学习, 确保其与全局图表示的高度一致性, 并强力区分其与负样本的差异; 同时, 适度降低了对边缘节点表示的优化强度, 从而有效抑制了边缘区域局部噪声的干扰, 并防止了模型在次要细节上的过拟合. 通过这种方式, 模型能够更有效地学习到反映网络关键结构和节点异质性的高质量表示.

综上所述, 本文的主要贡献可归纳为以下几点.

● 将图论中的核心-边缘结构概念引入图自监督学习领域, 从节点角色异质性的视角提出针对性解决方案, 以严格维护图对比学习的标签不变性假设.

● 设计基于核心-边缘结构感知的局部扰动策略及边缘感知对比损失函数, 有效避免了全局随机扰动所导致的语义失真, 并提高了模型对网络拓扑演化的鲁棒性.

● 在多个基准数据集以及下游任务中进行的广泛实验验证了本文提出的方法在节点分类预测任务上的显著优势, 证明了所提方法的有效性与鲁棒性.

本文第 1 节介绍图自监督学习的相关方法和研究现状. 第 2 节介绍本文构建的基于核心-边缘结构感知的图对比学习方法. 第 3 节通过对比实验验证了所提模型的有效性. 最后总结全文.

## 1 相关工作

图自监督学习旨在从大量无标签的图数据中学习富有信息的表示, 以减轻对数据标注的依赖. 近年来, 图对比学习 (GCL) 作为该领域中最主流和研究最广泛的范式[14−16], 已被成功应用于药物发现[17,18]、基因组学分析[19,20]等多个领域. 当前图对比学习方法成功的关键在于其图增强策略, 即通过对原始图进行变换以生成语义一致的正样本对, 从而引导模型学习到对标签信息不变的普适性表示[12,21]. 目前, 主流的图增强策略大致可分为基于规则的增强和基于学习的增强. 基于规则的增强方法遵循预定义规则来改变图数据, 这些方法通常对图的整体结构或特征进行操作, 例如随机扰动或掩盖部分节点与边[22,23]、基于随机游走等方式采样生成子图[24,25], 或是利用如个性化 PageRank 排名等图扩散过程来创建保留全局结构信息的视图[26,27]. 相对地, 基于学习的增强方法则以数据驱动的方式学习增强模式, 旨在从全局视角生成更优的对比视图, 例如学习一个用于下游任务的增强图结构[28−30]或通过对抗性训练生成具有挑战性的困难样本[31,32].

核心-边缘 (CP) 结构是网络科学中研究的一种重要的特征, 它将网络节点划分为一个连接紧密的核心 (core) 和一个连接稀疏的边缘 (periphery)[33,34]. 在这种结构中, 核心节点之间以及核心与边缘节点之间通常存在密集的连接, 而边缘节点之间的连接则相对稀疏. 另一种定义则认为核心节点是那些到网络中所有其他节点的距离都很短的节点. 这种核心-边缘模式在众多现实世界的网络中普遍存在, 例如在全球贸易网络中, 经济体量大的国家构成了核心, 而小经济体则形成边缘[35,36]; 在机场网络中, 大型枢纽机场为核心, 连接着其他主要机场及区域性机场, 而小型机场间航班较少[37,38]; 在学术引文网络里, 高影响力论文被广泛引用构成核心, 而普通论文之间互相引用较少[39−41]; 此外, 在社交网络[42−44]和生物网络[44−46]等领域也观察到了类似结构. 目前, 识别和量化网络核心-边缘结构的算法主要可以依据其核心思想分为两类[33,47]. 第 1 类是基于块模型的方法, 它们假设网络邻接矩阵或其生成概率矩阵具有特定的块状模式, 即核心密集、边缘稀疏. 典型方法包括最大化网络与理想 CP 结构模式矩阵的相关性[34], 或通过优化目标函数来最大化核心内部连接并最小化边缘内部连接[48], 也有方法利用随机块模型 (SBM) 进行似然最大化来推断节点归属[49], 以及基于点过程的生成模型来定义具有稠密核心的稀疏网络[50,51]. 第 2 类是基于传输或中心性的方法, 其核心思想是核心节点在网络中占据中心位置或与其他所有节点的距离较短. 例如, 有方法结合 $k$-core 分解与节点紧密度中心性来识别核心[8], 或通过评估节点移除对网络整体连通性 (如网络容量) 的影响来确定核心节点[52]. 此外, 还有一些采用其他思路的方法, 如基于随机游走的持续概率来定义边缘进而确定核心[53], 或利用基于路径的中心性度量来评估节点的“核心度”并指导划分[54]. 需要指出的是, 目前这些方法均侧

重于网络结构特征的识别与描述, 并主要解决如何从图的邻接矩阵直接提取出 CP 结构的问题, 而没有将核心-边缘结构信息作为一种知识或特征显式地融入现代图自监督学习框架中.

## 2 算法框架

现有的图对比学习方法通常利用图增强技术来生成不同的视图, 并隐含地假设这些视图在生成过程中保持了标签不变性. 标签不变性意味着增强操作 (如节点删除、边修改等) 不应改变图或节点的根本语义类别或功能角色, 这是对比学习能够有效学习区分性表示的基础. 然而, 许多工作采用全局性的、统一的扰动策略, 忽略了真实网络中广泛存在的核心-边缘异质结构. 这种无差别的扰动方式, 特别是当扰动触及构成网络信息枢纽的核心节点时, 很可能破坏图的关键结构和固有语义信息, 从而违背标签不变性假设, 限制了学习表示的质量.

为了克服这一限制, 本文提出了基于核心-边缘结构感知的图对比学习算法 (CP-GCL). 该算法的核心思路在于首先识别图中的核心节点与边缘节点, 然后将图增强操作有选择地仅应用于边缘节点. 这种策略主要基于两点考虑: 1) 通过保持核心结构的稳定性, 可以最大程度地维护图的本质语义, 从而更严格地满足标签不变性要求; 2) 针对边缘节点的删除等增强操作可以有效模拟真实世界网络动态演化的过程 (例如节点的加入或离开), 有助于模型捕获网络的动态稳定性与鲁棒性特征. 如图 2 所示, 该框架通过识别核心与边缘节点, 仅对边缘区域进行图增强以生成两个对比视图, 并以核心-边缘加权对比损失优化节点与图表示. 本文提出的框架具体执行流程如下: 1) 采用核心-边缘检测算法区分输入图中的核心节点与边缘节点; 2) 仅对识别出的边缘节点进行增强操作 (如随机删除部分边缘节点或添加新的边缘连接), 生成两个结构略有差异的对比视图; 3) 利用图神经网络 (GNN) 等图编码器分别对这两个视图进行编码, 得到节点级别的嵌入表示; 4) 本文引入设计的核心-边缘对比损失函数. 该损失函数会根据节点的核心度对不同节点在对比目标中的贡献进行加权, 更加关注核心节点表示与全局图表示的一致性, 同时适当降低边缘节点扰动可能引入的噪声影响, 从而引导模型学习到对核心-边缘结构具有感知能力的、更鲁棒的节点表示与图表示.

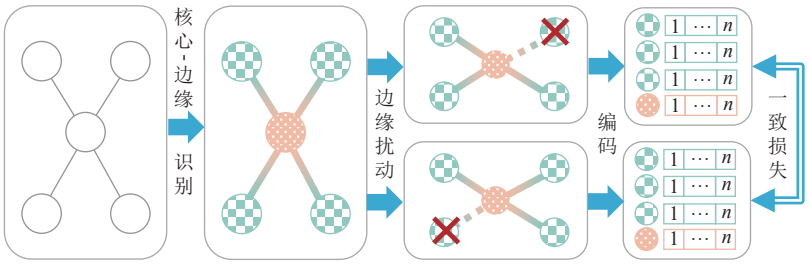



图 2 算法框架流程

### 2.1 问题定义

本文研究属性图上的自监督节点表示学习. 给定一个属性图 $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{X})$, 其中 $\mathcal{V}=\{v_1,\ldots,v_N\}$ 代表包含 $N$ 个节点的集合, $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ 代表边集, $\mathbf{A}\in\{0,1\}^{N\times N}$ 为邻接矩阵 (其中 $A_{ij}=1$ 表示节点 $v_i$ 和 $v_j$ 之间存在边), $\mathbf{X}\in\mathbb{R}^{N\times d}$ 为节点特征矩阵, $\boldsymbol{x}_i\in\mathbb{R}^d$ 是节点 $v_i$ 的 $d$ 维初始特征. 考虑到真实网络普遍存在的核心-边缘结构, 节点集 $\mathcal{V}$ ($\mathcal{V}=\mathcal{V}_C\cup\mathcal{V}_P$) 可被划分为核心节点子集 $\mathcal{V}_C$ 和边缘节点子集 $\mathcal{V}_P$, 其中核心节点内部连接稠密, 边缘节点连接相对稀疏. 本文的目标是在无监督范式下, 学习一个图编码器 $f_\theta\colon\mathcal{G}\to\mathbb{R}^{N\times k}$ ($\theta$ 为模型参数, $k\ll d$ 为嵌入维度), 为图中每个节点 $v_i$ 生成一个低维表示向量 $\boldsymbol{h}_i=f_\theta(\mathcal{G})_i\in\mathbb{R}^k$.

### 2.2 识别图的核心-边缘结构

在图对比学习框架中, 为了精准捕捉网络中节点角色的异质性并维护标签不变性假设, 首先需要对网络拓扑进行分析, 以区分承担不同结构角色的节点, 特别是识别出构成网络骨架的核心节点与处于外围的边缘节点. 传统的对核心-边缘结构的量化方法[34]通常旨在将节点划分到离散的“核心”或“边缘”类别中, 或者基于与单一理想化

模型 (例如, 核心全连接、边缘仅与核心连接) 的偏差来评分. 然而, 真实网络的结构往往更为复杂和多样, 可能包含不同规模、形态的核心, 甚至存在多个核心区域. 这些传统方法在处理此类复杂性时可能存在局限性. 因此, 为了更全面地捕捉节点在核心-边缘谱系中的位置, 本文采用了量化节点“核心度”的视角, 该视角能够为每个节点计算一个沿核心-边缘连续谱的数值[55]. 这种方法超越了简单的二元分类, 产生了基于核心-边缘结构的中心性度量, 能够更好地反映节点参与构成不同潜在核心结构的可能性和程度, 为后续根据节点的结构重要性进行差异化处理提供了更精细化的依据.

具体来说, 该识别方法的核心在于定义和优化一个核心质量函数 $R_\gamma$, 用以评估一个特定的节点排列相对于理想核心-边缘模式的匹配程度. 此处的 $\gamma$ 代表一组参数, 用于描述所考虑的理想核心结构的特征 (例如大小、形状等). 核心质量函数通常定义为网络中所有连接的加权贡献之和:

$$R_\gamma = \sum_{i,j} A_{ij} C_{ij} \tag{1}$$

其中, $A_{ij}$ 是图 $\mathcal{G}$ 的邻接矩阵元素 (对于无权图是 0 或 1, 对于有权图可以是边的权重), 表示节点 $i$ 和 $j$ 之间的连接强度. $C_{ij}$ 是一个由节点 $i$ 和 $j$ 的局部核心度值 $C_i$ 和 $C_j$ 共同决定的核心矩阵元素. $C_i \geqslant 0$ 代表节点 $i$ 在当前考虑的核心结构中的“核心”程度. $\gamma$ 则代表一组用于定义或约束核心结构形态的参数. 核心矩阵元素 $C_{ij}$ 的设计反映了理想核心-边缘结构中不同类型连接 (核心-核心、核心-边缘、边缘-边缘) 应有的贡献模式. 一种常用且直观的形式是乘积形式:

$$C_{ij} = C_i C_j \tag{2}$$

在这种形式下, 两个核心度值都高的节点之间的连接 ($C_iC_j$ 大) 对核心质量 $R_\gamma$ 的贡献最大, 而至少一个节点核心度值很低 (接近 0) 的连接 ($C_iC_j$ 小) 贡献则很小. 这符合核心节点内部紧密连接、边缘节点之间连接稀疏的直观预期.

为了确定每个节点的局部核心值 $C_i$, 本文不是直接优化 $C_i$, 而是首先定义一个理想化的目标核心向量 $C^*$. 这个向量的元素 $C_i^* = g(i)$ 由一个过渡函数 $g$ 决定, 通常 $i$ 代表节点的某种排序 (例如, 可以基于度或其他中心性指标的初步排序, 或者仅是节点索引). 过渡函数 $g$ 描绘了一种理想的核心度分布轮廓, 它通常包含参数 $\gamma$ 来控制这个轮廓的形状和大小. 在本文中使用锐利过渡函数定义为:

$$C_i^*(\alpha,\beta) = g_{\alpha,\beta}(i) = \begin{cases} \dfrac{i(1-\alpha)}{2\beta N}, & i \in \{1,\ldots,\beta N\} \\ \dfrac{(i-\beta N)(1-\alpha)}{2(N-\beta N)} + \dfrac{1+\alpha}{2}, & i \in \{\beta N+1,\ldots,N\} \end{cases} \tag{3}$$

其中, $N$ 是节点总数. 这里的参数 $\gamma = (\alpha,\beta)$ 中, $\beta \in [0,1]$ 控制了理想核心的相对大小 (核心大小从 $N$ 到 0), 而 $\alpha \in [0,1]$ 控制了核心与边缘之间过渡的“锐利”程度或核心度值的“跳跃”幅度 ($\alpha = 1$ 时产生最明显的两分结构). 在定义了目标核心向量 $C^*$ 后, 算法的目标是找到 $C^*$ 的一个最优排列将 $C^*$ 的各个元素值重新分配给图中的实际节点 $v_1,\ldots,v_N$, 得到实际的核心度向量 $C = (C_1,\ldots,C_N)$, 使得该分配下的核心质量 $R_\gamma$ (公式 (1)) 达到最大化. 这个寻找最优排列的过程通过模拟退火等启发式优化算法来求解.

通过对不同的参数组合 $\gamma = (\alpha,\beta)$ 重复上述优化过程, 可以得到一系列针对不同核心形态的最优核心度分配 $C(\gamma)$ 及其对应的最大核心质量 $R_\gamma$. 为了得到每个节点最终的、综合性的核心度评分, 可以计算其聚合核心分数 $CS(i)$:

$$CS(i) = Z \sum_{\gamma=(\alpha,\beta)} C_i(\gamma) \times R_\gamma \tag{4}$$

其中, $C_i(\gamma)$ 是节点 $i$ 在参数 $\gamma$ 下获得的最优核心度值, $R_\gamma$ 是对应的最大核心质量. $Z$ 是一个归一化因子, 通常选择使得 $\max_k[CS(k)] = 1$. 这个聚合核心分数 $CS(i)$ 综合考虑了节点在各种可能的高质量核心结构中的参与程度, 提供了一个稳健且具有信息量的连续核心度度量. $\alpha$ 和 $\beta$ 并不需要进行调优. 在本文的框架中, 该 $CS(i)$ 值可以用来区分核心与边缘节点. 具体来说, 通过设定一个阈值, 把分数高于阈值的节点核心放入节点集 $\mathcal{V}_C$, 反之则放入边

缘节点集 $\mathcal{V}_P$, 从而将核心-边缘结构信息更精细地融入图表示学习中.

### 2.3 图增强方法

在通过第 2.2 节所述方法识别出图 $\mathcal{G}$ 的核心节点集 $\mathcal{V}_C$ 与边缘节点集 $\mathcal{V}_P$ 后, 接下来将实施基于核心-边缘结构感知的图增强策略. 与现有 GCL 方法普遍采用的全局均匀扰动不同, 本框架仅对识别出的边缘节点子集 $\mathcal{V}_P$ 进行扰动操作, 以生成用于对比学习的两个视图 $\mathcal{G}'$ 和 $\mathcal{G}''$. 这种选择性增强策略具有如下优势: 1) 通过严格保护构成网络结构与功能核心的 $\mathcal{V}_C$ 及其关联边不被破坏, 能够最大限度地维持图的全局拓扑骨架和核心语义信息, 从而更可靠地满足对比学习所依赖的标签不变性假设. 2) 现实世界中的网络往往表现出边缘区域比核心区域更强的动态性 (例如社交网络中用户间的短期互动与关系解除等通常发生在边缘). 因此, 针对边缘节点的增强操作 (如节点/边的删除) 恰好模拟了这种网络演化的自然过程. 这不仅使得增强更具现实意义, 也符合动态系统研究中的光滑性原则, 即时间或结构上相近的图应具有相似的性质, 只有显著的变化才会导致性质的跃迁. 通过强制模型学习对这种边缘动态不敏感的表示, 可以有效提升其鲁棒性和对网络内在稳定结构的捕捉能力.

在本框架中, 本文采用了以下 4 种图增强技术, 将其限定作用于边缘节点 $\mathcal{V}_P$ 或与边缘节点相关的结构/特征, 以生成对比视图: 1) 边扰动: 该方法通过随机增加或删除一定比例的边来扰动图的拓扑结构. 在本文的框架下, 边扰动主要应用于以下两种情况: 边缘节点之间的边 (即 $\mathcal{V}_P\times\mathcal{V}_P$ 对应的子图) 和连接核心节点与边缘节点的边 (即 $\mathcal{V}_C\times\mathcal{V}_P$ 对应的连接). 形式上, 可以设定一个扰动率 $\alpha_e$, 并生成一个仅作用于上述相关边的扰动掩码 $\mathbf{E}_P$, 通过 $\tilde{\mathbf{A}}=\mathbf{A}\odot(\mathbf{1}-\mathbf{E}_P)+(\mathbf{1}-\mathbf{A})\odot\mathbf{E}_P$ (这里 $\odot$ 表示元素乘法, $\mathbf{1}$ 为全 1 矩阵) 得到增强后的邻接矩阵 $\tilde{\mathbf{A}}$. 这种方式模拟了网络中非核心连接的噪声或动态变化, 同时保持了核心内部连接的稳定性. 2) 边缘节点丢弃: 此方法随机移除图中的一部分节点及其关联的边. 本文将其限定为仅从边缘节点集 $\mathcal{V}_P$ 中按比例 $\alpha_n$ 随机选择节点进行丢弃. 具体地, 设被选中的边缘节点子集为 $\mathcal{V}_p'\subset\mathcal{V}_P$, 则增强后的图节点集为 $\tilde{\mathcal{V}}=\mathcal{V}\backslash\mathcal{V}_p'$, 对应的邻接矩阵和特征矩阵分别为 $\tilde{\mathbf{A}}=\mathbf{A}[\tilde{\mathcal{V}},\tilde{\mathcal{V}}]$ 和 $\tilde{\mathbf{X}}=\mathbf{X}[\tilde{\mathcal{V}},:]$. 这种操作模拟了网络边缘部分的节点自然流失现象, 由于丢弃的是重要性相对较低的边缘节点, 对图整体结构和功能的影响较小. 3) 边缘特征掩码: 该方法随机将节点特征矩阵中一部分元素替换为特定值 (如零、均值或随机噪声). 本文同样将其限定为仅对边缘节点集 $\mathcal{V}_P$ 中的节点特征进行掩码操作. 形式上, 对于每个边缘节点 $v_i\in\mathcal{V}_P$, 其特征向量 $\boldsymbol{x}_i$ 的每个维度以概率 $\alpha_f$ 被选中并替换. 设 $\mathbf{L}_P$ 为一个与 $\mathbf{X}$ 中边缘节点特征对应的掩码位置指示矩阵 (仅 $\mathcal{V}_P$ 行的对应元素可能为 1), $\mathbf{M}$ 为掩码值矩阵, 则增强后的特征矩阵 $\tilde{\mathbf{X}}$ 可表示为 $\tilde{\mathbf{X}}=\mathbf{X}\odot(\mathbf{1}-\mathbf{L}_P)+\mathbf{M}\odot\mathbf{L}_P$. 这模拟了边缘节点属性信息的不完整或存在噪声的情况. 4) 边缘节点特征混洗: 此方法在保持图拓扑结构不变的前提下, 随机打乱一部分节点的特征向量. 本文将其应用于边缘节点子集 $\mathcal{V}_p''\subset\mathcal{V}_P$ (例如随机选取比例为 $\alpha_s$ 的边缘节点). 定义一个随机置换 $\pi\colon\mathcal{V}_p''\to\mathcal{V}_p''$, 则增强后的特征矩阵 $\tilde{\mathbf{X}}$ 定义为: 对于所有 $i\in\mathcal{V}_p''$, $\tilde{\mathbf{X}}[i,:]=\mathbf{X}[\pi(i),:]$; 对于所有 $i\notin\mathcal{V}_p''$, $\tilde{\mathbf{X}}[i,:]=\mathbf{X}[i,:]$. 邻接矩阵保持不变, 即 $\tilde{\mathbf{A}}=\mathbf{A}$. 这种增强方式在不改变网络连接性的基础上, 扰乱了边缘区域节点与其自身属性的对应关系, 增加了学习表示的难度, 有助于模型关注结构信息. 通过组合或独立使用上述仅作用于边缘的增强方法, 可以为原始图 $\mathcal{G}$ 生成两个结构和特征略有差异但核心保持稳定的对比视图 $\mathcal{G}'$ 和 $\mathcal{G}''$.

### 2.4 核心-边缘对比损失

在通过第 2.3 节所述的图增强方法生成了两个结构略有差异但核心保持稳定的对比视图之后, 接下来需要设计一个有效的损失函数来指导图编码器学习节点与图表示. 传统的图对比学习方法通常平等对待所有节点, 最大化 (或最小化) 所有节点表示与图 (或其他节点) 表示之间的相似性. 然而, 正如前文所述, 真实网络普遍存在核心-边缘的异质结构. 核心节点作为网络的结构与功能骨架, 其局部信息理应与整个图的全局信息 (图表示) 高度相关且一致, 反映了它们在网络拓扑中的中心地位与关键作用. 相反, 边缘节点由于连接相对稀疏, 其在网络中的影响力有限, 并且它们的特征与表示可能更容易受到局部异常或数据噪声的干扰. 因此, 我们期望边缘节点的局部信息与全局信息的关联性相对较弱.

基于上述考虑, 本文将第 2.2 节识别出的节点核心度信息显式地融入学习过程, 提出了一种核心-边缘对比损失 (core-periphery contrastive loss, CPCL). 该损失函数的核心思想是将第 2.3 节得到的连续核心度分数向量

$\mathbf{CS}=[CS_1,\ldots,CS_n]^{\mathrm{T}}$ 作为权重, 融入对比学习的目标中, 以自适应地调控每个节点在损失计算中的贡献. 具体而言, 本文的目标是最大化一个视图中的节点表示与另一个视图中的图表示之间的互信息, 同时最小化与负样本 (例如, 来自特征混洗图的节点表示) 的互信息. 通过引入核心度权重 $CS_i$, 使得模型在优化过程中更加关注核心节点, 同时适度降低对边缘节点的关注.

设 $f_\theta$ 为图编码器 (如图神经网络 GNN), $p_\psi$ 为节点表示的投影头, $p_\phi$ 为图表示的投影头 (包含一个 READOUT 函数). 对于两个增强视图 $\mathcal{G}'$ 和 $\mathcal{G}''$, 首先通过编码器和投影头得到它们的节点表示矩阵 $\mathbf{H}'=p_\psi(f_\theta(\mathcal{G}'))\in\mathbb{R}^{n\times k}$ 和 $\mathbf{H}''=p_\psi(f_\theta(\mathcal{G}''))\in\mathbb{R}^{n\times k}$, 以及图表示向量 $\vec{h}'_g=p_\phi(\mathrm{READOUT}(f_\theta(\mathcal{G}')))\in\mathbb{R}^k$ 和 $\vec{h}''_g=p_\phi(\mathrm{READOUT}(f_\theta(\mathcal{G}'')))\in\mathbb{R}^k$. 为了构建负样本, 本文采用特征混洗策略. 具体来说先生成一个特征被随机置换的图 $\tilde{\mathcal{G}}=(\mathbf{A},\pi(\mathbf{X}))$, 其中 $\pi$ 是一个随机置换操作. 然后, 对 $\tilde{\mathcal{G}}$ 应用与生成 $\mathcal{G}'$ 和 $\mathcal{G}''$ 相同的增强过程 (或直接使用 $\tilde{\mathcal{G}}$ 作为基础), 得到对应的混洗节点表示 $\tilde{\mathbf{H}}'$ 和 $\tilde{\mathbf{H}}''$. 接着定义一个评分函数 (也称为判别器) $\mathcal{D}\colon\mathbb{R}^k\times\mathbb{R}^k\to\mathbb{R}$, 用于衡量节点表示和图表示之间的相似性 (在本文使用点积 $\mathcal{D}(\vec{u},\vec{v})=\vec{u}\cdot\vec{v}^{\mathrm{T}}$). 提出的核心-边缘对比损失旨在最大化正样本对 (一个视图的图表示与另一个视图的真实节点表示) 的评分, 同时最小化负样本对 (一个视图的图表示与另一个视图的混洗节点表示) 的评分, 并根据节点的核心度 $CS_i$ 进行加权. 采用基于 JSD (Jensen-Shannon divergence) 估计器的形式, 损失函数可以表述为最小化目标:

$$\begin{aligned}\mathcal{L}_{\mathrm{CPCL}}=-\frac{1}{N}\sum_{i=1}^{N}CS_i\Big[&\mathbb{E}_{(\mathcal{G}',\mathcal{G}'')}\Big(\log\sigma\Big(\mathcal{D}\Big(\vec{h}''_i,\vec{h}'_g\Big)\Big)+\log\sigma\Big(\mathcal{D}\Big(\vec{h}'_i,\vec{h}''_g\Big)\Big)\Big)\\&+\mathbb{E}_{(\mathcal{G}',\tilde{\mathcal{G}}''),(\mathcal{G}'',\tilde{\mathcal{G}}')}\Big(\log\Big(1-\sigma\Big(\mathcal{D}\Big(\tilde{\vec{h}}''_i,\vec{h}'_g\Big)\Big)\Big)+\log\Big(1-\sigma\Big(\mathcal{D}\Big(\tilde{\vec{h}}'_i,\vec{h}''_g\Big)\Big)\Big)\Big)\Big]\end{aligned}\tag{5}$$

其中, $N$ 是节点数量, $\vec{h}'_i$ 和 $\vec{h}''_i$ 分别是节点 $v_i$ 在视图 $\mathcal{G}'$ 和 $\mathcal{G}''$ 中的表示, $\tilde{\vec{h}}'_i$ 和 $\tilde{\vec{h}}''_i$ 是节点 $v_i$ 在特征混洗图对应视图中的表示, $\vec{h}'_g$ 和 $\vec{h}''_g$ 是视图 $\mathcal{G}'$ 和 $\mathcal{G}''$ 的图表示, $\sigma(x)$ 是 Sigmoid 函数, $\mathbb{E}$ 表示对图增强和混洗过程取期望. 本文提出的算法伪代码见算法 1.

**算法 1**. 基于核心-边缘结构感知的图对比学习算法.

输入: 无标签图 $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathbf{X})$; 图编码器 $f_\theta$; 核心度计算参数 (如 $\alpha$, $\beta$); 边缘增强算子集合 $A_{\mathrm{aug}}$; 训练周期数 $E_{\max}$; 学习率 $\eta$.

1. 初始化图编码器 $f_\theta$ 的参数;
2. 计算各节点 $v_j\in\mathcal{V}$ 的核心度 $C_j$;
3. 根据 $\{C_j\}$ 识别边缘节点集 $\mathcal{V}_P$;
4. **for** 训练周期 $e=1$ **to** $E_{\max}$ **do**
5. 从 $A_{\mathrm{aug}}$ 中随机选择增强算子 $aug_1$, $aug_2$;
6. $\mathcal{G}',\mathcal{G}''\leftarrow aug_1(\mathcal{G},\mathcal{V}_P),aug_2(\mathcal{G},\mathcal{V}_P)$; ▷ 对边缘节点增强生成视图
7. $\tilde{\mathcal{G}}',\tilde{\mathcal{G}}''\leftarrow\mathrm{Shuffle}(\mathcal{G}'),\mathrm{Shuffle}(\mathcal{G}'')$; ▷ 节点特征随机置换
8. $\mathbf{H}',\mathbf{H}''\leftarrow f_\theta(\mathcal{G}'),f_\theta(\mathcal{G}'')$; ▷ 视图对的节点表示
9. $h_{g'},h_{g''}\leftarrow\mathrm{READOUT}(\mathbf{H}'),\mathrm{READOUT}(\mathbf{H}'')$; ▷ 视图对的图表示
10. $\tilde{\mathbf{H}}',\tilde{\mathbf{H}}''\leftarrow f_\theta(\tilde{\mathcal{G}}'),f_\theta(\tilde{\mathcal{G}}'')$; ▷ 混淆视图对的节点表示
11. $\mathcal{L}\leftarrow 0$;
12. **for** $j=1$ **to** $|V|$ **do**
13. $\mathrm{term}_j\leftarrow C_j[\log\sigma(\mathcal{D}(h_{j''},h_{g'}))+\log\sigma(\mathcal{D}(h_{j'},h_{g''}))+\log(1-\sigma(\mathcal{D}(\tilde{h}_{j''},h_{g'})))+\log(1-\sigma(\mathcal{D}(\tilde{h}_{j'},h_{g''})))$
▷ $h'_j$ 为 $\mathbf{H}'$ 中节点 $v_j$ 的表示, $\tilde{h}'_j$ 为 $\tilde{\mathbf{H}}'$ 中节点 $v_j$ 的表示
14. $\mathcal{L}\leftarrow\mathcal{L}+\mathrm{term}_j$;
15. **end**

16. $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$; ▷ 更新编码器参数
17. **end**

提出的 CPCL 损失函数通过引入核心度 $CS_i$ 进行加权, 实现了对网络中不同角色节点的差异化处理. 对于核心节点 (具有较高的 $CS_i$ 值), 其对应的对比损失项被赋予了更大的权重. 这强制模型在优化过程中: 1) 优先确保核心节点的表示质量. 因为核心节点是网络的支柱部分, 其表示与全局表示 $h_g$ 的高度一致性对于捕捉网络关键结构和功能至关重要. 高权重放大了核心节点对损失的贡献, 促使模型学习到能准确反映其中心地位和信息的表示. 2) 强化结构信息的利用. 高 $CS_i$ 值本身就蕴含了节点在拓扑结构中的重要性. 将其作为权重, 相当于引入了结构先验, 引导模型不仅关注特征相似性, 更要尊重节点的结构角色. 3) 聚焦高信息量区域. 从信息论角度看, 核心节点通常连接更广, 携带更多关于网络整体结构的信息. 增大其权重符合优先处理信息密集区域的原则, 类似于人类认知中对关键要素的重点关注, 提高了学习效率. 同时, 对于负样本 (即节点 $i$ 与来自特征混洗图的节点 $k$ 的表示 $\tilde{z}'_k$ 或 $\tilde{z}''_k$), 较大的 $CS_i$ 权重也意味着模型需要更强力地将核心节点的表示与这些“伪”表示区分开. 因为核心节点一般体现了图的本质特征, 确保模型能清晰分辨真实核心节点与混淆样本之间的差异, 是学习鲁棒不变性表示的关键. 这种对核心节点正负样本对比都施加强约束的做法, 强化了模型对网络关键信号的辨识能力.

相应地, 对于边缘节点 (具有较低的 $CS_i$ 值), 其损失项的权重较小. 这意味着模型对边缘节点表示与全局表示的一致性要求相对放宽, 并且推开其与负样本表示的力度也相应减弱. 这种设计的优势在于: 1) 降低噪声干扰与增强鲁棒性. 边缘节点连接稀疏, 其特征和局部结构更容易受到噪声、异常值或图增强操作的影响. 低权重避免了模型过度拟合这些可能不稳定的边缘信息, 防止了噪声的放大和传播, 提高了模型整体的鲁棒性. 2) 防止过拟合与有效资源分配. 网络中边缘节点的数量可能远超核心节点. 如果对所有节点同等对待, 模型可能会在拟合大量边缘节点的细节上耗费过多容量, 导致过拟合. 降低边缘节点的权重, 使模型能够将“注意力”和计算资源集中于对全局结构更重要的核心节点上, 这是一种有效的正则化和资源优化策略. 3) 合理的负样本处理. 对于边缘节点, 由于其本身与全局表示的关联性就较弱 (即正样本对的相似度可能不高), 因此在负样本对比中, 也不需要施加过强的排斥力. 例如, 一个几乎孤立的边缘节点, 即使其特征被混洗, 其表示与全局表示的差异可能原本就很大, 无需通过巨大的损失权重来进一步拉远. 低权重使得模型对边缘节点的负样本处理更加“宽容”, 避免了不必要的优化负担.

综上所述, CPCL 损失函数通过核心度 $CS_i$ 的智能加权, 模拟了在理解复杂网络时关注核心、忽略边缘的认知过程, 引导模型在对比学习中侧重于把握网络的核心结构和关键信息, 同时对边缘的噪声和动态变化保持鲁棒. 这使得学习到的全局图表示 $h_g$ 更能抓住网络主旨, 节点表示 $z_i$ 更能反映其真实的结构重要性.

### 2.5 理论分析

本文的核心主张是“保护核心结构能更好地满足标签不变性假设”. 为从理论上论证为何对核心的扰动更易导致“标签偏移 (label shift)”, 我们引入随机块模型 (stochastic block model, SBM) 进行分析. SBM 是一种模拟网络社群结构的生成模型, 它允许我们从统计推断的视角, 分析移除不同角色的节点 (核心 vs. 边缘) 对网络中其他节点语义标签 (即社群归属) 稳定性的影响.

考虑一个 $N$ 节点的图 $\mathbf{A}$ (邻接矩阵) 具有以下参数: 一个块 (社群) 分配向量 $b = (b_1, \ldots, b_N)$, 其中 $b_i$ 是节点 $i$ 所属的社群 (即其语义标签); 以及一个 $C \times C$ 的亲和力矩阵 $\omega$, 其中 $\omega_{rs}$ 是社群 $r$ 和社群 $s$ 中节点间存在边的概率. 在一个具有清晰社群结构的网络中, 社群内连接概率 $p = \omega_{rr}$ 远大于社群间连接概率 $q = \omega_{rs}$ $(r \neq s)$.

在随机块模型 SBM 中, 目标是进行标签推断: 给定上述观测到的图 $\mathbf{A}$, 找到那个最有可能生成图 $\mathbf{A}$ 的“真实”社群分配 $b^*$. 这等价于最大化对数似然函数 $\mathcal{L}(b) = \log P(\mathbf{A}|b, \omega)$:

$$L(b) = \sum_{i<j} \left[ A_{ij} \log \omega_{b_i b_j} + (1 - A_{ij}) \log \left(1 - \omega_{b_i b_j}\right) \right] \tag{6}$$

$$b^* = \operatorname{argmax}_b \mathcal{L}(b)$$

这个对数似然函数 $\mathcal{L}(b)$ 可以被理解为一个“评分系统”, 用于判断一个猜测的语义分配 $b$ 有多合理. 如果节点

$i,j$ 属于同块 ($b_i = b_j$), 它们之间有边 ($A_{ij} = 1$) 会得到高分 (贡献 $\log p$), 没有边则得到“惩罚” (贡献 $\log(1-p)$); 如果节点 $i,j$ 属于异块 ($b_i \neq b_j$), 它们之间没有边 ($A_{ij} = 0$) 会得到高分 (贡献 $\log(1-q)$), 有边则得到“惩罚” (贡献 $\log q$). 因此, 最大化 $\mathcal{L}(b)$ 的过程, 就是在寻找一种节点分组 (标签分配) $b$, 使得“块内”连接尽可能多, “块间”连接尽可能少. 在可以计算节点的标签分配之后 $b^*$, “标签不变性”指的是当我们对图 $\mathbf{A}$ 进行轻微扰动 (如移除节点 $v$) 得到新图 $\mathbf{A}'$ 时, 在新图上推断出的最优标签分配 $b^*_{\mathbf{A}'}$, 对于所有剩余节点 $i \neq v$, 应保持 $b^*_{\mathbf{A}'}(i) = b^*(i)$.

下面本文分析删除核心/边缘节点对标签不变性的影响. 核心节点通常是高度连接的节点 (高 $d_i$). 在 SBM 中, 这意味着它在 $L_i(b)$ 的求和中包含大量非零项 $A_{ij}$. 因此, 核心节点 $i$ 对总似然 $\mathcal{L}(b)$ 的贡献 $L_i(b)$ 非常显著. 它为推断其自身及邻居的标签 $b_j$ 提供了大量的统计信息. 当这样一个核心节点被移除 (如图增强操作) 时, 相当于从 $\mathcal{L}(b)$ 中移除了一个高权重项 $L_i(b)$. 这会剧烈改变似然函数的形态, 导致优化后的新标签分配 $b^*_{\mathbf{A}'}$ 与原始 $b^*$ 相比发生显著变化, 即标签偏移. 相反, 边缘节点 $k$ 是稀疏连接的 (低 $d_k$), 其贡献 $L_k(b)$ 很小. 移除边缘节点 $k$ 仅轻微改变了 $\mathcal{L}(b)$, 剩余节点所依赖的统计信息基本保持不变. 因此, 新的最优分配 $b^*_{\mathbf{A}'}$ 将与 $b^*$ 几乎一致, 从而保持了标签不变性.

为了直观展示这一点, 考虑图 3 所示的具体网络实例. 该网络 (图 3(a)) 由 12 个节点组成, 明显分为两个集群 (社群 0 和社群 1). 节点 1 和 7 是各自社群的中心, 同时通过边 (1–7) 的连接成为关键的“桥接”核心节点. 其他节点 (如 2、3、4、5、6) 是边缘节点.

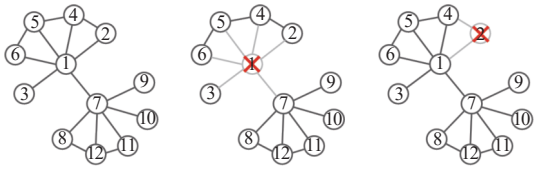



图 3 基于 SBM 对图进行不同扰动策略下的分析

我们通过最大化 SBM 对数似然函数, 得到 3 个网络图的最优语义标签向量.

1) 原始网络的最大似然分配: $b = [0,0,0,0,0,0,1,1,1,1,1,1]$.

2) 移除核心节点 1 后的分配: $b^{-1} = [x,2,0,2,2,2,1,1,1,1,1,1]$, 其中 $x$ 表示被移除的节点, 标签 2 表示新出现的语义类别.

3) 移除边缘节点 2 后的分配: $b^{-2} = [0,x,0,0,0,0,1,1,1,1,1,1]$, 其中 $x$ 代表被移除的节点.

可见, 当移除核心节点 1 后, 由于其所属集群 (社群 0) 失去了关键的结构信息, 导致原社群 0 中的 4 个节点 (2、4、5、6) 的标签发生了偏移, 被错误地归入了一个新的社群 2. 然而, 当移除边缘节点 2 时, 剩余所有节点的社群归属均未发生任何变化, 严格保持了标签不变. 因此上述 SBM 分析为我们的 CP-GCL 框架提供了理论支持. 核心节点对似然的影响大于边缘节点, 因此其移除导致的标签后验变化更显著.

### 2.6 计算时间复杂度

本文提出的 CP-GCL 框架的计算开销主要包含两部分: 核心度识别的预处理开销和图对比学习的训练开销. 核心度识别步骤的计算复杂度由模拟退火 (SA) 优化算法主导. 对于给定的参数组合 $\gamma$, SA 算法需要 $T$ 次迭代来寻找最优排列. 在每次迭代中, 更新排列 (例如交换两个节点) 后, 需要重新计算核心质量函数 $R_\gamma$ (公式 (1)) 的变化量. $R_\gamma$ 是对所有边的加权贡献求和, 图的边数为 $M$, 因此一次迭代的复杂度为 $\mathrm{O}(M)$ (或在稠密图下为 $\mathrm{O}(N^2)$). 若总共探索 $K$ 组 $\gamma$ 参数, 则核心度识别的总时间复杂度约为 $\mathrm{O}(K \cdot T \cdot M)$. 这部分计算作为预处理步骤, 在训练前一次性完成. 图对比学习的训练阶段与主流 GCL 方法类似, 每轮 (epoch) 的复杂度主要由对比损失计算 ($\mathrm{O}(N^2)$) 决定. 因此, 核心度识别是一个计算成本可控的预处理步骤, 不会增加模型训练阶段的迭代复杂度.

### 2.7 收敛性分析

本文的核心度识别算法涉及一个优化步骤, 即寻找一个最优的节点排列, 以最大化核心质量函数. 这是一个组

合优化问题, 在本文中, 我们采用模拟退火 (simulated annealing, SA) 等启发式优化算法来求解. 模拟退火是一种全局概率性优化算法. 从理论上讲, 只有当采用极慢的 (例如对数) 降温策略时, SA 才能保证渐近收敛到全局最优解. 然而在实际应用中, 为了计算效率, 通常采用更快的 (如几何) 降温策略. 这种策略虽然不保证找到全局最优解, 但它通过概率性的“跳出”机制有效避免了算法陷入局部最优解, 确保在有限的计算时间内收敛到一个高质量的稳定解. 对于核心度识别任务而言, 这种高质量的局部最优解已经足以反映网络中稳定且有意义的核心-边缘结构, 从而为后续的对比学习提供了鲁棒的结构先验.

## 3 实验分析

本节旨在通过一系列实验评估本文提出的基于核心-边缘结构感知的图对比学习方法 CP-GCL 的有效性. 本文将围绕以下几个问题展开.

● RQ1: 与当前最先进的图对比学习方法以及其他基准模型相比, 本文提出的 CP-GCL 在节点级/图级分类任务上的性能表现如何?

● RQ2: 本文方法的核心组成部分, 即核心-边缘感知的图增强策略和边缘感知对比损失函数, 各自对模型性能有何贡献? 与全局增强和标准损失相比, 它们的优势体现在哪里?

● RQ3: 本文提出的核心-边缘增强策略对关键超参数 (图增强的破坏率) 的敏感性如何? 模型性能是否在一定参数范围内保持鲁棒?

● RQ4: 通过可视化分析, 本文方法学习到的节点表示是否比基线方法更具判别性? 同时, 核心度识别算法能否在真实数据上有效地区分核心与边缘?

● RQ5: 本文引入的核心度识别预处理步骤带来了多少计算开销? 其运行时间是否在可接受范围内?

● RQ6: 本文的核心度识别方法是否优于其他替代方法 (如 $k$-core、PageRank)? CP-GCL 框架对核心识别的潜在“误判”是否具有鲁棒性?

● RQ7: CP-GCL 相较于最优基线方法的性能提升是否具有统计显著性?

● RQ8: 与近期“结构感知” GCL 方法 (如 LS-GCL、S3GCL) 相比, 本文方法的性能优势体现在哪里?

● RQ9: 用于划分核心-边缘节点的阈值参数 $\rho$ 的选择如何影响模型性能?

为了回答这些问题, 本文进行如下实验设置、性能比较、消融研究、参数敏感性分析、可视化、运行时间分析、鲁棒性分析、统计检验及对比分析.

### 3.1 实验数据

为了全面评估模型性能, 本文在图级分类和节点分类任务中广泛使用的基准数据集上进行了实验. 节点分类数据集具体包括: 1) 引文网络数据集 Cora、CiteSeer、PubMed[56]. 这些是 GNN 领域最常用的基准数据集, 节点代表论文, 边代表引用关系, 节点特征是词袋表示, 任务是预测论文类别. 2) 网页与知识图谱数据集 WikiCS[57]. 节点代表维基百科文章, 边代表超链接, 节点特征基于文本内容, 任务是预测文章类别. 3) 共现与共同购买网络数据集 Amazon-Computers (Amz-Comp.)、Amazon-Photo (Amz-Photo)[58]. 节点代表商品, 边代表共同购买关系, 节点特征是商品评论的词袋表示, 任务是预测商品类别. 4) 合著网络数据集 Coauthor-CS、Coauthor-Physics (Coauthor-Phy.)[58]. 节点代表作者, 边代表合著关系, 节点特征是作者论文关键词的词袋表示, 任务是预测作者的研究领域.

对于图分类任务, 本文选用了 TUDataset 基准[59]中被广泛使用的 8 个数据集, 涵盖了从生物信息学到社交网络的多个领域. 具体包括: 1) 分子化合物数据集 NCI1、MUTAG、DD、PROTEINS. NCI1 包含化合物分子图, 任务是预测化合物是否具有抗癌活性; MUTAG 包含芳香族和杂芳香族硝基化合物, 任务是预测化合物的致突变性; DD 包含蛋白质结构图, 节点代表氨基酸, 边代表氨基酸之间的空间距离, 任务是区分酶和非酶蛋白; PROTEINS 包含蛋白质图, 节点代表二级结构元素, 边代表邻近关系, 同样用于酶/非酶分类. 2) 社交网络数据集 COLLAB、IMDB-B. COLLAB 包含科学合作网络, 每个图代表一个研究者的自我中心网络, 任务是预测研究者所属的研究领域; IMDB-B 包含电影合作网络, 节点代表演员/演员, 边代表共同出演关系, 任务是预测电影类型. 3) Reddit 讨论主题数据集

RDT-B、RDT-M5K. 这些数据集中每个图代表一个在线讨论主题, 节点代表用户, 边代表回复关系, RDT-B 的图用于二分类任务, RDT-M5K 的图用于五分类任务, 目标是预测讨论所属的子版块. 这些数据集覆盖了不同规模、密度和领域, 能够有效检验模型的泛化能力.

### 3.2 评价指标及基准模型

本文遵循先前图对比学习方法在图级和节点级任务上的标准评估协议. 对于图分类任务: 本文报告 5 次运行后 10 折交叉验证的平均准确率. 使用线性支持向量机 (SVM) 作为下游分类器, SVM 在训练集上通过交叉验证进行训练, 并报告最佳平均准确率. 图分类任务数据集划分为 80% 训练集, 10% 验证集, 10% 测试集. 对于节点分类任务: 本文报告 50 次独立运行后在测试集上的平均准确率. 使用一个简单的线性神经网络模型作为下游分类器. 节点分类任务数据集划分为 10% 训练集, 10% 验证集, 80% 测试集.

对于节点级任务, 将提出的 CP-GCL 方法与以下几类基准模型进行比较: 1) 监督学习方法: 包括简单的多层感知机 MLP (仅使用节点特征) 和经典的图神经网络 GCN[56] (使用特征和结构). 它们代表了利用标签信息进行学习的上限参考. 2) 图嵌入方法: 包括基于随机游走的 DeepWalk[60]和 node2vec[61]. 这些是早期的无监督图表示学习方法. 3) 图对比学习方法: 这是本文主要比较的对象, 包括基于重构的方法 GAE、VGAE[62]; 基于互信息最大化的方法 DGI[63]、MVGRL[27]; 基于视图对比的方法 GRACE[21] (基础对比框架)、GCA[22] (引入自适应增强)、 BGRL[64] (非对称编码器)、CCA-SSG[65] (谱增强与正则化)、SUGRL[66] (自监督去噪).

图级任务的比较方法为: 1) 监督学习方法: 包括 GCN[56]和 GIN[67]. 它们是经典的图分类模型, 作为利用标签信息的性能上限. 2) 基于核的方法: 包括最短路径核 SP[68]、图谱核 GK[69]、Weisfeiler-Lehman 子树核 WL[70]以及深度图核 DGK[71]. 这类传统方法通过图核来衡量图之间的相似性, 在一些任务上仍有很强的竞争力. 3) 无监督方法: 包括 graph2vec[72]和 Sub2Vec[73]. 它们通过学习整个图或其子结构的嵌入来进行无监督表示. 4) 图对比学习方法: 这是本文最关注的比较对象, 包括基于互信息判别的 InfoGraph[74], 采用双视图对比的 GraphCL[12], 自动化增强策略的 AD-GCL[75]与 JOAO-v2[6], 利用关系感知增强的 RGCL[76], 使用简化正则的 SimGRACE[77], 构造语义分层视图的 SEGA[78], 以及结合神经架构搜索的 AutoGCL[29].

### 3.3 实验细节

本文使用 PyTorch Geometric 实现了本文方法和所有基线模型. 组合使用第 4.3 节中描述的针对边缘节点的增强策略, 破坏率从 $\{0.1, 0.2, \ldots, 0.9\}$ 中搜索最优值. 本文统一使用两层的 GNN 编码器作为所有对比学习方法 (包括提出的 CP-GCL) 的基础图编码器. 使用 Adam 优化器学习率从 $\{0.000\,1, 0.001, 0.01\}$ 中选择, 批量大小从 $\{16, 64, 128, 256, 512\}$ 中选择. 温度参数 $\tau$ 从 $\{0.1, 0.3, 0.5\}$ 中选择. 所有实验均在配备 NVIDIA GeForce RTX A6000 的服务器上使用 PyTorch 和 PyG 框架实现.

### 3.4 性能比较 (RQ1)

节点分类性能: 表 1 展示了 CP-GCL 与各基线方法在 8 个节点分类基准数据集上的准确率. 从结果可以看出, CP-GCL 在所有数据集上均显著优于现有的无监督和自监督对比学习方法. 例如, 在 Cora、CiteSeer 和 PubMed 这 3 个经典的引文网络上, CP-GCL 相较于次优的对比学习方法 (如 CCA-SSG 或 SUGRL) 分别取得了 1.82%、0.92% 和 1.62% 的性能提升. 在规模更大、结构更多样的数据集如 WikiCS、Amz-Comp.、Amz-Photo、 Coauthor-CS 和 Coauthor-Phy. 上, CP-GCL 的优势同样明显, 平均性能提升达到 1.03%. 值得注意的是, 在大部分数据集上 (例如 PubMed 和 Coauthor-CS 等), CP-GCL 的性能甚至超越了使用标签信息进行训练的监督方法 GCN, 这充分证明了本文提出的核心-边缘结构感知策略在挖掘图内在信息、学习高质量节点表示方面的有效性.

图分类性能: 表 2 报告了 CP-GCL 与相关基线在 8 个图分类基准数据集上的性能. 结果显示, CP-GCL 在大多数图分类数据集上也展现出了强大的竞争力. 相较于其他先进的自监督对比学习方法 (如 GraphCL、JOAOv2、RGCL、SimGRACE 等), CP-GCL 在 PROTEINS、DD、RDT-B、IMDB-B 等绝大多数数据集上均取得了最优或次优的性能, 平均准确率提升约 1.89%. 虽然在数据集 NCI1 上略低于特定的基线 (例如基于图核的 WL 方法), 但

总体而言, CP-GCL 展现了良好的泛化能力, 证明了核心-边缘结构感知对于学习图级别表示同样具有价值. 这些结果表明通过保护核心结构并模拟边缘动态, CP-GCL 能够学习到更具判别力的图级表示.

表 1 节点分类任务上的平均准确率 (%)

| Model | Cora | CiteSeer | PubMed | WikiCS | Amz-Comp. | Amz-Photo | Coauthor-CS | Coauthor-Phy. |
|---|---|---|---|---|---|---|---|---|
| MLP | 47.92±0.41 | 49.31±0.26 | 69.14±0.34 | 71.98±0.42 | 73.81±0.21 | 78.53±0.32 | 90.37±0.19 | 93.58±0.41 |
| GCN | 81.54±0.68 | 70.73±0.65 | 79.16±0.25 | 93.02±0.11 | 86.51±0.54 | 92.42±0.22 | 93.03±0.31 | 95.65±0.16 |
| DeepWalk | 70.72±0.63 | 51.39±0.41 | 73.27±0.86 | 74.42±0.13 | 85.68±0.07 | 89.40±0.11 | 84.61±0.22 | 91.77±0.15 |
| node2vec | 71.08±0.91 | 47.34±0.84 | 66.23±0.95 | 71.76±0.14 | 84.41±0.14 | 89.68±0.19 | 85.16±0.04 | 91.23±0.07 |
| VGAE | 77.27±0.86 | 67.46±0.20 | 76.02±0.52 | 75.55±0.22 | 86.40±0.30 | 92.13±0.12 | 92.10±0.31 | 94.43±0.20 |
| DGI | 82.24±0.63 | 71.82±0.61 | 76.80±0.30 | 75.42±0.17 | 84.05±0.42 | 91.62±0.37 | 92.14±0.55 | 94.54±0.52 |
| GMI | 82.40±0.57 | 71.74±0.12 | 79.28±0.94 | 74.79±0.16 | 82.24±0.39 | 90.81±0.15 | — | — |
| MVGRL | 83.37±0.65 | 73.29±0.36 | 80.33±0.61 | 77.55±0.06 | 87.45±0.17 | 91.77±0.21 | 92.24±0.31 | 95.30±0.13 |
| GRACE | 81.88±0.84 | 71.13±0.42 | 80.88±0.13 | 79.37±0.24 | 86.48±0.24 | 92.20±0.16 | 92.90±0.27 | 95.25±0.26 |
| GCA | 82.41±0.55 | 71.56±0.19 | 80.73±0.23 | 78.26±0.39 | 87.92±0.33 | 92.35±0.53 | 92.65±0.32 | 95.52±0.21 |
| BGRL | 81.86±0.32 | 72.10±0.31 | 80.65±0.42 | 79.28±0.45 | 89.21±0.47 | 92.28±0.44 | 92.73±0.41 | 95.31±0.26 |
| SUGRL | 83.29±0.38 | 73.11±0.22 | 81.96±0.49 | 78.88±0.35 | 88.98±0.20 | 92.87±0.19 | 92.84±0.24 | 94.80±0.24 |
| CCA-SSG | 84.17±0.44 | 73.27±0.30 | 81.91±0.41 | 77.67±0.29 | 88.88±0.22 | 93.14±0.43 | 93.23±0.16 | 95.29±0.11 |
| CP-GCL | **85.98±0.54** | **74.21±0.17** | **83.58±0.35** | **82.18±0.26** | **90.83±0.15** | **93.29±0.39** | **93.41±0.27** | **95.93±0.18** |

注: 最优无监督/自监督结果以粗体标出, 次优结果用下划线标出

表 2 TUDataset 上的监督和无监督分类准确率 (%)

| Model | NCI1 | PROTEINS | DD | MUTAG | COLLAB | RDT-B | RDT-M5K | IMDB-B |
|---|---|---|---|---|---|---|---|---|
| GCN | 80.20±0.14 | 74.92±0.33 | 76.24±0.14 | 85.63±0.24 | 79.01±0.18 | 50.00±0.10 | 20.00±0.10 | 70.45±0.37 |
| GIN | 82.75±0.19 | 76.28±0.28 | 78.91±0.13 | 89.47±0.16 | 80.23±0.19 | 92.44±0.25 | 56.50±0.24 | 73.70±0.60 |
| SP | 73.53±0.16 | 75.07±0.54 | — | 85.25±0.24 | — | 64.13±0.04 | 39.64±0.02 | 55.62±0.02 |
| GK | 66.06±0.12 | 71.67±0.55 | 78.53±0.03 | 81.71±0.21 | 71.81±0.31 | 77.32±0.02 | 41.06±0.08 | 65.93±0.10 |
| WL | **80.01±0.50** | 72.92±0.56 | 79.78±0.36 | 80.76±0.30 | 69.30±0.42 | 68.82±0.41 | 46.06±0.21 | 72.30±0.44 |
| DGK | 79.98±0.36 | 73.21±0.61 | 74.79±0.32 | 87.51±0.65 | 64.43±0.48 | 78.35±0.43 | 41.49±0.32 | 67.09±0.37 |
| node2vec | 54.93±0.16 | 57.58±0.36 | — | 72.62±1.02 | 56.12±0.02 | — | — | 50.25±0.09 |
| Sub2Vec | 52.82±0.15 | 53.06±0.56 | 54.33±0.24 | 61.17±1.59 | 55.26±0.15 | 71.48±0.42 | 36.68±0.42 | 55.34±0.15 |
| graph2vec | 73.21±0.18 | 73.33±0.21 | 79.32±0.29 | 83.28±0.93 | 71.10±0.54 | 75.78±0.96 | 46.86±0.26 | 71.16±0.05 |
| InfoGraph | 76.33±0.79 | 74.27±0.43 | 73.02±1.31 | 88.65±1.09 | 70.41±0.34 | 83.06±0.82 | 53.45±1.10 | 72.97±0.49 |
| GraphCL | 77.54±0.34 | 74.41±0.35 | 78.49±0.36 | 86.62±1.23 | 71.28±1.16 | 89.50±0.73 | 55.85±0.33 | 71.29±0.38 |
| AD-GCL | 74.15±0.62 | 73.33±0.36 | 75.74±0.68 | 88.70±1.66 | 72.01±0.53 | 90.04±0.76 | 54.55±0.40 | 70.35±0.59 |
| JOAOv2 | 78.36±0.21 | 74.02±0.53 | 77.36±0.29 | 87.23±1.12 | 69.05±0.22 | 86.36±0.30 | 55.77±0.23 | 70.12±0.25 |
| RGCL | 78.14±0.89 | 75.11±0.58 | 78.77±0.52 | 87.62±1.16 | 70.85±0.63 | 90.27±0.54 | 56.39±0.37 | 71.67±0.76 |
| SimGRACE | 79.06±0.15 | 75.26±0.24 | 77.13±0.88 | 89.23±1.25 | 71.26±0.51 | 89.41±0.33 | 55.91±0.23 | 71.22±0.30 |
| SEGA | 79.15±0.52 | 76.00±0.68 | 78.81±0.33 | 90.23±0.89 | 74.10±0.62 | 90.12±0.46 | 56.08±0.40 | 73.60±0.33 |
| AutoGCL | 78.39±0.58 | 69.80±0.38 | 75.76±0.93 | 85.22±1.43 | 71.40±0.26 | 86.56±1.31 | 55.68±0.23 | 72.26±0.34 |
| CP-GCL | 78.42±0.18 | **79.68±0.31** | **83.08±0.22** | **91.31±1.12** | **74.60±0.54** | **90.55±0.48** | **56.43±0.31** | **77.01±0.28** |

注: 无监督方法中表现最好的结果用粗体标出, 表现第2好的用下划线标出. “—”表示无结果

### 3.5 消融实验 (RQ2)

为了验证本文方法中各个核心组件的有效性, 本文进行了一系列消融实验. 具体来说, 本节研究以下两种变体: 1) w/o CP-Aug: 移除了核心-边缘增强策略, 转而采用标准的全局随机增强方法 (例如, 随机节点删除或边扰动作用于全图). 2) w/o CP-Loss: 移除了核心-边缘对比损失函数, 替换为标准的不考虑节点核心度的对比损失 (所有节点的损失权重相同). 实验结果如表 3 所示, 其中, 加粗数值代表所有对比算法中达到的最优性能.

从表 3 的结果可以观察到, 当移除核心-边缘增强策略并采用全局增强时 (行 “w/o CP-Aug”), 所有数据集上的

性能均出现下降. 例如, 在 CiteSeer 数据集上, 移除 CP-Aug 后性能下降了 1.66%. 这表明, 通过识别核心与边缘节点并仅对边缘进行扰动, 能够更有效地生成高质量的对比视图, 从而更好地保持标签不变性, 提升模型学习效果. 全局增强由于可能破坏核心结构, 反而损害了表示的质量.

表 3　消融实验结果 (节点分类准确率) (%)

| 方法 | Cora | CiteSeer | PubMed | WikiCS | Amz-Comp. | Coauthor-CS |
|---|---|---|---|---|---|---|
| CP-GCL | **85.98±0.54** | **74.21±0.17** | **83.58±0.35** | **82.18±0.26** | **90.83±0.15** | **93.41±0.27** |
| w/o CP-Aug | 84.25±0.48 | 72.55±0.36 | 82.32±0.19 | 80.73±0.31 | 89.14±0.26 | 92.21±0.40 |
| w/o CP-Loss | 83.64±0.39 | 73.07±0.22 | 82.16±0.27 | 81.02±0.12 | 89.03±0.15 | 91.55±0.23 |

类似地, 当使用标准对比损失替换核心-边缘对比损失时 (行“w/o CP-Loss”), 性能也普遍下降. 例如, 在 Cora 数据集上, 性能下降了 2.34%. 这表明了根据节点核心度对损失进行加权的策略是有效的. 它使得模型更加关注网络中的核心结构信息, 同时抑制了边缘节点可能带来的噪声影响, 从而学习到更鲁棒和有判别力的表示. 综上所述, 消融实验的结果说明了本文提出的核心-边缘增强策略和核心-边缘对比损失函数的核心贡献.

### 3.6　参数敏感性研究 (RQ3)

本节研究本文方法中关键超参数, 即核心-边缘增强中边缘节点破坏率 ($p_d$) 对模型性能的影响. 首先探究在核心-边缘增强策略中, 针对边缘节点的破坏率 (边缘节点丢弃比例, 记为 $p_d$) 如何影响模型在节点分类任务上的性能. 将 $p_d$ 的值从 0.1 变化到 0.7, 并在多个数据集上观察其准确率变化. 结果如表 4 所示, 其中, 加粗数值代表所有对比算法中达到的最优性能.

表 4　不同边缘节点破坏率 $p_d$ 下的节点分类准确率 (%)

| 破坏率 | Cora | CiteSeer | PubMed | WikiCS | Amz-Comp. | Coauthor-CS |
|---|---|---|---|---|---|---|
| 0.1 | 85.16±0.30 | 73.84±0.24 | 83.18±0.34 | 82.11±0.10 | 90.39±0.33 | 93.19±0.19 |
| 0.2 | 85.29±0.28 | **74.21±0.17** | 83.34±0.26 | 81.71±0.18 | 89.33±0.11 | 93.32±0.21 |
| 0.3 | **85.98±0.54** | 74.16±0.32 | 83.32±0.15 | **82.18±0.26** | **90.83±0.15** | 93.24±0.35 |
| 0.4 | 84.87±0.46 | 74.06±0.11 | **83.58±0.35** | 82.10±0.22 | 90.55±0.28 | **93.41±0.27** |
| 0.5 | 84.25±0.33 | 72.55±0.36 | 83.29±0.27 | 81.07±0.38 | 89.64±0.16 | 92.90±0.38 |
| 0.6 | 84.08±0.22 | 73.02±0.07 | 82.37±0.35 | 81.06±0.21 | 89.53±0.20 | 91.82±0.24 |
| 0.7 | 83.52±0.57 | 72.67±0.38 | 82.23±0.44 | 80.35±0.22 | 88.48±0.15 | 91.01±0.21 |

从表 4 可以看出, 当边缘节点破坏率 $p_d$ 处于较低的范围时 (例如 0.1–0.4), 模型性能相对稳定且表现较好. 这表明适度的边缘扰动有助于模型学习到对局部噪声具有鲁棒性的特征. 当 $p_d$ 过高时 (例如大于 0.5), 模型性能开始出现下降. 这可能是因为过度的边缘破坏虽然没有直接损害核心结构, 但也可能导致边缘区域信息损失过多, 或者使得对比视图与原始图的语义差异过大, 从而影响了对比学习的效果. 尽管如此, 在一定的参数范围内, 模型性能的波动不大, 显示出本文方法对边缘节点破坏率具有一定的鲁棒性.

### 3.7　可视化实验 (RQ4)

为了更直观地理解本文方法学习到的节点表示的质量以及核心度识别算法的有效性, 本节采用 t-SNE[79]算法将学习到的高维节点嵌入投影到二维空间进行可视化. 我们首先在 Cora 和 CiteSeer 数据集上进行节点嵌入可视化实验, 并与两种基线方法进行比较: 一个标准的监督学习方法 GCN[56]和一个先进的图对比学习方法 CCA-SSG[65].

节点嵌入可视化: 如图 4 第 1 行 (Cora 数据集) 和第 2 行 (CiteSeer 数据集) 所示, 图中不同的颜色代表数据集中不同的节点类别. 在这两个数据集上我们观察到了高度一致的现象: 监督方法 GCN (图 4(a)) 虽然能一定程度区分节点, 但各个簇之间距离较近, 交界处较为模糊. 基线对比学习方法 CCA-SSG (图 4(b)) 虽然簇间距离稍明显, 但簇内节点的分布相对松散. 相比之下, 本文方法 CP-GCL (图 4(c)) 的可视化结果在两个数据集上均表现优异, 产生的节点嵌入在不同类别之间形成了清晰和明确的边界, 各个类别的簇内聚合性也紧凑. 这验证了本文方法在不同数据集上均能学习到更具判别性的节点表示.

核心-边缘节点区分效果可视化: 为了直观展示本文核心度识别算法的有效性, 我们在 Cora 和 CiteSeer 数据集上绘制了节点核心度分数 $CS(i)$ 的分布效果图. 如图 5 所示, 可视化网络拓扑结构中, 节点的颜色不透明度与其核心度分数 $CS(i)$ 成正比: 颜色越深 (不透明度越高) 的节点, 其 $CS$ 分数越高, 代表其越接近“核心”; 颜色越浅 (越透明) 的节点, 其 $CS$ 分数越低, 代表其越接近“边缘”.

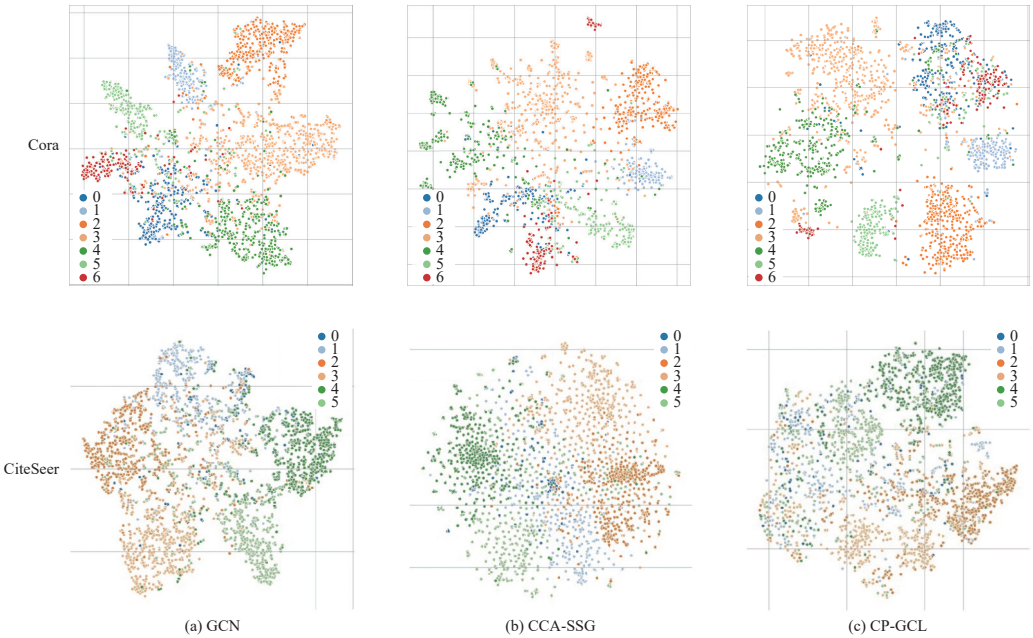



图 4 网络数据 Cora 和 CiteSeer 中节点的 t-SNE 嵌入可视化


图 5 Cora 和 CiteSeer 数据集的核心-边缘节点区分效果可视化

从图 4 和图 5 可以观察到, 本文提出的方法计算出的核心度分数与网络的拓扑密度高度一致. 在 Cora 数据集 (图 5(a)) 中, 得分最高 (颜色最深) 的节点高度集中在网络的中心区域, 形成了一个非常密集的核心簇. 在 CiteSeer 数据集 (图 5(b)) 中, 得分高的节点同样聚集在网络中连接最紧密的区域, 构成了网络的骨架. 在图 5(a) 和 (b) 中, 节点都呈现出从中心 (深红色) 向外围 (浅红色) 平滑过渡的趋势, 外围节点的连接相对稀疏. 这种可视化直观地说明了我们的核心度识别方法能够成功捕捉图的核心-边缘拓扑架构.

### 3.8 运行时间分析 (RQ5)

本文提出的 CP-GCL 框架在标准图对比学习流程之外, 引入了核心-边缘结构识别作为预处理步骤. 这一步是本方法的主要额外计算开销, 因此, 评估该步骤的计算效率对于衡量方法的实用性至关重要. 我们在配备 Intel(R) Core(TM) i7-12700 CPU 的服务器上, 统计了该预处理步骤在不同数据集上的运行时间, 如表 5 所示.

从实验结果中可以看出, 核心度识别的耗时主要与图的规模 (特别是边数 $M$) 紧密相关, 这与第 2.6 节中的时

间复杂度分析相一致. 例如, 在边数较多的 Amz-Comp. 和 Coauthor-Phy. 数据集上, 预处理时间相对较长. 然而, 由于核心-边缘结构识别是一个一次性离线计算 (one-time offline) 过程, 其在训练开始前一次性完成即可. 综合来看, 该步骤的耗时均在实际应用的可接受范围内, 证明了 CP-GCL 框架在引入结构先验知识的同时, 仍保持了良好的计算可行性.

表 5　CP-GCL 在不同数据集上的运行时间

| 数据集 | 节点数$N$ | 边数$M$ | 核心识别时间 (s) |
|---|---|---|---|
| Cora | 2 708 | 10 556 | 128.8 |
| CiteSeer | 3 327 | 9 104 | 109.1 |
| PubMed | 19 717 | 88 648 | 612.3 |
| WikiCS | 11 701 | 216 123 | 1 060.2 |
| Amz-Comp. | 13 752 | 491 722 | 1 211.6 |
| Amz-Photo | 7 650 | 238 162 | 410.7 |
| Coauthor-CS | 18 333 | 163 788 | 576.7 |
| Coauthor-Phy. | 34 493 | 495 924 | 1 509.0 |

### 3.9　核心度识别方法的可替代性与鲁棒性分析 (RQ6)

本节评估当核心-边缘方法采用其他常见核心识别方法时其性能表现如何. 我们选用了 3 种代表性方法: CP-GCL ($k$-core): 使用 $k$-core 分解. 将处于最内层 $k_{\max}$-core 的节点视为核心, 其余为边缘. CP-GCL (PageRank): 使用 PageRank 中心性. 将 PageRank 值超过阈值的节点视为核心, 其中阈值作为超参数. CP-GCL (SBM)[80]: 采用基于随机块模型的方法, 通过拟合 SBM 来推断每个节点的核心/边缘归属分数, 与 PageRank 类似使用阈值区分核心与边缘节点. 我们在节点分类数据集上进行了对比, 结果如表 6 所示, 其中, 加粗数值代表所有对比算法中达到的最优性能.

表 6　采用不同核心度识别方法时 CP-GCL 框架的节点分类准确率 (%)

| 方法 | Cora | CiteSeer | PubMed | Amz-Comp. | Coauthor-CS |
|---|---|---|---|---|---|
| CP-GCL (Ours) | **85.98±0.54** | **74.21±0.17** | **83.58±0.35** | **90.83±0.15** | **93.41±0.27** |
| CP-GCL ($k$-core) | 84.12±0.43 | 73.64±0.37 | 82.34±0.43 | 89.49±0.22 | 92.94±0.39 |
| CP-GCL (PageRank) | 84.35±0.57 | 74.06±0.35 | 82.16±0.38 | 90.02±0.13 | 93.27±0.25 |
| CP-GCL (SBM) | 85.10±0.44 | 73.78±0.23 | 81.81±0.35 | 89.35±0.28 | 92.82±0.43 |

从表 6 的结果中, 我们可以发现, 采用本文提出的连续核心度识别方法的 CP-GCL (Ours) 在所有数据集上均取得了最佳性能. 这证明了我们方法所捕捉到的连续、多尺度的核心-边缘结构 (通过优化 $R_\gamma$ 得到) 比传统的离散划分 ($k$-core) 或单一中心性 (PageRank、SBM) 能更精确地描述网络结构, 从而为对比学习提供了更优质的结构先验.

其次, 从结果可以发现核心检测的准确性确实会影响性能, 但识别核心与边缘这一模式本身具有良好的鲁棒性. $k$-core、PageRank 和 SBM 等方法识别的核心节点集合并不完全重叠, 每种方法相对于“真实”核心都可能存在一定程度的“误判”. 实验结果显示, 采用这些方法 (如 CP-GCL ($k$-core), CP-GCL (PageRank) 和 CP-GCL (SBM)) 时, 模型性能 (例如在 Cora 上的 84.12%, 84.35%, 85.10%) 虽然都低于我们的方法 (85.98%), 但仍然优于表 1 中的大多数基线 GCL 方法 (如 GRACE 的 81.88%, SUGRL 的 83.29%). 这说明了这一范式是鲁棒的. 只要识别出的核心具有一定的合理性 (即便是次优的), “保护核心、扰动边缘”的策略就是有效的. 我们的方法通过更精细的多尺度优化很大程度地减少了这种“误判”, 从而获得了最佳性能.

### 3.10　统计显著性检验 (RQ7)

为了进一步验证本文 CP-GCL 方法相较于最优基线模型的性能提升是否稳定且具有统计学意义, 而不仅是随机波动的结果, 本节进行统计显著性检验. 具体而言, 本文采用配对 t 检验 (paired t-test), 在 95% 的置信水平上 (即显著性水平 $\alpha = 0.05$), 比较了 CP-GCL 在多次独立运行的准确率上与表 1 (节点分类) 和表 2 (图分类) 中次优方法 (CCA-SSG 和 SEGA) 的差异.

检验结果的 p 值 (p-value) 汇总于表 7 和表 8. 从表中可以看出, 在节点分类任务中, CP-GCL 在全部 8 个数据集上的 p 值均小于 0.05, 其中 7 个数据集的 p 值甚至小于 0.01, 显示出其性能提升具有 (高度) 统计显著性. 在图分类任务中, CP-GCL 在 8 个数据集中有 7 个的 p 值小于 0.01, 同样达到了高度统计显著性. 唯一的例外是 NCI1 数据集, 其差异未通过显著性检验. 总体而言, 这一结果强有力地证明了本文提出的 CP-GCL 框架所带来的性能增益在绝大多数情况下是稳定且统计显著的.

表 7　节点分类任务上的配对 t 检验 p 值 (CP-GCL vs. CCA-SSG)

| 数据集 | p 值 |
|---|---|
| Cora | < 0.01 |
| CiteSeer | < 0.01 |
| PubMed | < 0.01 |
| WikiCS | < 0.01 |
| Amz-Comp. | < 0.01 |
| Amz-Photo | < 0.05 |
| Coauthor-CS | < 0.01 |
| Coauthor-Phy. | < 0.01 |

表 8　图分类任务上的配对 t 检验 p 值 (CP-GCL vs. SEGA)

| 数据集 | p 值 |
|---|---|
| NCI1 | 1.0 |
| PROTEINS | < 0.01 |
| DD | < 0.01 |
| MUTAG | < 0.01 |
| COLLAB | < 0.01 |
| RDT-B | < 0.01 |
| RDT-M5K | < 0.01 |
| IMDB-B | < 0.01 |

### 3.11　与结构感知 GCL 方法的对比分析 (RQ8)

为进一步考察本文方法在结构感知图对比学习中的有效性, 我们选取了两种代表性的结构感知 GCL 方法, LS-GCL[81]和 S3GCL[2], 进行性能对比. 这两种方法分别从不同角度引入结构信息: LS-GCL 聚焦于局部结构的尺度特征, 通过构建“语义子图”而非仅依赖一阶邻域来捕捉更丰富的局部上下文; S3GCL 则从图谱特性出发, 利用谱图滤波器生成高频与低频视图, 以应对图中不同程度的同配性与异配性. 尽管两者均具备结构感知能力, 但其视角与本文所提出的核心-边缘拓扑角色感知机制存在本质差异.

如表 9 所示 (加粗数值代表所有对比算法中达到的最优性能), 我们在 5 个基准数据集上对所提方法 CP-GCL 与上述两种方法进行了系统比较. 实验结果显示, CP-GCL 在绝大多数数据集上均取得了最优性能. 具体而言, 在 Cora、CiteSeer、PubMed 和 Amz-Comp. 这 4 个数据集上, CP-GCL 的分类准确率均显著优于 LS-GCL 和 S3GCL, 尤其在 Amz-Comp. 上提升最为显著. 综合以上结果, CP-GCL 相较于仅依赖局部子图构造或图谱滤波的方法, 展现出更全面的结构感知能力, 其基于核心-边缘拓扑角色的增强范式, 能够更灵活地捕捉图中不同节点所承担的结构功能, 从而在不同类型的图数据上实现稳定且优异的表示学习效果.

表 9　与结构感知 GCL 方法的分类准确率对比 (%)

| 方法 | Cora | CiteSeer | PubMed | Amz-Comp. | Coauthor-CS |
|---|---|---|---|---|---|
| LS-GCL | 83.71±0.71 | 72.16±0.19 | 80.18±0.41 | 82.43±0.29 | 90.38±0.37 |
| S3GCL | 84.68±0.65 | 72.27±0.12 | 83.43±0.34 | 82.55±0.10 | **94.56±0.15** |
| CP-GCL (Ours) | **85.98±0.54** | **74.21±0.17** | **83.58±0.35** | **90.83±0.15** | 93.41±0.27 |

### 3.12　阈值参数影响 (RQ9)

在 第 2.2 节中, 本文提出的算法计算出了每个节点的连续核心度分数 $CS(i)$. 为了将节点划分为核心集 $\mathcal{V}_C$ 与边缘集 $\mathcal{V}_P$, 我们引入了一个阈值超参数 $\rho$. 在实现中, 所有节点的 $CS(i)$ 分数被归一化到 $[0,1]$ 区间内, 得分 $CS(i)>\rho$ 的节点被视为核心节点, 反之则为边缘节点. $\rho$ 的最优值通过在 $[0,1]$ 范围内搜索确定.

为了研究模型性能对 $\rho$ 选择的敏感性, 我们在 Cora 和 CiteSeer 数据集上对 $\rho\in\{0.1,0.3,0.4,0.5,0.6,0.7,0.9\}$ 的取值进行了影响性分析. 实验结果如表 10 所示.

从表 10 中可以看到, 模型性能在 $\rho$ 处于 $[0.4,0.6]$ 这一中间区间内达到峰值, 并保持相对稳定. 这是因为当 $\rho$ 取值过高 (如 0.9) 时, 核心集 $\mathcal{V}_C$ 过小, 部分本应受保护的关键节点被错误划分为边缘节点, 导致其在增强过程中受到扰动, 违背了核心节点标签一致性的基本假设; 相反, 当 $\rho$ 取值过低 (如 0.1) 时, 核心集 $\mathcal{V}_C$ 过大, 导致可供扰

动的边缘节点太少, 生成的对比视图过于相似, 对比学习难以学到鲁棒的特征. 因此, [0.4,0.6] 的中间范围为划分核心与边缘提供了一个鲁棒且有效的选择.

表 10　不同核心度划分阈值 $\rho$ 下的节点分类准确率 (%)

| 数据集 | 0.1 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.9 |
|---|---|---|---|---|---|---|---|
| Cora | 83.55±0.51 | 85.02±0.39 | 85.97±0.48 | 85.29±0.35 | 85.75±0.62 | 84.90±0.44 | 83.41±0.54 |
| CiteSeer | 72.39±0.28 | 73.93±0.21 | 74.12±0.15 | 74.24±0.27 | 73.95±0.22 | 73.46±0.14 | 72.60±0.22 |

## 4 总结与未来工作

图对比学习 (GCL) 作为一种强大的无监督图表示学习范式, 在近年来取得了显著进展. 然而, 现有方法大多采用全局均匀的图增强策略, 忽视了真实网络中广泛存在的节点角色异质性, 特别是核心-边缘结构, 这可能导致对核心结构的破坏, 进而违反标签不变性假设, 限制模型性能. 针对这一挑战, 本文提出了一种基于核心-边缘结构感知的图对比学习 (CP-GCL) 框架. 该框架的核心思想是首先通过一种能够计算节点连续核心度的先进算法精准识别网络的核心与边缘区域, 然后仅对边缘节点施加增强操作, 以保护核心结构的完整性并模拟网络边缘的动态演化. 更进一步, 本文设计了一种核心-边缘感知对比损失函数, 该损失函数利用节点的核心度分数对局部-全局对比项进行加权, 从而引导模型更加关注对全局结构重要的核心节点, 同时降低边缘节点潜在的干扰. 在多个节点分类和图分类基准数据集上的大量实验结果表明, 提出的 CP-GCL 方法在性能上显著优于现有的多种先进自监督学习模型, 验证了其在学习高质量、鲁棒且具有良好泛化能力的图表示方面的有效性.

本文提出的 CP-GCL 框架主要基于一个核心假设即真实世界的网络普遍呈现核心-边缘 (CP) 的异质结构. 然而, 这一假设并非对所有图都成立. 例如, 在某些高度均质化的图 (如 Erdős-Rényi 随机图或部分规则的社交网络) 中, 节点之间的连接相对均匀, 缺乏明显的核心-边缘分层. 在这种情况下, 我们方法的有效性确实会受到限制. 当应用于非 CP 型图时, 本文的核心度识别算法可能难以区分出结构重要性差异, 导致所有节点的 $CS(i)$ 分数趋于一致. 这会带来两个主要后果: 首先, 核心-边缘对比损失中的权重 $CS_i$ 将几乎相同, 使损失函数退化为标准的、无加权的对比损失. 其次, 基于阈值 $p_c$ 划分的“边缘”节点将接近于随机抽样, 导致我们的局部增强策略退化为一种作用于部分节点的随机增强. 因此, 当图不具备明显 CP 特征时, 我们预期 CP-GCL 的性能优势将不复存在, 其表现将退化至接近于 GRACE 或 GraphCL 等采用全局随机增强的基线方法. 未来的工作将探索如何设计一个自适应模块, 首先量化输入图的“CP 结构强度”, 然后根据该强度自动调整核心保护策略的介入程度, 使模型在均质图上自动回退到全局增强, 从而提高框架的普适性.

**References**

[1] Zhao YH, Wang YJ, Wang ZK, Shan W, Huang MM, Wang XW. Graph contrastive learning with progressive augmentations. In: Proc. of the 31st ACM SIGKDD Conf. on Knowledge Discovery and Data Mining V.1. Toronto: ACM, 2025. 2079–2088. [doi: 10.1145/3690624.3709307]

[2] Wan GC, Tian YJ, Huang WK, Chawla NV, Ye M. S3GCL: Spectral, swift, spatial graph contrastive learning. In: Proc. of the 41st Int'l Conf. on Machine Learning. Vienna: JMLR.org, 2024. 2044.

[3] Tan SY, Li DY, Jiang RH, Zhang Y, Okumura M. Community-invariant graph contrastive learning. In: Proc. of the 41st Int'l Conf. on Machine Learning. Vienna: JMLR.org, 2024. 1938.

[4] Ghose A, Zhang YX, Hao JY, Coates M. Spectral augmentations for graph contrastive learning. In: Proc. of the 26th Int'l Conf. on Artificial Intelligence and Statistics. PMLR, 2023. 11213–11266.

[5] Lin L, Chen JH, Wang HN. Spectral augmentation for self-supervised learning on graphs. arXiv:2210.00643, 2022.

[6] You YN, Chen TL, Shen Y, Wang ZY. Graph contrastive learning automated. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 12121–12132.

[7] Kim DK, Baek J, Hwang SJ. Graph self-supervised learning with accurate discrepancy learning. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 1024.

[8] Holme P. Core-periphery organization of complex networks. Physical Review E, 2005, 72(4): 046111. [doi: 10.1103/PhysRevE.72.

046111]

[9] Verma T, Russmann F, Araújo NAM, Nagler J, Herrmann HJ. Emergence of core-peripheries in networks. Nature Communications, 2016, 7: 10441. [doi: 10.1038/ncomms10441]

[10] Yanchenko E, Sengupta S. Core-periphery structure in networks: A statistical exposition. Statistics Surveys, 2023, 17: 42–74. [doi: 10.1214/23-SS141]

[11] Mukherjee CS, Zhang JP. Balanced ranking with relative centrality: A multi-core periphery perspective. In: Proc. of the 2025 Int'l Conf. on Representation Learning. Appleton: OpenReview.net, 2025.

[12] You YN, Chen TL, Sui YD, Chen T, Wang ZY, Shen Y. Graph contrastive learning with augmentations. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 488.

[13] Holme P, Saramäki J. Temporal networks. Physics Reports, 2012, 519(3): 97–125. [doi: 10.1016/j.physrep.2012.03.001]

[14] Liu YX, Jin M, Pan SR, Zhou C, Zheng Y, Xia F, Yu PS. Graph self-supervised learning: A survey. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(6): 5879–5900. [doi: 10.1109/TKDE.2022.3172903]

[15] Wu LR, Lin HT, Tan C, Gao ZY, Li SZ. Self-supervised learning on graphs: Contrastive, generative, or predictive. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(4): 4216–4235. [doi: 10.1109/TKDE.2021.3131584]

[16] Yao X, Gao JY, Xu CS. Self-supervised graph contrastive learning for video question answering. Ruan Jian Xue Bao/Journal of Software, 2023, 34(5): 2083–2100 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6775.htm [doi: 10.13328/j.cnki.jos.006775]

[17] Wang JX, Guan JH, Zhou SG. Molecular property prediction by contrastive learning with attention-guided positive sample selection. Bioinformatics, 2023, 39(5): btad258. [doi: 10.1093/bioinformatics/btad258]

[18] Li SL, Zhou JB, Xu T, Dou DJ, Xiong H. GeomGCL: Geometric graph contrastive learning for molecular property prediction. Proc. of the AAAI Conf. on Artificial Intelligence, 2022, 36(4): 4541–4549. [doi: 10.1609/aaai.v36i4.20377]

[19] Zheng JH, Yang YD, Dai ZM. Subgraph extraction and graph representation learning for single cell Hi-C imputation and clustering. Briefings in Bioinformatics, 2024, 25(1): bbad379. [doi: 10.1093/bib/bbad379]

[20] Xiong ZH, Luo JW, Shi WW, Liu Y, Xu ZY, Wang B. scGCL: An imputation method for scRNA-seq data based on graph contrastive learning. Bioinformatics, 2023, 39(3): btad098. [doi: 10.1093/bioinformatics/btad098]

[21] Zhu YQ, Xu YC, Yu F, Liu Q, Wu S, Wang L. Deep graph contrastive representation learning. arXiv:2006.04131, 2020.

[22] Zhu YQ, Xu YC, Yu F, Liu Q, Wu S, Wang L. Graph contrastive learning with adaptive augmentation. In: Proc. of the 2021 Web Conf. Ljubljana: ACM, 2021. 2069–2080. [doi: 10.1145/3442381.3449802]

[23] Xu DK, Cheng W, Luo DS, Chen HF, Zhang X. InfoGCL: Information-aware graph contrastive learning. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. Curran Associates Inc., 2021. 2327.

[24] Qiu JZ, Chen QB, Dong YX, Zhang J, Yang HX, Ding M, Wang KS, Tang J. GCC: Graph contrastive coding for graph neural network pre-training. In: Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. ACM, 2020. 1150–1160. [doi: 10.1145/3394486.3403168]

[25] Jiao YZ, Xiong Y, Zhang JW, Zhang Y, Zhang TQ, Zhu YY. Sub-graph contrast for scalable self-supervised graph representation learning. In: Proc. of the 2020 IEEE Int'l Conf. on Data Mining (ICDM). Sorrento: IEEE, 2020. 222–231. [doi: 10.1109/ICDM50108.2020.00031]

[26] Jin M, Zheng YZ, Li YF, Gong C, Zhou C, Pan SR. Multi-scale contrastive Siamese networks for self-supervised graph representation learning. In: Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. 2021. 1477–1483. [doi: 10.24963/ijcai.2021/204]

[27] Hassani K, Khasahmadi AH. Contrastive multi-view representation learning on graphs. In: Proc. of the 37th Int'l Conf. on Machine Learning. JMLR.org, 2020. 385.

[28] Liu YX, Zheng Y, Zhang DK, Chen HX, Peng H, Pan SR. Towards unsupervised deep graph structure learning. In: Proc. of the 2022 ACM Web Conf. ACM, 2022. 1392–1403. [doi: 10.1145/3485447.3512186]

[29] Yin YH, Wang QZ, Huang SY, Xiong HY, Zhang X. AutoGCL: Automated graph contrastive learning via learnable view generators. In: Proc. of the 36th AAAI Conf. on Artificial Intelligence. AAAI Press, 2022. 8892–8900. [doi: 10.1609/aaai.v36i8.20871]

[30] Zhang SC, Hu ZN, Subramonian A, Sun YZ. Motif-driven contrastive learning of graph representations. IEEE Trans. on Knowledge and Data Engineering, 2024, 36(8): 4063–4075. [doi: 10.1109/TKDE.2024.3364059]

[31] Feng SY, Jing BY, Zhu YD, Tong HH. Adversarial graph contrastive learning with information regularization. In: Proc. of the 2022 ACM Web Conf. ACM, 2022. 1362–1371. [doi: 10.1145/3485447.3512183]

[32] Luo X, Ju W, Gu YY, Mao ZY, Liu LC, Yuan YH, Zhang M. Self-supervised graph-level representation learning with adversarial contrastive learning. ACM Trans. on Knowledge Discovery from Data, 2024, 18(2): 34. [doi: 10.1145/3624018]

[33] Csermely P, London A, Wu LY, Uzzi B. Structure and dynamics of core/periphery networks. Journal of Complex Networks, 2013, 1(2): 93–123. [doi: 10.1093/comnet/cnt016]

[34] Borgatti SP, Everett MG. Models of core/periphery structures. Social Networks, 2000, 21(4): 375–395. [doi: 10.1016/S0378-8733(99)00019-2]
[35] Krugman P. The Self Organizing Economy. Hoboken: John Wiley & Sons, Inc., 1996.
[36] Magone J, Laffan B, Schweiger C. Core-periphery Relations in the European Union: Power and Conflict in a Dualist Political Economy. London: Routledge, 2016. [doi: 10.4324/9781315712994]
[37] Lordan O, Sallan JM. Core and critical cities of global region airport networks. Physica A: Statistical Mechanics and Its Applications, 2019, 513: 724–733. [doi: 10.1016/j.physa.2018.08.123]
[38] Lordan O, Sallan JM. Analyzing the multilevel structure of the European airport network. Chinese Journal of Aeronautics, 2017, 30(2): 554–560. [doi: 10.1016/j.cja.2017.01.013]
[39] Zelnio R. Identifying the global core-periphery structure of science. Scientometrics, 2012, 91(2): 601–615. [doi: 10.1007/s11192-011-0598-0]
[40] Wedell E, Park M, Korobskiy D, Warnow T, Chacko G. Center-periphery structure in research communities. Quantitative Science Studies, 2022, 3(1): 289–314. [doi: 10.1162/qss_a_00184]
[41] Sedita SR, Caloffi A, Lazzeretti L. The invisible college of cluster research: A bibliometric core-periphery analysis of the literature. Industry and Innovation, 2020, 27(5): 562–584. [doi: 10.1080/13662716.2018.1538872]
[42] Yang JF, Zhang M, Shen KN, Ju XF, Guo XT. Structural correlation between communities and core-periphery structures in social networks: Evidence from twitter data. Expert Systems with Applications, 2018, 111: 91–99. [doi: 10.1016/j.eswa.2017.12.042]
[43] Cattani G, Ferriani S. A core/periphery perspective on individual creative performance: Social networks and cinematic achievements in the Hollywood film industry. Organization Science, 2008, 19(6): 824–844. [doi: 10.1287/orsc.1070.0350]
[44] Michailidis G. Statistical challenges in biological networks. Journal of Computational and Graphical Statistics, 2012, 21(4): 840–855. [doi: 10.1080/10618600.2012.738614]
[45] Alm E, Arkin AP. Biological networks. Current Opinion in Structural Biology, 2003, 13(2): 193–202. [doi: 10.1016/S0959-440X(03)00031-9]
[46] Girvan M, Newman MEJ. Community structure in social and biological networks. Proc. of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821–7826. [doi: 10.1073/pnas.122653799]
[47] Tang WL, Zhao LT, Liu W, Liu YP, Yan B. Recent advance on detecting core-periphery structure: A survey. CCF Trans. on Pervasive Computing and Interaction, 2019, 1(3): 175–189. [doi: 10.1007/s42486-019-00016-z]
[48] Brusco M. An exact algorithm for a core/periphery bipartitioning problem. Social Networks, 2011, 33(1): 12–19. [doi: 10.1016/j.socnet.2010.08.002]
[49] Zhang X, Martin T, Newman MEJ. Identification of core-periphery structure in networks. Physical Review E, 2015, 91(3): 032803. [doi: 10.1103/physreve.91.032803]
[50] Caron F, Fox EB. Sparse graphs using exchangeable random measures. Journal of the Royal Statistical Society Series B: Statistical Methodology, 2017, 79(5): 1295–1366. [doi: 10.1111/rssb.12233]
[51] Naik C, Caron F, Rousseau J. Sparse networks with core-periphery structure. Electronic Journal of Statistics, 2021, 15(1): 1814–1868. [doi: 10.1214/21-ejs1819]
[52] Da Silva MR, Ma HW, Zeng AP. Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. Proc. of the IEEE, 2008, 96(8): 1411–1420. [doi: 10.1109/JPROC.2008.925418]
[53] Rossa FD, Dercole F, Piccardi C. Profiling core-periphery network structure by random walkers. Scientific Reports, 2013, 3(1): 1467. [doi: 10.1038/srep01467]
[54] Cucuringu M, Rombach P, Lee SH, Porter MA. Detection of core-periphery structure in networks using spectral methods and geodesic paths. European Journal of Applied Mathematics, 2016, 27(6): 846–887. [doi: 10.1017/S095679251600022X]
[55] Rombach P, Porter MA, Fowler JH, Mucha PJ. Core-periphery structure in networks (revisited). SIAM Review, 2017, 59(3): 619–646. [doi: 10.1137/17M1130046]
[56] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, 2016.
[57] Mernyei P, Cangea C. Wiki-CS: A Wikipedia-based benchmark for graph neural networks. arXiv:2007.02901v2, 2022.
[58] Shchur O, Mumme M, Bojchevski A, Günnemann S. Pitfalls of graph neural network evaluation. In: Proc. of the 2018 Relational Representation Learning Workshop. NeurIPS, 2018.
[59] Morris C, Kriege NM, Bause F, Kersting K, Mutzel P, Neumann M. TUDataset: A collection of benchmark datasets for learning with graphs. arXiv:2007.08663, 2020.
[60] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2014. 701–710. [doi: 10.1145/2623330.2623732]

[61] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016. 855–864. [doi: 10.1145/2939672.2939754]
[62] Kipf T, Welling M. Variational graph auto-encoders. arXiv:1611.07308, 2016.
[63] Velickovic P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. arXiv:1809.10341, 2018.
[64] Thakoor S, Tallec C, Azar MG, Azabou M, Dyer EL, Munos R, Veličković P, Valko M. Large-scale representation learning on graphs via bootstrapping. 2022. https://hal.science/hal-05413315v1/document
[65] Zhang HR, Wu QT, Yan JC, Wipf D, Yu PS. From canonical correlation analysis to self-supervised graph neural networks. In: Proc. of the 35th Conf. on Neural Information Processing Systems (NeurIPS 2021). 2021. 76–89.
[66] Mo YJ, Peng L, Xu J, Shi XS, Zhu XF. Simple unsupervised graph representation learning. In: Proc. of the 36th AAAI Conf. on Artificial Intelligence. AAAI Press, 2022. 7797–7805. [doi: 10.1609/aaai.v36i7.20748]
[67] Xu KYL, Jegelka S, Hu WH, Leskovec J. How powerful are graph neural networks? In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.
[68] Borgwardt KM, Kriegel HP. Shortest-path kernels on graphs. In: Proc. of the 5th IEEE Int'l Conf. on Data Mining. Houston: IEEE, 2005. 8. [doi: 10.1109/ICDM.2005.132]
[69] Shervashidze N, Vishwanathan S, Petri T, Mehlhorn K, Borgwardt K. Efficient graphlet kernels for large graph comparison. In: Proc. of the 12th Int'l Conf. on Artificial Intelligence and Statistics. PMLR, 2009. 488–495.
[70] Shervashidze N, Schweitzer P, Van Leeuwen EJ, Mehlhorn K, Borgwardt KM. Weisfeiler-Lehman graph kernels. The Journal of Machine Learning Research, 2011, 12: 2539–2561.
[71] Yanardag P, Vishwanathan SVN. Deep graph kernels. In: Proc. of the 21st ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Sydney: ACM, 2015. 1365–1374. [doi: 10.1145/2783258.2783417]
[72] Narayanan A, Chandramohan M, Venkatesan R, Chen LH, Liu Y, Jaiswal S. graph2vec: Learning distributed representations of graphs. arXiv:1707.05005, 2017.
[73] Adhikari B, Zhang Y, Ramakrishnan N, Prakash BA. Sub2Vec: Feature learning for subgraphs. In: Proc. of 22nd Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining. Melbourne: Springer, 2018. 170–182. [doi: 10.1007/978-3-319-93037-4_14]
[74] Sun FY, Hoffman J, Verma V, Tang J. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv:1908.01000, 2019.
[75] Suresh S, Li P, Hao C, Neville J. Adversarial graph augmentation to improve graph contrastive learning. In: Proc. of the 35th Conf. on Neural Information Processing Systems (NeurIPS 2021). 2021. 15920–15933.
[76] Li SH, Wang X, Zhang A, Wu YX, He XN, Chua TS. Let invariant rationale discovery inspire graph contrastive learning. In: Proc. of the 39th Int'l Conf. on Machine Learning. PMLR, 2022. 13052–13065.
[77] Xia J, Wu LR, Chen JT, Hu BZ, Li SZ. SimGRACE: A simple framework for graph contrastive learning without data augmentation. In: Proc. of the 2022 ACM Web Conf. ACM, 2022. 1070–1079. [doi: 10.1145/3485447.3512156]
[78] Wu JR, Chen XY, Shi BW, Li SZ, Xu K. SEGA: Structural entropy guided anchor view for graph contrastive learning. In: Proc. of the 40th Int'l Conf. on Machine Learning. Honolulu: JMLR.org, 2023. 1552.
[79] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9(86): 2579–2605.
[80] Gallagher RJ, Young JG, Welles BF. A clarified typology of core-periphery structure in networks. Science Advances, 2021, 7(12): eabc9800. [doi: 10.1126/sciadv.abc9800]
[81] Yang K, Liu Y, Zhao ZJ, Ding PJ, Zhao WQ. Local structure-aware graph contrastive representation learning. Neural Networks, 2024, 172: 106083. [doi: 10.1016/j.neunet.2023.12.037]

## 附中文参考文献

[16] 姚暄, 高君宇, 徐常胜. 基于自监督图对比学习的视频问答方法. 软件学报, 2023, 34(5): 2083–2100. http://www.jos.org.cn/1000-9825/6775.htm [doi: 10.13328/j.cnki.jos.006775]

## 作者简介

**王业江**, 博士生, 主要研究领域为自监督学习, 图学习, 多标签学习.
**赵宇海**, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为数据库, 数据挖掘, 机器学习, 软件工程, 生物信息学.
**黄苗苗**, 博士生, CCF 学生会员, 主要研究领域为图机器学习, 生物分子预测.
**王梅霞**, 博士生, CCF 学生会员, 主要研究领域为图机器学习, 多模态学习.
**李芳婷**, 博士生, 主要研究领域为图机器学习, 进化计算, 分布式计算.
**王兴伟**, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为未来互联网, 云计算, 网络空间安全.