一种用于边缘计算环境中根因定位与分析联合任务的级联图神经网络

Duneesha Fernando、Maria A. Rodriguez、Rajkumar Buyya

arXiv:2603.01447v1 [cs.DC] 2 Mar 2026

摘要——**边缘计算环境承载着日益复杂的基于微服务的物联网应用，这些应用容易出现性能异常，并在依赖服务之间传播。识别故障组件（根因定位）及其底层故障类型（根因分析）对于及时缓解问题至关重要。目前，监督式图神经网络（GNN）代表了联合执行根因定位与分析的最先进水平。然而，现有方法依赖于在全系统图上进行集中式处理，导致推理延迟高，且在大型分布式边缘环境中的可扩展性有限。本文提出一种用于联合执行 RCL 和故障类型识别的级联图神经网络框架，明确应对这些可扩展性挑战。我们的方法采用通信驱动的聚类，将大型服务图划分为高交互社区，并利用包含两个子网络的级联网络，分层执行 RCL/RCA。通过将消息传递限制在规模更小、结构化的子图上，所提出的框架在保留关键信息的同时，显著降低了计算复杂度。我们在 MicroCERCL 基准以及使用 iAnomaly 仿真框架生成的大规模数据集上评估了所提出的方法。实验结果表明，与集中式图神经网络基线相比，该级联架构的诊断准确率相当；同时，随着图规模增大，其推理延迟仍近乎恒定，从而能够在边缘计算环境中实现可扩展且可执行的智能运维。**

索引词—**边缘计算、根因定位、根因分析、异常检测、微服务、图神经网络、性能诊断。**

一、引言

边云融合环境由地理位置分散、计算、存储和网络能力各异的设备组成。物联网应用通常采用微服务架构，并部署在这些分布式环境中，以满足各模块的服务质量（QoS）要求，同时优化资源利用率[1], [2]。随着时间推移，这些基于微服务的物联网应用容易受到资源占用（例如 CPU 或内存）和资源争用所引发的性能异常影响，从而损害其服务质量并违反服务等级协议 [3], [4]。因此，在边缘计算环境中检测基于微服务的物联网应用中的性能异常，并最终加以缓解，至关重要。

D. Fernando、M. A. Rodriguez 和 R. Buyya 隶属于墨尔本大学计算与信息系统学院的量子云计算与分布式系统（qCLOUDS）实验室，所在地为澳大利亚 Parkville。

然而，异常可能沿通信依赖和共置依赖传播，导致多个服务表现出异常，尽管其根本原因只有一个[5], [6]。确定导致观测到的异常的确切微服务，即根因定位（RCL），对于及时有效地缓解问题至关重要。除了定位问题组件之外，确定异常发生的原因同样重要。例如，根本原因可能是 CPU 占用、内存压力或网络延迟。确定异常原因的过程称为根因分析（RCA）。所有根因分析研究都会进行根因定位，而分析可以在不同层级开展 [7], [8]。一些根因分析研究侧重于识别导致异常的具体指标[9]–[11], ，另一些则专注于确定故障类型 [12]–[14]。此外，还有更细致的根因分析层级，会在源代码层面检查问题，这对于制定解决性能问题的长期方案很有价值 [15]。在本研究中，我们关注将故障类型诊断与根因定位相结合的根因分析类型。这种分析能够为自主智能运维（面向IT运维的人工智能）系统提供必要信息，帮助其有效解决性能问题 [14]。

虽然根因定位和根因分析常被分别讨论，但这两项任务彼此相互关联。它们都源于同一异常事件，并依赖相似的信号，例如时间性能模式、服务间依赖关系和异常传播行为。因此，近期研究提倡联合执行根因定位和根因分析，而非将二者孤立处理 [12]–[14]。目前，监督式图神经网络代表了联合根因定位与根因分析的最先进水平 [16]。通过联合学习根因定位和根因分析，模型可以共享前期特征提取器（例如 CNN 层和 GCN 网络层），从而捕捉有助于两项任务的有用时间模式和服务间依赖关系。与分别训练独立模型相比，这种方法能够实现更快速、更准确的定位和诊断 [12], [13]。

尽管效果显著，图神经网络的计算复杂度仍然很高 [5], [17]。在由大量设备和微服务组成的边缘计算环境中，这意味着需要构建包含大量节点的图，作为图神经网络的输入。此类大规模图会增加消息传递的计算成本，导致诊断速度变慢，缓解措施也会延迟。

为应对这一可扩展性挑战，我们提出了一种用于联合根因定位与根因分析的级联图神经网络架构，

专为大规模边缘部署而设计。我们根据微服务之间的通信依赖关系将其划分为多个簇，每个簇代表一组交互密切的微服务。级联图神经网络由两个网络组成：提议网络（P-Net）和输出网络（O-Net）。P-Net 在各个簇上运行并生成集群级输出，通过聚焦于有限区域来缩小搜索空间。O-Net 将 P-Net 生成的每个簇的潜在表示以及簇间边作为输入，构建表示边缘系统的图。通过将每个簇视为一个节点，O-Net 进一步缩小了搜索空间。由于与传统图神经网络相比，两个网络处理的搜索空间都更小，因此级联网络的整体复杂度更低。尽管在缩小后的感受野内运行可能会造成相较于传统图神经网络的一些信息损失，但我们的聚类技术能够根据通信依赖关系有效地将微服务分组，从而缓解这一损失。这确保了在提高效率的同时维持准确率。据我们所知，我们是首个提出计算高效的级联图神经网络框架、用于边缘计算环境下联合根因定位与根因分析的工作，并明确解决了大规模分布式边缘云系统中存在的可扩展性和延迟约束。

我们在公开的 MicroCERCL 基准数据集 [18]以及使用 iAnomaly 仿真框架 [19]生成的大规模数据集上评估了所提出的级联图神经网络方法。Micro-CERCL 上的实验结果表明，该级联模型的诊断准确率与集中式图神经网络基线相当；而 iAnomaly 上的可扩展性实验则表明，其推理时间近乎恒定，与延迟随图规模增大而增加的集中式模型形成对比。

本文其余部分安排如下。第二节回顾根因定位与根因分析、联合 RCL/RCA 技术以及面向效率的诊断方法的背景和相关工作。第三节介绍一种用于联合根因定位与根因分析的集中式图神经网络架构，并将其作为比较基线。第四节介绍所提出的级联图神经网络框架，包括通信驱动的聚类策略和分层推理设计。第五节报告在真实世界和大规模仿真数据集上的实验评估与可扩展性分析。最后，第六节总结全文并阐述未来研究方向。

II. 背景与相关工作

本节概述基于微服务的边缘与云系统中根因定位与分析的背景及相关工作。我们首先从诊断粒度的角度回顾现有的根因分析方法，涵盖从粗粒度定位到细粒度故障类型分析和代码级分析。随后，我们讨论近期联合执行根因定位与根因分析的技术，并重点介绍代表当前最先进水平的基于监督学习的图方法。最后，我们考察了以提升根因分析技术的效率和可扩展性为目标的相关工作，并指出现有方案在大规模边缘环境中的局限性。

这也说明了我们提出级联图神经网络框架的必要性。

A. 根因分析的层次

基于微服务的系统中的根因分析涵盖了不同粒度的诊断，从粗粒度定位到细粒度诊断。在最粗粒度层面，根因定位（RCL），也称为粗粒度定位，旨在确定导致已观察到的性能异常的微服务。根因定位通常被视为根因分析的首要且最基础步骤，几乎所有根因分析方法都会显式或隐式地执行这一环节 [7], [8]。一些研究专门关注此任务，包括 MicroRCA[20], 、MicroGBPM [6], 、MicroCERCL [18], 和 MicroEGRCL[21]。尽管通常可以快速完成粗粒度定位，但仅识别出故障微服务所提供的诊断价值有限，因为异常的根本原因仍然未知，站点可靠性工程师不得不手动调查内存耗尽、CPU争用或请求模式异常等问题 [11], [22]。此外，此层面的缓解措施（例如重启或重新部署整个服务）可能无法解决真正的根本原因，还可能对依赖服务造成不必要的干扰，并且与细粒度缓解措施相比，会延长恢复时间 [8]。

除了微服务级定位外，根因分析技术也日益纳入更细粒度的分析。一个中间层级是致因指标定位，旨在识别导致检测到的异常的具体性能指标（例如 CPU使用率、延迟、内存消耗）。这一类别的代表性方法包括 MicroDiag[9], AAMR [10], HeMIRCA [11], CausalRCA [8], PDiagnose[23],和 Wu等人 [22]。这一层级的分析有助于采取更有针对性的缓解措施，例如资源扩缩或限流；与重启服务相比，这些措施通常能更快恢复系统，并减少对系统的干扰 [8]。仅采用根因定位的方法，以及结合根因定位与致因指标定位的方法，通常都被建模为排序问题，即根据微服务或指标成为根因的可能性对其排序。

更高级别的根因分析侧重于故障类型识别，旨在确定异常本身的根本原因。在这一层级，异常会被归因于特定的故障类别，例如 CPU占用、内存泄漏、网络延迟或 I/O争用。DiagFusion [12], MicroIRC [13], DejaVu [14], Brandon等人 [24], 和 MicroHECL [25]等方法通过将异常分类为预定义的故障类型，明确解决了这一问题。与指标级诊断相比，故障类型识别具有一项关键优势：它能将观测到的异常行为直接映射到语义明确的系统故障，从而实现更精准、自动化程度更高的缓解措施。因此，这类方法通常被建模为分类问题。

在最细粒度上，根因分析（RCA）在源代码级别进行，目标是将性能异常定位到特定代码区域或组件。Nezha [15]是此类别的典型代表，可直接在代码级别识别性能

问题。尽管这类方法对于制定长期、永久性的修复方案很有价值，但通常需要对应用程序源代码或运行时系统进行大量插桩，这会在生产环境中引入显著开销 [22]。因此，源代码级根因分析通常不太适用于在线诊断和实时缓解，尤其是在对延迟敏感的系统中。

在本研究中，我们关注一种将故障类型识别与根因定位（RCL）相结合的根因分析（RCA）设置。这一层级的诊断为自主AIOps系统提供关键信息，使其能够有效应对性能问题，并与可操作诊断的概念密切契合 [14]。要真正具备可操作性，故障定位方案必须同时提供两项关键信息：（1）故障发生在哪里（通过根因定位确定故障组件）；（2）发生的是哪种故障（故障类型）。这两者结合通常称为故障单元 [14]。不同的故障类型（例如内存泄漏与CPU耗尽）需要截然不同的缓解策略。仅能定位故障服务的方法，无法自主判断是应该扩展资源、调整调度策略，还是应用内存管理技术。通过同时识别故障组件和故障类型，根因分析结果便可直接用于采取行动，使AIOps系统不再局限于简单告警，而能够实施有针对性的自动化缓解策略[26]。

B. 联合根因定位与根因分析技术

尽管根因定位和根因分析通常被视为两项独立任务，但二者相互关联。故障的位置和性质都源自同一异常事件，并由共同的系统信号驱动，包括时间性能模式、服务间依赖关系以及异常传播行为。因此，近年来，将根因定位与故障类型识别相结合的方法日益受到关注 [8], [12]–[14]。

Wu等人 [9], 指出，根因分析技术主要分为三类：1）深度学习，2）模式识别，以及3）因果推断。联合根因定位与根因分析主要通过深度学习和基于模式的方法进行探索 [8]。其中，深度学习方法尤其侧重于有监督的图神经网络，这类方法目前代表了联合根因定位与故障类型识别的最先进水平 [16]。典型案例包括 DiagFusion[12],，它利用图神经网络进行联合诊断；MicroIRC[13],，它将个性化随机游走算法与图神经网络模型相结合；以及 DejaVu [14],，它采用图注意力网络（图注意力网络）。另一方面，基于模式的方法（例如 Brandon等人 [24], 的研究）通过将异常子图与预定义的故障模式库进行匹配来实现联合诊断。

由于能够对复杂的服务间依赖关系进行建模，监督式图神经网络如今已成为联合根因定位与根因分析的最先进方法。尽管有人可能认为监督式方法难以识别此前未曾出现的故障类型，但实证证据表明，运行系统中的相当大比例故障会反复发生，有报告指出这一比例可能高达 74% [7]。尽管监督式方法通常需要大量标注数据和训练工作，而纯无监督方法往往无法实现完全自动化的端到端根因分析 [18]。因此，基于图神经网络的监督式方法可作为自动化智能运维的一种切实有效的起点，尤其适用于边缘环境。

C. 高效的根因定位与根因分析技术

尽管图神经网络效果显著，但众所周知，其会产生不可忽视的计算开销，这主要源于在图结构上进行迭代消息传递的成本 [5], [17]。边缘计算环境通常由大量分布式设备、微服务及其交互构成，因此输入图神经网络的图中会包含大量节点和边。随着图规模增大，图神经网络推理的计算成本会随节点数、边数和消息传递层数成比例增长，导致诊断延迟加长。具体而言，大规模图会加剧邻域聚合和特征传播的开销，而这些操作必须针对每个节点跨多个图神经网络层执行。这种额外的复杂度可能会拖慢根因识别和故障诊断，从而延误缓解措施。在对延迟敏感的边缘环境中，这种延迟尤其成问题，因为及时响应异常至关重要。

为应对这一可扩展性挑战，我们提出了一种用于联合根因定位与根因分析的级联图神经网络架构，专门面向大规模边缘部署。该框架由两个依次运行的网络组成，作用于基于通信依赖关系、由交互密切的微服务构成的簇。通过在精简且结构化的子图上应用级联模型的两个阶段，所提出的方法显著缩小了有效搜索空间，从而降低了相较于传统图神经网络的计算复杂度。基于通信的聚类能够保留关键依赖信息，最大限度地减少因感受野缩小而造成的信息损失，并维持诊断准确率。因此，所提出的框架在效率与准确率之间实现了有效平衡。

表 I 展示了云端和边缘根因定位与分析技术在对效率和准确率关注方面的比较。大多数关于云端 RCL/A 的研究主要侧重于准确率的提升，仅评估这一单一方面 [6], [27], [5], [11], [9], [8], [22], [13]。一些研究也涉及了效率，尤其是其所提方法的定位时间。例如，AAMR [10], MicroEGRCL [21],和 Grace [16] 声称能够提供更快的推理时间。然而，这些研究均未专门以效率为优先目标来设计其方法。此外，现有的面向边缘的解决方案，包括 MicroCERCL [18]以及 Kalinagac 等人 [29], 提出的根因分析框架，并未明确纳入效率方面的考量。特别是，MicroCERCL 报告的推理时间长于无监督启发式方法，这主要是由于网络的复杂性 [18]。

MicroHECL [25]和 PDiagnose [23] 是云端的 RCL/A 技术，专门致力于提升效率，尤其是提高定位速度。MicroHECL

表 I：云端和边缘根因定位与分析技术综述：准确率与效率侧重点

Work	仅限云端/云端-边缘	GNN	RCL/RCA 类型	准确率 侧重点	效率 重点	备注
[6], [27], [5], [11], [9], [8], [22], [13]	仅限云端	混合	根因定位；根因分析（指标 / 故障类型）	✓	×	仅关注准确率
[21], [20], [10], [16], [24], [14], [12], [28], [15]	仅限云端	混合	根因定位；根因分析（指标 / 故障 / 资源类型）	✓	△	声称推理速度更快；效率并非设计优先事项
PDiagnose [23], MicroHECL [25]	仅限云端	No	根因定位；根因分析（指标 / 故障类型）	△	✓	专注于提高效率；可能会降低准确率
MicroCERCL [18]	云边	Yes	仅根因定位	✓	×	侧重准确率；报告的推理时间较长
Kalinagac 等人 [29]	云边	No	根因分析（故障类型）	✓	×	仅关注准确率
我们的工作	云边	Yes	联合根因定位 +根因分析（故障类型）	✓	✓	带聚类的级联图神经网络；平衡准确率与效率

✓= 主要关注/已解决， △= 部分解决/可能折中， ×=未解决/被忽略

通过高效遍历服务依赖图，并在异常传播链分析期间使用剪枝技术排除无关的服务调用，从而提升效率。另一方面，PDiagnose 试图通过省去计算开销较大的依赖图构建阶段，并对异常队列采用基于投票的定位过程来提高效率。然而，这两种方法采用的策略都相对简单，可能会损害诊断准确率。鉴于需要在有效性和效率之间取得适当平衡，我们提出的方法采用基于图神经网络的方法，在保持高准确率的同时缩短诊断时间。

总体而言，现有根因分析技术在提升诊断准确率方面已取得实质性进展，尤其是通过基于监督学习的图神经网络联合根因定位与根因分析模型。然而，这些方法大多基于集中式处理假设设计，并在全系统图上运行，这在大规模边缘环境中会导致较高的计算开销。为更好地理解此类集中式设计的优势与局限，我们首先在下一节回顾一种具有代表性的集中式联合根因定位/根因分析图神经网络架构。该分析为引入我们所提出的级联框架奠定基础，该框架明确应对先前工作中指出的可扩展性与延迟挑战。

III. 集中式联合根因定位/根因分析图神经网络架构

本节介绍一种具有代表性的集中式图神经网络架构，用于联合进行 RCL 和故障类型识别，并作为基准，以便后续与我们提出的级联框架进行比较。我们首先介绍用于建模边缘计算环境及其微服务依赖关系的图表示（第 III-A 节），随后说明如何根据节点级和边级时间序列指标构建图神经网络输入表示（第 III-B 节）。接着，我们回顾集中式基于图神经网络的联合 RCL/RCA 模型的常见设计原则（第 III-C 节）。最后，介绍本文采用的整体集中式架构（第 III-D 节）。

A. 边缘计算环境的图建模

设边缘基础设施由一组边缘设备组成

$$\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\},$$

以及部署在该环境中的一组微服务

$$\mathcal{M} = \{m_1, m_2, \ldots, m_{|\mathcal{M}|}\}.$$

每个微服务 $m_i \in \mathcal{M}$恰好部署在一台边缘设备上。部署映射可定义为

$$\delta : \mathcal{M} \to \mathcal{E},$$

其中， δ（m_i）表示微服务 m_i 所部署的设备。

基于图神经网络的分析要求将系统表示为一张图，其中节点表示计算实体，边则反映结构关系或运行时关系。在边缘云微服务系统的背景下，我们使用拓扑图对环境进行建模

$$\mathcal{G} = (\mathcal{V}, \mathcal{L}),$$

其中，顶点集合 $\mathcal{V}$ 包含微服务及其部署所在的边缘设备：

$$\mathcal{V} = \mathcal{M} \cup \mathcal{E}$$

边集合 $\mathcal{L}$ 表示这些顶点之间的关系，由两种不同类型组成：

$$\mathcal{L} = \mathcal{L}_{\text{comm}} \cup \mathcal{L}_{\text{dep}}$$

1）通信边（L_{comm}）用于捕捉微服务之间的运行时交互，这些交互根据跟踪数据得出：

$$\mathcal{L}_{\text{comm}} \subseteq \mathcal{M} \times \mathcal{M},$$

其中，(m_i, m_j) $\in \mathcal{L}_{\text{comm}}$，当微服务 m_i 与 m_j 通信时。

2）部署边（L_{dep}）将每个微服务连接到其托管所在的边缘设备，这些信息来自部署配置：

$$\mathcal{L}_{\text{dep}} = \{(m_i, \delta(m_i)) \mid m_i \in \mathcal{M}\}.$$

这种统一的异构图结构能够对系统执行行为和物理部署约束进行联合推理。

节点和边均通过源自监控指标的时间序列特征进行丰富。微服务节点同时纳入资源级指标（例如 CPU 利用率、内存使用量、磁盘 I/O）以及反映服务行为的应用级指标，而边缘设备节点仅由描述设备级资源消耗的资源级指标表示。边特征刻画通信行为（例如 p50/p90/p99 延迟百分位数），从而反映运行时依赖关系及潜在的性能瓶颈。

B. 图神经网络输入表示

从建模角度来看，图神经网络在三个基本组成部分上运行：节点特征、图连通性，以及（可选的）边特征。设节点特征张量为

定义为

$$X \in \mathbb{R}^{|\mathcal{V}| \times F \times T},$$

其中，每个节点由在 T 个时间步内观测到的 F 个特征维度表示。由于该图包含 $|\mathcal{V}|$ 个节点，因此生成的特征张量维度为 $(|\mathcal{V}|, F, T)$。

系统的结构连接通过 edgeindex 编码，

$$\mathcal{L} \subseteq \mathcal{V} \times \mathcal{V},$$

用于捕获节点之间的通信关系和部署关系。

链路级属性和通信统计信息通过边特征张量纳入其中，该张量定义为

$$E \in \mathbb{R}^{|\mathcal{L}| \times F_e \times T},$$

其中，每条边由在 T 个时间步内测得的 F_e 个特征维度表征。给定图中的 $|\mathcal{L}|$ 条边，张量 E 的维度为 $(|\mathcal{L}|, F_e, T)$。

这些组成部分共同构成图神经网络模型的输入，使模型能够针对拓扑图的结构和时间特征进行表示学习。

C. 用于联合 *RCL/RCA* 的集中式图神经网络背景

在考察用于 RCL 和故障类型识别研究的监督式基于图神经网络的方法时，可以观察到两种主要的建模范式。第一种范式构建单个静态图：首先从时间序列数据中提取紧凑表示，以学习时间依赖关系；随后使用图神经网络进行空间依赖学习 [12]–[14]。第二种范式构建图序列（通常每个时间戳或时间窗口对应一张图），针对每张图独立学习空间特征，然后使用时间聚合或序列模型对整个图序列中的时间依赖关系进行建模 [16], [18]。

在第一类方法中，通常使用长短期记忆网络、卷积神经网络或其他序列编码器等技术进行时间特征提取 [14], [18], [27]。然后将学习得到的时间嵌入用作图学习中的节点和/或边级特征。鉴于第一种范式在计算效率、模型可扩展性以及明确分离时间学习与结构学习方面的优势，我们采用了这一范式；这些优势对于大规模微服务系统尤为有益 [12], [14]。

在完成时间特征提取并通过图神经网络层进行消息传递后，应用任务专用预测头，以支持根因定位和故障类型识别（根因分析）。这两项任务在粒度和目标上有所不同：根因定位旨在找出导致异常的具体微服务，而根因分析则用于确定异常类型或故障类别。

在基于 GNN 的框架中，RCL 问题通常被建模为节点级排序任务 [12], [18], [27]。模型会为每个微服务节点生成一个分数，并将其转换为所有节点上的一系列概率值（例如，通过 Softmax 激活），从而将最有可能的故障服务排在首位。相比之下，故障类型识别通常被视为图级多分类任务，其中图级读出层

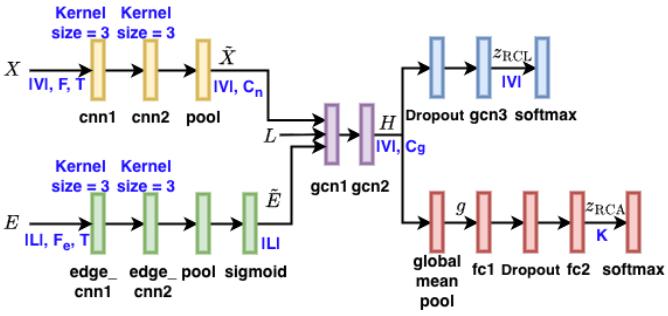

图 1：用于联合根因定位与根因分析的集中式图神经网络架构

会聚合学习到的节点表示（例如，使用均值/最大池化或基于注意力的池化），随后由分类头预测每种故障类型的概率 [12],[13]。

按照这一设计，采用两个输出头：用于根因定位的节点级预测头，以及用于根因分析的图级预测头；二者可以分别训练，也可以联合训练 [12], [27]。尽管一些研究将这些输出合并为单一预测任务 [13], [14], ，但我们采用双头形式。

D. 集中式联合根因定位/根因分析图神经网络架构概述如

图 1 所示，我们的基线图神经网络架构遵循 III-C 小节所述的第一种建模范式：先学习时间模式，再进行基于图的空间推理。该架构由三个连续阶段组成：（1）从节点和边提取时间特征；（2）使用基于图卷积网络的消息传递进行空间依赖学习；（3）设置两个任务专用预测头，分别用于根因定位和根因分析。前期各层（时间编码器和结构图编码器）由两项任务共享，使网络能够联合利用指标的时间演变和微服务依赖结构。

1）阶段 *1* - 时间特征提取

为从节点级和边级时间序列指标中获得紧凑表示，我们采用独立的基于一维卷积神经网络的时间编码器，这与前文讨论的特征处理方法一致。

(a) 节点特征编码器。输入节点张量 $X \in \mathbb{R}^{|\mathcal{V}| \times F \times T}$ 经过两个 CNN 层处理，随后沿时间维度进行池化：

$$X \xrightarrow{\text{CNN}_1} \xrightarrow{\text{CNN}_2} \xrightarrow{\text{Temporal Pool}} \tilde{X} \in \mathbb{R}^{|\mathcal{V}| \times C_n}, \quad (1)$$

其中，C_n 是时间池化后的输出嵌入维度。池化操作将时间信息聚合为定长表示，使下游图神经网络层能够处理经过时间压缩的嵌入，而非原始序列。

(b) 边特征编码器。同样，边属性张量 $E \in \mathbb{R}^{|\mathcal{L}| \times F_e \times T}$ 经过两个 CNN 层处理，随后沿时间维度进行池化：

$$E \xrightarrow{\text{CNN}_1} \xrightarrow{\text{CNN}_2} \xrightarrow{\text{Temporal Pool}} \tilde{E} \in \mathbb{R}^{|\mathcal{L}|}. \quad (2)$$

随后应用 Sigmoid函数，将边权重缩减为 $[0, 1]$，范围内的单一值，以便为后续消息传递阶段学习必要的信息。

这些嵌入构成图神经网络层的初始节点特征和边特征。

2）阶段 *2*——空间图学习
经过时间编码的嵌入通过两个堆叠的图卷积网络层，以学习拓扑图中的空间依赖关系：

$$(\tilde{X}, \mathcal{L}, \tilde{E}) \xrightarrow{\text{GCN}_1} \xrightarrow{\text{GCN}_2} H \in \mathbb{R}^{|\mathcal{V}| \times C_g}, \tag{3}$$

其中，C_g 是所得结构˜表征的隐藏维度。边嵌入 E 用作边权重，使图卷积网络在服务图中传播信息时能够纳入通信强度或延迟敏感性。

这些图卷积网络层充当结构编码器，供下游两个任务共享。这一共享表征至关重要，因为定位和故障类型推断都受益于对服务间影响模式及异常传播动态的理解。

3）阶段 *3*——任务专用预测头
完成空间编码后，网络分支为两个相互独立的输出头。

(a) RCL 头 – 节点级排序。该分支旨在识别发生故障的微服务，直接对节点嵌入进行操作：

$$H \xrightarrow{\text{Dropout}} \xrightarrow{\text{GCN}_3} z_{\text{RCL}} \in \mathbb{R}^{|\mathcal{V}|}, \tag{4}$$

z_{RCL} 是图中所有节点对应的 logit 值向量。Softmax 激活将 logit 值转换为概率，记作 $\hat{}p_{\text{RCL}}$。

(b) RCA 头 – 图级多分类。此处，通过全局均值池化聚合节点嵌入，以获得图级表示 g，随后将其输入两个带随机失活的全连接层，生成类别 logit：

$$H \xrightarrow{\text{GlobalMeanPool}} g \xrightarrow{\text{FC}_1} \xrightarrow{\text{Dropout}} \xrightarrow{\text{FC}_2} z_{\text{RCA}} \in \mathbb{R}^{K}, \tag{5}$$

其中，K 是异常/故障类型的数量。Softmax 激活将 logit 值转换为故障诊断所需的概率，记作 $\hat{}p_{\text{RCA}}$。

对于前文介绍的所有隐藏 GCN 和 CNN 层，我们均采用指数线性单元（ELU）作为激活函数。

4）损失函数构造
随后，使用联合损失函数同时优化两个预测头，

$$\mathcal{L}_{\text{total}} = \lambda_{\text{RCL}} \cdot \mathcal{L}_{\text{RCL}} + \lambda_{\text{RCA}} \cdot \mathcal{L}_{\text{RCA}}, \tag{6}$$

其中，$\mathcal{L}_{\text{RCL}}$ 和 $\mathcal{L}_{\text{RCA}}$ 分别表示用于根因定位和根因分析的节点级与图级负对数似然损失，λ_{RCL} 和 λ_{RCA} 为权重系数。

令$\hat{}y_{\text{RCL}}$ 表示指示真实根因节点的独热编码向量。单个图的根因定位损失定义为

$$\mathcal{L}_{\text{RCL}} = -\frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} y^i_{RCL} \log(p^i_{RCL}), \tag{7}$$

其中，y^i_{RCL} 是$\hat{}y_{\text{RCL}}$的 i^{th} 元素，p^i_{RCL} 表示节点 i 是根因的预测概率，并且 $|\mathcal{V}|$ 是图中的节点数。

类似地，对于根因分析任务，令$\hat{}y_{\text{RCA}}$ 表示在 K 种故障类别中指示故障类型的独热编码向量。单个图的根因分析损失定义为

$$\mathcal{L}_{\text{RCA}} = -\frac{1}{|\mathcal{K}|} \sum_{j=1}^{|\mathcal{K}|} y^j_{RCA} \log(p^j_{RCA}), \tag{8}$$

其中，y^j_{RCA} 是$\hat{}y_{\text{RCA}}$的第 j^{th} 个元素，p^j_{RCA} 表示故障类型 j 为根因的预测概率。

除非另有说明，我们在实现中设定 $\lambda_{\text{RCL}} = \lambda_{\text{RCA}} = 0.5$。

IV. 级联联合 RCL/RCA GNN 架构

尽管基准集中式图神经网络模型效果良好，但在大规模微服务部署中，其扩展效率并不理想。每个图卷积层的计算成本为 $\mathcal{O}(N^2)$，其中$N = |\mathcal{V}|$表示图 [5],[17]中的节点数。由于需要使用多个此类层进行全局消息传递，整体计算开销会进一步增大。随着微服务生态系统不断扩展，每一层都必须处理所有节点及其连接，导致诊断延迟迅速增加。在拥有数百或数千个分布式服务的边缘—云环境中，这种二次复杂度会成为关键瓶颈，使集中式推理的计算成本高昂，不适用于实时诊断。

为克服这一挑战，我们提出了一种面向联合根因定位与根因分析的级联图神经网络架构，专门针对大规模边缘部署进行了设计。我们的方法利用通信驱动聚类，根据微服务之间的交互强度将其划分为多个簇。这种聚类有两个目的：（i）缩小初始推理阶段的图规模；（ii）与现实世界中的异常传播模式相吻合，因为性能异常通常沿通信路径传播。因此，发生异常时，其影响更有可能局限于已识别簇的范围内，从而使我们的架构能够更高效地诊断根因，同时不影响准确率。

下文首先介绍用于将大型服务图划分为高交互社区的通信驱动聚类策略（第 IV-A 节）。随后介绍整体级联图神经网络架构，包括提议网络和输出网络的设计（第 IV-B 节），并进一步说明二者的联合优化。最后，分析所提框架的计算效率，并将其与集中式基线进行比较（第 IV-C 节）。

A. 通信驱动聚类

对于簇的形成，我们采用 Louvain 社区检测算法 [30], 一种非常适合大规模网络的模块度优化方法。算法 1 给出了我们的通信驱动的聚类算法。给定系统拓扑图 $\mathcal{G} = (\mathcal{V}, \mathcal{L})$，其中 $\mathcal{V} = \mathcal{M} \cup \mathcal{E}$ 由微服务和边缘设备组成，$\mathcal{L} = \mathcal{L}_{\text{comm}} \cup \mathcal{L}_{\text{dep}}$同时包含通信关系与部署关系，

算法 1 通信驱动的聚类算法

输入：系统图 $\mathcal{G} = (\mathcal{V}, \mathcal{L})$，部署映射 $\delta : \mathcal{M} \to \mathcal{E}$
输出：微服务的集群分配 $\phi(m)$ 和 $\psi(e)$ 边缘设备
构建通信子图 $\mathcal{G}_{\text{comm}} = (\mathcal{M}, \mathcal{L}_{\text{comm}})$
应用 Louvain 模块度优化，获得微服务社区 $\{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$
对每个社区 c_i **执行**
    对 每个微服务 m 在 c_i **执行**
        分配 $\phi(m) = c_i$
    结束循环
结束循环
/* 边缘设备集群分配 */
对每个边缘设备 $e \in \mathcal{E}$ **执行**
    令 $\mathcal{M}_e = \{m \in \mathcal{M} \mid \delta(m) = e\}$ {微服务已部署在 e} 上
    确定主导微服务集群
    $$c^* = \arg\max_{c_i} |\{m \in \mathcal{M}_e \mid \phi(m) = c_i\}|$$
    赋值 $\psi(e) = c^*$
结束 for
返回：对所有 $m \in \mathcal{M}$ 的 $\phi(m)$，对所有 $e \in \mathcal{E}$ 的 $\psi(e)$

我们导出用于聚类的仅通信子图。具体而言，我们构建

$$\mathcal{G}_{\text{comm}} = (\mathcal{M}, \mathcal{L}_{\text{comm}}),$$

即，我们提取一个仅包含微服务节点及其通信边的子图，因为异常传播主要沿运行时调用路径进行。对 $\mathcal{G}_{\text{comm}}$ 应用 Louvain 聚类，以获得节点到簇的分配：

$$\phi : \mathcal{M} \to \{c_1, c_2, \ldots c_{|\mathcal{C}|}\},$$

其中 $|\mathcal{C}|$ 是检测到的簇的数量。

为将聚类扩展到设备，我们将每个边缘设备分配到包含其已部署微服务中多数微服务的簇。形式上，对于设备 e，令 $\mathcal{M}_e = \{m \in \mathcal{M} \mid \delta(m) = e\}$，

我们选择主导社区

$$\psi(e) = \arg\max_{c_i} |\{\, m \in \mathcal{M}_e \mid \phi(m) = c_i \,\}| \,.$$

从而在微服务和边缘基础设施中形成一致的簇结构。这些由通信关系驱动的簇构成了我们级联 GNN 流水线的基础。

B. 级联联合 *RCL/RCA GNN* 架构概述

如图 2 所示，级联图神经网络由两个主要组件构成：提议网络（P-Net）和输出网络（O-Net）。我们将基线图神经网络拆分为两个网络：P-Net 在各个簇上运行并生成簇级输出；O-Net 将每个簇视为一个节点，在簇图上运行，并通过考虑簇间连接生成图级输出。此外，EdgeTemporalNet 模块由基线图神经网络架构中的边特征编码器组件构成。由于该模块由 P-Net 和 O-Net 共享，用于时序边特征嵌入，因此将其表示为一个独立组件。这种级联设计的主要动机是缩小各网络所处理的问题搜索空间，从而减轻计算负担并降低这些网络中图卷积网络层的计算负担。该策略旨在降低整体诊断延迟，同时保持较高的准确率。

1) 提议网络（*P-Net*）

为缩小搜索空间并提高计算效率，P-Net 在各个簇上运行。每个簇代表一个通信频繁的微服务组，如第 IV-A 小节所述。P-Net 的目标是学习簇内依赖关系并生成簇级输出。它会生成两种簇级输出。第一种是簇级嵌入，用于传递 RCA 所需的簇内信息。后续 O-Net 将以多个此类簇的嵌入作为输入。第二种输出是一个向量，其中包含与簇内节点对应的 logit 值。由于 RCL 被定义为节点级排序问题，且大多数异常传播局限于簇内（即，RCL 所需的全部信息通常都可在一个簇内获得），因此在簇级执行 RCL 已足够。因此，P-Net 负责执行 RCL。

P-Net 的输入源自完整图的属性 X、$\mathcal{L}$ 和 E，但仅限于各个簇。令 $\{c_1, c_2, \ldots c_{|\mathcal{C}|}\}$ 表示聚类阶段得到的簇集合。对于该集合中的每个簇 c_i，我们构建对应的子图，其中包含节点特征 X_{c_i}、边索引 $\mathcal{L}_{c_i}$ 和边特征 E_{c_i}，且只包含属于 c_i 的节点及簇内边。随后，P-Net 会分别处理这些特定于各簇的输入。如图 2 所示，P-Net 重用了基线图神经网络时间特征提取阶段的完整节点特征编码器。它还与 O-Net 共用基于 EdgeTemporalNet 的边特征编码器，如前文所述。空间图学习组件包含一个图卷积层。基线模型的 RCL 预测分支保留至图卷积层；之后对簇级输出进行聚合，并在所有节点（跨所有簇）上应用全局 Softmax，以生成最终的 RCL 分数。除了用于 RCL 的节点级输出外，P-Net 还会在其空间图学习阶段，对每个簇内的图卷积层输出应用全局平均池化，从而生成前文所述的簇级嵌入。

2）输出网络（*O-Net*）

O-Net 将每个簇视为一个节点，并将簇之间的连通关系视为边，从而生成图级输出。P-Net 的输出嵌入构成节点特征，而边特征则通过聚合簇间边的信息形成。由于根因分析被表述为一个图级分类问题，O-Net 负责为整个系统生成最终的根因分析预测。

对于 O-Net，节点特征 $X_{inter\ cluster}$ 是来自 P-Net 的嵌入，而边索引 $\mathcal{L}_{inter\ cluster}$ 则根据簇间边（即连接不同簇中节点的边）构建，此时索引映射到簇 ID，而非节点 ID。对应的边特征$E_{inter\ cluster}$ 首先由 EdgeTemporalNet 处理，以提取紧凑的时间序列表示。当两个簇之间存在多条边时，它们的编码特征

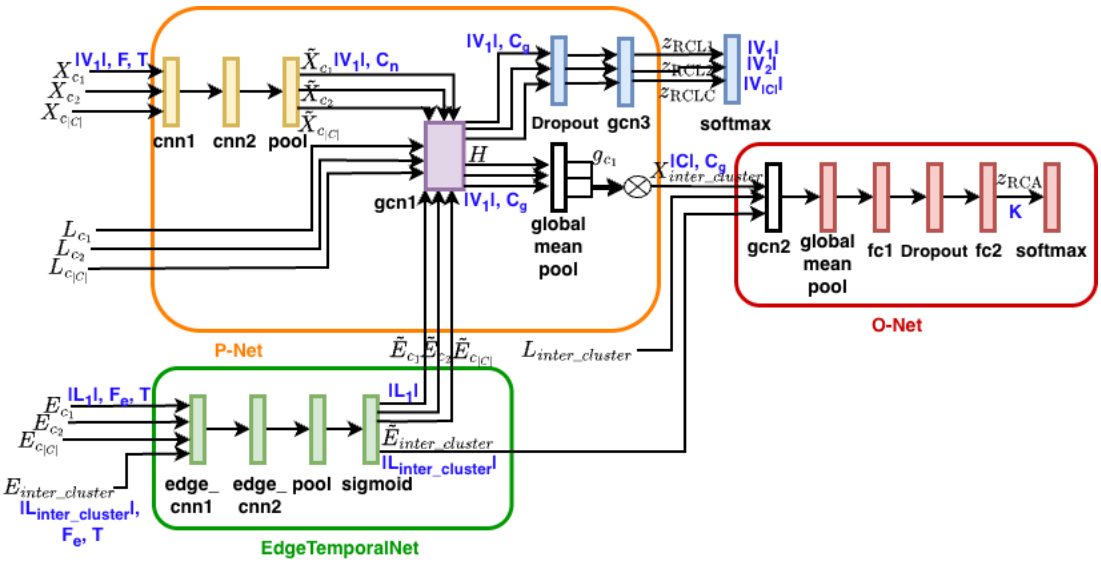

图 2：用于联合根因定位与根因分析的级联图神经网络架构

通过全局平均池化进行聚合，形成单个簇间边表示，然后传递给 O-Net。如图 2 所示，O-Net 由基线模型中通向 RCA 头的架构分支组成，并输出与故障类型相对应的类别概率。

3）联合任务优化

与基线架构类似，级联模型的根因定位和根因分析输出也使用第 III-D4 节中提到的相同多任务损失函数进行联合优化，从而使两个目标能够同时学习，并使共享表示惠及这两项任务。

C. 计算效率分析

为了评估级联设计在多大程度上实现了其目标，我们可以分析其相对于基线图神经网络的计算行为。尽管所提出的模型同样使用了总计三层图卷积层（P-Net 中两层，O-Net 中一层，与基线相同），但效率提升的关键在于每一层的运算位置。P-Net 并非在单个图中处理所有 N个节点，而是在每个簇内进行运算。将节点集划分为大小为 n_1、n_2、……，n_k的簇，使得 $N = \sum_{i=1}^{k} n_i$。对于消息传递复杂度为 $O(N^2)$ 的典型图卷积层，独立处理各个簇所得的总复杂度与 $N = \sum_{i=1}^{k} n_i$ [5], [17] 成正比。由于

$$N^2 = \left(\sum_{i=1}^{k} n_i\right)^2 = \sum_{i=1}^{k} n_i^2 + 2\sum_{i \neq j} n_i n_j.$$

以及所有 $n_i > 0$，由此可得：

$$N^2 > \sum_{i=1}^{k} n_i^2,$$

这意味着 P-Net 所执行的工作量严格低于一次性处理整个图。此外，各簇可以独立处理，因此能够并行化，进一步降低推理延迟。随后，O-Net 在包含 $|\mathcal{C}| \ll N$ 个节点的紧凑图上运行，从而进一步减少最后一个图卷积层的计算量。

尽管聚类会带来压缩，但准确性仍得以保留，因为大多数异常传播仍局限于簇内（因此 P-Net 运行时的上下文信息损失极小），而 O-Net 则通过聚合簇间边恢复跨簇信息。因此，级联架构能够实现预期目标：降低计算成本并加快诊断，同时保留准确性，下一节将对此进行验证。

V. 性能评估

本节从诊断效果和可扩展性两个方面评估所提出的级联联合 RCL/RCA GNN 架构。我们首先介绍 MicroCERCL 和 iAnomaly 的实验设置、数据集及实现细节（第 V-A 节）。随后，我们报告 MicroCERCL 上的准确率和延迟结果，并在中等规模设置下将级联模型与集中式基线进行比较（第 V-B 节）。接着，我们通过将图规模扩大至 10,000 个节点，并测量不同聚类配置下的端到端推理时间，分析 iAnomaly 上的可扩展性（第 V-C 节）。最后，我们开展消融实验，量化联合学习、聚类策略和图神经网络层类型对性能的影响（第 V-D 节）。

A. 实验设置与实现细节

我们使用公开的 MicroCERCL 数据集[1] [18]评估了所提出的级联图神经网络方法。该数据集是首个面向云边协同微服务系统的大规模基准数据集，至今仍是最全面的混合部署数据集。它包含从 81 个微服务中收集的数据，这些微服务属于四个应用：SockShop、Hipster、Bookinfo 和新引入的 AI-Edge。这些微服务部署在四台云服务器和两组各含两台边缘服务器的环境中，并遵循基于通信频率的应用放置策略。

[1]https://github.com/WDCloudEdge/MicroCERCL

为反映真实的生产环境，该数据集集成了多种异常。作者使用 ChaosMesh 注入应用级异常，例如容器中的 CPU 资源耗尽、内存泄漏和网络延迟。此外，他们还使用 Linux 内核流量控制（ TC ）模拟云与边缘节点之间的内核级网络故障，包括丢包、重复、损坏、乱序、延迟和抖动。这些注入使得能够在异构且复杂的故障条件下进行系统评估。

该数据集包含 682 个故障场景，在异常类型和强度方面具有较大的多样性。在实验中，我们采用了 60:20:20 的训练集、验证集和测试集划分。对于每个场景，我们提取分布式跟踪数据以构建服务依赖图，同时使用相应的系统指标和应用指标进行异常检测，并将其作为图神经网络模型的节点特征和边特征。

尽管 MicroCERCL 在部署真实性和异常多样性方面代表了大规模基准，但对于面向效率的评估而言，它可以被视为中等规模数据集。具体而言，其图规模不足以充分刻画在系统复杂性不断增加的情况下，RCL 和 RCA 模型的可扩展性表现。为了评估所提方法在更大输入规模下的计算效率，我们因此采用了一个基于 iAnomaly 框架 [19] 的数据集生成工具。该工具能够生成包含 50 至 10,000 个节点的服务依赖图，从而支持系统化的可扩展性分析。不同于依赖人工工作负载的纯合成模拟器，iAnomaly 生成器使用由 iAnomaly 仿真器产生的真实执行数据所派生的异常跟踪数据，为这些图填充内容。这确保了生成的数据集保留真实的时间特征和因果特征。

底层数据源自各种物联网应用，涵盖轻量级传感器处理流水线，以及基于摄像头的人脸检测与识别等计算密集型服务。这些应用在注入了多种异常的情况下运行，包括资源饱和、服务故障和网络拥塞。因此，生成的数据集为评估所提出的级联图神经网络框架的准确率和效率提供了一个真实且可扩展的测试平台。

我们使用 Scikit-learn 中基于 BIRCH 聚类的算法对这些数据集进行异常检测，并遵循 MicroCERCL 作者推荐的配置。具体而言，异常敏感度阈值 β 设为 0.07，以平衡异常检测准确率和降噪效果。

集中式和级联图神经网络模型均使用 Python 3.10 和 PyTorch Geometric 2.6.1[2] 实现。模型超参数采用基于树结构帕岑估计器（TPE）的贝叶斯优化方法进行优化[31]。所提出的通信驱动的聚类算法（算法 1）使用 Python 实现。所有实验，包括超参数调优，均在斯巴达 HPC 簇[3]上进行。模型训练使用 Adam 优化器 [32]。

B. MicroCERCL 评估结果

在将我们提出的级联图神经网络方法与集中式基线图神经网络模型进行比较之前，我们首先通过将集中式图神经网络基线与 MicroCERCL 论文中最初报告的结果进行基准测试，验证其准确率。值得注意的是，MicroCERCL 作者仅报告了两种应用（HH 和 SH）的根因定位性能，并采用 top-K 定位准确率进行衡量。其结果见表II。

表II：MicroCERCL论文中报告的根因定位准确率

应用	Acc@1	Acc@2	Acc@3	Acc@5	Acc@10
HH	0.632	0.756	0.796	0.833	0.896
SH	0.607	0.732	0.792	0.849	0.907

为验证我们的集中式基线具有竞争力，我们首先训练该模型仅执行根因定位。如表III所示，与MicroCERCL论文中报告的结果相比，我们的模型在前-K 项定位准确率上显著更高。这些结果证实，我们的集中式基线具有竞争力，适合与所提出的级联方法进行公平比较。

表III：我们的集中式图神经网络基线仅执行根因定位时的性能

	Acc@1	Acc@3	Acc@5	MAR	MRR
集中式（仅 RCL）	0.9203	0.9783	0.9928	1.1594	0.9534

我们还将集中式基线在 RCA 上作为一项独立任务进行评估，并将其表述为一个故障类型分类问题。结果如表 IV 所示。

表 IV：我们的集中式图神经网络基线仅执行 RCA 的性能

	准确率	精确率	召回率	F1分数
集中式（仅 RCA）	0.8478	0.8667	0.8478	0.8512

随后，我们使用共享表征和多任务目标训练基线模型，使其联合执行根因定位（RCL）和根因分析（RCA）。联合学习的结果汇总于表 V。与分别训练 RCL 和 RCA 模型相比，联合优化提升了定位性能和故障分类准确率。这表明，共享表示能够利用不同任务之间的互补信息，从而实现更有效的端到端根因诊断。

接下来，我们评估所提出的用于联合根因定位与根因分析的级联图神经网络架构，并将其与集中式联合模型进行比较。结果汇总于表 VI。总体而言，级联方法的定位和诊断性能与集中式基线相当。这表明，将全局图分解为通信驱动的簇并通过级联网络进行处理，不会降低诊断准确率。

为评估推理效率，我们测量了两种联合模型的端到端诊断时间。对于集中式联合图神经网络，平均和中位诊断时间分别为 17.779 ms

[2]https://pytorch-geometric.readthedocs.io/en/latest/

[3]https://dashboard.hpc.unimelb.edu.au/

表 V：我们的集中式基线的联合根因定位与根因分析性能

	RCL					RCA			
	Acc@1	Acc@3	Acc@5	MAR	MRR	准确率	精确率	召回率	F1分数
联合根因定位与根因分析	0.9275	0.9855	1.0000	1.1377	0.9550	0.8696	0.8877	0.8696	0.8715

表 VI：级联图神经网络的联合根因定位与根因分析性能

	RCL					RCA			
	Acc@1	Acc@3	Acc@5	MAR	MRR	准确率	精确率	召回率	F1分数
级联RCL与RCA	0.9130	0.9420	0.9493	1.9058	0.9316	0.8623	0.8703	0.8623	0.8640

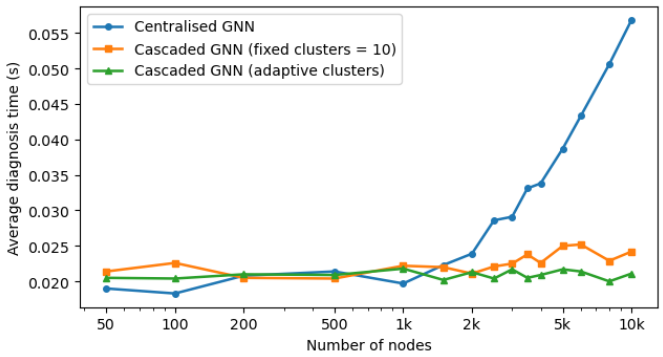

图3：iAnomaly数据集上的可扩展性比较

和 17.707 ms。相比之下，级联联合图神经网络的平均诊断时间为 20.237 ms，中位诊断时间为 20.273 ms。这些结果表明，在 MicroCERCL 数据集的规模（约 50 个节点）下，级联架构尚未表现出相较于集中式基线的明显延迟改进。

这一观察结果并不意外，因为在效率评估方面，MicroCERCL数据集属于中等规模数据集，无法充分考验集中式图处理的可扩展性极限。在这种情况下，聚类和多阶段推理带来的开销可能会抵消潜在的计算节省。因此，尽管级联图神经网络保持了相当的诊断准确率，但其效率优势在这一规模下未能充分体现。

为进一步考察可扩展性，我们在iAnomaly数据集上开展了更多实验。该数据集包含规模显著更大的服务图。这些实验旨在对集中式模型进行压力测试，并评估随着图规模增大，级联架构的表现。接下来我们讨论这些结果。

*C. iAnomaly*数据集上的可扩展性分析

使用iAnomaly数据集，我们将服务图规模从50个节点逐步增加到10,000个节点，以便系统地比较图规模不断增大时集中式联合图神经网络与级联联合图神经网络的表现。

我们测量了三种配置下的平均端到端诊断时间：(i) 集中式联合RCL/RCA模型；(ii) 簇数量固定为 10 的级联联合模型；(iii) 簇数量自适应的级联联合模型，其中簇数量根据图规模按比例选取（约每 20 个节点对应一个簇）。

图 3 总结了结果。随着节点数量增加，集中式模型的诊断时间明显上升，这表明随着图规模不断增大，全局消息传递会限制其可扩展性。簇数量固定的级联模型减缓了这一增长，但随着簇规模随图规模扩大，延迟仍略有增加。相比之下，根据图规模自适应控制簇数量时，级联图神经网络在所有评估规模下的诊断时间都基本保持恒定。

这些结果证实，所提出的级联图神经网络架构能够在实现尺度不变推理的同时保持诊断准确率，因此非常适用于边缘和云环境中的大规模微服务部署。

D. 消融实验

为分析所提出的级联图神经网络架构中各项设计选择的作用，我们围绕联合学习、聚类策略和图神经网络层类型开展了一系列消融实验。

表 VII 展示了消融实验的结果。第一行对应所提出的完整级联模型，该模型采用联合学习、通信驱动聚类和图卷积网络层。

a) 联合学习的影响。
移除联合学习并训练任务特定模型会导致性能持续下降。对于根因定位，Acc@1 从 0.9130 降至 0.8696。对于根因分析，影响更为显著：F1分数从 0.8640 大幅降至 0.6692。这一显著下降表明，定位和诊断任务之间的共享表示能够提升特征泛化能力和判别能力。

b) 聚类策略的影响。
用随机聚类替代通信驱动聚类会显著降低 RCL 性能。Acc@1 从 0.9130 降至 0.7971，MAR 从 1.9058 升至 2.6884，表明排序质量变差。尽管 RCA 的 F1分数降幅较小（从 0.8640 降至 0.8512），但两项任务的性能均有所下降，这证实了在图分解过程中保留通信依赖关系对于维持定位准确率至关重要。

c) 图神经网络层类型的影响。
将图卷积网络层替换为图注意力网络层后，结果喜忧参半。虽然 Acc@5 略有提升（从 0.9493 升至 0.9855），平均排名（MAR）有所改善（从 1.9058 降至 1.3333），但 Acc@1有所下降（从 0.9130 降至 0.8478），根因分析（RCA）的 F1分数也从 0.8640

表 VII：所提出的级联图神经网络架构的消融实验

类别	变体	RCL					RCA			
		Acc@1	Acc@3	Acc@5	MAR	MRR	Acc.	精确率	Rec.	F1
所提出的（完整模型）		**0.9130**	**0.9420**	**0.9493**	**1.9058**	**0.9316**	**0.8623**	**0.8703**	**0.8623**	**0.8640**
联合学习	仅根因定位	0.8696	0.9420	0.9638	1.8116	0.9123	–	–	–	–
	仅根因分析	–	–	–	–	–	0.6739	0.6760	0.6739	0.6692
聚类	随机	0.7971	0.8623	0.8841	2.6884	0.8386	0.8478	0.8667	0.8478	0.8512
图神经网络层	GAT	0.8478	0.9493	0.9855	1.3333	0.9067	0.8333	0.8422	0.8333	0.8346

降至 0.8346。鉴于图注意力网络会带来额外的注意力计算开销，这些结果表明，增加复杂度并未带来稳定的性能提升。总体而言，在级联架构中，图卷积网络层在效果与计算效率之间实现了更好的平衡。

VI. 结论与未来工作

本文研究了大规模边缘—云微服务环境中的根因定位与故障类型识别联合问题。尽管近期基于图神经网络的方法展现出较强的诊断准确率，但它们依赖于在全系统图上进行集中式处理，导致分布式边缘环境中的推理延迟较高且可扩展性有限。为应对这一挑战，我们提出了一种级联图神经网络架构，通过通信驱动聚类和层次化图推理来分解诊断任务。

我们的框架由一个提议网络和一个输出网络组成。提议网络在各个通信感知簇上运行，以生成簇级输出；随后，输出网络将每个簇视为一个节点，在簇图上运行，并通过考虑簇间连接生成图级输出。通过将消息传递限制在规模较小且结构连贯的子图中，并在紧凑的簇级图上运行，所提出的方法显著缩小了有效搜索空间，同时保留了关键的依赖关系信息。

在 MicroCERCL 基准以及使用 iAnomaly 生成的大规模数据集上进行的广泛评估表明，与集中式图神经网络模型相比，级联架构在实现相当诊断准确率的同时，显著提升了可扩展性。具体而言，在大型服务图上的实验表明，随着系统规模扩大，所提出的方法能够保持推理延迟近乎恒定；相比之下，集中式方法的延迟则迅速增长。

未来工作中，我们计划研究所提出的级联框架的分布式部署策略。尽管当前设计是在集中位置执行推理，但将模型组件直接部署在边缘设备上，可通过尽量减少数据传输开销进一步降低诊断延迟。为支持这种去中心化运行，我们将探索联邦学习技术，以协调分布式节点上的提议网络和输出网络。此外，我们还计划扩展当前的通信驱动聚类机制，同时考虑通信和共置依赖，从而实现更具部署感知能力的大规模边缘环境中的分区。

参考文献

[1] F. Al-Doghman、N. Moustafa、I. Khalil、N. Sohrabi、Z. Tari 和 A. Y. Zomaya，“边缘计算中的人工智能赋能安全微服务：机遇与挑战”，*IEEE*服务计算汇刊，第16卷，第1485–1504页，2023年。[2] C. Wu、Q. Peng、Y. Xia、Y. Jin 和 Z. Hu，“面向边缘计算环境中经济高效且稳健的人工智能微服务部署”，未来一代计算机系统，第141卷，第129–142页，2023年。[3] S. Becker、F. Schmidt、A. Gulenko、A. Acker 和 O. Kao，“面向边缘计算环境中的智能运维”，载于*2020*年*IEEE*国际大数据会议论文集（*Big Data*）。美国佐治亚州亚特兰大：IEEE，2020年，第3470–3475页。[4] M. Soualhia、C. Fu 和 F. Khomh，“边缘云环境中的基础设施故障检测与预测”，载于第*4*届*ACM/IEEE*边缘计算研讨会论文集，系列：SEC ’19。弗吉尼亚州阿灵顿：计算机协会，2019年，第222–235页。[5] D. Scheinert、A. Acker、L. Thamsen、M. K. Geldenhuys 和 O. Kao，“学习分布式云应用中的依赖关系以识别和定位异常”，载于*2021*年*IEEE/ACM*云智能国际研讨会论文集（*CloudIntelligence*），2021年，第7–12页。[6] W. Tian、H. Zhang、N. Yang 和 Y. Zhang，“具有保护机制的微服务系统中的基于图的根因定位”，国际软件工程与知识工程杂志，第33卷，第08期，第1211–1238页，2023年。[7] T. Wang 和 G. Qi，“（微）服务根因分析综述：方法、挑战与趋势”，2024年。

[8] [ ] 2408 00803 在线。可访问：https://arxiv.org/abs/。R. Xin、P. Chen 和 Z. Zhao，“CausalRCA：基于因果推断的微服务应用精准细粒度根因定位”，《系统与软件杂志》，第203卷，C辑，2023年9月。[9] L. Wu、J. Tordsson、J. Bogatinovski、E. Elmroth 和 O. Kao，“Microdiag：微服务系统的细粒度性能诊断”，载于*2021*年*IEEE/ACM*云智能国际研讨会论文集（*CloudIntelligence*），2021年，第31–36页。[10] Z. Zhang、B. Li、J. Wang 和 Y. Liu，“AAMR：云原生环境中的自动化异常微服务排序”，载于*SEKE’21*论文集，2021年，第86–91页。[11] Z. Zhu、C. Lee、X. Tang 和 P. He，“HEMiRCA：利用异构数据源对微服务进行细粒度根因分析”，*ACM*软件工程与方法学汇刊，第33卷，第8期，2024年11月。[12] S. Zhang、P. Jin、Z. Lin、Y. Sun、B. Zhang、S. Xia、Z. Li、Z. Zhong、M. Ma、W. Jin、D. Zhang、Z. Zhu 和 D. Pei，“通过多模态数据实现微服务系统的稳健故障诊断”，*IEEE*服务计算汇刊，第16卷，第6期，第3851–3864页，2023年。[13] Y. Zhu、J. Wang、B. Li、Y. Zhao、Z. Zhang、Y. Xiong 和 S. Chen，“MicroRCA：微服务系统的实例级根因定位”，《系统与软件杂志》，第216卷，C辑，2024年10月。[14] Z. Li、N. Zhao、M. Li、X. Lu、L. Wang、D. Chang、X. Nie、L. Cao、W. Zhang、K. Sui、Y. Wang、X. Du、G. Duan 和 D. Pei，“面向在线服务系统重复发生故障的可操作且可解释的故障定位”，载于第*30*届*ACM*欧洲软件工程会议与软件工程基础研讨会联合会议论文集，系列：ESEC/FSE 2022。美国纽约：计算机协会，2022年，第996–1008页。[15] G. Yu、P. Chen、Y. Li、H. Chen、X. Li 和 Z. Zheng，“Nezha：基于多模态可观测性数据的微服务可解释细粒度根因分析”，载于第*31*届*ACM*欧洲软件工程会议与软件工程基础研讨会联合会议论文集

软件工程基础，系列：ESEC/FSE 2023，美国加利福尼亚州旧金山，2023年，第553–565页。[16] R. Ren、Y. Wang、F. Liu、Z. Li、G. Tyson、T. Miao 和 G. Xie，“Grace：基于图卷积网络的微服务可解释根因分析”，载于2023年IEEE/ACM第31届国际服务质量研讨会论文集（IWQoS），2023年，第1–4页。[17] N. Fu、G. Cheng、Y. Teng、G. Dai、S. Yu 和 Z. Chen，“微服务系统中的智能根因定位：综述与新视角”，ACM计算调查，第57卷，第12期，2025年7月。[18] Y. Zhu、J. Wang、B. Li、X. Tang、H. Li、N. Zhang 和 Y. Zhao，“云边协同环境中的微服务系统根因定位”，2024年。 [在线]。可访问：https://arxiv.org/abs/ 2406 13604 。[19] D. Fernando、M. A. Rodriguez 和 R. Buyya，“iAnomaly：用于生成边云一体化计算环境性能异常数据集的工具包”，载于第17届IEEE/ACM实用计算与云计算国际会议论文集，系列：UCC’24。阿联酋沙迦：计算机协会，2024年。[20] L. Wu、J. Tordsson、E. Elmroth 和 O. Kao，“MicroRCA：微服务性能问题的根因定位”，载于NOMS 2020——2020年IEEE/IFIP网络运营与管理研讨会论文集，2020年，第1–9页。[21] R. Chen、J. Ren、L. Wang、Y. Pu、K. Yang 和 W. Wu，“MicroEGRCL：一种用于微服务系统根因定位的基于边注意力的图神经网络方法”，载于《面向服务的计算：第20届国际会议（ICSOC 2022），西班牙塞维利亚，2022年11月29日至12月2日，论文集》。柏林、海德堡：施普林格出版社，2022年，第264–272页。[22] L. Wu、J. Bogatinovski、S. Nedelkoski、J. Tordsson 和 O. Kao，“使用深度学习诊断云微服务中的性能问题”，载于《面向服务的计算——ICSOC 2020研讨会：智能运维、CFTIC、STRAPS、AI-PA、AI-IOTS及卫星活动，阿联酋迪拜，2020年12月14日至17日，论文集》。柏林、海德堡：施普林格出版社，2020年，第85–96页。[23] C. Hou、T. Jia、Y. Wu、Y. Li 和 J. Han，“利用异构数据源诊断微服务中的性能问题”，载于2021年IEEE并行与分布式处理及应用、 大数据与云计算、可持续计算与通信、社会计算与网络国际会议论文集（ISPA/BDCloud/SocialCom/SustainCom），2021年，第493–500页。[24] ´Alvaro Brand´on、M. Sol´e、A. Hu´elamo、D. Solans、M. S. P´erez 和 V. Munt´es-Mulero，“面向服务架构与微服务架构的基于图的根因分析”，《系统与软件杂志》，第159卷，第110432页，2020年。[25] D. Liu、C. He、X. Peng、F. Lin、C. Zhang、S. Gong、Z. Li、J. Ou 和 Z. Wu，“MicroHECL：大型微服务系统中的高效根因定位”，载于第43届国际软件工程会议：软件工程实践论文集，系列：ICSE-SEIP ’21。IEEE出版社，2021年，第338–347页。[26] J. Soldani 和 A. Brogi，“基于（微）服务的云应用中的异常检测与故障根因分析：综述”，ACM计算调查，第55卷，第3期，2022年。[27] C. Lee、T. Yang、Z. Chen、Y. Su 和 M. R. Lyu，“Eadro：面向多源数据微服务的端到端故障排查框架”，载于第45届国际软件工程会议论文集，系列：ICSE ’23。IEEE出版社，2023年，第1750–1762页。[28] G. Yu、P. Chen、H. Chen、Z. Guan、Z. Huang、L. Jing、T. Weng、X. Sun 和 X. Li，“Microrank：利用扩展频谱分析定位微服务环境中的端到端延迟问题”，载于2021年万维网大会论文集，系列：WWW ’21。计算机协会，2021年，第3087–3098页。[29] O. Kalinagac、W. Soussi、Y. Anser、C. Gaber 和 G. G¨ur，“边缘物联网服务微服务架构中的根因与责任分析”，载于ICC 2023——IEEE国际通信会议论文集，2023年，第3277–3283页。[30] V. D. Blondel、J.-L. Guillaume、R. Lambiotte 和 E. Lefebvre，“大型网络中的社区快速展开算法”，《统计力学杂志：理论与实验》，第2008卷，第10期，第P10008页，2008年10月。[31] J. Bergstra、R. Bardenet、Y. Bengio 和 B. K´egl，“超参数优化算法”，载于第25届神经信息处理系统进展年会论文集，西班牙格拉纳达，2011年。[32] D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，2017年。 [在线]。可访问：https://arxiv.org/abs/1412.6980