几何深度学习在模拟电路迭代分类与筛选中的应用

AnthonySirico Jr.*，**Daniel R. Herber**系统工程系科
罗拉多州立大学，科罗拉多州柯林斯堡 80523 邮箱:
{anthony.sirico，daniel.herber}@colostate.edu

arXiv:2303.09770v4 [cs.CE] 3 Aug 2023

摘要

许多复杂的工程系统都可以用拓扑形式表示，例如图。本文采用一种名为几何深度学习（*GDL*）的机器学习技术，帮助设计人员应对具有挑战性的、以图为中心的设计问题。本文提出的策略是利用图数据，通过几何深度学习有效且高效地寻找性能最佳且可实现的解决方案，同时降低计算成本。本案例研究关注模拟电路的综合设计，旨在使电路在特定频率范围内匹配指定的频率响应

f

。先前研究采用枚举技术生成了 *43,249* 个独特的无向图，这些图对应有效的潜在电路。然而，确定许多电路的尺寸参数和性能可能成本过高。为在准确率方面实现量化的权衡并降低计算成本，研究将部分电路图及其性能作为数据输入面向分类的几何深度学习模型。随后，可利用该几何深度学习模型低成本地预测其余图，从而帮助决策者搜索最佳图解。本文结果表明，增加基于图的特征是有益的；仅使用 *10%* 的图，即可实现 *80%* 的有利总体分类准确率；迭代构建的几何深度学习模型还可将图进一步划分为目标群组，其群组中位数显著接近最佳值，并且平均而言，使用 *25 %* 的图即可包含表现最佳的前 *100* 个图中的 *88.2* 个。

关键词：机器学习、几何深度学习、图分类、基于图的设计、电路综合

1 引言

数学图能够捕捉离散的组成信息和关系信息，因此可用于表示许多系统和决策。数十年来，研究一直采用不同的图表示来描述各自的问题 [1–8], ；一个多世纪以来，研究人员也一直利用图枚举来理解工程设计问题，以辅助决策 [2, 3, 9–11]。如今，许多工程设计问题（包括“系统建筑学”的构建）在范围和复杂性上不断增加，传统的离散和连续表示已不足以描述该系统 [2]。

系统架构是一种概念模型，用于描述产品、流程或元素的结构、行为、规则等 [12] ，通常也是其设计、构建和运行的基础。许多研究致力于运用图论有效表示系统架构 [13–15], ，其目标通常是得到一组满足约束且可行的实用架构。此外，还可以为备选架构确定一个或多个价值指标（例如性能或成本），以便设计者对候选方案进行筛选。生成所有这些选项的一种方法是图枚举，即针对某种指定结构，完整且有序地列出所有潜在的图 [9, 16, 17]。然而，视具体问题和所选表示方式而定，这种方法可能产生数量庞大的潜在解决方案。再加上现代系统日益复杂，计算成本随之呈指数级增长，使设计者或系统架构师的决策流程愈发困难。这些挑战促使人们需要一种能够辅助解决规模更大、复杂度更高的以图为中心的设计问题的方法。

本文中，我们探讨深度学习，特别是几何深度学习（图深度学习），作为解决这些问题的一种潜在策略。深度学习是指由多个处理层组成、能够学习的机器学习模型

*通讯作者，anthony.sirico@colostate.ed u

能够学习具有多个抽象层次的数据表示 [18], ； “深度”则指神经网络中隐藏层数量较多。如今，图深度学习是一个统称，涵盖了一种新兴技术，可将神经网络推广到欧几里得域和非欧几里得域，例如图、流形、网格或字符串表示 [19], ，并使用图神经网络。本质上，图深度学习涵盖了将输入变量的结构空间及对称性纳入考虑，并利用这些信息提升模型所捕获数据质量的方法。图深度学习潜力巨大，已广泛应用于其他科学领域，包括分子表示 [20–22], 材料科学 [23],建筑学 [24], 和医学领域 [25, 26]。在工程设计领域，图神经网络已应用于翼型设计 [27], 基于网格的物理仿真的结构力学 [28], 风电场功率估算 [29], 分布式电路设计 [30], 以及共享出行系统中的决策流程 [31]。此外，通过基于装配关系和流动关系进行训练，图神经网络还助力实现了部件功能分类 [32]。

本文所述的大多数工程问题都没有庞大的数据集。或许正是由于缺少相关的、以图为基础的设计数据集，尽管已有相关工具，且这些工具在其他领域中已取得有据可查的成功，但图深度学习或其他类似的基于图的机器学习方法在工程设计筛选流程中的应用仍然有限 [33]。许多研究利用机器学习在三维空间中表示和选择对象，通常应用于计算机辅助设计（CAD） [34, 35]。然而，网格与其他图表示之间存在一些关键差异，包括大小、结构、局部性以及几何解释（或缺乏几何解释）。本研究旨在提供另一种图设计示例和数据集 [36] ，以展示图深度学习如何助力工程设计流程。

本文尤其关注以图为中心的设计问题：生成大量候选图是可行的，但确定每种方案的价值或性能却成本过高。先前研究在本文所用的特定数据集上取得了可喜成果，但需要规模庞大的训练集和较长时间 [37]。本文提出的方法利用图深度学习，以较小的训练集为代价，在分类准确率上作出权衡，从而降低对可能“优”或“劣”方案进行分类的计算成本。

本文其余部分安排如下：第2 节介绍图设计和图深度学习的必要背景。第3节介绍所提出的方法，包括图分类、机器学习模型，以及用于确定模型性能的指标。第4节讨论电路频率响应匹配图设计的案例研究，第5 节介绍为探究图深度学习在该问题上的应用而开展的若干实验。最后，我们在第6 节总结全文，并讨论未来工作。

2 背景

为更好地理解所提方法背后的动机与技术，本节将重点介绍图论、通过图进行系统表示、图深度学习和图神经网络的其他关键方面。

2.1 图论

图G 是由两个集合组成的有序对（V，E），其中$E \subseteq [V]^2$（即，E 是V的一个二元素子集）。E 表示图的边，而V 是其顶点或节点的集合 [38]。图具有各种属性。例如，图的阶（记作$n = |G|$）是图中顶点的数量。每个顶点都对应一个度值，即与该顶点相邻的邻居或边的数量，记作$d_G(v) = d(v)$。

虽然在数学上表示图的方法有多种，但这里我们重点介绍一种称为邻接矩阵$\mathbf{A} = (a_{i,j})_{n\times n}$ 的方法，其定义如下：

$$a_{i,j} := \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

对于简单无向图，矩阵是对称的，$\mathbf{A}$ 的对角线元素全为零，这表明该图没有自环。

另一个重要概念是图同构。假设我们有两个图 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$。如果存在一个从 $V_1 \to$ 到 V_2 的双射 φ，使得对于所有 v_i、$v_j \in \in V$ [39]，$(v_i, v\,j) \in E_1 \leftrightarrow$ 当且仅当 $(\varphi(v_i), \varphi(v_j)) \in E_2$，那么这两个图就是同构的，记作 $G1 \cong G_2$, 。我们还会考虑特征图同构（或标记图同构）。这里，我们考虑前一定义中的相同两个图，但它们包含第三种特征。具体而言，我们将此特征视为顶点标签，记作 X，其中 $G_1 = (V_1, E_1, X_1)$ 和 $G_2 = (V_2, E_2, X_2)$。只要在某个有效双射 φ 下保留顶点标签属性，这些图就同构。更一般地，特征矩阵 $X \in \mathbb{R}^{n\times c}$ 可以包含任意数量的列，以表示与每个顶点关联的 c 个特征。

2.1.1 将电路表示为图。上一节中的概念可用于表示工程系统。本案例研究采用的特定系统类型是电路*RLC* ，因此将用它们来说明如何将其表示为图。观察图1左侧，可以看到一个基本的*RLC* 电路原理图。右侧则以图形方式将同一电路表示为图。图中的每个顶点现在都标记为相应的元件，其中*I* 和*O*分别表示输入和输出节点，*R* 表示电阻器，*L* 表示电感器，*C* 表示电容器，最后，*N* 表示电压恒定的电压节点。此外，同一电路还可以有其他图表示。

对于图1，表示该图的邻接矩阵**A** 表示

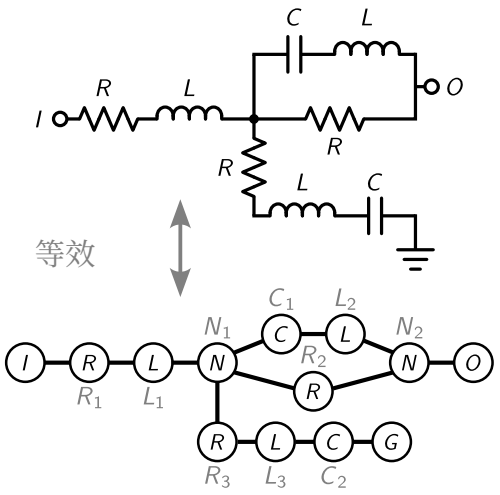

图1：以顶点标记图表示的电路原理图。

其结构及表示顶点标签的顶点列表**V** 如下：

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} I \\ R \\ L \\ N \\ C \\ R \\ L \\ N \\ O \\ R \\ L \\ C \\ G \end{bmatrix} \tag{2}$$

其中，1 表示两个顶点之间存在连接。

2.2 图枚举与优化

设计人员和系统架构师在探索以图为中心的问题时，仍常常依赖工程直觉和试错法。然而，在时间限制下，这些方法可能导致无法找到新颖且令人满意的解决方案。

一种潜在的系统性策略是图枚举，即生成（有时也只是统计）具有特定属性的非同构图的技术 [40]。这些属性可能多种多样，有时称为网络结构约束 [17]。例如，限制图中顶点可能具有的度，或限制与图相关的最大成本 [10]。需要注意的是，有效图的数量通常呈组合增长；因此，仅生成枚举结果的计算成本就可能非常高。不过，两种不同的图枚举算法可能都能生成所需的枚举结果，但其中一种效率可能更高，而另一种可能会更高效地完成这一任务 [10]。

无论如何，在若干重要的工程应用中，对图进行枚举（通常包含数千至数百万个条目）是可行的。即使无法枚举图，生成一个部分列表（通过算法或手动生成）仍可能有助于探索潜在解决方案。本文假设已有一组生成的图，记为 $\mathcal{G}$，其中包含感兴趣的图。对于本案例研究，该集合是基于 [1]，对具有特定属性的*RLC* 电路图进行的枚举。

尽管拥有一组新颖的图本身就很有用，但在许多工程领域中，图仅仅是一种中间表示，用于计算一个或多个感兴趣的价值指标。例如，图可能对应于一个基于物理原理的系统模型，用于确定汽车悬架的舒适性和操控性 [41]。在本研究中，我们将考虑一个性能指标，记作 $J(G_i)$，它是图的函数；自然的目标是寻找能够使这一“性能”指标最小化的图：

$$\underset{G_i}{\text{minimize:}} \quad J(G_i) \tag{3}$$

图工程设计中的一个挑战是，有时计算 $J(G_i)$ 的成本相当高，甚至可能比生成图G_i 本身高出许多个数量级。成本来源可能多种多样，包括高保真仿真、优化、以人为中心的评估和物理实验。本研究关注的是成本高到难以承受的情况。

基于 [42], 提出的分类，我们定义以下三种以图为中心的问题：

- 类型 *0* — 所有期望的图均可生成，其性能指标$J(G_i)$也可在T时间内计算
- 类型 *1* — 所有期望的图均可生成，但只有部分性能指标$J(G_i)$可在T时间内评估；性能评估的代价过高
- 类型 *2* — 无法在T时间内生成所有期望的图

其中，T 是完成图设计研究所分配的时间。本研究聚焦于类型 1 问题的方法（使用来自大型类型 0 研究的数据）。

2.3 几何深度学习

深度学习模型在图像 [43], 文本 [44], 和语音 [45], 上进行训练时表现十分出色，但这些数据都具有底层欧几里得结构。图与大多数深度学习应用（语言、图像和视频）中使用的常见欧几里得数据有着根本区别。欧几里得数据具有底层网格状结构。例如，图像可以映射到（x，y，z）笛卡尔坐标系中，其中每个像素都位于一个（x，y）坐标上，而 z 表示颜色。但是，这种方法并不适用于所有问题，因为某些操作需要许多维度。这种“扁平化”的

表示方式存在局限，例如无法表示层级结构和其他直接关系。

这促使人们研究用于图表示的双曲空间，或非欧几里得学习 [46,47],，它们有望更灵活地执行相同的操作；同时，图深度学习方法能够利用此类数据丰富的结构，学习更好的表示并做出更准确的预测 [19]。例如，许多现实世界数据集天然以图的形式组织，其中数据点之间的关系比单个数据点更为关键。基于欧几里得的方法难以处理此类数据，因为它们是为处理扁平、无结构的数据而设计的。另一方面，图深度学习方法可以直接利用数据的结构，从而取得更好的效果。

自诞生以来，图深度学习便可应用于所有以几何先验表示的数据形式 [48]。几何先验编码了有关数据几何的信息，例如平滑性、数据的稀疏性，或数据与其他变量之间的关系，使我们能够处理维度更高的数据。对称性在图深度学习中至关重要，通常以不变性和等变性来描述。不变性是某些数学对象的一种性质，指它们在特定变换下保持不变；等变性则是某些关系的一种性质，指它们在特定变换下彼此保持相同的相对位置 [49,50]。例如，考虑第2.1节中的图同构性质。几何深度学习的另一项优势是能够灵活适应更广泛的数据类型和问题。这种灵活性使其成为解决基于欧几里得的方法无法处理的现实世界问题的有力工具。

2.4 图神经网络

与采用卷积神经网络（CNNs）的传统深度学习模型类似，图深度学习利用图神经网络（GNNs）来学习图结构数据的表示，例如社交网络、分子和计算机程序。图神经网络与卷积神经网络非常相似，能够提取局部特征并将其组合起来构建表示 [51]。如前所述，两者的关键区别在于，图神经网络能够处理非欧几里得数据，以及图像或文本这类结构不均匀的数据。最后，图神经网络还能从未标记的数据中学习，这称为无监督学习。
这很重要，因为许多现实世界的数据集都没有标注。

训练用于图分类（即判断图是否具有某种特征）的图神经网络相对简单，通常需要经过三个步骤：1）通过多轮消息传递为每个节点生成嵌入；2）将节点嵌入聚合为统一的图嵌入；3）在图嵌入上训练最终的分类器。我们将在第3.3节中进一步探讨这一过程，并讨论此处所用图神经网络的各层。

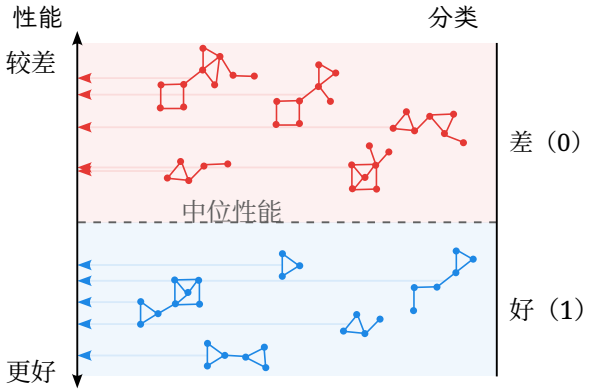

图2：基于性能值的图分类示意图。

3 方法

将图深度学习应用于基于图的设计问题时，将采用图分类方法。本节将讨论这一概念，以及所选模型、超参数和用于评估模型性能的指标。

3.1 图分类

图分类是指根据选定的标签对图进行分类。在此，我们考虑监督式图分类，即我们拥有一组图$\mathcal{G}$，每个图都有其标签$J(G_i)$，并使用额外的图属性（例如结构、嵌入和节点数据）来训练模型。这些特征有助于模型区分不同的图，从而提高标签预测的准确性。图分类示意图见图2。在此，中位性能值可用作判定阈值，将图划分为二元类别：观测到的性能值中位数以下的“优质”图，以及中位数以上的“较差”图。

与回归相比，采用这种方法的理由是，它能更好地体现设计者在概念设计早期阶段的意图（即灵活筛选）。通常，目标并非将潜在的图缩小到某一个特定的图，而是选出一组“优质”或有前景的图，以便进一步分析（由于图设计期间在建模及其他方面所作的假设，通常会采用更高的保真度进行分析）。此外，案例研究中的性能值$J(G_i)$存在显著差异，且许多值较为粗略。因此，重点在于展示图属于特定组的“归属程度”，而不是为图指定具体数值。

图分类有多种方法，包括深度图卷积神经网络（DGCNN）[52],对图结构和节点特征进行编码的隐藏层表示[53], EigenPooling [54], 以及可微池化[55]。这些方法已用于许多不同的研究，例如学习分子指纹 [56], 文本分类[57], 加密

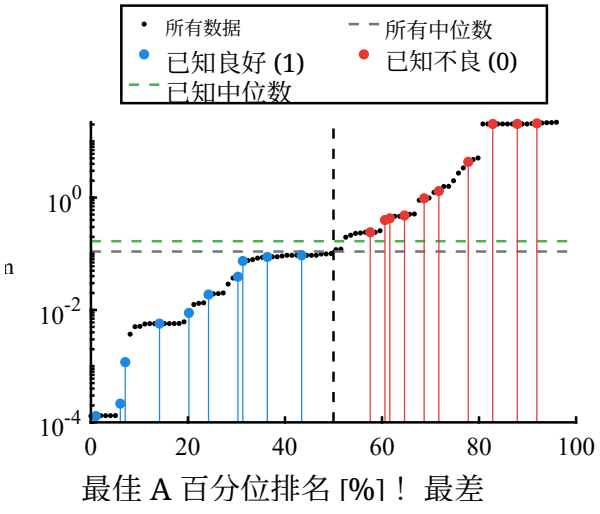

图3：展示 100 个图的 $\mathcal{G}_{all}$ 和 $\mathcal{G}_{known}$ 、 $\mathcal{G}_{known}$的基于性能的分类，以及中位数之间可能存在的差异。

交通分析 [58], 与癌症研究 [59]。

3.2 数据集

基于此处第 2.2 节所讨论的第1类问题分类，我们将考虑仅已知部分性能值 $J(G_i)$ 的情况，其中 $G_i \subset \mathcal{G}$ 。这将把图分为以下两个集合：

$$\mathcal{G} \equiv \mathcal{G}_{all} = \mathcal{G}_{known} \cup \mathcal{G}_{unknown} \tag{4}$$

其中， $\mathcal{G}_{known}$ 是已知$J(G_i)$值的图集合，而$\mathcal{G}_{unknown}$ 表示$J(G_i)$值未知的图（这正是 GDL 模型的用途）。我们将这些集合的大小分别记为 $|\mathcal{G}_{all}| = N_{all}$、 $|\mathcal{G}_{known}| = N_{known}$ 和 $|\mathcal{G}_{unknown}| = N_{unknown}$。 它们还满足：

$$N_{all} = N_{known} + N_{unknown} \tag{5}$$

我们还要注意，这与机器学习中使用的其他类型数据集略有不同，因为潜在输入的数量已知且有限。这里的目标是开发准确的模型，其中$N_{known} \ll N_{unknown}$。

与机器学习中的常见做法一样，我们将从 $\mathcal{G}_{known}$ 中创建两个子集：

$$\mathcal{G}_{known} = \mathcal{G}_{training} \cup \mathcal{G}_{validation} \tag{6}$$

其中， $\mathcal{G}_{training}$ 是训练数据集， $\mathcal{G}_{validation}$ 是验证集。模型使用 $\mathcal{G}_{training}$进行拟合，并用拟合后的模型预测 $\mathcal{G}_{validation}$ [60]中观测值的响应。例如，每次迭代后，模型都会相应地调整权重，并在验证集上进行测试，这有助于了解模型性能，并在必要时调整选项。模型定稿后，会使用未知集测试模型在其未曾见过的数据上的表现。

如图2所示，$\mathcal{G}_{known}$ 中图的二元分类基于性能值中位数（也可能基于使用已知数据确定的其他分界线）。由于我们只知道$J(G_i)$ 在 $\mathcal{G}_{known}$ 中，而不是在 $\mathcal{G}_{all}$中，因此用于初始标记的性能值中位数可能与拥有 $\mathcal{G}_{all}$的所有性能值时的真实中位数不同。然而，随着 $\mathcal{G}_{known}$ 的相对规模增大，这一误差会逐渐减小。本节中的概念如图3所示。这里， $\mathcal{G}_{all}$ 中有100个图，并为$\mathcal{G}_{known}$随机选取了20个图。真实中位数与基于 $\mathcal{G}_{known}$的中位数有所不同，两条线之间有两个图。

3.3 模型

本研究所用的模型包含五层：三个图卷积层、 [61], 一个均值池化层，以及用于最终读出的线性层。为实现图分类目标，模型通过以下步骤进行处理：1）通过消息传递，利用卷积层嵌入每个节点；2）聚合节点嵌入，生成图嵌入；3）使用图嵌入训练分类层。

3.3.1 图卷积层。 图卷积层（GCN）层通过以下方式确定输出特征$\mathbf{X}'$：

$$\mathbf{X}'_i = \mathbf{W}_1 \mathbf{X}_i + \mathbf{W}_2 \sum_{j \in \mathrm{N}(i)} e_{j,i} \cdot \mathbf{X}_j \tag{7}$$

其中， $\mathbf{X}$ 表示每个节点的输入特征（如第2.1节所述），$(\mathbf{W}_1, \mathbf{W}_2)$表示训练过程中每次迭代后调整的权重，而$e_{j,i}$ 表示从源节点j 到目标节点i [61]的边权重。每个图卷积网络层还使用 ReLU 激活函数：

$$f(x) = \max(0, x) \tag{8}$$

当接收到任何负类输入时，该函数返回 0；而对于任何正值x，它会返回该值。最后，将公式（7）的结果输入激活函数：

$$\mathbf{X}_{i+1} = f(\mathbf{X}_i) \tag{9}$$

这有助于我们获得局部节点嵌入。

节点嵌入是将图的节点或顶点映射到 N维数值空间中的过程。该流程根据图结构生成数值特征向量。彼此相似的节点在空间中彼此靠近，形成所谓的社区。例如，我们可以使用 Node2Vec [62]计算节点嵌入，并使用图 4 中的图进行可视化，该图取自图 1。比较这两幅图，可以看出 Node2Vec 如何分析输入电路，并通过嵌入表示相关节点。在图1中，三个节点（I_1, R_1, L_1）在图 4 中显示为一组，此外还有并联元件社区（$N_1, C_1, L_2, R_2, N_2, O$）；这与实际情况一致，因为这些元件彼此相连且距离相近。

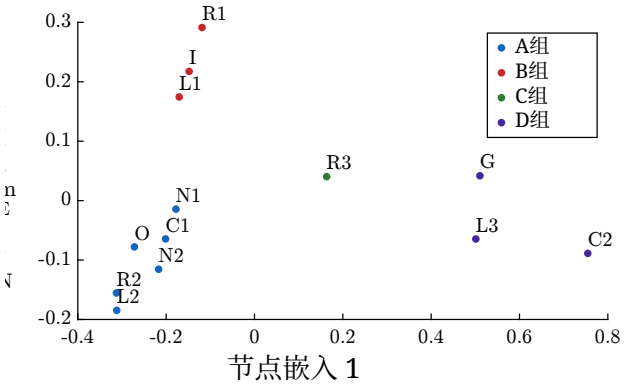

图4：计算得到的图中节点嵌入，见图1。

3.3.2 全局平均池化层。下一层是全局平均池化，它接收图卷积网络各层的最后一个输出，并通过对节点特征$\mathbf{X}$ 取平均，得到输出$\mathbf{r}$ ，从而纳入所有节点N的节点嵌入，为每个图生成图嵌入：

$$\mathbf{r}_i = \frac{1}{N_i}\sum_{n=1}^{N_i}\mathbf{X}_n \tag{10}$$

3.3.3 线性层。最后一层是分类层，它将均值池化层的输出作为输入，并应用以下线性变换：

$$\mathbf{y} = \mathbf{r}\mathbf{A}^T + \mathbf{b} \tag{11}$$

不过，在这一层之前会应用丢弃，按照指定的概率，从伯努利分布中抽取样本，随机将输入张量中的某些元素置零 [63]。

3.4 模型超参数和选项

对于选定的模型架构，可以调整若干参数，使模型能够针对所提供的数据类型进行良好训练。有关超参数及其调优的更多信息，请参阅以下资料 [60, 64, 65]。

3.4.1 学习率。模型的学习率（学习率）是一个正标量值，用于确定每轮次更新权重时，针对估计误差所作响应的步长大小；它被认为是较为重要的超参数之一[65]。通常，该值介于 0.0 到 1.0 之间。取值过小可能导致训练时间延长，而取值过大则可能导致训练流程不稳定。在这些案例研究中，我们使用的学习率为 0.001。

3.4.2 训练轮次数。一个训练轮次是指对整个训练数据集进行训练迭代的次数 [66]。每个训练轮次中，数据都会输入模型，并更新权重。训练轮次数通常可从数百次到数千次不等，使模型能够持续训练，直到模型中的误差最小化。案例研究中的不同实验将探讨训练轮次数。

3.4.3 优化算法。 优化算法是一种用于寻找函数输入参数的流程，使函数取得最小值或最大值 [66]。在此模型中，我们选择使用自适应矩估计（ADAM）算法，这是一种随机优化方法，可针对梯度的一阶矩和二阶矩的不同参数分别计算学习率 [67]。

3.4.4 损失函数。 损失函数用于计算模型当前输出与预期输出之间的距离。然后，这一度量指标衡量模型的表现。损失函数有多种，而此模型使用的是交叉熵 损失函数，通常用于分类问题。它衡量给定变量的两个概率分布之间的差异。

3.4.5 批次大小。 训练时，我们可以设定批次大小，即在更新模型权重之前需要处理的数据点数量。这种技术称为小批量处理，在训练深度学习模型时十分有利，因为它能让模型更好地扩展以处理大量数据。小批量处理并非逐个处理数据点或一次性处理全部数据，而是将一组中等规模的数据点归为一批。案例研究将探讨这一超参数。

3.5 指标

本节讨论用于评估模型性能的指标。

3.5.1 混淆矩阵。 如图 1所示，混淆矩阵（CM）是一个二维矩阵，其中每一列包含分类器（模型）的输出样本，每一行包含真实类别中的样本（数据） [68]。对于二元分类器，左上方的方格表示真正例 (T_P)，即被正确分类为 1 的数据点。右上方表示假正例 (F_P)，即被分类为 1、但实际上为 0 的数据点。左下方表示假负例 (F_N)，即被分类为 0、但实际上为 1 的数据点。最后，右下方表示真负例(T_N)，即被正确分类为 0 的数据点。该矩阵中的数值用于计算下文介绍的许多指标。

3.5.2 准确率。 准确率是正确预测的数量除以预测总数，见公式 (12)。

$$Accuracy = \frac{T_P + T_N}{N} \tag{12}$$

根据表1中的数据，我们可以计算出分类器的准确率为 0.86。

表1：混淆矩阵示例（含示例值）。数据

Model		实际 正类（1）	实际 负类（0）
	预测 正类 (1)	2,050（吨$_P$）	450（F_P）
	预测 负类 (0)	250（F_N）	2,250（T_N）

3.5.3 精确率 这个度量指标也称为阳性预测值（PPV），表示正类判定中实际正确的比例。当数据存在类别不平衡（例如，0 比 1 多）时，这个度量指标尤其有用。在某一类别的样本数量多于另一类别的数据集上进行训练时，模型可能会将所有数据都判定为最常见的类别，从而获得“较高的准确率”。这时就需要用到精确率，因为它是一个按类别定义的度量指标，定义如下：

$$Precision = \frac{T_P}{T_P + F_P} \tag{13}$$

再次以我们的示例为例，我们的分类器精确率为 0.82，这意味着当它将某个数据点预测为正类时，有 82% 的概率是正确的。相同的度量指标也可以应用于负类值，但在本文中，我们将使用精确率，因为我们的案例研究侧重于寻找最佳 建筑学。

3.5.4 召回率。该度量指标的计算方式如下：

$$Recall = \frac{T_P}{T_P + F_N} \tag{14}$$

这表示正确识别出的实际正类判定所占的比例。根据表 1，召回率 为 0.89，这表明模型正确识别了所有正类的 89%。

3.5.5 F1 分数。它将精确率和召回率结合为一个度量指标，计算二者的调和平均值，公式如下：

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{15}$$

模型的*F1* 分数 较高，意味着精确率 和召回率 都较高。继续这个示例，该模型的*F1* 分数 为 0.85。

3.5.6 马修斯相关系数（MCC）。在用于评估分类模型性能的指标中，这是一项最重要的指标。再次利用混淆矩阵中的数值，*MCC* 会在模型于混淆矩阵的所有格子中都取得良好结果时给出较高分数 [69,70]；这意味着 T_P 和 T_N 值较高，而F_P 和F_N 值较低。MCC 的计算方式如下：

$$MCC = \frac{T_P \cdot T_N - F_P \cdot F_N}{\sqrt{(T_P + F_P)(T_P + F_N)(T_N + F_P)(T_N + F_N)}} \tag{16}$$

马修斯相关系数 的取值范围为 −1 到 1，其中 1 表示模型能够完美地做出预测，−1 表示每个预测都不正确，而 0 表示模型的表现与随机猜测一样好。

最后，在我们的示例中，该模型取得的*MCC* 为 0.72，通常被认为是一个表现良好的模型。

3.5.7 总集合准确率。将数据集拆分为各自的子集（$\mathcal{G}_{known}$、$\mathcal{G}_{unknown}$）后，我们仍然希望知道预测结果与 $\mathcal{G}_{all}$ 相比表现如何（如果该数据可用）。因此，我们将总集合准确率定义如下：

$$Total\ Set\ Accuracy = \frac{T_P^{(u)} + T_N^{(u)} + T_P^{(k)} + T_N^{(k)}}{N_{all}} \tag{17}$$

其中，$(T(u)_P$，$T(u)_N)$ 是使用训练后的模型在 $\mathcal{G}_{unknown}$ 上确定的；而 $(T(k)_P$，$T(k)_N)$ 则使用 $\mathcal{G}_{known}$ 确定，由于第 3.2 节所述的中位数差异，其结果可能并非完全正确。如果基于不同中位数没有发生错误分类，那么 $T^{(k)}{}_P + T^{(k)}{}_N = N_{known}$。这项度量指标反映了一种潜在现实场景中的结果：设计者同时使用已知和未知数据对所有图进行分类。

4 案例研究：电路频率响应图设计问题

为探究图深度学习在基于图的工程设计问题中的能力，我们采用了一项频率响应匹配研究的结果 [1]。用于复现这些研究的数据集和代码见参考文献 [36]。

4.1 电路图枚举

模拟电路设计综合的自动化长期以来一直是备受关注的问题，人们为此进行了许多尝试 [71–73]。然而，在面对陌生而复杂的设计问题时，这些方法往往无法满足需求。即使引入启发式方法 [74] 和知识库 [75]，，确定全局解、备选方案集合及模式也并非易事，因此我们转而考虑枚举。

如第 2.2 节所述，我们将考虑参考文献 [1]中的电路枚举。具体而言，集合$\mathcal{G}$ 包含 43,249 个无向图，涵盖了最多含有 6 个阻抗子电路、带有*RC* 元件且要求连接至接地的所有拓扑结构 [1]。图 5展示了两个示例，其中 *RC* 的取值针对下一节所述的设计问题进行了优化选择。

许多研究人员采用了电路问题中的枚举技术 [11,76,77]，也将其用于其他数据可表示为图或其他可枚举对象的问题领域 [78–80]。如前所述，此处的每个电路都表示为顶点标记图，这意味着每个顶点都有一个关联标签，用于表示某种电路概念。每个图的大小介于 6 到 20 个节点之间，且所有图都保证具有不同的

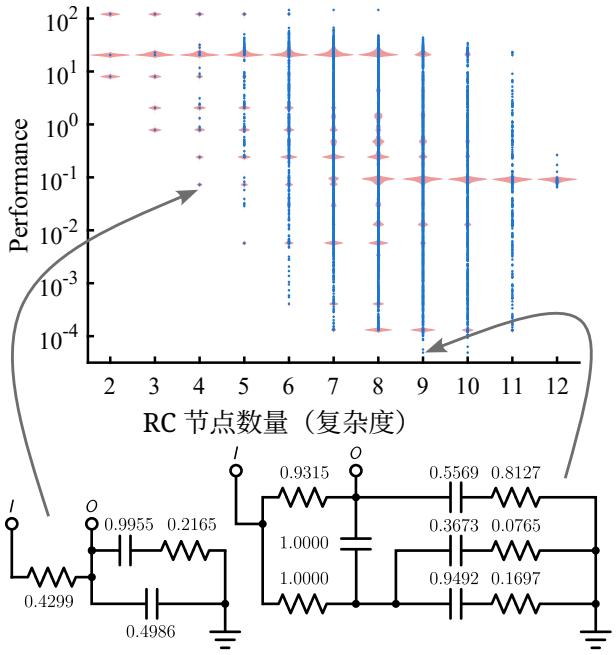

图 5：频率响应匹配优化案例研究数据集中全部 43,249 个图的 RC 节点数量（复杂度度量指标）与性能$J(G_i)$的汇总，图中显示了各个数据点及其密度。

拓扑结构各不相同（即不存在第2.1节所述的带标记图同构），这对于图深度学习来说相对较小。

4.2 频率响应匹配优化

此处的性能值$J(G_i)$是目标频率响应与所选电路所提供频率响应之间的误差，其定义最初基于参考文献 [71] 中的研究。电路图G_i本身并没有固有的J值；相反，该值取决于给定图中可调参数（RLC 系数）的取值，以及所选的优化问题及其目标函数和约束条件。根据参考文献 [1], ，我们考虑其中定义的“问题集 1”：

$$\underset{\mathbf{z}_i=\{\mathbf{R}_i,\mathbf{C}_i\}}{\text{minimize:}} \quad J=\sum_k \left(\log|H_i(j\omega_k,\mathbf{z}_i)|-\log|F(j\omega_k)|\right)^2 \tag{18a}$$

$$\text{subject to:} \quad 10^{-2}\le R_j\le 10^{0} \quad \text{for all } R_j \text{ in } G_i \tag{18b}$$

$$10^{-2}\le C_j\le 10^{0} \quad \text{for all } C_j \text{ in } G_i \tag{18c}$$

$$\text{where:} \quad |F(j\omega)|=\sqrt{\frac{2\pi}{10\omega}} \quad 0.2\le\frac{\omega}{2\pi}\le 5 \tag{18d}$$

其中，$|F(j\omega)|$是目标频率响应，$H_i(j\omega_k,\mathbf{z}_i)$是电路图$G_i$的频率响应，$\mathbf{z}_i$是图$G_i$中电阻器$R$和电容器$C$的优化变量集合，而 ω 在 500 个对数间隔评估点处进行采样。

求解这一非线性约束最小二乘优化问题可能耗费大量计算资源；原始研究中，每个图的优化都需要超过 8 小时，才能确定$J(G_i)$。因此，减少计算成本以甄别优劣电路图是可取的。

这里的性能值范围很大，在4×10^{-4}至2×10^{2}之间，如图5所示。分类依据抽样的$\mathcal{G}_{known}$进行，使用已知性能值的中位数，详见第3.2节。由于我们掌握所有图的性能值，这个问题可作为一个很好的示例，用于探索图深度学习在此类问题中的潜在有效性，从而与使用$\mathcal{G}_{all}$进行分类的效果作比较。

5 案例研究实验与结果

本节介绍针对第4节中的工程图设计问题数据集开展的实验。工具和计算架构详见附录A [36]。

5.1 实验1：建立基线

在第一个实验中，我们首先创建一个基线模型，以供后续比较。这里，我们将训练集 $\mathcal{G}_{training}$设为72%（31,139个图），验证集$\mathcal{G}_{validation}$设为18%（7,784个），剩余10%（4,324个）设为$\mathcal{G}_{unknown}$。这绝不是推荐的数据划分比例，而是设定一种可供GDL模型使用大量数据的情境，以探索可能实现的效果，以及如何在准确率与效率之间取得平衡。

本次实验的混淆矩阵见表2，使用了 $\mathcal{G}_{unknown}$ 中的 4,324 个图，这些图从未以任何方式用于模型训练（实际上，您不会有这个混淆矩阵）。根据公式（12），我们得出准确率为 79.1%，与基线模型在 $\mathcal{G}_{known}$ 上的最终准确率 79.5% 相当接近。接下来，根据公式（13），我们计算得出基线模型的精确率为 77%。根据公式（14），我们计算得出基线模型的召回率为 79.9%。最后，根据公式（15）和公式（16），我们分别计算得出$F1$ 分数为 78.6%，以及马修斯相关系数为 58%。

5.2 实验 2：额外的基于图的特征

这里，我们考虑在 $\mathbf{X}$ 中添加额外的基于图的特征，超越当前的顶点标签 R、C、G 等。这些额外特征应能更好地解释训练数据中的方差（这称为特征工程和特征选择）。其理论依据是，在训练轮次相同的情况下，这些特征将提升模型的性能。这里添加的两个特征是特征向量中心性和介数中心性。

特征向量中心性根据节点邻居的中心性计算该节点的中心性，是衡量图中节点影响力的指标。特征向量中心性得分越高，意味着该节点连接的高分节点越多。节点 v 的特征向量中心性是向量 v 的第i个归一化

表 2：用于预测$\mathcal{G}_{unknown}$的基线模型的混淆矩阵。

Model \ Data	实际 正类（1）	实际 负类 (0)
预测 正类 (1)	1,667	487
预测 负类 (0)	419	1,751

表3：*3* 特征模型的混淆矩阵，预测$\mathcal{G}_{unknown}$。

Model \ Data	实际 正类 (1)	实际 负类 (0)
预测 正类 (1)	1,901	213
预测 负类 (0)	451	1,759

元素，计算公式如下：

$$\mathbf{Av} = \lambda \mathbf{v} \tag{19}$$

其中 $\mathbf{A}$ 是邻接矩阵，λ 是最大特征值[81]。

现在，节点的介数中心性是经过该节点的所有节点对最短路径所占比例之和：

$$c_B(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{20}$$

其中 V 是节点集合，$\sigma(s,t)$ 是最短路径的数量，而 $\sigma(s,t|v)$ 是经过某个除 (s,t) 之外的节点 v 的路径数量 [82]。

我们还要指出，计算这些额外特征的计算成本，与计算 $J(G_i)$ 相比相当低，因此对每个图而言，增加的成本相对较小。加入这两个额外特征后，每个图对应的新特征矩阵 $\mathbf{X}$从 $\mathbf{X} \in \mathbb{R}^{n\times 1}$ 变为 $\mathbf{X} \in \mathbb{R}^{n\times 3}$。本次实验的混淆矩阵见表 3。加入这两个额外特征后，模型的准确率达到 85%，精确率达到 89.9%，召回率 达到 80.8%，*F1* 分数 达到 85%，马修斯相关系数 达到 70%。因此，图深度学习模型在相同的 $\mathcal{G}_{known}$ 数据集上能够做出更好的预测，因为所有指标均持平或有所提升。接下来我们将使用全部三项特征。

5.3 实验 3：训练轮次数与已知图数量

此前的实验假设训练轮次数较多，但为了降低计算成本，我们希望了解实际需要多少轮次才能有效训练模型。这里我们将观察最多 1,000 轮次的训练行为

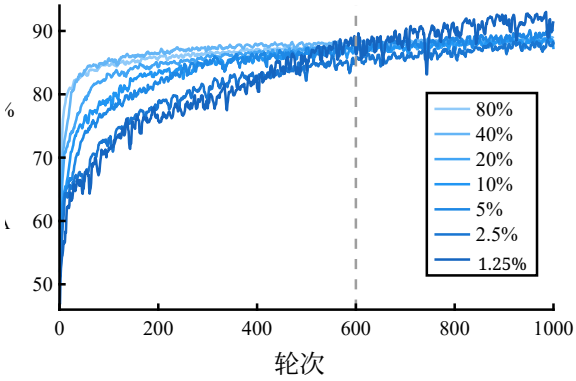

图 6：使用数据集的不同比例进行训练时，七种不同模型的准确率，用于确定约 600 轮的停止点。

并为后续实验确定一个上限。

此外，像前几次实验通常采用的做法那样，将 80% 的数据用于训练并不可取，尤其是每个 $J(G_i)$ 的成本相当高。因此，本实验还将探究 ff的不同百分比 $\mathcal{G}_{known}$，这表示使用昂贵的公式（18）确定尺寸的电路数量较少。目标是针对这个特定数据集，确定 $\mathcal{G}_{known}$ 相对于 $\mathcal{G}_{all}$ 的合适比例，在维持适当准确率的同时降低总体计算负担（包括模型训练时间以及生成 $\mathcal{G}_{known}$所需的时间）。预计随着训练数据量减少，模型性能也会下降。

5.3.1 训练轮次数。使用不同的ff个模型进行训练，各模型采用不同的ff个N_{known}，取值范围为 1.25% 至 80%。观察图 6中的结果，可以清楚地看到，在约 600 轮次时，许多准确率指标趋于平稳。对于一些较小的训练集，准确率在此后仍会继续提高，但观察发现，这主要是对小型训练数据集的过拟合。因此，将轮次上限设为 600；通常这能使模型得到充分训练，同时避免浪费额外的迭代次数。

5.3.2 已知图数量。使用相同的设置，我们可以探究在 权衡中纳入不同数量（此处以百分比表示）的图所带来的取舍 $\mathcal{G}_{known}$。减少 N_{known} 会导致 $N_{unknown}$ 增加，根据式（5），因此需要预测更多图的分类结果。因此，本实验旨在确定模型要保持较高准确率所需的最少数据量。

结果显示在表4 中，所有指标均使用未知图计算得出 $\mathcal{G}_{unknown}$。正如预期，N_{known} 越大，模型的准确率越高。现在，只需将数据量从 80% 减半至 40%，我们仍能达到 ≈83% 的高准确率，且 马修斯相关系数 仍为 0.68 左右。

表 4：不同已知图数量$_{known}$ 取值下的结果，所有指标均针对未知图 $\mathcal{G}_{unknown}$计算。

已知图数量$_{known}$ [图的数量]	34,599	17,300	8,650	4,325	2,162	1,081	541	270	135
% 的N_{all}	80%	40%	20%	10%	5%	2.5%	1.3%	0.6%	0.3%
训练时间 [秒]	4,294	2,239	1,257	720	464	334	269	243	229
准确率 [%]	83.80	83.99	82.70	81.73	78.68	76.15	72.28	66.70	60.69
精确率 [%]	84.10	88.37	80.29	79.11	77.18	76.51	70.97	70.54	67.70
召回率 [%]	83.22	78.52	86.85	86.30	81.37	75.49	75.41	57.39	40.90
F1 分数 [%]	83.66	83.15	83.44	82.55	79.22	75.99	73.13	63.29	51.00
MCC	0.68	0.68	0.66	0.64	0.57	0.52	0.45	0.34	0.23

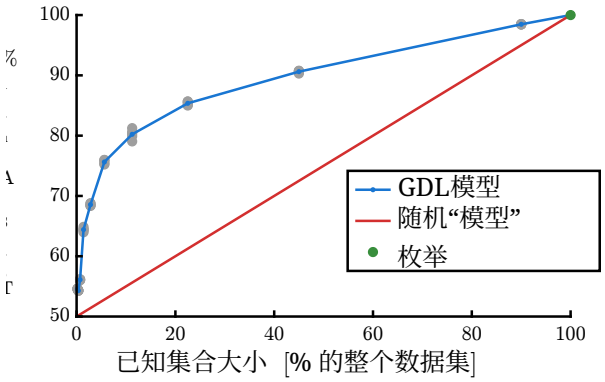

图 7：不同已知集合大小下五次运行的总集合准确率平均得分 N_{known}。

然而，一个关键的结论是训练时间。使用 80% 数据的模型耗时 4,294 秒（约 72 分钟），而使用 40% 数据的模型仅耗时 37 分钟；将数据量减半，训练时间也减半。我们还应考虑构建 $\mathcal{G}_{known}$ 所需的计算成本；使用 40% 的数据，确定所有必需的 $J(G_i)$ 所需时间名义上将减半。在权衡总计算成本与模型效果（例如准确率）时，这两个因素对于理解理想的 N_{all} 值至关重要。再看较小的 N_{known} 值，结果也符合预期：N_{known} 越小，许多得分就越低。在运行次数的极限情况下，仅使用 135 个图（0.3%）时，模型的准确率仅为 ≈60%，马修斯相关系数 也低得多，仅为 0.23。

为了更好地了解在案例研究中使用几何深度学习模型所需的权衡取舍，我们将式 总集合准确率（17）中的度量指标可视化，如图 7 所示。由于该方法包含若干随机因素，图中展示了五次不同运行的平均值，不同的灰色圆点则表示各次运行的数值。总体而言，各次运行之间的变化相对较小。除了这条曲线外，我们还加入了一个理论上的随机“模型”：假设所有已知值均预测正确，其余值则被随机分配为好／坏分类。最后，当已知集合大小占数据集的 100% 时，我们已经枚举了所有图，因此总集合的总集合准确率自然为 100%。因此，应以这些额外标准为参照来比较图深度学习。

总体而言，图深度学习模型的表现远胜随机模型，准确率约高出 1%–40%；但随着已知图数量增加，差距开始缩小，并且所有方法都趋近于 N_{known} 增大时的 100%/100% 穷举点。因此，一般建议将 N_{known} 设为 N_{all} 的 20%；不过，具体取值当然可能因问题而异，也取决于设计者如何权衡总集合准确率 与计算成本（通常与N_{known}成正比）。

5.4 实验4：迭代式 GDL 分类

即使分类完全准确，目前为止所述的方法也只能确定 $\mathcal{G}_{all}$ 中表现排名前 50% 的图。尽管这一结果在实践中无疑很有用，但通常我们希望进一步缩小图的范围，筛选出表现最佳的图。不过，我们没有理由只能构建一个分类模型。

在本实验中，我们希望从 $\mathcal{G}_{all}$ 中筛选出一个规模更小、中位数表现更好的图集，方法是按以下步骤迭代构建图深度学习模型：

1. 设置 $k=1$，并创建一个初始 $\mathcal{G}^k_{known}$。 2. 使用 $\mathcal{G}^k_{known}$ 创建一个 GDL模型$m^k(G_i)$；根据 $\mathcal{G}^k_{known}$ 的中位数 J 值，自然地将其划分为“已知 1”和“已知 0”两组。 3. 使用 $m^k(G_i)$ 预测 $\mathcal{G}^k_{unknown}$ 的类别，得到“预测为 1”和“预测为 0”两组，分别包含预测为优质（1）或较差（0）的图。 4. 目标是识别优质图，因此令 $\mathcal{G}^{k+1}_{known}$ 等于“已知 1”，令 $\mathcal{G}^{k+1}_{unknown}$ 等于“预测为 1”（并假定其余图表现较差，将其移除）。 5. 设置 $k \to k+1$，并重复步骤 2，直至 $k=n$。

每次迭代中，$\mathcal{G}^k_{known}$ 的大小都会减半，从而降低其中位性能值，因此，$m^k(G_i)$的优劣分类阈值也应相应调整。

本实验以已知 20% 的数据初始化，结果汇总于图8。首先，最左侧的箱

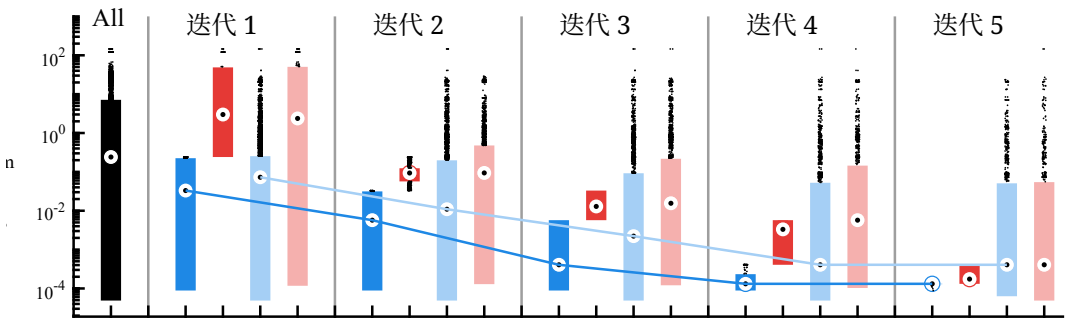

图 8：第 5.4 节所述方法的五次迭代，利用连续的、以性能最小化为目标的图子集（括号中的数字表示相应集合中的图数量）重新训练 GDL模型，同时展示所有数据点的分布 $J(\mathcal{G}_{all})$ 以供比较。

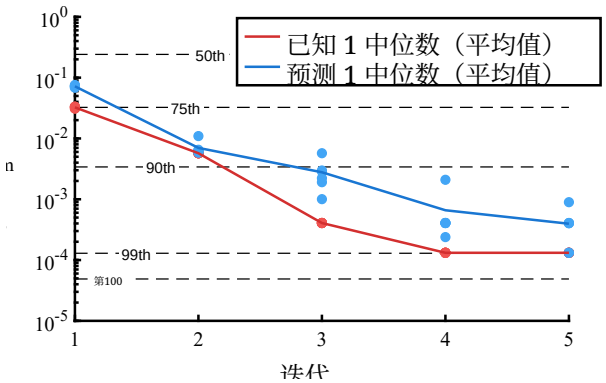

图 9：采用迭代图深度学习分类方法，针对所示的不同性能百分位数，对“已知 1”和“预测 1”集合在六次运行中的中位数取平均值，$J(\mathcal{G}_{all})$。

线图展示了$J(\mathcal{G}_{all})$的分布。接下来的两个箱线图展示了基于迭代 1 分类阈值对 $\mathcal{G}_{known}$ 进行划分的结果。迭代 1 中最后两个箱线图分别展示了 GDL模型预测为 1 和 0 的结果；请注意，虽然确实存在识别错误的图，但其中位数彼此相距较远，并且与对应的“已知”结果一致。随后的箱线图展示了各次迭代的结果，直至n = 5。我们观察到，在迭代 5 时，当前有N_{known} = 540 个图，分类效果相当差，因为“预测”箱线图的分布相似。两个中位数的趋势线也体现了这一点：直到第五次迭代，中位数都大幅下降。这种行为或许可以预料，因为我们没有从初始 $\mathcal{G}^1_{known}$中添加任何新数据，而且数量用于训练的图数量会减少。根据“预测 1”在每次迭代中添加更多数据是未来的工作。

评估第 4 次迭代结束时的结果（由于第 5 次分类效果不佳），最终“优质”集合中共有 2,920 个图，中位数为 3×10^{-4}，远低于初始分类的中位数 2×10^{-1}。此处假设设计者会对“预测为 1”的所有图评估 $J(G_i)$；因此，总共有 11,029 个图经过了优化。对于这个最终集合，排名前 10 的图全部保留，排名前 100 的图保留了 87 个，排名前 1000 的图保留了 727 个，而计算成本仍仅为完全枚举的 25%。此外，与从 $\mathcal{G}_{all}$中随机抽取 11,029 个图相比，迭代 GDL 模型的结果远超预期平均值：随机集合中包含排名前 10、100 和 1000 的图的预期数量分别为 2.25 、 22.51 和 225.13。

为了更好地理解这种迭代方法的统计显著性，我们采用不同的抽样，额外完成了五次随机运行。次运行中，平均有 11282.2 个图被判定为已知或“优化”（标准差为 492.8）。在排名前 100 的图中，平均有 88.2 个（标准差为 2.5）保留下来，而预期值为 22.96。最后，在排名前 1000 的图中，平均有 751.2 个（标准差为 39.5）保留下来，而预期值为 229.6。此外，图9 展示了六次运行中每次迭代的“已知 1”和“预测 1”图集合的中位数。这里仍可看到中位数下降，表明迭代式 GDL 方法正将潜在“优质”图的范围缩小到性能较高的百分位区间。事实上，在迭代 4 时，“预测 1” 的所有中位数都优于第 90 百分位数，均值则更接近第 95 百分位数。然而，如前文在图8中所讨论的，第 5 次迭代带来的收益微乎其微。

如图7所示，若要通过减少对 的必要评估次数来降低计算成本，并同时减少 GDL 任务，就需要权衡取舍。因此，我们开展了另一项研究，仅减少起始 $J(G_i)$ 的数量。初始时仅使用 10% 的 $\mathcal{G}_{all}$，迭代方法最终得到平均 8118.2 个（标准差为 969.1）需要依据式（18）进行优化的图。五次随机运行的结果汇总如下：排名前 10 的图中有 9.2 个（标准差为 0.4）保留下来，排名前 100 的图中有 85.2 个（标准差为 6.3）保留下来，排名前 1000 的图中有 715.2 个（标准差为 76.9）保留下来。不同起始 规模之间的另一项差异是不同起始 $\mathcal{G}_{known}$规模（即 20% 与 10%）下的迭代式 GDL 分类总成本，其中包括每次迭代训练各个 GDL 模型。两者的差异为：20% 需要 45 分钟，10% 需要 36 分钟$\mathcal{G}^1_{known}$。由于两种情况下汇总的顶级图结果相差不大，进一步研究如何减少用于 $\mathcal{G}^1_{known}$ 的图数量，对于了解设计研究的总计算成本与满足不同设计者目标之间的权衡至关重要。

6 结论

本文提出了一种几何深度学习（GDL）方法，用于对基于图的模拟电路进行分类和筛选，并依据其在工程设计问题中的性能评估结果，将其筛选至性能更优的解集。在所呈现的图设计案例研究中，所有图均已知，但其基于优化的性能值计算成本高昂。我们展示了 GDL 在性能数据有限的情况下帮助区分“优质”和“劣质”图的潜力，也揭示了一些需要调整的关键超参数。此外，加入特征值中心性和介数中心性这两种基于图的特征后，许多关键指标均有所改善。

结果表明，对于这一案例研究，GDL 可作为一种有效且高效的方法来缩小图集合，因为模型实现了 80% 的总体集合分类准确率。此外，迭代式几何深度学习分类方法以完全枚举 25% 的计算成本，识别出了排名前 10 的 9.0 个、排名前 100 的 88.2 个，以及排名前 1000 的 751.2 个。

未来工作的重点包括：研究如第5.4节所述，迭代地向数据集中添加新图；探索使用训练好的 GDL 模型预测 $\mathcal{G}_{unknown}$ 类别的替代方法（改变每次迭代中要移除的图）；以及在类似问题上进行迁移学习 [1]。此外，还需研究如何将此方法应用于其他工程领域的数据集，例如包含有向图和多个性能指标的数据集。与许多基于图神经网络的方法一样，仍需进一步探索最佳神经网络架构和超参数，以降低数据输入要求并提高模型准确性。此外，与第5.2节类似，添加其他基于图的特征可能带来更理想的结果（例如，电流流接近中心性、接近活力和调和中心性 [83, 84]）。针对这个小型工程图大型数据集取得的良好结果表明，迭代分类式图深度学习前景广阔，有望更广泛地应用于其他工程图设计问题。

参考文献

[1] Herber, D. R., 2017. “组合架构、被控对象与控制设计方面的进展”。博士论文，伊利诺伊大学厄巴纳-香槟分校，美国伊利诺伊州厄巴纳。[2] Selva, D., Cameron, B., and Crawley, E., 2016. “系统架构决策中的模式”。《系统工程》，19(6)，第 477–497 页。doi: 10 1002 21370 . / sys. Foster, R. M., . “电网络的几何电路”。美国电气工程师学会汇刊，[3] 1932 ***51*** 2 309 317 10 1109 1932 5056068 ( )，第 – 页。doi: . /T-AIEE. . Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., and Yin, D., 2019. “用于社交推荐的图神经网络”。载于 WWW ’19：万维网大会，第 417–426 页。doi: 10 1145 3308558 3313488 . / . Zhou, Y., Liu, S., Siow, J., Du, X., and [4] [5] Liu, Y., 2019. “Devign：通过图神经网络学习全面的程序语义，实现有效的漏洞识别”。载于神经信息处理系统会议。[6] Cheng, X., Wang, H., Hua, J., Xu, G., and Sui, Y., 2021. “Deep-wukong：利用深度图神经网络静态检测软件漏洞”。《ACM软件工程与方法论汇刊》，30(3)。doi: 10 1145 3436877 . / Yang, W., Ding, H., and Zhang, D., 2018. “行星齿轮系的新型图表示”。美国机械工程师学会机械设计期刊，[7] 140(1)，第 012303 10 1115 1 4038303 页。doi: . / . Hsu, C.-H., and Lam, K.-T., 1992. “一种用于行星直齿轮系自动运动学分析的新型图表示”。美国机械工程师学会机械设计期刊，[8] 114(1)，第 196–200 页。doi: 10 1115 1 2916916 . / . Herber, D. R., Guo, T., and Allison, J. T., 2017. “利用完美匹配进行架构枚举”。美国机械工程师学会机械设计期刊，[9] ***139*** 5 051403 10 1115 1 4036132 ( )，第 . 页。doi: . / . Herber, D. R., . “对基于完美匹配的图枚举方法在工程挑战中的改进”。载于美国机械工程师学会国际设计工程技术会议。[10] 2020 doi: 10 1115 2020 22774 . /DETC - Macmahon, P. A., . “电阻的组合”。《离散应用数学》，[11] 1994 54(2)，第 225–228 页。doi: 10.1016/0166-218 94 90024 8 X( ) - Maier, M. W., and Rechtin, E., 2009.《系统架构设计艺术》，第 3 版。CRC出版社。[12] [13] Arney, D. C., and Wilhite, A. W., 2014. “利用图论对航天系统架构建模”。《航天器与火箭》，51(5)，第 1413 1429 10 2514 1 32578 – 页。doi: . / .A Taft, J., .《系统架构的数学表示》。技术报告 PNNL-27387，巴特尔纪念研究所为美国能源部、太平洋西北国家实验室编制，3月。[14] 2018 [15] Potts, M., Pia, S., Johnson, A., and Bullock, S.《隐藏结构：运用图论探索复杂的系统之系统架构》。[16] Schmidt, L. C., Shetty, H., and Chase, S. C., 1999. “一种用于机构结构综合的图文法方法”。美国机械工程师学会机械设计期刊，***122*** 4 371 376 10 1115 1 1315299 ( )，第 – 页。doi: . / . Wyatt, D. F., Wynn, D. C., Jarrett, J. P., and Clarkson, P. J., 2012. “通过计算设计综合支持产品架构设计 [17]

中的网络结构约束”。《研究工程设计》，23，第 17–52 页。doi: 10.1007/s00163-011-0112-y [18] LeCun, Y., Bengio, Y., and Hinton, G., 2015. “深度学习”。《自然》，*521*，436 444 10 1038 14539 第 – 页。doi: . /nature [19] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Van- dergheynst, P., 2017. “几何深度学习：超越欧几里得数据”。《IEEE信号处理杂志》，34(4)，第 18–42 页。doi: 10 1109 2017 2693418 . /msp. . Atz, K., Grisoni, F., and Schneider, G., 2021.《分子表示上的几何深度学习》。arXiv: 2107 12375 . Gainza, P., [20] Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bron- stein, M. M., and Correia, B. E., 2020. “利用几何深度学习从蛋白质分子表面解读相互作用指纹”。《自然》，17，第 184–192 页。doi: 10.1038/s41592-019-0666-6[22] Segler, M. H. S., Kogej, T., Tyrchan, C., and Waller, M. P., 2018. “利用循环神经网络生成用于药物发现的定向分子库”。《ACS中心科学》，4(1)，第 120–131 页。doi: 10 1021 7 00512 . /acscentsci. b Krokos, V., [23] Bordas, S. P. A., and Kerfriden, P., 2022.《一种基于图的概率几何深度学习框架，结合在线物理校正以预测多孔材料中缺陷的临界性》。arXiv: 2205 06562 . Fedorova, S., Tono, A., Nigam, M. [24] S., Zhang, J., Ahmadnia, A., Bolognesi, C., and Michels, D., 2021. “用于建筑学领域几何深度学习的合成数据生成流程”。《国际摄影测量、遥感与空间信息科学档案》，XLIII-B2-2021，第 337–344 10 5194 2 2021 337 202 页。doi: . / [25] isprs-archives-XLIII-B - - - Thiery, A. H., Braeu, F., Tun, T. A., Aung, T., and Girard, M. J. A., 2022.《几何深度学习在青光眼诊断中的医学应用》。arXiv: 2204 07004 . Sarasua, I., Lee, J., and [26] Wachinger, C., 2021.《利用解剖网格进行阿尔茨海默病预测的几何深度学习》。arXiv:2104.10047[27] Wong, J. C., Ooi, C. C., Chattoraj, J., Lestandi, L., Dong, G., Kizhakkinan, U., Rosen, D. W., Jhon, M. H., and Dao, M. H., 2022. “用于几何设计的物理仿真图神经网络代理模型”。载于IEEE计算智能研讨会系列，第 1469–1475 页。doi: 10 1109 51031 2022 10022022 . / [28] SSCI . . Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. W., 2021.《利用图网络学习基于网格的仿真》。arXiv:2010.03409[29] Park, J., and Park, J., 2019. “物理诱导图神经网络：在风电场功率估算中的应用”。《能源》，187，第 115883 10 1016 2019 115883 页。doi: . / [30] 2019 j.energy. . Zhang, G., He, H., and Katabi, D., . “Circuit-GNN：用于分布式电路设计的图神经网络”。载于国际机器学习会议，第 7364–7373 页。[31] Xiao, Y., Ahmed, F., and Sha, Z., 2023. “基于图神经网络的共享出行系统设计决策支持”。美国机械工程师学会机械设计期刊，*145* 9 091703 10 1115 1 4062666 ( )，第 . 页。[32] 2021 doi: . / . Ferrero, V., DuPont, B., Hassani, K., and Grandi, D., . “利用图神经网络对产品装配体中的组件功能进行分类”。美国机械工程师学会机械设计期刊，144(2)，第 021406 页。doi: 10 1115 1 4052720 . / . Regenwetter, L., [33] Nobari, A. H., and Ahmed, F., 2022. “工程设计中的深度生成模型：综述”。美国机械工程师学会机械设计期刊，*144* 7 071704 10 1115 1 4053859 ( )，第 . 页。doi: . / . [34] Ranjan, A., Bolkart, T., Sanyal, S., and Black, M. J., 2018.《利用卷积网格自编码器生成三维人脸》。arXiv:1807.10267[35] Cheng, S., Bronstein, M., Zhou, Y., Kotsia, I., Pantic, M., and Zafeiriou, S., 2019.《MeshGAN：非线性三维人脸可变形模型》。arXiv:1903.10384 [36] https://github.com/anthonysirico/GDL-for-Engineering-Design。[37] Guo, T., Herber, D. R., 和 Allison, J. T., 2019年。“使用生成对抗网络（GAN）进行电路综合”。载于 AIAA 科技论坛。doi: 10 2514 6 2019 2350 . / . - Diestel, R., .《图论》。施普林格。[38] 2017 doi: 10.1007/978-3- 662-53622-3[39] Godsil, C., 和 Royle, G., 2001年。《代数图论》。施普林格。doi: 10 1007 978 1 4613 0163 9 . / - - - - Borkar, [40] V. S., Ejov, V., 和 Nguyen, G. T., 2012年。《哈密顿回路问题与马尔可夫链》。施普林格。doi: 10.1007/978-1-4614- 3232-6[41] Herber, D. R., 和 Allison, J. T., 2019年。“一种结合架构、被控对象与控制设计的问题类别，并将其应用于车辆悬架”。美国机械工程师学会机械设计期刊，141(10)，第 101401 页。doi: 10 1115 1 4043312 . / . Guo, T., Herber, D. R., 和 Allison, J. T., 2018年。[42] “利用主动学习降低电路综合的评估成本”。载于美国机械工程师学会国际设计工程技术会议，第 DETC 2018 85654 10 1115 2018 85654 - [43] 2012 号。doi: . /DETC - Krizhevsky, A., Sutskever, I., 和 Hinton, G. E., . “使用深度卷积神经网络进行 ImageNet 分类”。《神经信息处理系统进展》25，第 1097–1105 页。[44] Wang, T., Wu, D. J., Coates, A., 和 Ng, A. Y., 2012年。“使用卷积神经网络进行端到端文本识别”。载于国际模式识别会议，第 3304–3308 页。[45] Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., Gong, Y., 和 Acero, A., 2013年。“微软语音研究中深度学习的最新进展”。载于 IEEE 国际声学、语音与信号处理会议，第 8604 8608 10 1109 2013 6639345 – 页。doi: . / [46] 2017 ICASSP. . Nickel, M., 和 Kiela, D., . 用于学习层次表示的庞加莱嵌入。arXiv: 1705 08039 。Chamberlain, B. P., [47] Clough, J., 和 Deisenroth, M. P., 2017年。双曲空间中的图神经嵌入。arXiv: 1705 10359 。Bronstein, [48] 2021 M. M., Bruna, J., Cohen, T., 和 Velickovic, P., . 几何深度学习：网格、群、图、测地线与规范。arXiv:2104.13478[49]Cohen, T. S., 和 Welling, M., 2016年。可转向卷积神经网络。arXiv:1612.08498[50] Cohen, T. S., Geiger, M., Koehler, J., 和 Welling, M., 2018年。球面卷积神经网络。doi: 1801.10130[51] Lecun, Y., Bottou, L., Bengio, Y., 和 Haffner, P., 1998年。“将基于梯度的学习应用于文档识别”。《IEEE 论文集》，*86* 11 2278 2324 10 1109 5 726791 ( )，第 – 页。doi: . / . Zhang, M., [52] Cui, Z., Neumann, M., 和 Chen, Y., 2018年。“用于图分类的端到端深度学习架构”。载于 AAAI 人工智能会议，第 32 卷，第 4438– 4445 10 1609 32 1 11782 页。doi: . /aaai.v i . Kipf, T. N., [53] 2016 和 Welling, M., . 基于图卷积网络的半监督分类。arXiv: 1609 02907 。Ma, Y., Wang, S., [54] 2019 Aggarwal, C. C., 和 Tang, J., . 基于 EigenPooling 的图卷积网络。arXiv: 1904 13107 。Ying, R., You, J., [55] Morris, C., Ren, X., Hamilton, W. L., 和 Leskovec, J., 2018年。利用可微池化进行层次图表示学习。arXiv: 1806 08804 。 [56] Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., 和 Adams, R. P., 2015年。用于学习分子指纹的图卷积网络。arXiv: 1509 09292 。 [57] Rousseau, F., Kiagias, E., 和 Vazirgiannis, M., 2015年。“将文本分类视为图分类问题”。载于计算语言学协会年会和第七届国际自然语言处理联合会议，第 1702–1712 页。doi: 10 3115 1 15 1164 . /v /P - Shen, M., Zhang, J., [58] Zhu, L., Xu, K., 和 Du, X., 2021年。“准确的

利用图神经网络通过加密流量分析识别去中心化应用”。IEEE 信息取证与安全汇刊，16，第 2367 2380 10 1109 2021 3050608 – 页。[59] doi: . /TIFS. . Hashemi, A., 和 Pilevar, A. H., . “利用图分类检测肺部CT图像中的肿块”。电气与电子工程杂志，3(3)。[60] James, G., Witten, D., Hastie, T., 和 Tibshirani, R., 2021年。《统计学习导论》。施普林格。doi: 10.1007/978-1-0716- 1418-1[61] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., 和 Grohe, M., 2018年。Weisfeiler和Leman走向神经网络：高阶图神经网络。arXiv: 1810 02244 。[62] Grover, A., 和 Leskovec, J., . node vec：面向网络的可扩展特征学习。arXiv: 1607 00653 。[63] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., 和 Salakhutdinov, R., 2012年。通过防止特征检测器协同适应来改进神经网络。arXiv: 1207 0580 。[64] Bengio, Y., . 深度架构基于梯度训练的实用建议。arXiv: 1206 5533 。[65] Goodfellow, I., Bengio, Y., 和 Courville, A., 2016年。《深度学习》。麻省理工学院出版社。[66] Chollet, F., 2018年。《Python深度学习》。Manning 出版社。[67] Kingma, D. P., 和 Ba, J. L., 2015年。“Adam：一种随机优化方法”。载于国际学习表征会议。[68] Simske, S. J., 2013年。《元算法学：构建稳健、低成本、高质量系统的模式》。约翰·威立父子出版公司。[69] Jurman, G., Riccadonna, S., 和 Furlanello, C., 2012年。“多分类预测中马修斯相关系数（MCC）与 CEN 误差度量的比较”。《公共科学图书馆·综合》，*7* 8 41882 10 1371 0041882 ( )，第 e 页。[70] doi: . /journal.pone. Chicco, D., . “计算生物学机器学习的十条实用建议”。BioData Min.，10(35)。doi: 10.1186/s13040-017-0155-3[71] Grimbleby, J. B., 1995年。“使用遗传算法自动进行模拟网络综合”。载于第一届工程系统遗传算法国际会议：创新与应用，第 53 58 10 1049 19951024 – 页。doi: . /cp: Das, A., 和 Vemuri, R., . “ 使用遗传算法的自动无源模拟电路综合框架”。载于 IEEE计算机学会年度超大规模集成电路研讨会，第 145–152 页。doi: 10 1109 2007 22 . /ISVLSI. . Grimbleby, J. B., . “使用遗传算法自动进行模拟电路综合”。IEE 电路与器件系统学报，147(6)，第 319–323 页。doi: 10 1049 20000770 . /ip-cds: Sussman, G., 和 Stallman, R., 1975年。“计算机辅助电路分析中的启发式技术”。IEEE 电路与系统汇刊，*22* 11 857 865 10 1109 1975 1083985 ( )，第 – 页。doi: . /TCS. . Harjani, R., Rutenbar, R., 和 Carley, L., . “OASYS：模拟电路综合框架”。IEEE 计算机辅助设计汇刊，8(12)，第 1247 1266 10 1109 43 44506 – 页。doi: . / . Lomnicki, Z. A., . “二端串并联网络”。《应用概率进展》，*4* 1 109 150 10 2307 1425808 ( )，第 – 页。doi: . / Isokawa, Y., . “串并联电路与连分数”。《应用数学科学》，10(27)，第 1321–1331 页。doi: 10 12988 2016 63103 . /ams. . Bayrak, A. E., Ren, Y., 和 Papalambros, P. Y., 2016年。“面向混合动力汽车架构设计的拓扑生成”。美国机械工程师学会机械设计期刊，*138* 8 081401 10 1115 1 4033656 ( )，第 页。doi: . / . del Castillo, J. M., . “ 基于功能约束的 -DOF 行星齿轮系图枚举 ”。美国机械工程师学会机械设计期刊，*124* 4 723 732 10 1115 1 1514663 ( )，第 – 页。doi: . / . Ma, W., Trusina, A., El-Samad, H., Lim, W. A., 和 Tang, C., 2009年。“定义能够实现生化适应的网络拓扑结构”。《细胞》，138(4)，第 760–773 页。doi: 10.1016/j.cell.2009.06.013

[81] Bonacich, P., 1987年。“权力与中心性：一族度量指标”。《美国社会学杂志》，*92* 5 1170 1182 10 1086 228631 ( )，第 – 页。doi: . / Freeman, L. C., .“基于中介中心性的一组中心性度量指标”。《社会计量学》，*40* 1 35 41 10 2307 3033543 ( )，第 – 页。[83] doi: . / Koschützki, D., Lehmann, K. A., Peeters, L., Richter, S., Tenfelde- Podehl, D., and Zlotowski, O., 2005年。《中心性指标》。施普林格，第 16 61 10 1007 978 3 540 31955 9_3 – 页。doi: . / - - - - Boldi, P., and Vigna, S., .“中心性公理”。《互联网数学》，*10*， 222 262 10 1080 15427951 2013 865686 第 – 页。doi: . / . . Fey, M., and Lenssen, J. E., .“ 使用 PyTorch Geometric 快速进行图表示学习”。载于 ICLR 图与流形表示学习研讨会。[86] Paszke, A., and et al., 2019年。“PyTorch：一种命令式风格的高性能深度学习库”。载于《神经信息处理系统进展》第 32 卷。[87] Van Rossum, G., and Drake, F. L., 2009年。《Python 3 参考手册》。CreateSpace，斯科茨谷，CA。[88] Hagberg, A. A., Schult, D. A., and Swart, P. J., 2008年。“使用 NetworkX 探索网络结构、动态特性与功能”。载于 Python 科学会议，第 11–15 页。[89] pandas 开发团队，2020年。pandas-dev/pandas：Pandas。doi: 10.5281/zenodo.3509134

A 工具与计算架构

主要使用的工具是 PyTorch-Geometric (PyG) [85]，因为 PyG 能够广泛应用于图数据。它构建于 PyTorch 之上 [86], 。PyTorch 是一个开源机器学习框架，还包含丰富的工具库，可用于模型操作和数据分析。由于这两种工具主要与 Python 配合使用 [87], ，因此 Python 将作为模型的首选语言。我们还使用 Networkx，这是一个用于创建和操作复杂网络的 Python 软件包 [88]。作出这一选择，是因为 PyG 要求数据采用特定类型的实例，例如 SciPy 稀疏矩阵或 Trimesh 实例；数据必须采用 PyG 可读取的形式。选择 Networkx 是因为它可以为图添加信息，而且能够轻松转换为 PyG 实例。随后，我们使用 Pan-das [89] 的数据组织功能导入数据，并将其准备好，以便传递给 Networkx，最后传递给 PyG。有关所用工具及其版本的完整列表，请参见表 5。

所有工具均在一台个人工作站上运行，该工作站配备英特尔酷睿 i9-9900K 中央处理器 @ 3.60GHz、32GB 已安装内存和英伟达 GeForce RTX 2060 Super 图形处理器。

表 5：主要工具及其版本列表。

Tool	版本
Python	3.9
NetworkX	2.8.7
PyTorch	1.12.1
PyTorch Geometric	2.1.0
SciPy	1.9.1
Pandas	1.5.0