

# 面向分布偏移的多维节点级图神经网络架构搜索

2252090 王麒懿\*, 2250420 陈君\*, 2250758 林继申\*

同济大学

**摘要** 图神经架构搜索 (GraphNAS) 在缓解图神经网络 (GNN) 在分布偏移下的性能退化方面表现出潜力。现有方法通常采用权重共享机制, 以端到端方式为每个图生成定制的 GNN 架构。然而, 这些方法往往忽视图间的分布特征, 且依赖大量训练数据, 在训练图稀缺或仅有单一图时难以学习图与架构的最优映射, 导致泛化性能不佳。本文提出了一种节点级别的图神经架构搜索框架 NodeNAS, 可为每个节点定制聚合策略, 实现节点拓扑与图分布的解耦, 从而在有限数据条件下实现灵活架构适配。在此基础上, 提出多维 NodeNAS (MNNAS), 引入基于注意力机制的自适应聚合器, 学习具备更强泛化能力的节点级架构定制器。MNNAS 通过扩展搜索空间的垂直深度, 支持在多维上同时进行节点架构定制。此外, MNNAS 对不同同配性条件下的节点度数遵循幂律分布进行建模, 从结构不变性中提取指导信号, 辅助多维架构选择。在多个有监督与无监督任务中的实证结果表明, MNNAS 显著优于现有方法, 在分布偏移场景下展现出强鲁棒性和优秀的泛化能力。

**Keywords:** 图神经网络架构搜索 · 分布偏移 · 架构定制

## 1 引言

图神经网络 (Graph Neural Networks, GNNs) 在图结构学习任务中取得了显著成功, 包括图分类、图划分与社区发现等 [40,24,36]。GNN 的核心机制是消息传递 (Message Passing), 其中每个节点通过聚合其局部邻居的信息, 逐步更新自身表示。这一机制虽使 GNN 能够捕捉复杂的关系依赖, 但也带来了根本性的脆弱性: GNN 的表示学习与预测性能高度依赖于图结构本身。因此, 在测试图分布与训练图分布不一致 (即“分布偏移”) 的情况下, GNN 模型的性能往往大幅下降。为应对分布偏移带来的挑战, 图神经架构搜索 (Graph Neural Architecture Search, GraphNAS) 成为一个有前景的研究方向。GraphNAS 通过数据驱动的搜索自动设计 GNN 架构, 使模型能够探索超越人工设计启发式的多样聚合与传播策略。早期的 GraphNAS 方法 [6,48] 将架构设计建模为离散搜索空间中的黑盒优化问题。尽管这些方法在架构灵活性上取得进展, 但大多假设训练与测试数据满足独立同分布 (IID) 假设, 而这一假设在实际图数据中常常不成立。

近期的 GraphNAS 研究通过引入可微搜索与架构权重共享等技术, 构建出更具表达力的搜索框架 [29,43]。这些方法以图整体为单位建模, 通过端到端方式动态生成图特定的 GNN 架构。它们将候选操作的选择过程松弛为可学习的软权重机制, 在一定程度上提升了模型在分布偏移下的鲁棒性。然而, 即使取得了这些进展, 当前方法仍存在几个关键局限。首先, 现有 GraphNAS 方法通常依赖大量训练数据, 才能有效建模图实例间的架构偏好差异。在训练图稀疏甚至仅有单一图的情况下, 模型难以学习出可靠的图到架构映射, 导致泛化能力不足。其次, 大多数方法在架构自适应方面采用图级粒度, 即所有节点共享同一聚合策略。这忽视了

<sup>1</sup> \* 所有作者都对本文做出了同样的贡献。

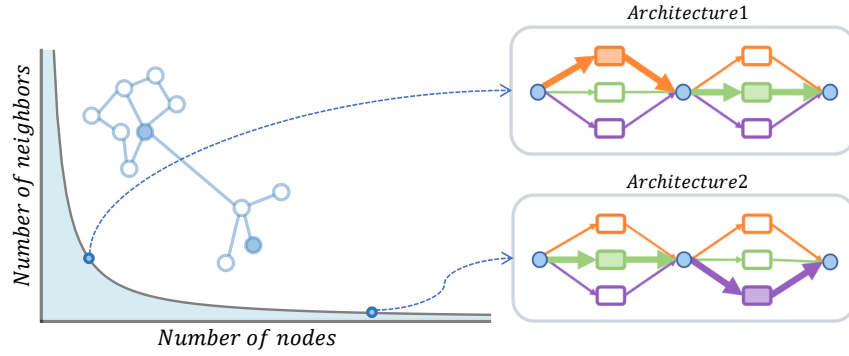


图 1: NodeNAS 的架构

现实图中一个基本现象：节点度分布呈长尾特性，即少数高阶（head）节点与大量低阶（tail）节点并存。统一的聚合函数难以适应这种结构异质性。目前尚无 GraphNAS 框架在节点粒度上实现基于结构差异的架构个性化。

为解决上述问题，本文提出 NodeNAS——一种支持节点级聚合策略定制的图神经架构搜索框架。如图 1 所示，NodeNAS 通过为每个节点学习其在候选操作上的概率分布，实现节点到架构的个性化关联。该设计使模型能够根据节点的度数或结构角色赋予其不同的聚合机制。此外，NodeNAS 显式地将节点度分布与图的全局属性（如同配性）解耦，有助于在学习节点-架构映射过程中屏蔽虚假结构模式，从而提升模型在未知分布图上的泛化能力。在此基础上，我们进一步提出多维 NodeNAS（Multi-dimension NodeNAS, MNNAS），在语义维度上扩展搜索空间，以捕捉更深层次、可迁移的结构模式。具体而言，MNNAS 构建了一个多维架构搜索网络，使每个节点可在多个独立的搜索维度（Search Dimensions, S-Dims）中被同时关联至多个候选架构。这些维度对应架构变化的潜在独立因子，从而增强模型的代表多样性。为保证参数高效性并提升泛化能力，我们将操作到嵌入的映射机制在所有 S-Dims 中共享。候选操作首先被投影至连续的嵌入空间，再通过可微的混合权重进行架构选择。此外，MNNAS 融入了由链接模式编码器引导的自适应聚合注意力机制，该机制能够感知不同同配性下的结构不变模式，支持并行进行多维节点级架构定制，同时抑制结构伪相关。最终，节点表示通过整合多个定制架构的输出而得。该设计不仅通过权重共享实现了端到端训练的高效性，也显著增强了模型对分布偏移的鲁棒性。我们还从信息论的角度，基于信息瓶颈（Information Bottleneck, IB）原理，对 MNNAS 的可解释性与泛化行为进行了理论分析。在多个有监督与无监督任务上的实验结果表明，MNNAS 在分布偏移环境下显著优于现有 GraphNAS 与 GNN 模型，验证了其多维节点级架构定制机制的有效性与先进性。

本文的主要贡献如下：

- 我们提出了一种新颖的节点级图神经架构搜索框架 NodeNAS，可为每个节点学习个性化的嵌入更新策略。不同于以往对整图分配统一架构的 GraphNAS 方法，NodeNAS 支持在节点粒度上进行细致的架构定制，使模型能够有效适应具有异质结构和未知分布属性的图数据，从而缓解因结构差异带来的性能退化。
- 我们进一步提出多维 NodeNAS（Multi-dim NodeNAS, MNNAS），对 NodeNAS 进行多语义维度扩展，支持在多个潜在语义维度上同时进行架构定制。为此，我们设计了一种自适应聚合注意力机制，显式解耦节点度数分布（遵循幂律规律）与图级同配性。所得多维

搜索网络可实现具有可解释性且性能优越的节点级架构定制，显著提升模型在分布偏移条件下的泛化能力。

- 据我们所知，MNNAS 是首个即使在极端单图训练场景下仍能实现强泛化性能的 NAS 框架。同时，它也是首个将神经架构搜索扩展至无监督图任务（如社区发现）的方法，并在多类分布外图场景中保持稳健的泛化表现。

## 2 预备知识

### 2.1 分布外泛化

给定图空间  $\mathcal{G}$  和标签空间  $\mathcal{Y}$ ，我们定义训练图数据集  $\mathcal{G}_{\text{tr}} = \{g_i\}_{i=1}^{N_{\text{tr}}}, g_i \in \mathcal{G}$ ，以及对应的标签集  $\mathcal{Y}_{\text{tr}} = \{y_i\}_{i=1}^{N_{\text{tr}}}, y_i \in \mathcal{Y}$ 。类似地，测试图数据集表示为  $\mathcal{G}_{\text{te}} = \{g_i\}_{i=1}^{N_{\text{te}}}$ ，标签集为  $\mathcal{Y}_{\text{te}} = \{y_i\}_{i=1}^{N_{\text{te}}}$ 。分布外泛化的目的是利用训练数据集  $\mathcal{G}_{\text{tr}}$  和  $\mathcal{Y}_{\text{tr}}$  学习一个模型  $F: \mathcal{G} \rightarrow \mathcal{Y}$ ，使其在测试数据集  $\mathcal{G}_{\text{te}}$  和  $\mathcal{Y}_{\text{te}}$  上表现良好，假设训练数据和测试数据的分布不一致，即  $P(\mathcal{G}_{\text{tr}}, \mathcal{Y}_{\text{tr}}) \neq P(\mathcal{G}_{\text{te}}, \mathcal{Y}_{\text{te}})$ ，其中  $P(\mathcal{G}, \mathcal{Y})$  表示图及其标签的分布。分布外泛化的目标可以表示为：

$$\arg \min_F \mathbf{E}_{\mathcal{G}, \mathcal{Y} \sim P(\mathcal{G}_{\text{te}}, \mathcal{Y}_{\text{te}})} [l(F(\mathcal{G}), \mathcal{Y}) \mid \mathcal{G}_{\text{tr}}, \mathcal{Y}_{\text{tr}}], \quad (1)$$

其中  $l: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  是一个损失函数。在本文中，我们研究一种设置，即在训练阶段无法获取测试图  $\mathcal{G}_{\text{te}}$  及其对应的标签  $\mathcal{Y}_{\text{te}}$ 。

### 2.2 可微分图神经架构搜索 (Differentiable GraphNAS)

与传统的图神经架构搜索 (GraphNAS) 方法不同，后者将选择最佳架构视为离散域中的黑箱优化问题，可微分 GraphNAS[29,43] 将离散的搜索空间松弛为连续空间，并允许通过梯度下降进行高效优化。我们定义候选操作集合为  $\mathcal{O} = \{o_1, o_2, \dots, o_K\}$ ，其中每个  $o_k \in \mathcal{O}$  表示搜索空间中的一个操作， $K$  是  $\mathcal{O}$  中操作的总数。

此外，可微分 GraphNAS 将  $\mathcal{O}$  中操作的刚性选择松弛为软选择，为每个候选操作分配一个概率。公式 (2) 展示了在架构空间中可微分搜索的一个示例。第  $l$  层的输出可以表示为：

$$\mathbf{h}_i^{(l+1)} = \sum_{o \in \mathcal{O}} p^o o(\mathbf{h}_i^{(l)}), \quad (2)$$

其中， $\mathbf{h}_i^{(l)}$  表示第  $l$  层中节点  $i$  的嵌入， $p^o$  是与相应候选操作  $o$  相关联的概率。在这一维度上，概率分布被归一化，使得  $\sum_{o \in \mathcal{O}} p^o = 1$ 。在每个图中，所有节点共享相同的概率集合，最终架构可以通过在测试过程中保留概率最高的候选操作来获得。

### 2.3 幂律分布与同配性

在许多现实世界的图中，例如社交网络和分子网络，节点的度分布通常遵循幂律分布。这种分布表明，节点具有  $k$  个连接的概率  $P(k)$  与  $k^{-\alpha}$  成正比，其中  $\alpha$  是一个正的常数：

$$P(k) \propto k^{-\alpha} \quad (3)$$

**幂律分布**反映了节点连接的异质性，其中少数节点（即枢纽节点）占据了大部分的边，而大多数节点呈现出长尾的度分布，如图 1 所示。为了最大化图神经网络 (GNNs) 的性能，可以针

对幂律分布中不同位置的节点采用不同的消息聚合机制。然而, 不同类型图的全局拓扑特征差异要求对幂律分布进行差异化处理。引入全局视角能够使模型识别当前图结构与度分布之间的虚假模式。为了在不同大小和密度的图之间一致地比较节点的连接性, 我们对度向量进行归一化。这种归一化确保每个节点的度是相对于图中观察到的最大度来考虑的。节点  $v$  的归一化度  $\hat{d}_v$  由下式给出:

$$\hat{d}_v = \frac{d_v}{\max(d)} \quad (4)$$

其中  $d_v$  是节点  $v$  的度,  $\max(d)$  是图中的最大度。这一过程保持了图中节点的相对重要性, 并提高了模型在不同拓扑和大小的图上的鲁棒性。

**同配性**是衡量图中节点倾向于与具有某些特定属性相似的其他节点相连的程度。这一指标通常揭示了重要的结构模式, 例如社交网络中个体倾向于与与自己相似的人交往。形式上, 同配性系数可以定义为:

$$q_j = \frac{jp_j}{\sum_k kp_k}, \quad (5)$$

$$\gamma = \frac{\sum_{jk} jk(e_{jk} - q_j q_k)}{\sigma_q^2}, \quad (6)$$

其中,  $p_j$  是度为  $j$  的节点比例,  $e_{jk}$  是图中连接度为  $j$  的节点和度为  $k$  的节点的边的比例,  $\sigma_q$  是  $q$  的标准差。 $\gamma$  通常介于  $-1$  到  $1$  之间, 正值的  $\gamma$  表示高度节点倾向于与其他高度节点相连, 而负值的  $\gamma$  则表明它们倾向于与低度节点相连。我们利用  $\gamma$  使 NodeNAS 能够学习具有分布偏移的图的不变表示, 从而实现搜索架构的泛化能力。

### 3 节点特定的图神经架构搜索

NodeNAS 从传统 GNN 架构中引入了范式转变。传统架构中, 每一层内所有节点都统一应用单一的嵌入更新方法。相比之下, NodeNAS 提出了一种灵活且自适应的框架, 能够根据给定图中每个节点的具体需求动态调整合适的节点信息聚合方法。

具体而言, 给定一个图  $\mathcal{G} = \{V, E\}$ , 其中  $V$  表示节点集合,  $E$  表示边集合, NodeNAS 的目标是学习一个架构映射函数  $\Phi: \mathcal{G} \rightarrow \mathcal{A} \times W_{\mathcal{A}}$ , 其中  $\mathcal{A}$  表示每个节点的架构, 即  $\mathcal{A} = \{A_1, A_2, \dots, A_N\}$ ,  $W_{\mathcal{A}}$  表示对应的权重。 $N = |V|$  表示图  $\mathcal{G}$  中的节点数量。

为了确保 NodeNAS 的可微性, 我们引入了权重共享机制, 并为每个节点分配一个概率向量  $\mathbf{p}$ 。 $\mathbf{p}$  表示不同操作 (如 *GATConv*) 被应用于该节点的概率。定义候选操作集合为  $\mathcal{O} = \{o_1, o_2, \dots, o_K\}$ , 其中每个  $o_k \in \mathcal{O}$  表示搜索空间中的一个操作,  $K$  是操作的总数。 $A_i$  可以进一步表示为  $A_i = \{(o, p_i^o)\}$ , 其中  $o$  是满足  $o \in \mathcal{O}$  的候选操作,  $p_i^o$  是对应的概率。在第  $l$  层中, 节点  $i$  的嵌入更新可以表示为:

$$\mathbf{h}_i^{(l+1)} = \sum_{o \in \mathcal{O}} p_i^o o(\mathbf{h}_i^{(l)}), \quad (7)$$

其中,  $\mathbf{h}_i^{(l)}$  表示第  $l$  层中节点  $i$  的嵌入。操作  $o$  的权重在不同图之间共享, 这确保了每个图的节点特定架构可以端到端地进行定制, 同时通过梯度下降实现高效优化。

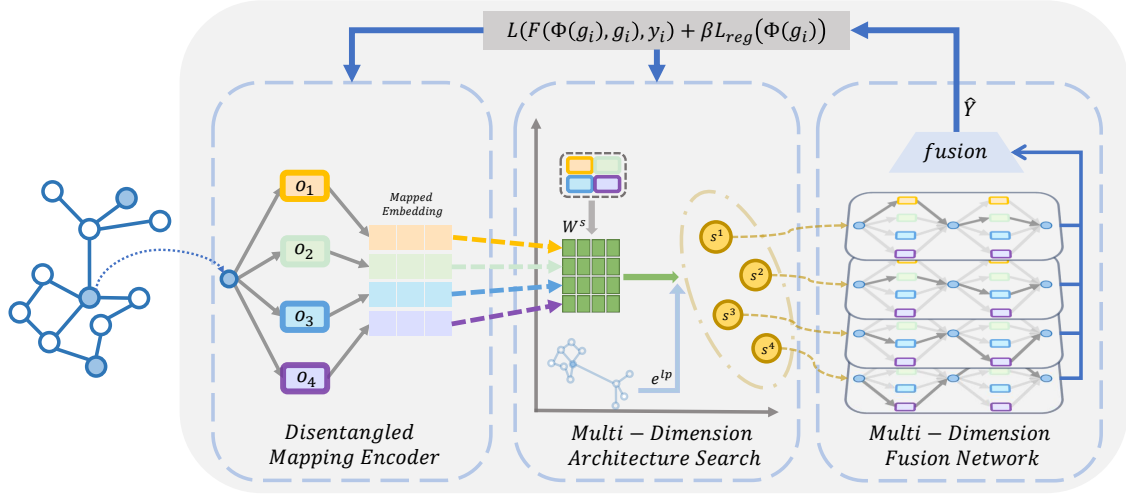


图 2: MNNA 模型的架构图。

## 4 基于自适应聚合注意力的多维节点级图神经架构搜索

### 4.1 框架

在我们提出的方法中，通过最大化从有限数据集中学习节点与图分布之间的内在关系，为每个图定制独特的节点特定架构。特别是，我们的方法支持在多个维度上搜索架构，从而在架构定制中更灵活地表达学习到的解耦信息。

具体而言，我们旨在学习一个架构映射函数  $\Phi: G \rightarrow A \times \mathcal{W}_A$ 。与第 3 节不同，节点  $i$  的  $A_i$  包含由多个搜索维度组成的架构集合，即  $A_i = \{A_i^1, A_i^2, \dots, A_i^Z\}$ ，其中  $Z$  是搜索维度的数量（S-Dims）， $A_i^z = \{(o_k, p^z, o_k)\}$  表示在第  $z$  个 S-Dim 中搜索到的架构。对于任意  $z$ ， $|A_i^z| = |\mathbf{p}| = K$  始终成立，这表明每个 S-Dim 中的搜索空间都包含完整的候选操作集合，从而使得模型能够为不同图中的不同节点可微地学习架构偏好。 $\Phi$  可以进一步分解为每个 S-Dim 的映射集合，即  $\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_Z\}$ ，每个  $\Phi_z$  将图映射到第  $z$  个 S-Dim 中的合适架构  $A_z$  及对应的权重  $\mathcal{W}_{A_z}$ 。因此，公式 1 可以转化为以下形式：

$$\min_{\Phi \in S_p} \sum_{(g_i, y_i) \in \mathcal{G}_{tr}} [\mathcal{L}(F(\sum_{z=1}^Z \Phi_z(g_i), g_i), y_i) + \beta \mathcal{L}_{reg}(\Phi(g_i))], \quad (8)$$

其中， $\beta$  是一个超参数， $\mathcal{L}_{reg}(\Phi(g_i))$  是架构的正则化项。 $S_p$  表示搜索空间，是一个由操作平面和概率平面组成的二维空间， $F(\cdot)$  用于在定制的架构和权重下为下游任务获取输出。

对于每个  $g_j \in G_{te}$ ，我们利用训练好的  $\Phi$  为其生成独特的架构，使其在分布偏移下表现良好。特别是对于某些无监督任务（如社区检测和图划分）， $\mathcal{G}_{tr}$  只包含一个图。

### 4.2 解耦映射编码器

不同图具有独特的分布，这给依赖局部节点聚合的 GNN 带来了挑战，往往导致其丢失全局结构信息，进而学习到虚假模式并表现出较差的泛化能力。为解决这一问题，我们在每一层的节点表示中融入全局上下文，以捕捉局部和全局特征，从而提高模型的鲁棒性。

编码器的目标是快速聚合整体语义特征。然而，在 GNN 中，信息主要通过边传播，需要一个深度卷积网络才能捕获全局信息。此外，多层固定的 GNN 架构加剧了输入特征与邻接

矩阵之间的非线性依赖关系, 这种依赖关系是由于分布中的虚假模式引起的。因此, 我们的编码器仅以特征作为输入, 并手动聚合全局语义信息, 以加速节点之间的信息传播。具体设计如下:

$$\mathbf{h}_i^{(l)} = \text{ENCODER}(\mathbf{h}_i^{(l-1)} + \eta \text{ LINEAR}(\frac{1}{N} \sum_{i=1}^N \mathbf{h}_i^{(l-1)})), \quad (9)$$

其中,  $\eta$  是一个初始超参数,  $\mathbf{h}_i^{(l)}$  是第  $l$  层中节点  $i$  的嵌入,  $\mathbf{h}_i^{(0)}$  初始化为节点  $i$  的特征  $\mathbf{x}_i$ 。

在获得节点嵌入后, 我们将候选操作映射为每个节点的可学习嵌入。具体而言, 对于  $\mathcal{O}$  中的每个操作  $o$ , 我们学习一个对应的嵌入  $\mathbf{e}_i^o$ 。在架构搜索过程中, 我们将  $\mathcal{O}$  替换为由映射嵌入组成的集合  $E_i$ 。操作到嵌入的映射表示为  $\Phi_{\mathcal{O} \rightarrow E}$ , 可以表示为:

$$E_i = [\mathbf{e}_i^1, \dots, \mathbf{e}_i^K] = [o_1(\mathbf{h}_i^{(l)}), \dots, o_K(\mathbf{h}_i^{(l)})], \quad (10)$$

其中,  $\mathbf{e}_i^k$  表示第  $l$  层中节点  $i$  对应操作  $o_k$  的映射嵌入,  $k$  是候选操作的数量。这里省略了  $l$  以简化表达。

$\Phi_{\mathcal{O} \rightarrow E}$  使得模型能够在单维度计算复杂度下进行多维度搜索。这是因为映射在不同的 S-Dim 中是共享的, 即在第  $a$  个 S-Dim 和第  $b$  个 S-Dim 中, 始终有  $\mathbf{e}_{i(\dim_a)}^k = \mathbf{e}_{i(\dim_b)}^k = \mathbf{e}_i^k$ 。每个操作可以同时参与多个 S-Dim 的架构搜索, 但仅计算一次。

此外, 为了避免模式坍塌 (即不同操作的映射嵌入在训练过程中趋于不可区分), 我们引入了一个正则化项, 利用余弦距离来保持映射嵌入的多样性:

$$L_{\cos} = \sum_i \sum_{\substack{o, o' \in \mathcal{O} \\ o \neq o'}} \frac{\mathbf{e}_i^o \cdot \mathbf{e}_i^{o'}}{\|\mathbf{e}_i^o\|_2 \|\mathbf{e}_i^{o'}\|_2}, \quad (11)$$

其中,  $\mathbf{e}_i^o$  和  $\mathbf{e}_i^{o'}$  分别表示节点  $i$  对应操作  $o$  和  $o'$  的嵌入。  $L_{\cos}$  确保了  $\Phi_{\mathcal{O} \rightarrow E}$  的正交性, 有助于在图中针对性地编码解耦信息。

### 4.3 多维度架构搜索

我们提出了一种多维度架构定制方法, 将每个图映射到多个节点特定的 GNN 架构中, 即为每个节点定制多组  $A_i^k$ 。该方法包括两个组件: 链接模式编码器 (link pattern encoder), 用于学习遵循幂律分布的不同图分布中的结构不变信息; 以及自适应聚合注意力机制 (adaptive aggregation attention), 引导模型同时在多个 S-Dims 中搜索架构。

**链接模式编码器:** 在有限的数据集中, 节点特定的架构搜索能够更好地表达解耦信息, 而多维度架构则进一步放大了 NodeNAS 带来的泛化能力。然而, 捕捉隐藏在分布中的这种解耦信息是一个挑战。在处理训练集中的虚假相关性时, 区分相似图中不同度节点的偏好, 以及不同图中相似度节点的偏好至关重要。

为了解决这些问题, 我们将不同节点的度分布纳入链接模式编码器, 并使用同配性系数 (assortativity) 来量化整体图结构, 从而促进节点和图在拓扑上的解耦。具体而言, 链接模式编码器可以表示为:

$$\mathbf{e}_i^{lp} = \text{ENCODER}(\gamma_g, \frac{d_i^2}{\bar{d}^2}, \frac{d_i}{\bar{d}}, \frac{1}{|E|} \sum_{(a,b) \in E} d_a d_b), \quad i \in g, \quad (12)$$

其中,  $d_a$  和  $d_b$  分别是节点  $a$  和  $b$  的度, 它们通过边相连;  $\bar{d}$  和  $\bar{d}^2$  分别是图  $g$  中所有节点的平均度和平均平方度, 节点  $i$  位于图  $g$  中。

特别地, 我们将同配性系数近似为度统计量的函数, 即重写公式 13 如下:

$$\gamma_g \approx \frac{\frac{1}{|E|} \sum_{(a,b) \in E} d_a d_b - \left[ \frac{1}{|E|} \sum_{(a,b) \in E} \frac{1}{2} (d_a + d_b) \right]^2}{\frac{1}{|E|} \sum_{(a,b) \in E} \frac{1}{2} (d_a^2 + d_b^2) - \left[ \frac{1}{|E|} \sum_{(a,b) \in E} \frac{1}{2} (d_a + d_b) \right]^2}, \quad (13)$$

其中, 分子表示连接节点的度的实际差异与期望差异, 分母表示度分布的统计特性。

**自适应聚合注意力机制:** 自适应聚合注意力机制结合了链接模式向量  $\mathbf{e}^{lp}$ , 并为每个 S-Dim 中的候选操作搜索概率值。具体而言, 对于第  $z$  个 S-Dim 中的节点  $i$ ,  $\mathcal{O}$  中不同操作的概率可以按以下方式计算:

$$\mathbf{s}_i^z = \mathbf{e}_i^{lp} \odot \frac{\mathbf{e}_i^z W^s(E_i)^\top}{\sqrt{d}}, \quad p_i^{z,o} = \frac{\exp(s_i^{z,o})}{\sum_{o' \in \mathcal{O}} \exp(s_i^{z,o'})}, \quad (14)$$

其中,  $\odot$  表示 Hadamard 乘积,  $\mathbf{s}_i^z$  表示操作偏好向量, 代表架构子空间, 即候选操作在第  $z$  个 S-Dim 上的投影。  $p_i^{z,o}$  表示节点  $i$  在第  $z$  个 S-Dim 中操作  $o$  的概率。  $d = |\mathbf{e}_i^z|$  是公式 10 中映射嵌入的维度,  $W^s$  是权重矩阵, 满足  $W^s \in \mathbb{R}^{d \times d}$ 。公式 14 约束了搜索维度的数量必须始终等于候选操作集合  $\mathcal{O}$  中操作的数量, 即  $|\mathcal{S}_i| = Z = |\mathcal{O}| = K$ 。注意力机制间接提供了归一化约束, 防止在有限数据集的背景下, 由于无限制扩展搜索维度而引发的优化困境。

自适应聚合注意力机制利用链接模式编码器解耦节点特征与特定图分布之间的关系, 并进一步量化  $E_i$  中每个映射嵌入的重要性。本质上, 该机制基于  $att \leftarrow q \times k$ , 以不同的映射嵌入为中心, 最大化通过各种操作学习到的相似表示的概率。

#### 4.4 多维度融合网络

我们将从不同 S-Dims 搜索到的架构整合到一个连续空间中, 综合考虑所有 S-Dims 中的架构来学习最终的节点表示。多维度架构融合后的节点表示可以按以下方式计算:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\frac{1}{Z} \sum_{z=1}^Z \sum_{o \in \mathcal{O}} p_i^{z,o} o(\mathbf{h}_i^{(l)})\right) + \mathbf{h}_i^{(l)}, \quad (15)$$

其中,  $\mathbf{h}_i^{(l)}$  表示第  $l$  层中节点  $i$  的初始嵌入,  $\mathbf{h}_i^{(l+1)}$  表示输出的节点表示,  $\sigma$  表示激活函数。

由于映射函数  $\Phi_{\mathcal{O} \rightarrow E}$  的存在, 我们在每个 S-Dim 中的  $S_p$  中为映射嵌入分配概率, 而不是为操作分配概率, 从而避免了冗余计算和不必要的可学习参数。公式 15 中使用了快捷连接, 以防止过拟合并减轻无关特征的影响。

我们的方法通过在图之间共享权重并保持连续架构, 消除了 NAS 方法中常见的重新训练和架构离散化步骤。这使得模型能够通过共享的网络参数进行端到端执行, 有效地创建了一个绕过单独架构特定训练的集成模型。

#### 4.5 理论分析

在本节中, 我们将从信息瓶颈理论 (information bottleneck theory) 的视角, 对自适应聚合注意力机制如何增强模型泛化能力进行理论分析。

考虑节点及其操作映射集合为, 对应的输出表示为。映射函数可表示为。假设, 其中为含噪声的输入变量, 为目标不变特征, 为高斯噪声方差。信息瓶颈原理可表示为:

$$f_{IB}(Z_i|E_i) = \arg \min_{f(Z_i|E_i)} I(E_i, Z_i) - I(Z_i, E'_i), \quad (16)$$

其中表示互信息。

基于公式 (14), 我们定义注意力得分为:

$$S_i = Q(E_i)K^\top(E_i, \mathbf{e}^{lp}), \quad (17)$$

其中, 我们将参数分解为, 注意力机制表示为:

$$Z_i = \sum_{o \in O} \frac{\exp(\mathbf{e}_i^z W^q)(\mathbf{e}_i^o (e_{i,o}^{lp} W^o)^\top)}{\sum_{o' \in O} \exp(\mathbf{e}_i^z W^q)(\mathbf{e}_i^{o'} (e_{i,o'}^{lp} W^o)^\top)} \mathbf{e}_i^o = \sum_{o \in O} p_i^{z,o} \mathbf{e}_i^o, \quad (18)$$

依据文献 [47,8], 我们可进一步阐明公式 (18) 与公式 (16) 的内在联系。将信息瓶颈原理表达为聚类问题, 如节点的局部图结构, 其概率更新迭代过程可描述为:

$$p_t(c|i) = \frac{\log p_{t-1}(c)}{Z(i)} \exp[-D_{KL}[p(E_i|i)|p_{t-1}(E_i|c)]], \quad (19)$$

其中表示 KL 散度。

假设, 我们有:

$$D_{KL}[p(E_i|i)|p_{t-1}(E_i|c)] \propto (\mathbf{e}_i - \mu_t - \mathbf{1}^{(c)})^\top [\Sigma_{t-1}^{(c)}]^{-1} (\mathbf{e}_i - \mu_t - \mathbf{1}^{(c)}). \quad (20)$$

通过将此表达式代入迭代方程, 可得到自适应聚合注意力形式:

$$\mu_c^{(t)} = \sum_{i=1}^n \frac{\exp(2[\mu_{t-1}^{(c)}]^\top [\Sigma_{t-1}]^{-1} \mathbf{e}_i)}{\sum_{c'} \exp(2[\mu_{t-1}^{(c')}]^\top [\Sigma_{t-1}]^{-1} \mathbf{e}_i)} \mathbf{e}_i. \quad (21)$$

记, 可得最终的注意力表示形式为:

$$Z_i = \sum_{o \in O} \frac{\exp(Q(E_i, m_{e_i}^o)^\top K(E_i, E_{lp}, m_{e_i}^o))}{\sum_{o \in O} \exp(Q(E_i, m_{e_i}^o)^\top K(E_i, E_{lp}, m_{e_i}^o))} m_{e_i}^o. \quad (22)$$

由此可见, 自适应聚合注意力机制与信息瓶颈原理密切相关, 通过优化公式 (16) 的目标, 显式学习节点的泛化表示, 并增强了 GNAS 模型的泛化能力。这种机制允许模型更有效地捕获并保持跨数据分布的不变节点特征, 进而提高模型的泛化性能。

#### 4.6 复杂度分析

设  $|V|$  和  $|E|$  分别表示图中的节点数和边数。我们定义  $d_i$  为输入特征的维度,  $d_e$  为初始节点嵌入的维度,  $d_m$  为不同操作在  $\mathcal{O}$  中的映射嵌入的维度,  $d_o$  为输出的维度。

**可学习参数数量:** 在我们的框架中, 节点编码器模块包含  $O(2d_i d_e)$  个参数, 学习映射嵌入的模块包含  $O(|\mathcal{O}| d_e d_m)$  个可学习参数, 链接模式编码器包含  $4|\mathcal{O}|$  个参数, 自适应聚合注意力机制包含  $O(d_m^2)$  个参数, 多维度融合网络没有可学习参数。最终输出层包含  $O(d_m d_o)$  个参数。对于一个  $\eta$  层网络, 可学习参数的总数为:

$$O(2d_i d_e + d_m d_o + \eta(d_m^2 + |\mathcal{O}|(4 + d_e d_m))).$$



表 1: 数据集统计信息。

Dataset	Motif	hiv	bace	sider	Cora	Cite	Pub	BA	ER	RR	NW
Setting	Sup.	Sup.	Sup.	Sup.	Unsup.	Unsup.	Unsup.	Unsup.	Unsup.	Unsup.	Unsup.
Task	Graph	Graph	Graph	Graph	Node	Node	Node	Node	Node	Node	Node
Graphs	18,000	41,127	1,513	1,427	1	1	1	1	1	1	1
Avg. Nodes	26.1	25.5	34.1	33.6	2,708	3,327	19,717	10,000	10,000	10,000	10,000
Avg. Edges	36.3	27.5	36.9	35.4	10,556	9,104	88,648	99,950	99,950	100,000	22,092
Classes	3	2	2	2	7	6	3	-	-	-	-

**时间复杂度：**节点编码器的时间复杂度为  $O(|V|d_id_e)$ 。对于映射嵌入模块，其时间复杂度为  $O(|V||\mathcal{O}|d_ed_m)$ 。多维度架构搜索模块的时间复杂度为  $O(|V||\mathcal{O}|(d_m^2 + |\mathcal{O}|^2))$ ，多维度融合模块的时间复杂度为  $O(|V||\mathcal{O}|^2d_m)$ 。输出层的时间复杂度为  $O(|V|d_md_o)$ 。此外， $L_{\cos}$  的时间复杂度为  $O(|V||\mathcal{O}|^2d_e)$ 。

由于不同的 S-Dims 通过  $\Phi$  共享操作权重，并且考虑到  $|\mathcal{O}|$  是一个小常数，多维度搜索的时间复杂度与单维度搜索的时间复杂度相当，因此不会增加额外的训练成本。因此，总时间复杂度可以简化为：

$$O(|V|(d_id_e + |\mathcal{O}|d_ed_m + |\mathcal{O}|d_m^2 + d_md_o)).$$

## 5 实验

在本节中，我们报告实验结果以验证我们模型的有效性。

**数据集：**我们构建了合成数据集和真实世界数据集，以评估 MNNAS 在监督图分类以及无监督社区检测和逆图划分任务中的性能，特别是在涉及分布偏移的条件下。

对于图分类任务，我们使用了集成基础形状和模式形状的合成数据集 Spurious-Motif(Motif)，以及用于分子性质预测的 OGBG-Mol 数据集（包括 hiv、bace、sider）。社区检测任务使用了 Cora、CiteSeer 和 PubMed 数据集。逆图划分任务则使用了合成数据集，包括 Erdős-Rényi (ER)、随机正则 (RR)、Barabási-Albert (BA) 和 Newman-Watts-Strogatz (NW) 图。

**基线模型：**我们将我们的模型与以下基线模型进行比较。

1. **优秀的 GNN 模型：**常用的手动设计的 GNN 模型包括 GCN[13]、GAT[33]、GIN[40]、GraphSAGE[9]、GraphConv[23]、GAP[24] 和 ClusterNet[36]。最近提出的方法，如 ASAP[30]、DIR[37]、PNA[3] 和 GSAT[21]，已在具有分布外 (OOD) 设置的图级任务中展现出强大的性能。

2. **优秀的 NAS 模型：**我们评估了六个先进的 NAS 基线模型，包括 DARTS[20] 和五个基于 GraphNAS 的算法：GraphNAS[6]、PAS[35]、GASSO[28]、GRACES[29] 和 DCGAS[43]，其中 DCGAS 是当前图神经架构搜索 (GNAS) 领域的最新技术。

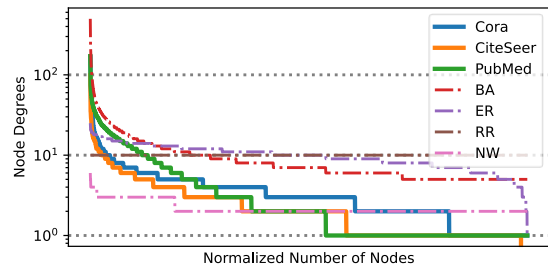


图 3: 数据集的度分布。

表 2: 在 swious-motif 上测试准确性, 并在 OGBG-mol 上测试 AUC-ROC。

Method	Spurious-Motif (Accuracy)			OGBG-Mol (AUC-ROC)		
	b=0.7	b=0.8	b=0.9	hiv	sider	bace
GCN	48.39 $\pm$ 1.69	41.55 $\pm$ 3.88	39.13 $\pm$ 1.76	75.99 $\pm$ 1.19	59.84 $\pm$ 1.54	68.93 $\pm$ 6.95
GAT	50.75 $\pm$ 4.89	42.48 $\pm$ 2.46	40.10 $\pm$ 5.19	76.80 $\pm$ 0.58	57.40 $\pm$ 2.01	75.34 $\pm$ 2.36
GIN	36.83 $\pm$ 5.49	34.83 $\pm$ 3.10	37.45 $\pm$ 3.59	77.07 $\pm$ 1.49	57.57 $\pm$ 1.56	73.46 $\pm$ 5.24
SAGE	46.66 $\pm$ 2.51	44.50 $\pm$ 5.79	44.79 $\pm$ 4.83	75.58 $\pm$ 1.40	56.36 $\pm$ 1.32	74.85 $\pm$ 2.74
GraphConv	47.29 $\pm$ 1.95	44.67 $\pm$ 5.88	44.82 $\pm$ 4.84	74.46 $\pm$ 0.86	56.09 $\pm$ 1.06	78.87 $\pm$ 1.74
ASAP	54.07 $\pm$ 13.85	48.32 $\pm$ 12.72	43.52 $\pm$ 8.41	73.81 $\pm$ 1.17	55.77 $\pm$ 1.18	71.55 $\pm$ 2.74
DIR	50.08 $\pm$ 3.46	48.22 $\pm$ 6.27	43.11 $\pm$ 5.43	77.05 $\pm$ 0.57	57.34 $\pm$ 0.36	76.03 $\pm$ 2.20
DARTS	50.63 $\pm$ 8.90	45.41 $\pm$ 7.71	44.44 $\pm$ 4.42	74.04 $\pm$ 1.75	60.64 $\pm$ 1.37	76.71 $\pm$ 1.83
GraphNAS	55.18 $\pm$ 18.62	51.64 $\pm$ 19.22	37.56 $\pm$ 5.43	-	-	-
PAS	52.15 $\pm$ 4.35	43.12 $\pm$ 5.95	39.84 $\pm$ 1.67	71.19 $\pm$ 2.28	59.31 $\pm$ 1.48	76.59 $\pm$ 1.87
GRACES	65.72 $\pm$ 17.47	59.57 $\pm$ 17.37	50.94 $\pm$ 8.14	77.31 $\pm$ 1.00	61.85 $\pm$ 2.56	79.46 $\pm$ 3.04
DCGAS	87.68 $\pm$ 6.12	75.45 $\pm$ 17.40	61.42 $\pm$ 16.26	<b>78.04<math>\pm</math>0.71</b>	63.46 $\pm$ 1.42	81.31 $\pm$ 1.94
<b>MNNAS</b>	<b>97.53<math>\pm</math>3.65</b>	<b>98.42<math>\pm</math>1.65</b>	<b>93.19<math>\pm</math>6.17</b>	76.55 $\pm$ 3.04	<b>65.46<math>\pm</math>1.18</b>	<b>84.69<math>\pm</math>3.67</b>

### 5.1 合成数据集与真实数据集上的图分类

**实验设置**对于 Spurious-Motif 数据集, 我们遵循了 [29] 中描述的实验设置。此外, 我们还在真实数据集 OGBG-Mol 上进行了实验。每个实验均使用不同的随机种子重复了十次, 结果以平均值和标准差的形式呈现。

**定性结果:** 表 2 显示, 我们的模型在合成数据集和真实数据集上均优于基线方法。传统 GNN 在合成数据集上表现不佳, 这是由于虚假相关性和分布偏移导致的。在真实数据集中, GNN 的有效性因图的特性而异。尽管 NAS 方法在手动设计的 GNN 基础上略有改进, 但在面对分布变化时仍面临挑战。相比之下, MNNAS 有效减少了图分布中的虚假相关性, 显著提升了性能, 尤其是在合成数据集上。

### 5.2 社区检测

**实验设置**我们在真实数据集上评估社区检测的性能。根据 [36] 中描述的方法, 每个数据集被划分为 10 个社区。为了在不同数据集中统一特征维度, 我们使用非负矩阵分解 (Non-negative Matrix Factorization) 代替原始节点特征进行训练。这种方法使得模型能够在不同图环境中进行训练和测试, 从而检验模型在多样化图环境中的泛化能力。

**定性结果**表 3 显示, 手动设计的 GNN 模型在测试阶段的泛化性能显著下降。这些模型通常在训练数据集上表现良好, 但在测试数据集上无法保持其有效性。类似的趋势也出现在可微分 NAS 方法中, 这表明在整个图中应用统一的 GNN 方法会因节点偏好的差异而降低泛化能力。相比之下, 我们的模型在训练和测试阶段均表现出色。其优异的性能主要归功于节点特定的架构搜索能力, 这使得模型能够有效分离出特定于节点的不变特征。

表 3: 在现实世界数据集上测试模块度。

Method	Cora <sub>tr</sub>	Cite <sub>te</sub>	Pub <sub>te</sub>	Cite <sub>tr</sub>	Cora <sub>te</sub>	Pub <sub>te</sub>	Pub <sub>tr</sub>	Cite <sub>te</sub>	Cora <sub>te</sub>
GCN	0.65 $\pm$ 0.02	0.56 $\pm$ 0.01	0.50 $\pm$ 0.02	0.65 $\pm$ 0.01	0.58 $\pm$ 0.02	0.50 $\pm$ 0.02	0.64 $\pm$ 0.01	0.55 $\pm$ 0.03	0.54 $\pm$ 0.02
GAT	0.59 $\pm$ 0.03	0.52 $\pm$ 0.05	0.42 $\pm$ 0.04	0.61 $\pm$ 0.04	0.50 $\pm$ 0.05	0.48 $\pm$ 0.02	0.60 $\pm$ 0.02	0.52 $\pm$ 0.01	0.49 $\pm$ 0.04
GIN	0.58 $\pm$ 0.08	0.52 $\pm$ 0.06	0.45 $\pm$ 0.03	0.66 $\pm$ 0.03	0.52 $\pm$ 0.03	0.41 $\pm$ 0.05	0.56 $\pm$ 0.08	0.53 $\pm$ 0.07	0.49 $\pm$ 0.08
SAGE	0.60 $\pm$ 0.03	0.47 $\pm$ 0.02	0.44 $\pm$ 0.03	0.64 $\pm$ 0.03	0.51 $\pm$ 0.05	0.50 $\pm$ 0.03	0.60 $\pm$ 0.03	0.48 $\pm$ 0.04	0.48 $\pm$ 0.03
GraphConv	0.57 $\pm$ 0.03	0.42 $\pm$ 0.03	0.41 $\pm$ 0.04	0.45 $\pm$ 0.22	0.30 $\pm$ 0.16	0.29 $\pm$ 0.15	0.53 $\pm$ 0.02	0.41 $\pm$ 0.03	0.39 $\pm$ 0.03
MLP	0.64 $\pm$ 0.02	0.48 $\pm$ 0.03	0.28 $\pm$ 0.03	0.66 $\pm$ 0.04	0.47 $\pm$ 0.05	0.30 $\pm$ 0.06	0.58 $\pm$ 0.04	0.44 $\pm$ 0.05	0.34 $\pm$ 0.03
ClusterNet	0.59 $\pm$ 0.02	0.56 $\pm$ 0.06	0.42 $\pm$ 0.04	0.70 $\pm$ 0.03	0.46 $\pm$ 0.01	0.31 $\pm$ 0.06	0.61 $\pm$ 0.02	0.58 $\pm$ 0.06	0.47 $\pm$ 0.03
DARTS	0.66 $\pm$ 0.02	0.53 $\pm$ 0.03	0.42 $\pm$ 0.06	0.67 $\pm$ 0.02	0.53 $\pm$ 0.03	0.47 $\pm$ 0.02	0.62 $\pm$ 0.04	0.54 $\pm$ 0.02	0.51 $\pm$ 0.03
GASSO	0.63 $\pm$ 0.03	0.58 $\pm$ 0.04	0.52 $\pm$ 0.03	0.68 $\pm$ 0.03	0.57 $\pm$ 0.04	<b>0.53<math>\pm</math>0.04</b>	0.66 $\pm$ 0.04	0.61 $\pm$ 0.03	0.59 $\pm$ 0.03
MNNAS	<b>0.69<math>\pm</math>0.02</b>	<b>0.63<math>\pm</math>0.02</b>	<b>0.53<math>\pm</math>0.01</b>	<b>0.72<math>\pm</math>0.02</b>	<b>0.60<math>\pm</math>0.01</b>	0.52 $\pm$ 0.03	<b>0.68<math>\pm</math>0.01</b>	<b>0.61<math>\pm</math>0.01</b>	<b>0.61<math>\pm</math>0.02</b>

表 4: 合成数据集上子图间的割边比。

Method	BA1000 <sub>tr</sub>	BA10000 <sub>te</sub>	RR10000 <sub>te</sub>	ER10000 <sub>te</sub>	NW10000 <sub>te</sub>
GCN	0.95 $\pm$ 0.01	0.86 $\pm$ 0.10	0.87 $\pm$ 0.07	0.87 $\pm$ 0.06	0.81 $\pm$ 0.11
GAT	0.95 $\pm$ 0.01	0.87 $\pm$ 0.05	0.75 $\pm$ 0.07	0.75 $\pm$ 0.07	0.72 $\pm$ 0.09
SAGE	<b>0.99<math>\pm</math>0.00</b>	0.95 $\pm$ 0.01	0.92 $\pm$ 0.04	0.90 $\pm$ 0.05	0.80 $\pm$ 0.09
GIN	<b>0.99<math>\pm</math>0.00</b>	0.92 $\pm$ 0.00	0.80 $\pm$ 0.05	0.81 $\pm$ 0.04	0.66 $\pm$ 0.09
GraphConv	<b>0.99<math>\pm</math>0.00</b>	0.94 $\pm$ 0.01	0.87 $\pm$ 0.05	0.87 $\pm$ 0.04	0.81 $\pm$ 0.04
MLP	0.98 $\pm$ 0.00	0.92 $\pm$ 0.01	0.91 $\pm$ 0.01	0.90 $\pm$ 0.01	0.76 $\pm$ 0.06
GAP	0.96 $\pm$ 0.01	0.91 $\pm$ 0.02	0.90 $\pm$ 0.02	0.90 $\pm$ 0.02	0.81 $\pm$ 0.10
MNNAS	<b>0.99<math>\pm</math>0.00</b>	<b>0.96<math>\pm</math>0.00</b>	<b>0.98<math>\pm</math>0.00</b>	<b>0.97<math>\pm</math>0.00</b>	<b>0.84<math>\pm</math>0.05</b>

### 5.3 逆图划分

**实验设置**为了评估模型在典型幂律分布数据集之外的泛化能力，我们使用了四种具有不同分布的合成图：BA、ER、RR 和 NW。训练在包含 1,000 个节点的 BA 图上进行，而测试则使用包含 10,000 个节点的更大图进行。每个图被划分为 10 个子图，以最大化子图之间的边数并最小化子图内的边数。

**定性结果**表 4 的结果表明，使用单一 GNN 在具有不同节点分布的数据集之间进行泛化会显著降低性能。这是由于模型需要在全局范围内适应不同的图分布，同时在局部适应不同结构的节点特征。我们提出的 MNNAS 模型通过引入链接模式编码器解决了这一挑战，该编码器能够解耦不变的结构因素，从而在节点级别实现定制化的架构定制。

### 5.4 可解释性

在图 4 中，我们展示了四个数据集中节点的架构偏好，这些节点根据其度从高到低被分为五组。对于 Cora 数据集，高度节点倾向于线性架构，而低度节点更喜欢 GINConv。在 Citeseer 数据集中，随着度分布的变化，架构偏好从 GCNConv 转变为 GraphConv；Pubmed 和 BA 数据集也观察到了类似的模式。这些结果表明，每个图中的节点表现出受其度影响的不同架

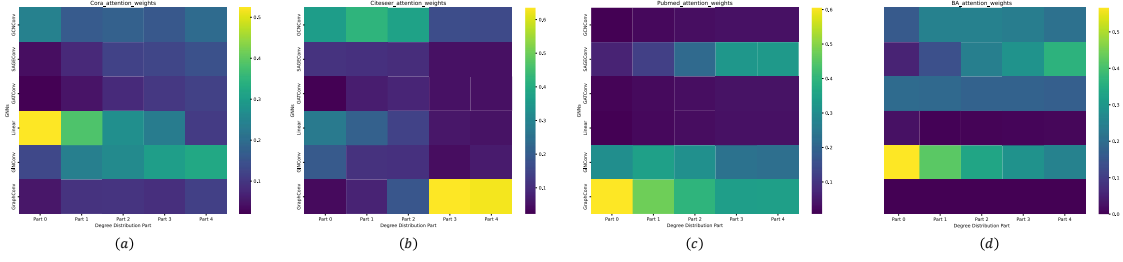


图 4: (a)-(d) 展示了 MNNAS 在不同数据集的度分布中统计得出的架构定制模式

构偏好, 突显了幂律分布的重要性。此外, 不同图中的节点表现出不同的偏好, 这进一步证明了在区分不同图中的幂律分布时, 引入图级别的拓扑特征 (如同配性) 的有效性。

## 6 相关工作

### 6.1 图神经架构搜索 (Graph Neural Architecture Search)

近年来, 神经架构搜索 (Neural Architecture Search, NAS) 在自动化发现最优模型结构方面展现出巨大潜力, 有效减轻了人工调参负担, 并能在无需丰富专家经验的前提下找到性能优越的解决方案。已有多种搜索策略被提出以快速定位最佳设计方案, 包括基于强化学习的 NAS 方法 [49,27,2,1], 基于进化算法的 NAS 方法 [22,31,19,38], 以及基于贝叶斯优化的 NAS 方法 [12]。ENAS [27] 首次提出参数共享的概念, 指出所有结构共享权重可显著提高搜索效率。随后, 一些研究将离散搜索转化为可微搜索, 以实现端到端加速, 如 DARTS [20] 和 SNAS [39], 它们在搜索空间中对各个操作赋予可学习的权重, 并融合了优秀的 GNN 架构操作。

受 NAS 成功经验启发, 图神经架构搜索 (GraphNAS) 近年来受到广泛关注, 涌现出一系列代表性成果。GraphNAS [6] 首次引入强化学习策略, 以汇聚优秀 GNN 架构构建搜索空间。AGNN [48] 借助循环神经网络控制器, 消除了聚合函数变化带来的架构搜索噪声。MoHINRec [45] 提出一种简化搜索空间并应用于推荐系统的 GraphNAS 方法, 显著提升了推荐准确性。此外, 一些研究致力于优化搜索流程, 如基于进化算法的架构搜索 [32], 通过生成多样性结构提升多样性; 随机搜索方法 [44], 以等概率采样多个设计方案; 以及一系列其他优秀方法 [4,25,18]。

值得注意的是, PDNAS [46] 首次将可微搜索应用于 GraphNAS, 借助 Gumbel-Sigmoid 技术将离散搜索空间扩展为连续空间, 对本工作具有启发意义。近年来, 针对图分类任务的 GraphNAS 方法也受到广泛研究, 如 [35,11,26], 已被成功应用于蛋白分子等化学属性分类场景。GRACES [29] 是首个考虑分布偏移问题的 GraphNAS 方法, 通过引入解耦编码器对非独立同分布图进行架构自适应定制。

### 6.2 GNN 泛化能力 (GNN Generalization)

图神经网络在表示学习中对结构具有高度依赖性, 因而难以在未知图结构下实现有效泛化, 尤其在非独立同分布图中表现不佳 [10,14]。近年来, 一些前沿工作致力于设计具有更强泛化能力的 GNN 架构, 以缓解因分布偏移引起的性能下降问题。这些方法包括结合随机注

意力机制的 GNN [21]、引入随机傅里叶特征的模型 [42]、基于 Hilbert-Schmidt 独立性准则的解耦方法 [17] 等。

也有部分研究聚焦于图的局部结构特性，探索 GNN 的分布外泛化能力，如构建泛化边界 [7]、边屏蔽生成器 [5] 以及局部结构感知机制 [41] 的设计等。

此外，还有研究尝试针对不同图数据设计差异化处理策略。例如，Policy-GNN [15] 通过强化学习为每个节点动态选择聚合层数；OODGNN [16] 采用样本重加权方式学习解相关的图表示；Customized-GNN [34] 为不同图生成专属训练参数。GRACES [29] 通过可微搜索实现图级架构自适应，是首个将架构搜索用于提升 GNN 泛化能力的算法。基于 GRACES，DCGAS [43] 引入基于扩散模型的数据增强模块，显著提升了 GNN 在非独立同分布图上的分类准确性。

## 7 结论

本文提出了多维节点级图神经架构搜索框架 (Multi-dimension Node-specific graph Neural Architecture Search, MNNAS)，旨在提升图神经架构搜索 (GNAS) 在分布偏移场景下的泛化能力。MNNAS 通过定制能够反映图结构内在变异性的节点级架构，克服了现有 GNAS 方法依赖大量训练数据、难以适应跨图分布差异的局限性。大量实验结果表明，MNNAS 在结合自适应聚合注意力机制与幂律分布建模的基础上，在多种有监督与无监督任务中均表现出色，显著提升了模型在分布外条件下的鲁棒性与泛化能力。

## 参考文献

1. Baker, B., Gupta, O., Naik, N., Raskar, R.: Designing neural network architectures using reinforcement learning. arXiv preprint arXiv:1611.02167 (2016)
2. Cai, H., Chen, T., Zhang, W., Yu, Y., Wang, J.: Efficient architecture search by network transformation. In: Proc. of AAAI (2018)
3. Corso, G., Cavalleri, L., Beaini, D., Liò, P., Veličković, P.: Principal neighbourhood aggregation for graph nets. Proc. of NeurIPS pp. 13260–13271 (2020)
4. Ding, Y., Yao, Q., Zhang, T.: Propagation model search for graph neural networks. arXiv preprint arXiv:2010.03250 (2020)
5. Fan, S., Wang, X., Mo, Y., Shi, C., Tang, J.: Debiasing graph neural networks via learning disentangled causal substructure. Proc. of NeurIPS pp. 24934–24946 (2022)
6. Gao, Y., Yang, H., Zhang, P., Zhou, C., Hu, Y.: Graph neural architecture search. In: Proc. of IJCAI (2021)
7. Garg, V., Jegelka, S., Jaakkola, T.: Generalization and representational limits of graph neural networks. In: Proc. of ICML. pp. 3419–3430 (2020)
8. Guo, K., Wen, H., Jin, W., Guo, Y., Tang, J., Chang, Y.: Investigating out-of-distribution generalization of gnns: An architecture perspective. In: Proc. of KDD. pp. 932–943 (2024)
9. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Proc. of NeurIPS (2017)
10. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. Proc. of NeurIPS pp. 22118–22133 (2020)

11. Jiang, S., Balaprakash, P.: Graph neural network architecture search for molecular property prediction. In: 2020 IEEE International conference on big data (big data). pp. 1346–1353 (2020)
12. Jin, H., Song, Q., Hu, X.: Auto-keras: An efficient neural architecture search system. In: Proc. of KDD. pp. 1946–1956 (2019)
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
14. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., et al.: Wilds: A benchmark of in-the-wild distribution shifts. In: Proc. of ICML. pp. 5637–5664 (2021)
15. Lai, K.H., Zha, D., Zhou, K., Hu, X.: Policy-gnn: Aggregation optimization for graph neural networks. In: Proc. of KDD. pp. 461–471 (2020)
16. Li, H., Wang, X., Zhang, Z., Zhu, W.: Ood-gnn: Out-of-distribution generalized graph neural network. IEEE Transactions on Knowledge and Data Engineering (2022)
17. Li, H., Zhang, Z., Wang, X., Zhu, W.: Disentangled graph contrastive learning with independence promotion. IEEE Transactions on Knowledge and Data Engineering (2022)
18. Li, Y., King, I.: Autograph: Automated graph neural network. In: Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part II 27. pp. 189–201 (2020)
19. Liu, H., Simonyan, K., Vinyals, O., Fernando, C., Kavukcuoglu, K.: Hierarchical representations for efficient architecture search. arXiv preprint arXiv:1711.00436 (2017)
20. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
21. Miao, S., Liu, M., Li, P.: Interpretable and generalizable graph learning via stochastic attention mechanism. In: Proc. of ICML. pp. 15524–15543 (2022)
22. Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzian, A., Duffy, N., et al.: Evolving deep neural networks. In: Artificial intelligence in the age of neural networks and brain computing, pp. 269–287 (2024)
23. Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M.: Weisfeiler and leman go neural: Higher-order graph neural networks. In: Proc. of AAAI. pp. 4602–4609 (2019)
24. Nazi, A., Hang, W., Goldie, A., Ravi, S., Mirhoseini, A.: Gap: Generalizable approximate graph partitioning framework. arxiv 2019. arXiv preprint arXiv:1903.00614
25. Nunes, M., Pappa, G.L.: Neural architecture search in graph neural networks. In: Intelligent Systems: 9th Brazilian Conference, BRACIS 2020, Rio Grande, Brazil, October 20–23, 2020, Proceedings, Part I 9. pp. 302–317 (2020)
26. Peng, W., Hong, X., Chen, H., Zhao, G.: Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: Proc. of AAAI. pp. 2669–2676 (2020)
27. Pham, H., Guan, M., Zoph, B., Le, Q., Dean, J.: Efficient neural architecture search via parameters sharing. In: Proc. of ICML. pp. 4095–4104 (2018)
28. Qin, Y., Wang, X., Zhang, Z., Zhu, W.: Graph differentiable architecture search with structure learning. Proc. of NeurIPS pp. 16860–16872 (2021)
29. Qin, Y., Wang, X., Zhang, Z., Xie, P., Zhu, W.: Graph neural architecture search under distribution shifts. In: Proc. of ICML. pp. 18083–18095 (2022)

30. Ranjan, E., Sanyal, S., Talukdar, P.: Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In: Proc. of AAAI. pp. 5470–5477 (2020)
31. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: Proc. of AAAI. pp. 4780–4789 (2019)
32. Shi, M., Tang, Y., Zhu, X., Huang, Y., Wilson, D., Zhuang, Y., Liu, J.: Genetic-gnn: Evolutionary architecture search for graph neural networks. Knowledge-Based Systems p. 108752 (2022)
33. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
34. Wang, Y., Ma, Y., Jin, W., Li, C., Aggarwal, C., Tang, J.: Customized graph neural networks. arXiv preprint arXiv:2005.12386 (2020)
35. Wei, L., Zhao, H., Yao, Q., He, Z.: Pooling architecture search for graph classification. In: Proc. of CIKM. pp. 2091–2100 (2021)
36. Wilder, B., Ewing, E., Dilkina, B., Tambe, M.: End to end learning and optimization on graphs. Proc. of NeurIPS (2019)
37. Wu, Y.X., Wang, X., Zhang, A., He, X., seng Chua, T.: Discovering invariant rationales for graph neural networks. In: ICLR (2022)
38. Xie, L., Yuille, A.: Genetic cnn. In: Proc. of ICCV. pp. 1379–1388 (2017)
39. Xie, S., Zheng, H., Liu, C., Lin, L.: Snas: stochastic neural architecture search. arXiv preprint arXiv:1812.09926 (2018)
40. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
41. Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.i., Jegelka, S.: Representation learning on graphs with jumping knowledge networks. In: Proc. of ICML. pp. 5453–5462 (2018)
42. Xu, K., Zhang, M., Li, J., Du, S.S., Kawarabayashi, K.i., Jegelka, S.: How neural networks extrapolate: From feedforward to graph neural networks. arXiv preprint arXiv:2009.11848 (2020)
43. Yao, Y., Wang, X., Qin, Y., Zhang, Z., Zhu, W., Mei, H.: Data-augmented curriculum graph neural architecture search under distribution shifts (2024)
44. You, J., Ying, Z., Leskovec, J.: Design space for graph neural networks. Proc. of NeurIPS pp. 17009–17021 (2020)
45. Zhao, H., Zhou, Y., Song, Y., Lee, D.L.: Motif enhanced recommendation over heterogeneous information network. In: Proc. of CIKM. pp. 2189–2192 (2019)
46. Zhao, Y., Wang, D., Gao, X., Mullins, R., Lio, P., Jamnik, M.: Probabilistic dual network architecture search on graphs. arXiv preprint arXiv:2003.09676 (2020)
47. Zhou, D., Yu, Z., Xie, E., Xiao, C., Anandkumar, A., Feng, J., Alvarez, J.M.: Understanding the robustness in vision transformers. In: Proc. of ICML. pp. 27378–27394 (2022)
48. Zhou, K., Huang, X., Song, Q., Chen, R., Hu, X.: Auto-gnn: Neural architecture search of graph neural networks. Frontiers in big Data p. 1029307 (2022)
49. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proc. of CVPR. pp. 8697–8710 (2018)