arXiv:2601.00229v1 [cs.LG] 1 Jan 2026

利用对抗图提示实现鲁棒图微调

Ziyan Zhang、Bo Jiang* 和 Jin Tang

摘要—**参数高效微调（PEFT）方法已成为将预训练图神经网络模型适配到下游任务的主流范式。然而，现有参数高效微调方法通常容易受到图拓扑和节点属性/特征上的各种噪声与攻击的影响。为解决这一问题，我们首次提出将对抗学习融入图提示，并构建一种新颖的对抗图提示（AGP）框架，以实现鲁棒图微调。AGP 包含两个关键方面。首先，我们将 AGP 的一般问题形式化为极大极小优化问题，并提出一种交替优化方案来求解。针对内层最大化，我们提出联合投影梯度下降（JointPGD）算法，以生成强对抗噪声。针对外层最小化，我们采用一个简单而有效的模块来学习最优节点提示，以抵御对抗噪声。其次，我们从理论上证明了所提出的 AGP 能够应对图拓扑噪声和节点噪声。这证实了 AGP 微调方法在各种图噪声下的通用性和鲁棒性。值得注意的是，所提出的 AGP 是一种通用方法，可与各种预训练图神经网络模型结合，以增强它们在下游任务中的鲁棒性。在多个基准任务上进行的大量实验验证了 AGP 方法相较于最先进方法的鲁棒性和有效性。**

索引词—**图神经网络、参数高效微调、图提示学习、对抗学习。**

I. 引言

RECENTLY，图神经网络（GNNs）已成为学习和表示图结构数据的主流方法。它们已应用于众多现实世界的应用中，包括推荐系统 [1], 、分子分类 [2], 以及计算机视觉 [3]。尽管取得了成功，图神经网络的性能通常依赖于大规模标注数据集，而这类数据集在实践中往往难以获取且成本高昂。为减轻这种依赖，广泛使用的“预训练与微调”范式已成功扩展到图学习领域 [4]–[6]。该范式首先进行预训练

在大量无标签图数据上训练图神经网络，以学习通用结构模式，然后通过特定任务微调使模型适应下游任务。然而，全参数微调计算成本高昂，且容易导致灾难性遗忘，即遗忘预训练期间获得的知识。近年来，人们探索了多种参数高效微调（PEFT）算法来解决这些问题。

参数高效微调为全参数微调提供了一种高效且灵活的替代方案。其目标是通过引入少量可学习参数或轻量级模块，使冻结的预训练图神经网络模型适应微调下游任务。在图领域，参数高效微调技术大致可分为三种代表性范式，即基于适配器的方法、基于 LoRA 的方法和基于提示的方法。基于适配器的方法通常会在图神经网络架构中加入紧凑的可训练适配器，以捕获任务特定知识，同时保留预训练阶段学到的表示 [7], [8]。低秩适配（LoRA）方法通过对权重矩阵进行低秩分解来更新模型，从而将参数更新限制在低维子空间中 [9]。基于提示的方法会引入可学习提示，并将其注入节点 [10]–[12], 、边 [13],或子图 [14]，以缩小预训练与下游任务之间的任务或结构差距。由于简单且可扩展，基于提示的方法已成为图学习中应用最广泛的参数高效微调策略。

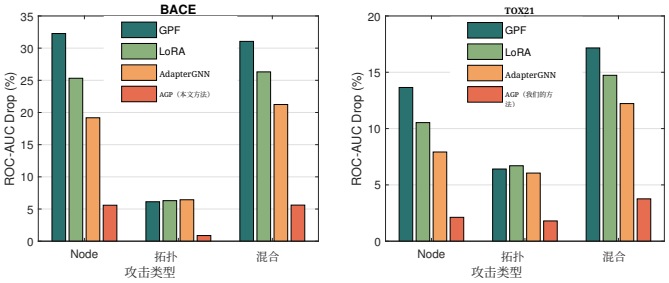


图 1. BACE 和 TOX21 数据集上不同对抗攻击目标（节点、拓扑和混合）下 ROC-AUC 降幅的比较。“ROC-AUC 降幅”表示干净基线与含噪基线之间的性能差距。现有参数高效微调方法（GPF [10], LoRA [15] 和 AdapterGNN [7]）在所有攻击类型下均出现较大降幅，而所提出的 AGP 始终表现出最小的性能退化，展现出对不同类型对抗噪声的强大鲁棒性。

在许多应用中，下游应用中的图数据往往会受到各种噪声和攻击的影响，例如伪相关拓扑结构或受损节点特征，这些问题源于环境干扰、人为错误、不完善的数据处理或恶意攻击等。 [16]然而，对于预训练图神经网络模型和现有参数高效微调方法而言，一个关键挑战是它们容易受到对抗扰动以及

* 通讯作者 Ziyan Zhang 和 Jin Tang 任职于中国合肥 230009 安徽大学计算机科学与技术学院（电子邮箱：zhangziyanahu@163.com,ahu tj@163.com）。Bo Jiang 任职于安徽大学光电信息获取与防护技术国家重点实验室、计算机科学与技术学院（电子邮箱：jiangbo@ahu.edu.cn）

0000–0000/00$00.00 © 2021 年 IEEE

图数据中各种噪声的影响。如图 1 所示，这通常会导致在图拓扑或节点特征遭受攻击和噪声干扰时，模型性能大幅下降。现有参数高效微调研究主要强调从预训练图神经网络中进行知识迁移的有效性。然而，在下游微调场景中的鲁棒学习方面，相关研究仍在很大程度上尚未得到充分探索。针对鲁棒参数高效微调，Song 等人 [17] 提出了一种鲁棒图提示方法，用于抵御拓扑扰动的微调。然而，该方法基于一些人工制定的噪声识别规则来检测噪声，因而无法泛化到某些隐蔽的对抗扰动。此外，该方法仅限于处理拓扑噪声，对节点特征噪声和混合噪声则未作处理。因此，如何设计一个鲁棒的图微调参数高效微调框架，以防御各种（拓扑和节点）攻击与噪声，仍然是一个悬而未决的问题。

为弥补这一差距，我们提出了一种新的鲁棒图提示学习方案，称为**对**抗**图**提示**法**（AGP），通过将对抗学习引入图提示调优问题。需要指出的是，对抗学习策略通常用于传统的图神经网络学习领域，以增强图神经网络模型的鲁棒性 [18], [19]。然而，据我们所知，这种策略尚未用于鲁棒图提示学习问题。具体而言，**首先**，我们明确提出了 AGP 的一般问题表述，将其建模为一个最小—最大优化问题。内层最大化问题旨在寻找能够降低提示模型性能的强对抗噪声，而外层最小化问题则旨在学习鲁棒图提示，以抵消对抗性（拓扑和节点）攻击/噪声的影响。**其次**，我们从理论上证明，所提出的 AGP 能够同时应对图拓扑噪声和节点噪声，这验证了 AGP 微调方法在不同类型图噪声下的通用性和鲁棒性。**第**三，我们针对 AGP 问题提出了一种有效算法，通过内层与外层子问题之间的交替优化来求解。对于内层最大化，我们推导出一种高效的联合投影梯度下降（JointPGD）算法，用于生成对抗性节点噪声和拓扑噪声。对于外层最小化，我们设计了一个简单而有效的模块，用于学习最优节点提示。

本文的主要贡献总结如下：

- 我们将对抗学习策略引入图提示问题，并提出了一种新颖的对抗图提示（AGP）方法。AGP 是一种通用方案，可与各种预训练图神经网络模型结合，在下游任务上实现鲁棒微调。
- 我们从理论上证明，所提出的 AGP 能够应对图拓扑和节点噪声，证实了 AGP 微调方法在不同类型图噪声下的通用性和鲁棒性。
- 我们通过对内层与外层子问题进行交替优化，推导出一种有效算法来求解所提出的 AGP 问题。对于内层最大化问题，我们提出了一种新的联合投影梯度下降（JointPGD）算法，可高效生成对抗性拓扑噪声和节点噪声。

为评估所提出的 AGP 方法的有效性，我们将其集成到多个预训练图神经网络模型中，并应用于各种下游任务。在多个基准任务上进行的大量实验表明，与当前最先进的图微调方法相比，AGP 方法具有良好的有效性和鲁棒性。

II. 相关工作

A. 图提示调优

提示学习旨在不进行完整参数微调的情况下，使大型预训练模型适应下游任务 [10], [20],，近年来在图领域受到越来越多的关注。现有图提示学习技术通常可分为两类：输入级提示和输出级提示。输入级提示旨在通过对齐预训练与下游任务的数据分布来弥合二者之间的差距，主要通过向输入空间或中间图神经网络层注入可学习组件来实现。例如，图提示特征（GPF） [10] 引入共享提示向量，通过可学习的加权机制调节输入节点特征。GraphControl [21] 和 EdgePrompt [13] 通过引入边级提示来优化输入图拓扑。All-in-One [14] 利用信息丰富的子图作为提示，使模型适应特定任务。广义图提示（GGPT） [12], 、MultiGPrompt [22],和 ProNoG [23]等方法则设计任务特定提示向量，并通过逐元素加法或乘法将其与隐藏层特征结合，引导模型实现目标。与这些输入级提示不同，输出级提示通过学习能够构建共享语义空间的原型来对齐预文本任务与下游任务。GPPT [20]和 HetGPT [24] 等代表性方法采用这一范式，使冻结的预训练模型适应下游任务。然而，现有图提示框架仍容易受到图攻击以及图拓扑或节点特征噪声的影响，导致性能显著下降，如图 1 所示。为克服这一局限，我们提出了一种鲁棒的对抗式图提示调优框架。

B. 图上的对抗训练

对抗训练是增强图神经网络抵御恶意攻击鲁棒性的基本防御策略。对抗训练的关键在于构建极小极大优化目标，通过优化模型参数，使其能够抵御最坏情况下的对抗噪声。根据攻击目标的不同，现有的对抗式图神经网络可分为三类：基于特征的、基于拓扑的和混合式对抗鲁棒图神经网络。基于特征的对抗训练侧重于学习作用于节点特征的扰动矩阵。通过向节点特征空间注入噪声，这些方法训练图神经网络在特征级攻击下维持性能，例如 Nettack [25]。基于拓扑的对抗训练仅通过边添加、删除或重连等操作扰动图拓扑。随后，模型在受到扰动的图拓扑上进行训练，以降低对结构噪声的敏感性。典型示例包括 Metattack [26],

GraphPGD [18] 和图结构攻击 [27]。混合对抗训练同时攻击图拓扑和节点特征。这种方法会生成更全面、更具挑战性的对抗样本，从而使模型能够抵御各种类型的攻击噪声，例如 IG-FGSM [19] 和双目标对抗鲁棒图神经网络 [28]。尽管取得了成功，现有的对抗鲁棒图神经网络方法都是针对特定图神经网络架构设计的，并且从头开始训练，无法直接应用于普遍采用的“预训练与微调”范式。

尽管近期研究 [29] 已在图提示学习框架中采用对抗训练，提出了一种新的后门攻击方法，但如何在图提示框架下确保对抗鲁棒性仍是一个悬而未决的问题。为解决这一问题，我们提出了一种新颖的基于图提示的对抗训练框架，使图神经网络能够在多种攻击场景下进行鲁棒微调。

III. 对抗图提示模型

A. 预备知识

令 $G(\mathbf{X}, \mathbf{A})$表示一个具有节点特征$\mathbf{X} \in \mathbb{R}^{N\times D}$和邻接矩阵$\mathbf{A} \in \{0,1\}^{N\times N}$的图，其中 N和 D分别表示节点数和特征维度。在下游标签稀缺的场景中，通常采用“预训练与微调”范式来增强图模型的表达能力。在这一范式下，首先通过无监督方式 [4], [5],在源数据集上对GNN骨干网络进行预训练，随后通过微调使其适应下游任务。图提示学习是一种有效的微调方法 [10], [12], ，只需优化少量额外的参数化图提示，即可保留预训练知识。本文聚焦于广泛使用的输入级图提示方法。令 $\mathcal{G}(\cdot;\Theta^*, \mathcal{P})$表示提示增强图学习模型，该模型将具有固定参数的预训练骨干网络 $\Theta^* = \{\Theta^{(l)}\}_{l=0}^{L-1}$ 与一组可学习提示$\mathcal{P} = \{\mathbf{P}^{(l)}\}_{l=0}^{L-1}$ 相结合，以使模型适应特定的下游任务。这里，$\mathbf{P}^{(l)}$ 表示第 l 层的提示组件，可采用多种形式实例化，例如特征向量、拓扑结构或子图。基于上述符号，输入级图提示学习问题可表述为：

$$\min_{\mathcal{P}} \mathcal{L}_{task}\big(\mathcal{G}(\mathbf{X}, \mathbf{A}; \Theta^*, \mathcal{P}), \mathbf{Y}\big) \tag{1}$$

其中， $\mathcal{L}_{task}$ 表示下游任务损失函数，$\mathbf{Y}$表示标签。

B. AGP模型

在真实场景中，下游任务中的图数据不可避免地会受到噪声和恶意攻击。这些问题表现为拓扑异常或特征损坏，来源多种多样，例如环境干扰、人工标注错误、不完善的数据预处理以及对抗攻击 [16]。预训练图神经网络模型通常对这类图噪声或攻击较为敏感，可能导致其在下游任务中的性能大幅下降 [17]。为解决这一问题，我们提出了一种简单而有效的鲁棒图提示微调方法，称为**对抗图提示**（AGP），该方法将对抗训练引入图提示，以实现鲁棒微调。具体而言，AGP 将图提示学习过程建模为一个极大极小优化问题。在这一框架中，对抗噪声与图提示以相互促进的方式迭代更新：通过模拟最坏情况攻击来优化扰动，使其最大化下游任务损失；与此同时，训练图提示以抵御此类扰动，从而最小化该损失。这种对抗式交互促使提示模型具备抵御噪声的鲁棒性。

本文主要关注将针对节点特征和图拓扑的对抗攻击构造为加性噪声分量。具体而言，令$\mathbf{E}_x \in \mathbb{R}^{N\times D}$ 和$\mathbf{E}_a \in \mathbb{R}^{N\times N}$分别表示针对节点特征和图拓扑的对抗扰动矩阵。在图提示学习的一般形式（式（1））基础上，我们的AGP一般表述为：

$$\min_{\mathcal{P}} \max_{\mathbf{E}_x\in\mathcal{C}_x, \mathbf{E}_a\in\mathcal{C}_a} \mathcal{L}_{task}\big(\mathcal{G}(\mathbf{X}+\mathbf{E}_x, \mathbf{A}+\mathbf{E}_a; \Theta^*, \mathcal{P}), \mathbf{Y}\big) \tag{2}$$

其中， Θ^* 表示预训练图神经网络模型的冻结参数。 $\mathcal{C}_x$ 和 $\mathcal{C}_a$ 分别定义了节点特征和图拓扑的可行扰动约束。需要注意的是，上述AGP是一种通用方案。它可以与各种预训练图神经网络模型和特定的提示策略相结合，从而在下游任务上实现鲁棒微调。

IV. AGP模型优化

AGP模型（式（2））的优化构成了一个双层优化问题，包括：（i）内层最大化，即在特定约束下合成对抗噪声，以最大化任务损失；（ii）外层最小化，即优化图提示，使预训练图神经网络能够抵御此类攻击。我们推导了一种用于优化该问题的交替算法。后续小节将详细介绍每个步骤的推导过程。

A. 内层最大化

内层最大化阶段侧重于合成最坏情况扰动。其主要目标是确定对图拓扑和节点特征的对抗性修改，从而最大化下游任务损失。固定图提示 $\mathcal{P}^* = \{\mathbf{P}^{(l)^*}\}_{l=0}^{L-1}$ 后，这个内层最大化子问题可形式化表示为：

$$\max_{\mathbf{E}_x\in\mathcal{C}_x, \mathbf{E}_a\in\mathcal{C}_a} \mathcal{L}_{task}\big(\mathcal{G}(\mathbf{X}+\mathbf{E}_x, \mathbf{A}+\mathbf{E}_a; \Theta^*, \mathcal{P}^*), \mathbf{Y}\big) \tag{3}$$

下文将首先介绍噪声约束的细节，然后推导出优化该约束的算法。

1）约束定义：对于基于特征的攻击，噪声矩阵 $\mathbf{E}_x \in \mathbb{R}^{N\times D}$ 应受到约束，以确保其幅度保持在半径为 ϵ、范数为 q 的球内，从而保证扰动不可察觉，正如文献所建议的那样

在 [30], [31] 中所述。因此，节点特征攻击的约束可表述为

$$\mathbf{E}_x \in \mathcal{C}_x = \{\|\mathbf{E}_x\|_q \leq \epsilon\} \tag{4}$$

对于离散拓扑攻击，我们引入一个二值指示矩阵$\mathbf{B} \in \{0,1\}^{N\times N}$来表示边翻转，其中$\mathbf{B}_{ij} = \mathbf{B}_{ji} = 1$表示节点 i 与节点 j 之间的连接发生了变化（删除或添加边）。这一表述使我们能够将拓扑攻击建模为加性噪声$\mathbf{E}_a$ [18]。为确保对抗图拓扑具有稀疏性，假设边扰动指示矩阵$\mathbf{B}$ 满足稀疏约束。因此，基于拓扑的攻击可表示为：

$$\begin{aligned} \mathbf{E}_a &= \mathbf{B} \odot (\mathbf{1}\mathbf{1}^\top - \mathbf{I} - 2\mathbf{A}) \\ s.t.\ \ \mathbf{B} &\in \mathcal{C}_a = \{\mathbf{B}_{ij} \in \{0,1\}, \|\mathbf{B}\|_0 \leq r\|\mathbf{A}\|_0\} \end{aligned} \tag{5}$$

其中，$\odot$ 表示逐元素乘积运算。$\mathbf{1}$ 表示全一向量，$\mathbf{I}$ 表示单位矩阵。$\|\cdot\|_0$ 表示ℓ_0-范数，用于统计矩阵中的非零元素个数。

2）目标优化：基于上述约束定义，我们可以将上述内层最大化子问题（式（3））改写如下：

$$\begin{aligned} &\mathbf{E}_x^*, \mathbf{E}_a^* = \underset{\mathbf{E}_x\in\mathcal{C}_x, \mathbf{B}\in\mathcal{C}_a}{\arg\max}\ \mathcal{L}_{task}(\mathcal{G}(\mathbf{X}+\mathbf{E}_x, \mathbf{A}+\mathbf{E}_a; \Theta^*, \mathcal{P}^*), \mathbf{Y}) \\ &s.t.\ \mathcal{C}_x = \{\|\mathbf{E}_x\|_q \leq \epsilon\}, \mathcal{C}_a = \{\mathbf{B}_{ij} \in \{0,1\}, \|\mathbf{B}\|_0 \leq r\|\mathbf{A}\|_0\} \end{aligned} \tag{6}$$

有许多攻击方法可以用来解决这一问题。本文将标准的投影梯度下降（PGD） [30]扩展为一种新的联合投影梯度下降（JointPGD），旨在同时攻击节点特征和图拓扑。具体而言，JointPGD 沿损失函数的联合梯度方向移动 K 步，同时更新图拓扑和节点特征的对抗噪声矩阵。为简便起见，我们将 $\mathbb{A} = \mathbf{1}\mathbf{1}^\top - \mathbf{I} - 2\mathbf{A}$ 记作，并令 $\mathcal{L}^{(k)}$ 表示

$$\mathcal{L}^{(k)} = \mathcal{L}_{task}(\mathcal{G}(\mathbf{X}+\mathbf{E}_x^{(k-1)}, \mathbf{A}+\mathbf{E}_a^{(k-1)}); \Theta^*, \mathcal{P}^*), \mathbf{Y}) \tag{7}$$

其中，$k = 1, 2, \ldots, K$ 表示扰动步数。随后，我们的 JointPGD 攻击的第 k 步可表示为：

$$\mathbf{E}_x^{(k)} = \mathbf{E}_x^{(k-1)} + \alpha \cdot \text{sign}(\nabla_{\mathbf{X}_{adv}^{(k)}} \mathcal{L}^{(k)}) \tag{8}$$

$$\mathbf{E}_a^{(k)} = (\mathbf{B}^{(k-1)} + \beta \cdot \nabla_{\mathbf{B}^{(k-1)}} \mathcal{L}^{(k)}) \odot \mathbb{A} \tag{9}$$

其中，$\mathbf{X}_{adv}^{(k)} = \mathbf{X} + \mathbf{E}_x^{(k-1)}$和 α, β 为学习率。$\mathbf{E}_x^{(0)}$初始化为一个服从均匀分布 $\mathcal{U}(-\epsilon, \epsilon)$ 的随机矩阵，而 $\mathbf{B}^{(0)}$ 初始化为零矩阵。为确保满足约束条件 $\mathcal{C}_x$，我们在每次迭代后应用以下投影函数：

$$\text{Proj}_x(\mathbf{E}_x, \epsilon) = min(-\epsilon, max(\mathbf{E}_x, \epsilon)) \tag{10}$$

约束 $\mathcal{C}_a$ 同时涉及 ℓ_0-范数约束 $\|\mathbf{B}\|_0 \leq r\|\mathbf{A}\|_0$ 和离散二值约束 $\mathbf{B}_{ij} \in \{0,1\}$。为满足 ℓ_0-范数约束，我们在每次迭代后采用阈值函数，具体如下：

$$\text{Proj}_a(\mathbf{B}, r)_{ij} = \begin{cases} 0, & \mathbf{B}_{ij} < \eta \\ \mathbf{B}_{ij}, & \mathbf{B}_{ij} \geq \eta \end{cases} \tag{11}$$

其中，η 表示$\mathbf{B}$ 和 $q = \lfloor r\|\mathbf{A}\|_0 \rfloor$中的第 q 大值。为满足二值约束，我们对上一次迭代得到的 $\mathbf{B}^{(K)}$ 进行伯努利分布采样 [32]，以获得最终的离散扰动指示矩阵 $\mathbf{B}^*$。完整的攻击过程 JointPGD 总结于算法**1**中。

算法1 JointPGD 混合攻击

1: 输入： 图数据 G（$\mathbf{X}$，$\mathbf{A}$）、图标签 $\mathbf{Y}$、学习率 α 和 β、扰动半径 ϵ、扰动比率 r，扰动步数 K、提示矩阵 $\mathcal{P}^*$、预训练图神经网络模型及其参数 Θ^*

2: 初始化： 初始化噪声和指示矩阵：
$\mathbf{E}_x^{(0)} \sim \mathcal{U}(-\epsilon, \epsilon)$, $\mathbf{B}^{(0)} \leftarrow \mathbf{0}$

3: **对于** $k = 1$ **到** K，**执行**

4: 执行梯度下降：
$\mathbf{E}_x^{(k)} \leftarrow \mathbf{E}_x^{(k-1)} + \alpha \cdot$ 符号$(\nabla_{\mathbf{X}_{adv}^{(k)}} \mathcal{L}^{(k)})$
$\mathbf{B}^{(k)} \leftarrow \mathbf{B}^{(k-1)} + \beta \cdot \nabla_{\mathbf{B}^{(k-1)}} \mathcal{L}^{(k)}$

5: 通过投影函数更新噪声：
$\mathbf{E}_x^{(k)} \leftarrow \text{Proj}_x(\mathbf{E}_x^{(k)}, \epsilon)$，$\mathbf{B}^{(k)} \leftarrow$ 投影$_a(\mathbf{B}^{(k)}, r)$

6: **结束循环**

7: 生成最终的对抗图噪声：
$\mathbf{E}_x^* \leftarrow \mathbf{E}_x^{(K)}$, $\mathbf{E}_a^* \leftarrow$ 伯努利分布$(\mathbf{B}^{(K)}) \odot \mathbb{A}$

8: 返回对抗图噪声 $\mathbf{E}_x^*$ 和 $\mathbf{E}_a^*$

B. 外层最小化

外层最小化问题旨在学习鲁棒图提示，使模型在遭受对抗攻击时仍能保持性能，从而将最坏情况下的任务损失降至最低。给定经过优化的对抗图噪声 $\mathbf{E}_x^*$ 和 $\mathbf{E}_a^*$，外层最小化子问题可表述为：

$$\min_{\mathcal{P}} \mathcal{L}_{task}\big(\mathcal{G}(\mathbf{X}+\mathbf{E}_x^*, \mathbf{A}+\mathbf{E}_a^*; \Theta^*, \mathcal{P}), \mathbf{Y}\big) \tag{12}$$

请注意，式 (12) 给出了一种通用的提示学习公式，此处可以采用各种具体的图提示策略。本文采用图节点提示策略，即在每个预训练图神经网络层中，将提示向量添加到每个节点的特征上，以增强该节点，即

$$\mathbf{X}_p^{(l)} = \mathbf{X}^{(l)} + \mathbf{P}^{(l)} \tag{13}$$

$\mathbf{P}^{(l)}$ 表示第 l 层中可学习的图节点提示，其中 $l = 0, \mathbf{1}, \cdots, L-1$ 和 $\mathbf{X}^{(0)} = \mathbf{X}$。请注意，该策略通常可以统一多种图提示范式 [10]。具体而言，我们将在 § V 中提供理论分析，以证明该策略能够缓解拓扑扰动和节点特征噪声。式 (13) 中的提示 $\mathbf{P}^{(l)}$ 既可以直接学习 [10]，[12]，也可以通过参数化函数 [33]学习。本文引入一种更紧凑、轻量级的瓶颈函数来计算它：

$$\mathbf{P}^{(l)} = \text{BatchNorm}\left(\sigma\big(\mathbf{X}^{(l)}\mathbf{W}_d^{(l)}\big)\mathbf{W}_u^{(l)}\right) \tag{14}$$

其中 $\mathbf{W}_d^{(l)} \in \mathbb{R}^{D^{(l)}\times d}$、$\mathbf{W}_u^{(l)} \in \mathbb{R}^{d\times D^{(l)}}$ 和 $d \ll D^{(l)}$。$\sigma(\cdot)$ 是非线性激活函数，BatchNorm(·) 表示批归一化操作。图 2 展示了所提出的图提示学习模块的架构。基于

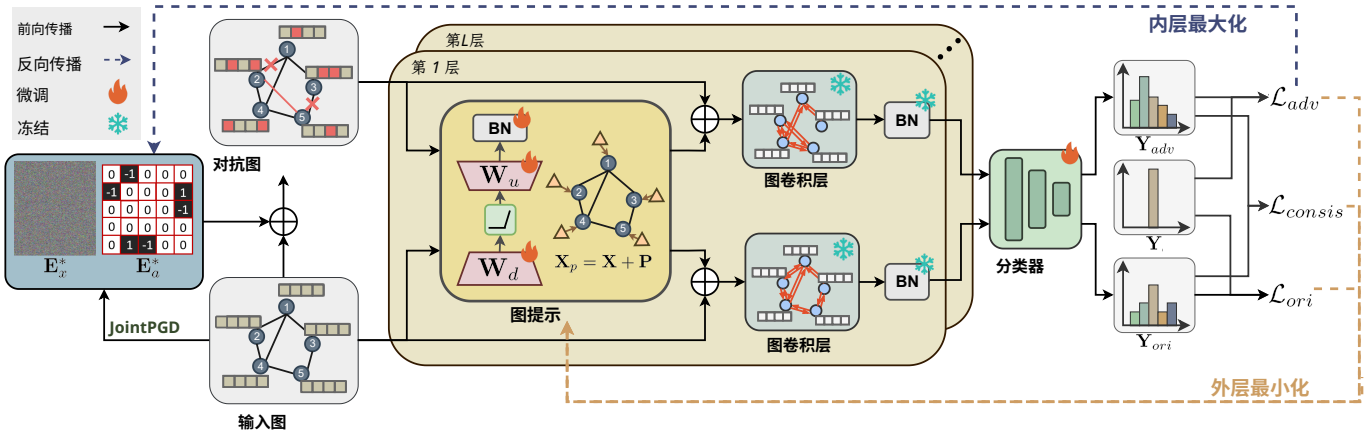


图 2. 所提出的对抗图提示（AGP）框架的整体架构。微调过程包含两个目标：(1) 最大化（蓝色虚线）：通过 JointPGD 算法生成更具挑战性的对抗噪声$\mathbf{E}_x^*$ 和$\mathbf{E}_a^*$，从而最大化对抗损失 $\mathcal{L}_{adv}$。(2) 最小化（红色虚线）：最小化对抗损失 $\mathcal{L}_{adv}$、原始损失 $\mathcal{L}_{ori}$ 和一致性损失 $\mathcal{L}_{consis}$，以调整提示模块和分类器，同时保持 GNN 骨干网络冻结。

式 (14)，式 (12) 中的外层最小化转化为学习最优参数 $\mathcal{W}=\{\mathbf{W}_d^{(l)}, \mathbf{W}_u^{(l)}\}_{l=0}^{L-1}$，使模型在对抗噪声 $\mathbf{E}_x^*$ 和 $\mathbf{E}_a^*$ 下仍能保持模型性能，即:

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \mathcal{L}_{task}\big(\mathcal{G}(\mathbf{X}+\mathbf{E}_x^*, \mathbf{A}+\mathbf{E}_a^*; \Theta^*, \mathcal{W}), \mathbf{Y}\big) \tag{15}$$

这可以通过梯度下降算法求解，如下所示:

$$\mathcal{W}' = \mathcal{W} - \lambda \nabla_{\mathcal{W}} \mathcal{L}_{total} \tag{16}$$

其中，λ 表示学习率。总体损失函数$\mathcal{L}_{total}$ 定义为

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \gamma \mathcal{L}_{ori} + \eta \mathcal{L}_{consis} \tag{17}$$

其中，γ 和 η 为超参数。$\mathcal{L}_{adv}$ 表示定义如下的对抗损失:

$$\mathcal{L}_{adv} = \mathcal{L}_{task}(\mathbf{Y}_{adv}, \mathbf{Y}) \tag{18}$$

其中，$\mathbf{Y}_{adv}$ 表示基于对抗输入的模型预测。为在鲁棒学习过程中保持模型的标准推理能力，模型还基于原始干净数据上的任务损失进行联合优化，具体如下:

$$\mathcal{L}_{ori} = \mathcal{L}_{task}(\mathbf{Y}_{ori}, \mathbf{Y}) \tag{19}$$

其中，$\mathbf{Y}_{ori}$ 表示来自原始图的预测。此外，如文献 [31]，所建议，我们还引入一致性损失 $\mathcal{L}_{consis}$，以强制原始输入与对抗输入之间的预测保持一致，即

$$\mathcal{L}_{consis} = \mathcal{L}_{task}(\mathbf{Y}_{adv}, \mathbf{Y}_{ori}). \tag{20}$$

完整的训练过程如算法2所示，AGP的总体框架如图2所示。

V. 理论分析

在本节中，我们将进行理论分析，并严格证明所提出的 AGP 如何解决节点噪声和拓扑噪声问题。具体而言，我们有

定理1。假设输入图数据 $G(\hat{\mathbf{X}}、\hat{\mathbf{A}})$ 同时受到节点特征噪声和拓扑噪声 $\{\mathbf{E}_x、\mathbf{E}_a\}$的破坏。

算法2对抗图提示

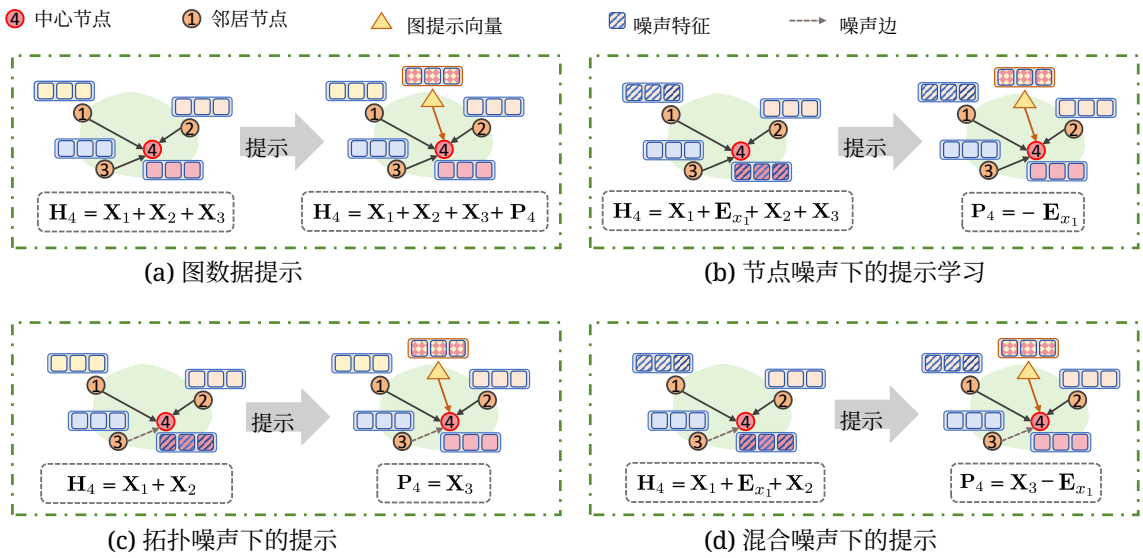
1：输入：图数据 G(X, A)、图标签 Y、预训练 GNN参数 Θ^*、学习率 γ、初始提示参数 $\mathcal{W}^{(0)}$、迭代次数 T、超参数γ、 η2：对于 $t=1$ 到 T 执行 3：计算干净图上的预测：$\mathrm{Y}_{ori} \leftarrow \mathcal{G}$(X, A；$\Theta^*$、$\mathcal{W}^{(t-1)}$) 4：通过内层最大化生成对抗噪声：$\mathrm{E}_x^*$、$\mathrm{E}_a^* \leftarrow$ JointPGD(G(X, A) $\mathcal{G}$(·；Θ^*、$\mathcal{W}^{(t-1)}$)) 5：构造对抗样本：$\mathrm{X}_{adv}^* \leftarrow$ X $+\mathrm{E}_x^*$、$\mathrm{A}_{adv}^* \leftarrow$ A $+\mathrm{E}_a^*$6：计算对抗图上的预测：$\mathrm{Y}_{adv} \leftarrow \mathcal{G}(\mathrm{X}_{adv}^*$、$\mathrm{A}_{adv}^*$；$\Theta^*$、$\mathcal{W}^{(t-1)}$)) 7：计算总损失：$\mathcal{L}_{total} \leftarrow \mathcal{L}_{adv} + \gamma\mathcal{L}_{consis} + \eta\mathcal{L}_{ori}$8：通过梯度下降更新提示参数：$\mathcal{W}^{(t)} \leftarrow \mathcal{W}^{(t-1)} - \lambda\nabla_{\mathcal{W}^{(t-1)}}\mathcal{L}_{total}$9：结束循环 10：返回：优化后的提示参数 $\mathcal{W}^{(T)}$

对于具有预训练参数 $\Theta^* = \{\Theta^{(l)}\}_{l=0}^{L-1}$ 的预训练图神经网络模型 $\mathcal{G}(\cdot)$，存在最优节点提示 $\mathcal{P} = \{\mathbf{P}^{(l)}\}_{l=0}^{L-1}$，使得:

$$\mathcal{G}(\hat{\mathbf{A}}, \hat{\mathbf{X}}; \Theta^*, \mathcal{P}) = \mathcal{G}(\hat{\mathbf{A}} - \mathbf{E}_a, \hat{\mathbf{X}} - \mathbf{E}_x; \Theta^*) \tag{21}$$

证明。在以下证明中，我们以预训练 GIN [2] 为例。其他一些预训练模型，如 GCN [34],和 GraphSAGE [35],，也可以用类似的方法得到。为简单起见，令 $\tilde{\hat{\mathbf{A}}} = \mathbf{A} + (1+\varepsilon)\mathbf{I}$，其中 $\mathbf{I}$ 表示单位矩阵，ε 是一个可学习标量参数。给定初始输入 $\hat{\mathbf{X}}^{(0)} = \hat{\mathbf{X}}$，带有节点提示$\mathbf{P}^{(l)}$ 的 GIN [2] 的逐层传播可表示为

$$\begin{aligned}\mathcal{G}^{(l)}(\hat{\mathbf{A}}, \hat{\mathbf{X}}^{(l)}; \Theta^{(l)}, \mathbf{P}^{(l)}) &= \tilde{\mathbf{A}}(\hat{\mathbf{X}}^{(l)} + \mathbf{P}^{(l)})\Theta^{(l)} \\ &= \tilde{\mathbf{A}}\hat{\mathbf{X}}^{(l)}\Theta^{(l)} + \tilde{\mathbf{A}}\mathbf{P}^{(l)}\Theta^{(l)} = \mathbf{X}^{(l+1)} + \tilde{\mathbf{A}}\mathbf{P}^{(l)}\Theta^{(l)}\end{aligned} \tag{22}$$



(a) 图数据提示　(b) 节点噪声下的提示学习

(c) 拓扑噪声下的提示　(d) 混合噪声下的提示

图 3. 示例说明了所提出的图提示在邻域聚合过程中缓解节点噪声和拓扑噪声的能力。

首先，从输入层（$l = 0$）开始。为确保 Eq.(21)成立，最优输入层提示$\mathbf{P}^{(0)}$ 必须满足以下方程:

$$\mathcal{G}^{(0)}(\hat{\mathbf{A}}, \hat{\mathbf{X}}^{(0)}; \Theta^{(0)}, \mathbf{P}^{(0)}) = \mathcal{G}^{(0)}(\hat{\mathbf{A}} - \mathbf{E}_a, \hat{\mathbf{X}}^{(0)} - \mathbf{E}_x) \quad (23)$$

根据式 (22)，式 (23) 的左侧项为

$$\mathcal{G}^{(0)}(\hat{\mathbf{A}}, \hat{\mathbf{X}}^{(0)}; \Theta^{(0)}, \mathbf{P}^{(0)}) = \mathbf{X}^{(1)} + \tilde{\mathbf{A}}\mathbf{P}^{(0)}\Theta^{(0)} \quad (24)$$

式 (23) 的右侧项可展开为

$$\mathcal{G}^{(0)}(\hat{\mathbf{A}} - \mathbf{E}_a, \hat{\mathbf{X}}^{(0)} - \mathbf{E}_x; \Theta^{(0)}) = (\tilde{\mathbf{A}} - \mathbf{E}_a)(\hat{\mathbf{X}}^{(0)} - \mathbf{E}_x)\Theta^{(0)}$$
$$= \mathbf{X}^{(1)} - \mathbf{E}_a\hat{\mathbf{X}}^{(0)}\Theta^{(0)} - \tilde{\mathbf{A}}\mathbf{E}_x\Theta^{(0)} + \mathbf{E}_a\mathbf{E}_x\Theta^{(0)} \quad (25)$$

通过对齐式 (24) 和式 (25)，我们可以发现，当.....时，式 (23) 成立

$$\tilde{\mathbf{A}}\mathbf{P}\Theta^{(0)} = \mathbf{E}_a\mathbf{E}_x\Theta^{(0)} - \mathbf{E}_a\hat{\mathbf{X}}^{(0)}\Theta^{(0)} - \tilde{\mathbf{A}}\mathbf{E}_x\Theta^{(0)} \quad (26)$$

因此，我们可以得到最优的$\mathbf{P}^{(0)}$ ，即

$$\mathbf{P}^{(0)} = \tilde{\mathbf{A}}^{-1}\mathbf{E}_a(\mathbf{E}_x - \hat{\mathbf{X}}^{(0)}) - \mathbf{E}_x \quad (27)$$

其次，基于上述分析，我们可以证明，节点特征噪声$\mathbf{E}_x$ 会通过输入提示$\mathbf{P}^{(0)}$被吸收。因此，对于后续的中间隐藏层，我们只需考虑拓扑噪声$\mathbf{E}_a$ ，因为（受扰动的）ˆ拓扑$\mathbf{A}$ 会参与GIN模型的所有层，即式（21）变为:

$$\mathcal{G}^{(l)}(\hat{\mathbf{A}}, \hat{\mathbf{X}}^{(l)}; \Theta^{(l)}, \mathbf{P}^{(l)}) = \mathcal{G}^{(l)}(\hat{\mathbf{A}} - \mathbf{E}_a, \hat{\mathbf{X}}^{(l)}) \quad (28)$$

其中 $l = 1, \ldots, L-1$。按照类似的推导，当满足以下条件时，式（28）成立:

$$\mathbf{P}^{(l)} = -\tilde{\mathbf{A}}^{-1}\mathbf{E}_a\hat{\mathbf{X}}^{(l)}, \ l = 1, \cdots, L-1 \quad (29)$$

最后，根据上述式（27）和式（29），我们可以找到一个最优节点提示集合 $\mathcal{P}$ ，以补偿预训练GIN模型中的拓扑噪声和节点噪声。 □

注。如上述证明所分析的，当节点噪声 $\mathbf{E}_x$ 和拓扑噪声 $\mathbf{E}_a$ 同时存在时，我们需要对所有预训练图神经网络层进行提示学习，如式（13）所示。然而，当仅存在节点噪声$\mathbf{E}_x$ 时，只需设置 $\mathbf{P}^{(0)} = -\mathbf{E}_x$ 和 $\mathbf{P}^{(l)} = \mathbf{0}, l = 1, \ldots, L-1$，即可完全吸收其影响。也就是说，我们只需在输入层进行提示学习。在实验中，我们将这一简单变体记为 AGP-S。

证明。为了直观地理解所提出的提示机制如何缓解拓扑噪声和节点噪声，我们以图3中标记为节点4的代表性目标节点为例，展示邻域聚合过程。图3（a）展示了由提示增强的标准聚合过程。图3（b）—（d）展示了具体的鲁棒性场景：（b）节点噪声：当邻居节点1的特征被噪声向量$\mathbf{E}_{x_1}$扰动时，提示$\mathbf{P}_4$ 通过设置$\mathbf{P}_4 = -\mathbf{E}_{x_1}$来消除这一扰动；（c）拓扑噪声：当节点4与节点3之间的边被切断时，提示$\mathbf{P}_4$ 通过设置$\mathbf{P}_4 = \mathbf{X}_3$有效地恢复缺失的连接；（d）混合噪声：在同时包含节点噪声和拓扑噪声的混合攻击下，提示$\mathbf{P}_4 = \mathbf{X}_3 - \mathbf{E}_{x_1}$ 可同时抵消这两种噪声。上述案例直观地展示了定理1中的理论发现，表明最优提示能够有效地抵御各种噪声，保持表示完整性。

VI. 实验

为全面评估所提出 AGP 的鲁棒性和有效性，我们在七个分子数据集和三种不同的图攻击场景下，将其与五种具有代表性的微调方法进行比较。

A. 实验设置

1) 数据集：骨干模型在从

ZINC15数据库中采样的 200 万个无标签分子图上进行了预训练 [36]在下游评估中，我们采用了来自化学领域的七个广泛使用的分子性质预测数据集 [37]。这些数据集涵盖了多种生化任务，包括药物毒性预测、生物活性分类和副作用识别。在所有数据集中，每个样本都表示为一个分子图，其中节点对应原子，边表示从分子结构中提取的化学键。详细的数据集统计信息见表 I。所有数据集均采用分子图学习基准中普遍使用的标准基于骨架的数据划分 [4]。

表 I 下游任务数据集的统计信息。

数据集	#图数	#平均节点数	#平均边数	#任务
BACE	1513	34.09	73.72	1
BBBP	2039	24.06	51.91	1
CLINTOX	1478	26.16	55.77	2
HIV	41127	25.51	54.94	1
SIDER	1427	33.64	70.72	27
TOX21	7831	18.57	38.59	12
TOXCAST	8575	18.78	38.52	617

2) 基线方法：我们将所提出的方法与五种具有代表性的微调方法进行比较：

- 全量微调（FT）。这是一种传统的微调方法，会在微调阶段更新所有模型参数。
- BitFit [38]：该方法仅微调模型中的偏置项，同时保持预训练权重不变。
- 图提示微调（GPF） [10]。该方法引入一个可学习的提示向量，并将其添加到每个节点的原始特征中。微调期间，仅更新提示向量的参数。
- LoRA [15]。该方法采用带有瓶颈架构的并行LoRA模块。与先前工作 [7], 所采用的设置一致，LoRA模块被放置在每个图神经网络层中消息传递操作的前后。
- AdapterGNN [7]：该方法在预训练图神经网络旁并行引入双适配器模块。它会微调多层感知机中的偏置项、批归一化层以及适配器模块的参数。

对于所有基线方法，我们采用其各自原始论文中报告的最优超参数设置。为确保公平比较，所有基线均在经过100轮次微调后进行评估。

3）攻击设置：为全面评估 AGP 的鲁棒性，我们测试了模型在以下三种不同对抗噪声场景下的 ROC-AUC：

- 节点攻击：使用标准 PGD 方法实施， [30] 扰动步长为 $K = 10$ ，扰动半径为$\epsilon = 0.8$。对于数据集 BBBP 和 CLINTOX，我们将学习率设为 $\alpha = 0.005$。对于其他数据集，我们将学习率设为 $\alpha = 0.01$。
- 拓扑攻击：采用离散 PGD 算法进行，扰动步长为 $K = 10$ ，扰动比率为 $r = 40\%$。对于数据集 BBBP 和 CLINTOX，我们将学习率设为 $\beta = 30$。对于其他数据集，我们将学习率设为 $\beta = 100$。
- 混合攻击：使用所提出的 JointPGD 方法实施，参数与上述单独的节点攻击和拓扑攻击保持一致。

为评估模型的有效性，我们采用“ROC-AUC”作为主要评估指标。我们分别报告干净数据和对抗数据上的结果，以衡量标准性能和鲁棒性。由于节点攻击中初始噪声矩阵具有随机性，且拓扑攻击中包含概率采样步骤，因此在每种网络初始化下，我们均执行五次攻击并报告平均结果。

4）模型设置：对于化学数据集上的图分类，我们采用具有 5 层的 GIN [2] 骨干网络，并使用“mean”池化策略，这遵循了以往工作的常见做法 [7],[10]。我们在每个 GIN 层的输入处分别插入提示模块，以帮助抵御对抗噪声。在模型末尾添加一个三层多层感知机，用于下游分类。隐藏维度设为 300，每个提示模块中的瓶颈维度设为 64。微调期间，预训练 GIN 编码器的参数保持冻结，仅更新图提示模块和分类器，如图 2 所示。模型使用 Adam 优化器 [39], 进行训练，学习率设为 0.001，权重衰减系数设为 0。为确保对抗优化稳定，AGP 在对抗微调前先使用干净数据进行 30 个预热轮次，以初始化提示参数。超参数 γ 和 η分别设为 0.3 和 0.6。为进行公平比较，预训练模型通过 Infomax 策略 [5]从同一个公开可用的预训练 GIN 模型 [37],初始化，该模型可从网站 [1]下载。

B. 比较结果

表II-V 分别报告了全参数微调和四种参数高效微调策略在七个分子性质预测数据集上针对节点攻击、拓扑攻击和混合攻击的性能。值得注意的是，所提出的 AGP 不仅在应对各种攻击时实现了更高的平均鲁棒性，还提升了在干净数据集上的泛化性能。具体而言，(i) 在所有三种对抗攻击下，所提出的 AGP 在大多数数据集上始终实现最高的鲁棒性，明显优于现有的微调和参数高效微调方法。这一趋势表明，将对抗训练融入参数高效微调框架能够显著增强模型鲁棒性，使模型更好地抵御节点攻击和拓扑攻击。(ii) AGP 在原始干净数据集上也取得了具有竞争力或更高的准确率。我们认为，这些小幅提升主要归因于两个因素。首先，我们的鲁棒提示学习可能有效缓解了原始数据集中的固有噪声。其次，对于样本量有限的下游任务，生成的对抗样本可作为有效的数据增强形式，

[1]https://github.com/snap-stanford/pretrain-gnns

表 II 七个数据集在节点特征攻击下的鲁棒性表现比较。粗体和下划线分别表示最佳方法和次佳方法。“平均值”表示所有数据集上的平均值。

方法	BACE	BBBP	TOX21	TOXCAST	SIDER	HIV	CLINTOX	**Avg.**
微调 (100%)	61.66±1.75	58.99±1.20	68.36±0.35	60.85±0.15	56.49±0.95	66.88±1.50	65.47±2.28	62.67±1.17
BitFit（0.2%）	47.14±2.96	53.67±1.18	64.08±0.70	57.74±0.73	53.95±0.99	59.42±2.31	57.76±1.67	56.25±1.51
GPF (0.01%)	48.53±2.11	53.29±1.40	64.24±1.14	57.80±0.36	54.41±0.42	59.64±2.02	57.55±4.85	56.49±1.76
LoRA（5.0%）	55.85±1.92	55.82±0.60	67.04±1.04	59.43±0.50	54.79±1.25	59.81±0.99	62.69±3.68	59.35±1.43
AdapterGNN（5.7%）	62.40±1.38	59.58±1.06	68.40±0.44	60.20±0.37	55.26±0.55	65.34±0.76	66.10±2.58	62.47±1.02
AGP-S (0.5%)	70.75±2.42	59.40±1.06	73.83±0.24	62.94±0.59	60.17±0.49	69.26±0.97	65.17±1.48	65.93±1.04
AGP（2.6%）	**77.41±2.24**	**65.00±0.88**	**75.46±0.46**	**63.92±0.44**	**60.70±0.62**	**75.60±0.49**	**69.12±2.56**	**69.60±1.10**

表 III 拓扑攻击下七个数据集的鲁棒性表现对比。粗体表示最佳方法。“平均值”表示所有数据集的平均值。

方法	BACE	BBBP	TOX21	TOXCAST	SIDER	HIV	CLINTOX	**Avg.**
微调 (100%)	74.06±1.25	60.50±2.53	70.33±0.63	60.63±0.21	59.66±1.49	67.74±0.68	57.70±1.94	64.37±1.25
BitFit（0.2%）	73.82±1.57	59.95±1.58	71.16±0.32	60.86±0.53	62.49±0.85	65.02±1.50	56.32±1.73	64.23±1.15
GPF (0.01%)	74.67±1.91	59.71±0.62	71.48±0.35	60.49±0.29	62.57±0.84	65.17±1.20	56.99±3.09	64.44±1.19
LoRA（5.0%）	74.87±1.16	62.52±1.04	70.87±0.49	61.15±0.41	61.67±1.15	65.05±0.83	57.10±4.54	64.75±1.37
AdapterGNN（5.7%）	75.14±1.51	62.31±2.08	70.23±0.00	60.25±0.54	58.92±0.99	67.77±1.14	60.90±1.94	65.07±1.17
AGP（2.6%）	**81.24**±1.76	**70.47**±1.18	**75.87**±0.35	**63.80**±0.68	**63.77**±1.28	**79.55**±1.52	**71.93**±3.74	**72.38**±1.50

表 IV 混合攻击下七个数据集的鲁棒性表现比较。粗体表示最佳方法。“平均值”表示所有数据集的平均值。

方法	BACE	BBBP	TOX21	TOXCAST	SIDER	HIV	CLINTOX	**Avg.**
微调 (100%)	59.42±1.66	54.38±1.54	64.11±0.37	57.70±0.18	55.40±1.38	60.46±1.07	50.03±2.52	57.36±1.25
BitFit（0.2%）	49.10±2.58	50.75±1.32	60.85±0.60	54.99±0.63	56.46±0.53	54.61±1.84	49.09±2.52	53.69±1.43
GPF（0.01%）	49.75±1.75	50.75±0.99	60.73±0.98	54.89±0.40	56.34±0.79	54.51±1.96	47.98±1.82	53.56±1.24
LoRA（5.0%）	54.86±1.80	52.94±0.44	62.84±0.83	56.79±0.30	56.29±1.68	54.90±0.80	51.31±3.89	55.69±1.39
AdapterGNN（5.0%）	60.34±1.85	57.20±1.36	64.10±0.44	57.20±0.40	54.38±1.18	59.63±0.93	56.28±2.17	58.45±1.19
AGP（2.6%）	**76.39±1.42**	**66.18±1.12**	**73.88±0.52**	**62.86±0.69**	**57.07±1.10**	**75.27±0.87**	**70.86±2.03**	**68.93±1.11**

从而提升学习性能。(iii) 对于 AGP-S（AGP 的单提示层变体），结果表明它也能增强模型在节点攻击下的鲁棒性（表 II），这与 §*V* 中的分析一致。然而，由于可学习参数有限，其鲁棒性和准确率的平均提升幅度略低于所提出的 AGP。

C. 模型分析

1) 预训练模型：为验证我们的方法对各种预训练骨干网络的普适性，我们在 BACE 和 TOX21 数据集上，采用四种不同的预训练方法（包括 ContextPred [37],EdgePred [35], Masking [37] 和 SimGrace [40]），评估其针对节点攻击、拓扑攻击和混合攻击的鲁棒性。为确保公平比较，所有预训练模型参数均来自公开可用网站[1][2]。鲁棒性通过“ROC-AUC下降”量化，定义为从干净数据到受攻击数据时的性能下降幅度。数值越低，表示鲁棒性越高。如图4所示，一个重要发现是，标准微调（FT）和现有参数高效微调方法（例如，GPF [10], LoRA [15], 和 AdapterGNN [7]）在所有预训练模型上都极易受到三种对抗攻击的影响。例如，在 BACE 数据集上，混合攻击通常会导致现有模型的性能下降超过 20%。相比之下，所提出的 AGP 和 AGP-S 方法在所有预训练模型上均表现出更强的鲁棒性。即使在相同的攻击条件下，它们在大多数数据集上的 ROC-AUC 下降幅度也保持在 5 % 以内。这一结果表明，AGP 能够在不同预训练场景下提升模型的泛化鲁棒性。

[2]https://github.com/junxia97/SimGRACE

表 V 七个数据集上的 ROC-AUC 结果比较。“AGP-NODE”“AGP-TOPOLOGY”和“AGP-HYBRID”分别表示 AGP 在节点特征攻击、图拓扑攻击和混合攻击场景下通过对抗训练获得的结果。粗体表示最佳方法。“平均值”表示所有数据集上的平均值。

方法	BACE	BBBP	TOX21	TOXCAST	SIDER	HIV	CLINTOX	**Avg.**
微调 (100%)	80.68±1.10	67.59±1.13	76.24±0.49	64.49±0.24	63.96±1.09	75.66±1.08	74.77±1.76	71.91±0.98
BitFit（0.2%）	79.34±1.26	65.07±0.63	77.46±0.44	65.32±0.57	65.20±0.98	72.40±1.98	72.77±1.93	71.08±1.11
GPF (0.01%)	80.80±1.89	64.69±0.88	77.89±0.62	65.23±0.35	65.93±0.56	72.96±1.25	72.50±3.93	71.43±1.35
LoRA（5.0%）	81.17±0.94	66.26±0.69	77.57±0.70	65.08±0.48	65.42±1.27	72.20±0.95	74.61±3.67	71.76±1.24
AdapterGNN（5.0%）	81.58±1.16	67.90±1.14	76.32±0.56	64.28±0.48	63.51±0.60	75.32±0.93	76.55±3.29	72.21±1.17
AGP-S（0.5%）	82.03±1.30	67.27±0.69	**78.15±0.32**	**66.16±0.54**	**67.11±0.24**	75.92±0.64	73.08±1.58	72.82±0.76
AGP-节点 (2.6%)	**83.00±1.73**	**70.40±0.59**	77.82±0.31	65.53±0.42	64.17±0.58	**78.20±0.42**	76.45±2.34	**73.65±0.91**
AGP-拓扑（2.6%）	82.12±0.96	70.24±1.20	77.19±0.19	65.34±0.32	65.47±0.89	77.47±1.24	75.47±3.54	73.33±1.19
AGP-混合 (2.6%)	82.00±0.77	70.38±0.39	77.64±0.39	65.14±0.60	64.58±1.74	76.21±1.19	**77.05±2.18**	73.29±1.04

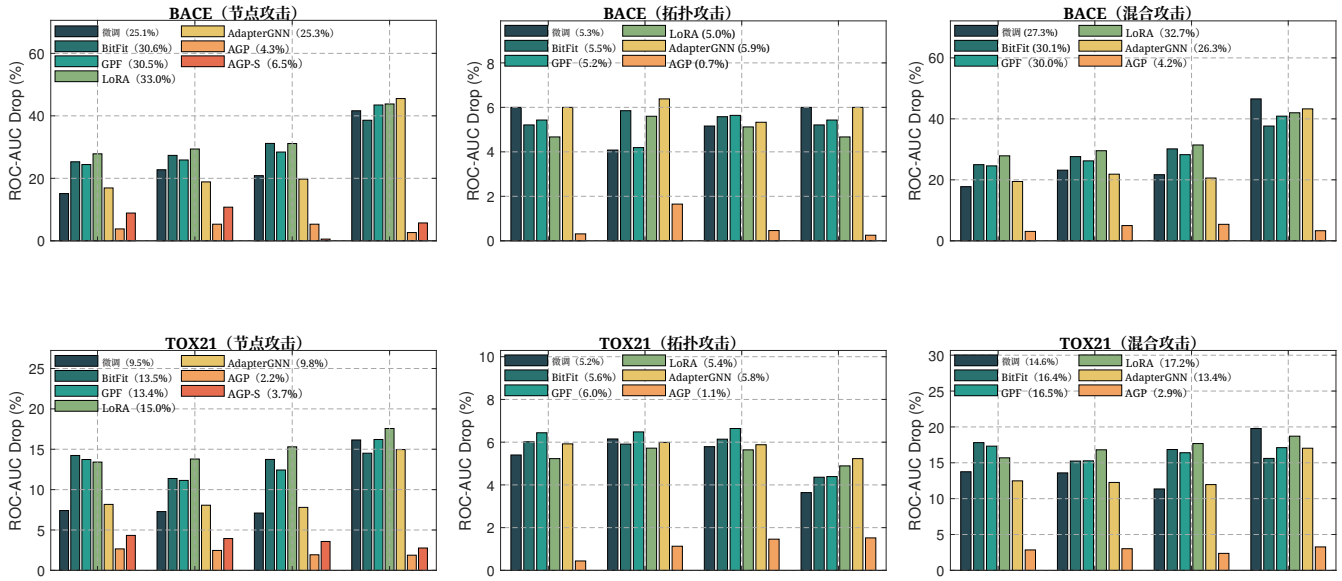


图4. 不同预训练模型下各微调策略在 BACE 和 TOX21 数据集上的鲁棒性评估。图例展示了不同预训练策略下的平均结果。

2）效率分析：为分析所提出的 AGP 的效率，我们从可调参数数量、GPU内存使用量和推理延迟等方面将其与基线方法进行比较。表II-V列出了各方法的可调参数与全量微调参数之比。图5展示了各方法在 TOX21 和 HIV 数据集上的 GPU内存使用量和推理延迟。在可调参数数量方面，AGP 仅需全量微调（FT）参数量的 2.6%，而简化的单层变体 AGP-S 进一步将这一比例降至 0.5%，低于 LoRA（5.0%）和 AdapterGNN（5.7%）等常见参数高效微调基线方法。在计算效率方面，AGP 和 AGP-S 在 GPU内存使用量和推理延迟方面均表现出良好的竞争力。具体而言，AGP-S 的推理时间为 0.67μ秒和 0.65μ秒，与轻量级基线方法 GPF 在 TOX21 和 HIV 数据集上的推理时间（0.67μ秒和 0.64μ秒）相当。尽管 AGP 带来轻微的尽管多层架构会增加延迟，但其效率仍然很高。此外，AGP-S 的内存占用较低，与三种轻量级方法（FT、GPF 和 BitFit）相近。与 LoRA 和 AdapterGNN 等其他多层架构相比，AGP 的 GPU 内存效率也具有竞争力。总体而言，这些结果表明，所提出的方法在效率方面能够在现有参数高效微调方法中保持较强的竞争力。

3）瓶颈维度：本节探讨瓶颈维度对 AGP 鲁棒性和准确率的影响。图6展示了在混合攻击下，AGP 在 BACE 和 TOX21 数据集上采用不同瓶颈维度时的性能。具体而言，维度过小（例如 $d = 1$）会限制模型的表征能力，导致欠拟合。相反，维度过大（例如 $d = 150$）则会引发过拟合，导致

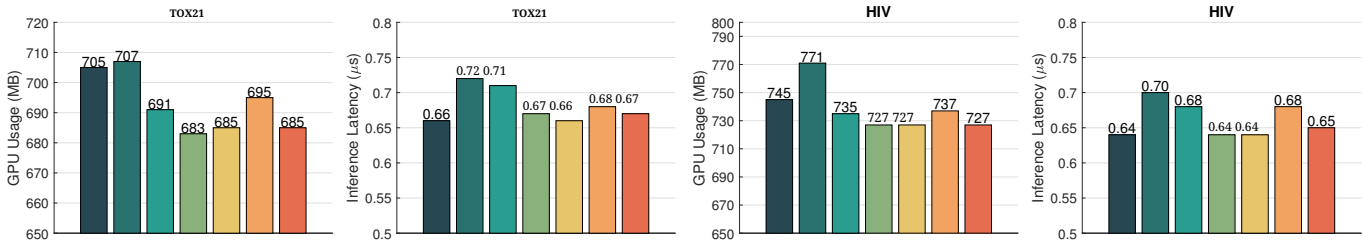


图 5. 不同微调方法下 GPU 内存用量和推理延迟的比较。

性能下降。因此，瓶颈维度约为 60 时，参数效率与模型性能之间能够达到最佳平衡。值得注意的是，即使瓶颈维度受到严格限制（例如 $d = 1$），AGP 在两个数据集上的表现仍优于全量微调基线。

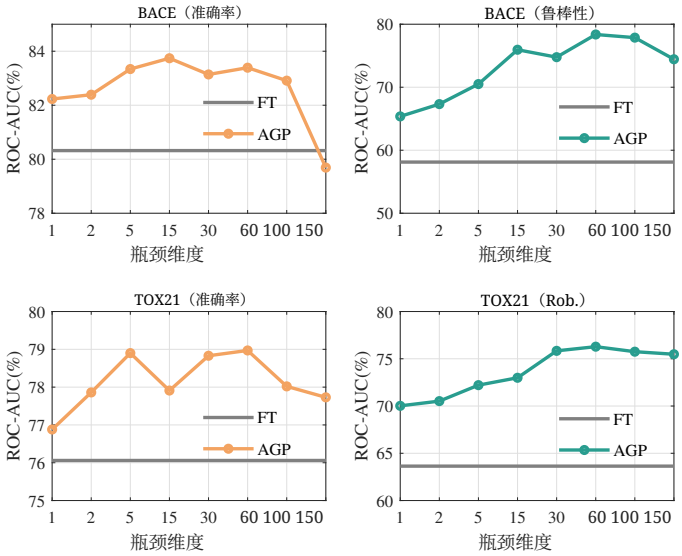


图 6. 模型训练期间损失函数的收敛情况。“Acc.” 和 “Rob.” 分别表示模型在干净图数据和对抗图数据上的表现。

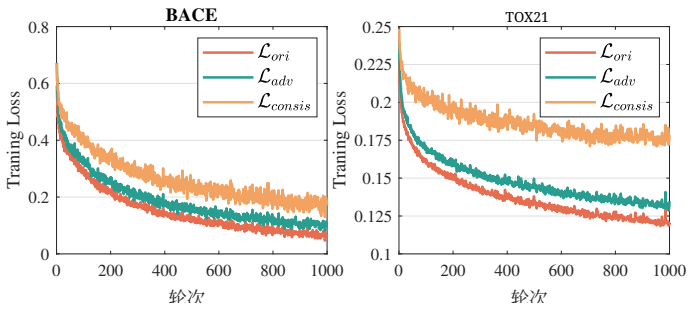


图 7. BACE 和 TOX21 数据集上三种不同损失函数的收敛曲线。

4）损失分析：图 7 展示了 BACE 和 TOX21 数据集上三个不同损失分量的收敛曲线。所有曲线持续下降并最终趋于稳定，验证了所提出的

表 VI BACE 和 TOX21 数据集上三个损失分量的消融研究。“准确率”和“鲁棒性”分别表示模型在干净图数据和对抗图数据上的表现。

$\mathcal{L}_{ori}$	$\mathcal{L}_{adv}$	$\mathcal{L}_{consis}$	BACE		TOX21	
			Rob.	Acc.	Rob.	Acc.
√			54.52	81.46	62.5	75.8
	√		77.42	78.67	72.55	75.16
		√	68.97	72.47	65.48	76.17
√	√		72.71	80.32	70.97	76.66
√		√	74.32	82.51	69.06	76.99
	√	√	**78.27**	81.26	**74.51**	76.84
√	√	√	78.12	**82.86**	74.03	**77.99**

AGP 框架的训练稳定性。为进一步剖析各分量的作用，表 VI 展示了 BACE 和 TOX21 上鲁棒性（“鲁棒性”）和干净准确率（“准确率”）的结果。可以观察到几项一致的趋势。首先，仅使用 $\mathcal{L}_{ori}$ 进行训练时，两个数据集上的鲁棒性均最低，这证实了仅干净数据监督不足以抵御对抗攻击。其次，加入 $\mathcal{L}_{adv}$ 后，“鲁棒性”显著提升（例如，BACE 上从 54.52% 提升至 72.71%），验证了对抗监督的有效性。然而，这种提升往往以准确率下降为代价。值得注意的是，引入一致性正则化（$\mathcal{L}_{consis}$）能够缓解这一权衡：将其与 $\mathcal{L}_{ori}$或 $\mathcal{L}_{adv}$结合时，有助于稳定性能并带来更均衡的提升。最后，同时优化全部三项损失的完整目标函数取得了最佳总体表现，在两个基准数据集上均实现了具有竞争力的鲁棒性和准确率。这些结果表明，一致性约束能够有效对齐干净域和对抗域，使模型在增强鲁棒性的同时保持干净准确率。

VII. 结论

本文提出了一种新颖的框架——对抗图提示（AGP），旨在提升参数高效的图神经网络微调在面对混合对抗噪声时的鲁棒性。我们将所提出的 AGP 表述为一个极大极小优化问题，并通过在内层最大化与外层最小化之间迭代优化来求解。我们还提供了理论分析，说明 AGP 如何缓解

图拓扑和节点特征中的噪声。大量实验表明，在鲁棒性和准确率方面，AGP 均显著优于五种具有代表性的微调方法。

参考文献

[1] Y. Xu、L. Zhu、J. Li、F. Li 和 H. T. Shen，“面向高效推荐的时序社交图网络哈希”，*IEEE*知识与数据工程汇刊，第 36 卷，第 7 期，第 3541–3555 页，2024 年。[2] K. Xu、W. Hu、J. Leskovec 和 S. Jegelka，“图神经网络的能力有多强？”，载于国际学习表征会议，2019 年。[3] C. Chen、Y. Wu、Q. Dai、H.-Y. Zhou、M. Xu、S. Yang、X. Han 和 Y. Yu，“计算机视觉中的图神经网络与图Transformer综述：面向任务的视角”，*IEEE*模式分析与机器智能汇刊，第 46 卷，第 12 期，第 10297–10318 页，2024 年。[4] Y. You、T. Chen、Y. Sui、T. Chen、Z. Wang 和 Y. Shen，“利用数据增强进行图对比学习”，载于神经信息处理系统进展，第 33 卷，第 5812–5823 页，2020 年。[5] F.-Y. Sun、J. Hoffman、V. Verma 和 J. Tang，“Infograph：通过互信息最大化进行无监督和半监督图级表示学习”，载于国际学习表征会议，2020 年。[6] G. Chu、X. Wang、C. Shi 和 X. Jiang，“Cuco：采用课程式对比学习的图表示”，载于第*30*届国际人工智能联合会议论文集，第 2300–2306 页，2021 年。[7] S. Li、X. Han 和 J. Bai，“AdapterGNN：参数高效微调提升图神经网络的泛化能力”，载于*AAAI*人工智能会议论文集，第 38 卷，13608，2024 年，第 13600 页。– 。[8] A. Gui、J. Ye 和 H. Xiao，“G-adapter：面向图Transformer网络的结构感知参数高效迁移学习”，载于*AAAI*人工智能会议论文集，第 12226–12234 页，2024 年。[9] Z.-R. Yang、J. Han、C.-D. Wang 和 H. Liu，“Graphlora：面向跨图迁移学习的结构感知对比式低秩适配”，载于第*31*届*ACM SIGKDD*知识发现与数据挖掘会议论文集第*1*卷，第 1785–1796 页，2025 年。[10] T. Fang、Y. Zhang、Y. Yang、C. Wang 和 L. Chen，“面向图神经网络的通用提示调优”，载于第*37*届神经信息处理系统会议，第 36 卷，第 52464–52489 页，2023 年。[11] Z. Liu、X. Yu、Y. Fang 和 X. Zhang，“Graphprompt：统一图神经网络的预训练与下游任务”，载于*2023*年*ACM Web*会议论文集，第 417–428 页，2023 年。[12] X. Yu、Z. Liu、Y. Fang、Z. Liu、S. Chen 和 X. Zhang，“广义图提示：迈向图上预训练与下游任务的统一”，*IEEE*知识与数据工程汇刊，第 36 卷，第 11 期，第 6237–6250 页，2024 年。[13] X. Fu、Y. He 和 J. Li，“面向图神经网络的边提示调优”，载于第十三届国际学习表征会议，2025 年。[14] X. Sun、H. Cheng、J. Li、B. Liu 和 J. Guan，“All in one：面向图神经网络的多任务提示学习”，载于第*29*届*ACM SIGKDD*知识发现与数据挖掘会议论文集，第 2120–2131 页，2023 年。[15] E. J. Hu、P. Wallis、Z. Allen-Zhu、Y. Li、S. Wang、L. Wang、W. Chen、等，“LoRA：大语言模型的低秩适配”，载于国际学习表征会议，2021 年。[16] L. Sun、Y. Dou、C. Yang、K. Zhang、J. Wang、P. S. Yu、L. He 和 B. Li，“图数据上的对抗攻击与防御：综述”，*IEEE*知识与数据工程汇刊，第 35 卷，第 8 期，第 7693–7711 页，2023 年。[17] S. Song、P. Li、M. Dun、M. Huang、H. Cao 和 X. Ye，“GPromptshield：提升图提示调优抵御对抗攻击的韧性”，载于第十三届国际学习表征会议，2025 年。[18] K. Xu、H. Chen、S. Liu、P.-Y. Chen、T.-W. Weng、M. Hong 和 X. Lin，“图神经网络的拓扑攻击与防御：优化视角”，载于第*28*届国际人工智能联合会议论文集，第 3961–3967 页，2019 年。[19] H. Wu、C. Wang、Y. Tyshetskiy、A. Docherty、K. Lu 和 L. Zhu，“图数据的对抗样本：深入剖析攻击与防御”，载于第*28*届国际人工智能联合会议论文集

，第 4816–4823 页，2019年。

[20] M. Sun、K. Zhou、X. He、Y. Wang 和 X. Wang，“GPPT：图预训练与提示调优以实现图神经网络的泛化”，载于《第28届ACM SIGKDD知识发现与数据挖掘会议论文集》，第1717–1727页，2022年。[21] Y. Zhu、Y. Wang、H. Shi、Z. Zhang、D. Jiao 和 S. Tang，“GraphControl：为通用图预训练模型添加条件控制以实现图领域迁移学习”，载于《ACM Web会议2024论文集》，第539–550页，2024年。[22] X. Yu、C. Zhou、Y. Fang 和 X. Zhang，“MultiGPrompt：面向图的多任务预训练与提示学习”，载于《ACM Web会议2024论文集》，第515–526页，2024年。[23] X. Yu、J. Zhang、Y. Fang 和 R. Jiang，“非同配图预训练与提示学习”，载于《第31届ACM SIGKDD知识发现与数据挖掘会议论文集》第1卷，第1844–1854页，2025年。[24] Y. Ma、N. Yan、J. Li、M. Mortazavi 和 N. V. Chawla，“HetGPT：在预训练异构图神经网络中发挥提示调优的力量”，载于《ACM Web会议2024论文集》，第1015–1023页，2024年。[25] D. Z¨ugner、A. Akbarnejad 和 S. G¨unnemann，“面向图数据的神经网络对抗攻击”，载于《第24届ACM SIGKDD知识发现与数据挖掘国际会议论文集》，第2847–2856页，2018年。[26] D. Z¨ugner 和 S. G¨unnemann，“通过元学习对图神经网络实施对抗攻击”，载于《国际学习表征会议》，2019年。[27] B. Wang 和 N. Z. Gong，“通过操纵图结构攻击基于图的分类”，载于《2019年ACM SIGSAC计算机与通信安全会议论文集》，第2023–2040页，2019年。[28] H. Kwon 和 D.-J. Kim，“图神经网络逃逸攻击中的双目标对抗样本”，《Scientific Reports》，第15卷，第1期，第3912页，2025年。[29] M. Lin、Z. Zhang、E. Dai、Z. Wu、Y. Wang、X. Zhang 和 S. Wang，“你使用的是可靠的图提示吗？图神经网络上的木马提示攻击”，载于《第31届ACM SIGKDD知识发现与数据挖掘会议论文集》第2卷，第1729–1740页，2025年。[30] A. Madry、A. Makelov、L. Schmidt、D. Tsipras 和 A. Vladu，“迈向能够抵抗对抗攻击的深度学习模型”，载于《国际学习表征会议》，2018年。[31] H. Zhang、Y. Yu、J. Jiao、E. Xing、L. E. Ghaoui 和 M. Jordan，“鲁棒性与准确率之间具有理论依据的权衡”，载于《第36届国际机器学习会议论文集》，第97卷，第7472–7482页，2019年。[32] S. Liu、S. P. Chepuri、M. Fardad、E. Mas¸azade、G. Leus 和 P. K. Varshney，“具有相关测量噪声的估计中的传感器选择”，《IEEE信号处理汇刊》，第64卷，第13期，第3509–3522页，2016年。[33] J. Li、J. Li 和 C. Zhang，“实例感知图提示学习”。arXiv:2411.17676，2024年。[34] T. N. Kipf 和 M. Welling，“利用图卷积网络进行半监督分类”，载于《国际学习表征会议》，2017年。[35] W. Hamilton、Z. Ying 和 J. Leskovec，“大规模图上的归纳式表示学习”，载于《神经信息处理系统进展》，第30卷，第1025–1035页，2017年。[36] T. Sterling 和 J. J. Irwin，“Zinc 15——人人都能进行的配体发现”，《Journal of chemical information and modeling》，第55卷，第11期，第2324–2337页，2015年。[37] W. Hu、B. Liu、J. Gomes、M. Zitnik、P. Liang、V. Pande 和 J. Leskovec，“图神经网络预训练策略”，载于《国际学习表征会议》，2020年。[38] E. B. Zaken、Y. Goldberg 和 S. Ravfogel，“BitFit：面向基于Transformer的掩码语言模型的简单参数高效微调”，载于《第60届计算语言学协会年会论文集》（第2卷：短论文），第1–9页，2022年。[39] D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，载于《国际学习表征会议》，2015年。[40] J. Xia、L. Wu、J. Chen、B. Hu 和 S. Z. Li，“SimGRACE：一种无需数据增强的简单图对比学习框架”，载于《ACM Web会议2022论文集》，第1070–1079页，2022年。