图形表示学习的结构感知变换器

陈德雄 [*1 2] **Leslie O'Bray** [*1 2] **Karsten Borgwardt** [1 2]

摘要

近来，变换器架构在图表示学习领域受到越来越多的关注，因为它避免了图神经网络（GNN）严格的结构归纳偏差，而只通过位置编码对图结构进行编码，从而自然地克服了图神经网络（GNN）的一些局限性。在这里，我们证明了变换器通过位置编码生成的节点代表并不一定能捕捉到它们之间的结构相似性。为了解决这个问题，我们提出了结构感知变换器（Structure-Aware Transformer），这是一类建立在新的自我关注机制上的简单灵活的图变换器。这种新的自我关注机制在计算关注度之前，会先生成一个根植于每个节点的子图表示，从而将结构形成纳入原始的自我关注机制。我们提出了几种自动生成子图表示的方法，并从理论上证明了所生成的表示至少与子图表示具有同样的表现力。实际上，我们的方法在五个图预测基准上达到了最先进的性能。我们的结构感知框架可以利用任何现有的 GNN 来提取子图表示，我们的研究表明，相对于基础 GNN 模型，它能系统地提高性能，成功地结合了 GNN 和 Transformers 的优势。我们的代码可在 https://github.com/BorgwardtLab/SAT.

1. 导言

图形神经网络（GNN）已成为图形表示学习的强大而灵活的工具、

[*]同等贡献[1] 瑞士联邦理工学院生物系统科学与工程系[2] SIB 瑞士生物信息学研究所。通讯作者：陈德雄陈德雄 <dexiong.chen@bsse.ethz.ch>, Leslie O'Bray <leslie.obray@bsse.ethz.ch>。

第 39 届[th] 国际机器学习大会论文集，美国马里兰州巴尔的摩，PMLR 162，2022。

在药物发现（Gaudelet等人，2021年）、蛋白质设计（Ingraham等人，2019年）、社交网络分析（Fan等人，2019年）等领域都有成功应用。一大类 GNN 建立了多层模型，其中每一层都在上一层的基础上运行，利用消息传递机制生成新的表示（Gilmer 等人，2017 年），以聚合本地邻域信息。

虽然已经提出了许多不同的信息传递策略，但在这类 GNN 中也发现了一些关键的局限性。这些限制包括 GNN 的表达能力有限（Xu 等人，2019 年；Morris 等人，2019 年），以及过度平滑（Li 等人，2018 年；2019 年；Chen 等人，2020 年；Oono & Suzuki，2020 年）和过度压扁（Alon & Yahav，2021 年）等已知问题。过度平滑是指在足够多层之后，所有节点的表示都会趋同于一个常数，而过度挤压则是指来自远处节点的信息无法有效地通过图中的某些 "瓶颈 "进行传播，因为太多的信息会被压缩成一个固定长度的向量。因此，设计超越邻域聚合的新架构对于解决这些问题至关重要。

在自然语言理解（Vaswani 等人，2017 年）、计算机视觉（Dosovitskiy 等人，2020 年）和生物序列建模（Rives 等人，2021 年）中取得成功的 Transformers（Vaswani 等人，2017 年）具有解决这些问题的潜力。Transformer 架构不仅能在信息传递机制中聚合本地邻域信息，还能通过单个自我关注层捕捉任何节点对之间的交互信息。此外，与 GNN 不同的是，Transformer 避免在中间层引入任何结构归纳偏差，从而解决了 GNN 的表达能力限制问题。相反，它只将节点的结构或位置信息编码到输入节点特征中，但限制了从图结构中学习信息的数量。因此，在图表示学习领域，将图结构信息整合到变换器架构中的做法越来越受到关注。然而，现有的大多数方法只对节点之间的位置关系进行编码，而不是对结构关系进行明确编码。因此，它们可能无法识别节点之间的结构相似性，也无法模拟节点之间的*结构交互*（见图 1）。这可以解释为什么它们的性能

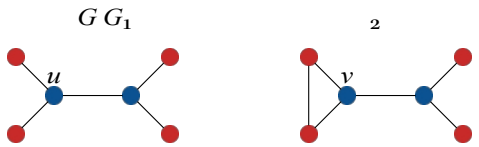


图 1：*位置感知与结构感知*：使用分别基于 G_1 和 G_2 中最短路径的位置编码（假设所有边的权重相同），节点 u 和 v 将获得相同的编码，因为在这两个图中，它们到所有其他节点的最短路径相同。但是，它们的结构不同，v 与其红色邻居构成一个三角形。

在一些任务中，稀疏 GNN 占主导地位（Dwivedi 等人，2022 年）。

贡献 在这项工作中，我们解决了如何在跨前架构中编码结构信息这一关键问题。我们的主要贡献在于引入了一种灵活的*结构感知*自关注机制，该机制明确考虑了图结构，从而确保了节点之间的结构交互。由此产生的变换器类别，我们称之为结构感知变换器（Structure-Aware Transformer，SAT），可以提供结构感知的图表示，这与现有的大多数用于图结构数据的位置感知变换器截然不同。具体来说

- 我们将 Vaswani 等人（2017 年）的自我关注机制重新表述为一种核平滑器，并通过提取以每个节点为中心的子图表示，扩展了节点特征的原始指数核，从而也考虑到了局部结构。
- 我们提出了几种自动生成子图表征的方法，使生成的核平滑器能够同时捕捉节点之间的结构和属性相似性。理论上可以保证生成的表示至少与子图表示一样富有表现力。
- 我们在五个图和节点属性预测基准上展示了 SAT 模型的有效性，表明它比最先进的 GNN 和 Transformers 性能更好。此外，我们还展示了 SAT 如何轻松利用任何 GNN 来计算包含子图信息的节点表示，并超越基础 GNN，使其成为任何现有 GNN 的轻松增强器。
- 最后，我们表明，我们可以将性能的提升归功于我们架构的结构感知方面，并展示了 SAT 如何比采用绝对编码的经典变换器更具可解释性。

我们将在第 2 和第 3 部分介绍相关工作和相关背景，然后在第 4 部分介绍我们的方法，并在第 5 部分介绍我们的实验结果。

2. 相关工作

我们在此介绍与我们最相关的工作，即源自消息传递 GNN、图上位置表示法和图变换器的工作。

消息传递图神经网络 消息传递图神经网络近来已成为图表示学习的主要方法之一。早期的开创性例子是基于对图进行卷积的 GCN（Kipf & Welling，2017 年）。Gilmer 等人（2017 年）将早期的 GNN 重新表述为消息传递 GNN 框架，自此成为当今使用的 GNN 的主要框架，并有大量实例（Hamilton 等人，2017 年；Xu 等人，2019 年；Corso 等人，2020 年；Hu 等人，2020b 年；Velic˘kovic´ 等人，2018 年；Li 等人，2020a 年；Yang 等人，2022 年）。然而，如上所述，它们存在表达能力有限、过度平滑和过度挤压等问题。

绝对编码 由于 GNN 的表达能力有限，最近有一些关于使用*绝对编码的*研究（Shaw 等人，2018 年），其中包括向输入节点特征添加或串联位置或结构表示。虽然它被称为*绝对位置编码*，但我们更普遍地将其称为*绝对编码*，以包括在图建模中都很重要的位置编码和结构编码。绝对编码主要考虑节点之间的位置或位置关系。基于位置的方法包括拉普拉斯位置编码（Dwivedi & Bresson，2021；Kreuzer 等人，2021）、基于 Weisfeiler-Lehman 的位置编码（Zhang 等人，2020）和随机漫步位置编码（RWPE）（Li et al、2020b；Dwivedi 等人，2022），而基于距离的方法包括到一组预定义节点的距离（You 等人，2019）和节点对之间的最短路径距离（Zhang 等人，2020；Li 等人，2020b）。Dwivedi 等人（2022 年）通过使用可训练的绝对编码扩展了这些想法。

图变换器 虽然上述绝对编码方法可用于消息传递 GNN，但它们在（图）变换器架构中也起着至关重要的作用。Graph Transformer（Dwivedi & Bresson，2021 年）是将 Transformer 架构推广到图的早期范例，它使用拉普拉卡特征向量作为绝对编码，并计算每个节点邻域的注意力，而不是整个图。SAN（Kreuzer 等人，2021 年）也使用拉普拉奇特征向量计算绝对编码，但计算的是整个图的注意力，同时区分真实边和创建边。许多图变换器方法也使用*相对编码*（Shaw 等人，2018 年）。

除了绝对编码之外。这种策略将图中节点之间的相对位置或距离直接纳入自我注意机制，而绝对编码则仅将图中节点之间的相对位置或距离纳入自我注意机制。

对输入节点特征应用一次。米亚隆等人（2021 年）该研究提出通过对图形的核进行相对编码，使自我关注计算产生偏差，然后通过核函数的选择将位置信息纳入变换器。近期的其他研究试图将结构信息纳入图变换器，例如将一些精心挑选的图论属性（如中心性度量和最短路径距离）编码为位置表示（Ying 等人，2021 年），或使用 GNN 整合图结构（Rong 等人，2020 年；Jain 等人，2021 年；Mialon 等人，2021 年；Shi 等人，2021 年）。

在这项工作中，我们结合了网格传递 GNN 和 Transformer 架构的优点。我们既采用了绝对编码，也采用了明确包含图结构的新型相对编码，从而设计出一种同时考虑局部和全局信息的 Transformer 架构。

3. 背景介绍

在下文中，我们将一个图称为 $G = (V, E, \mathbf{X})$，其中节点 $u \in V$ 的节点属性表示为 $x_u \in X \subset \mathbb{R}^d$，对于有 n 个节点的图，所有节点的节点属性都存储在 $\mathbf{X} \in \mathbb{R}^{n \times d}$ 中。

3.1. 图形上的变换器

GNN 明确使用图结构，而 Transformer 则去除这种明确的结构，而是利用节点属性来推断节点之间的关系。从这个意义上说，Transformer（Vaswani 等人，2017 年）忽略了图结构，而是将图视为一组（多）节点，并使用自我关注机制来推断节点之间的相似性。变换器本身由两个主要模块组成：一个自我注意模块和一个前馈神经网络。在自注意模块中，输入节点特征 $\mathbf{X}$ 首先通过线性投影被投影到查询矩阵（$\mathbf{Q}$）、键矩阵（$\mathbf{K}$）和值矩阵（$\mathbf{V}$）中，分别为 $\mathbf{Q} = \mathbf{X}\mathbf{W}_\mathbf{Q}$、$\mathbf{K} = \mathbf{X}\mathbf{W}_\mathbf{K}$ 和 $\mathbf{V} = \mathrm{X}\mathrm{W}_\mathrm{V}$。我们可以通过以下方法计算自关注度

$$\mathbf{Q}\mathbf{K}^\mathsf{T} \qquad n \times d$$

如下图所示：

$$\mathbf{X}' = \mathbf{X} + \mathrm{Attn}(\mathbf{X}), \\ \mathbf{X}'' = \mathrm{FFN}(\mathbf{X}') := \mathrm{ReLU}(\mathbf{X}'\mathrm{W})\mathbf{W}_2。 \tag{2}$$

多个层可以叠加形成一个变换器模型，最终提供图的节点级表示。由于自关注度等价于输入节点的排列组合，因此变换器将始终为具有相同属性的节点生成相同的表示，而不管它们在图中的位置和周围结构如何。因此，有必要将此类信息纳入变换器，通常是通过绝对编码。

绝对编码 绝对编码是指在主变换器模型之前将图的位置或结构表示添加或连接到输入节点特征中，例如拉普拉斯位置编码（Dwivedi & Bresson，2021 年）或 RWPE（Dwivedi 等人，2022 年）。这些编码方法的主要缺点是，它们通常不提供节点与其邻域之间结构相似性的度量。

作为核平滑的自我注意 正如米-阿隆等人（2021 年）所指出的，公式 (1) 中的自我注意可以改写为核平滑器

$$\mathrm{Attn}(x_v) = \sum_{u \in V} \frac{\kappa_{\exp}(x_v, x)_u}{\sum_{w \in V} \kappa_{\exp}(x_v, x_w)} f(x_u)，\ \forall v \in V、 \tag{3}$$

其中，$f(x) = \mathbf{W}_\mathrm{V}\, x$ 是线性值函数，$\kappa_{\exp}$ 是 $\mathbb{R}^d \times \mathbb{R}^d$ 上的（非对称）指数核，由 $\mathbf{W}_\mathbf{Q}$ 和 $\mathbf{W}_\mathbf{K}$ 参数化：

$$\kappa_{\exp}(x, x') := \exp\left(\langle \mathbf{W}_\mathbf{Q} x, \mathbf{W}\, \mathbf{x}_\mathbf{K}' \rangle / \sqrt{d_{\text{out}}}\right)、 \tag{4}$$

其中 $\langle \cdot, \cdot \rangle$ 是 $\mathbb{R}^d$ 上的点积。通过这种形式，Mialon 等人（2021 年）提出了一种相对位置编码策略，即通过该核与图上的扩散核的乘积来捕捉节点间的位置相似性。不过，这种方法只能感知位置，与我们将在第 4 节介绍的结构感知编码不同。

4. 结构感知变压器

$$\mathrm{Attn}(\mathbf{X}) := \mathrm{softmax}(\frac{}{\text{输出}})\mathbf{V} \in \text{出来} \quad (1)$$

其中 d 向外 指 $\mathbf{Q}$ 的维数、 和 $\mathbf{W_Q}, \mathbf{W_K}, \mathbf{W_V}$ 是可训练参数。通常使用*多头*注意力，将公式（1）的多个实例串联起来，在实践中证明是有效的（Vaswani 等人，2017 年）。然后，自我注意力的输出由一个跳过连接和一个前馈网络（FFN）跟进，这两个网络共同组成了一个 Trans-

在本节中，我们将介绍如何编码图
在自我关注机制中加入结构，并提供一个
基于该框架的变形器模型类。

4.1. 结构意识自我关注

如上所述，变换器中的自我关注可以改写为内核平滑器，其中内核是一个定义在节点特征上的可训练指数内核，它只有

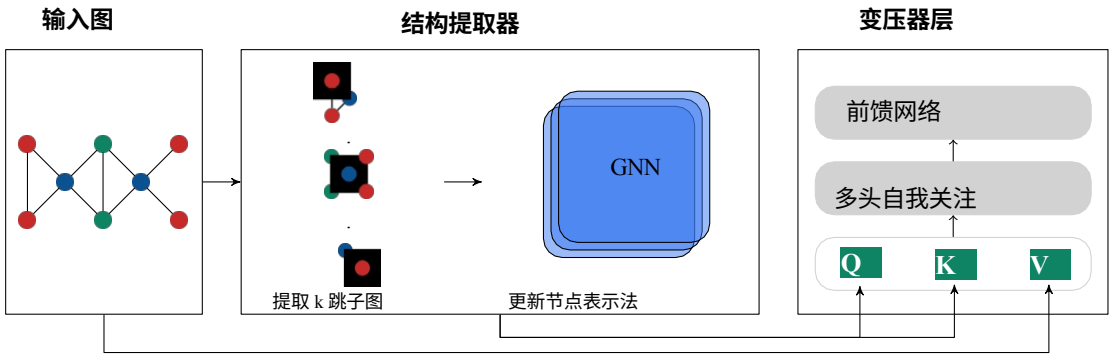


图 2：使用 k *子图*GNN 提取器作为结构提取器的 SAT 层示例概览。结构提取器生成结构感知节点表示，用于计算变换层中的查询（**Q**）和键（**K**）矩阵。结构感知节点表示是在 *k 子图*GNN 提取器中生成的，首先提取以每个节点为中心的 *k* 跳子图（此处 $k = 1$），然后在每个子图上使用 GNN，利用完整的子图信息生成节点表示。虽然结构提取器可以使用任何类别的子图，但这里说明的定义在 *k 跳子图*类别上的结构提取器在计算量和执行量之间有一个合理的权衡。

捕捉一对节点之间的属性相似性。这种核平滑器的问题在于，当节点特征相同或相似时，它无法过滤掉结构上不同于相关节点的节点。为了同时考虑节点间的结构相似性，我们考虑了一种更广义的核，该核还考虑了每个节点周围的局部子结构。通过引入一组以每个节点为中心的子图，我们将结构感知注意力定义为：

$$\text{SA-attn}(v) := \sum_{u \in V} \frac{\kappa_{\text{graph}}(S_G(v), S_G(u))}{\sum_{w \in V} \kappa_{\text{graph}}(S_G(v), S_G(w))} f(x_u), \quad (5)$$

其中 $S_G(v)$ 表示 G 中以节点 v *为*中心、与节点特征 **X** 相关联的子图，κ_{graph} 可以是比较一对子图的任何核。这种新的自我关注函数不仅考虑了属性相似性，还考虑了子图之间的结构相似性。因此，正如我们将在第 4.4 节中展示的那样，它生成的节点表示比原始的自我关注更具表现力。此外，这种自我关注不再等价于任何节点的排列，而只等价于特征与子图重合的节点，这是一个理想的特性。

在本文的其余部分，我们将考虑以下形式的 κ_{graph}，它已经包含了一大类表现力强、计算简单的模型：

$$\kappa_{\text{graph}}(S_G(v), S_G(u)) = \kappa_{\text{exp}}(\phi(v, G), \phi(u, G)), \quad (6)$$

其中，$\phi(u, G)$ 是一个结构*提取器*，用于提取以 u 为中心、具有节点特征 **X** 的子图的向量表示。值得注意的是，我们的结构感知自我关注可以灵活地与任何生成子图表示的模型相结合，包括 GNN 和（可微分）图核。对于符号

为简单起见，我们假设不存在边缘属性，但只要结构提取器能容纳边缘属性，我们的方法就能轻松将其纳入。因此，在自我关注计算中不考虑边缘属性，而是将其纳入结构感知节点表示中。在本文介绍的结构提取器中，这意味着只要基本 GNN 能够处理边缘属性，边缘属性就会被包含在内。

-子树 GNN 提取器

k

在节点 u 上获取局部结构信息是为了应用任何将现有的 GNN 模型转化为带有节点特征的输入图

X 并将 u 处的输出节点表示法视为 u 处的子图表示法。更正式地说，如果我们用 $\mathrm{GNN}^{(k)}$ 表示应用于*具有*节点特征 **X 的** G 的具有 k 层的任意 GNN 模型，则

$$j(u, G) = \mathrm{GNN}^{(k)}(u)_{G}。 \quad (7)$$

这种提取器能够表示以 u 为根的 k *子树*结构（Xu 等人，2019 年）。虽然这类结构提取器的计算速度很快，而且可以灵活利用任何现有的 GNN，但由于消息传递 GNN 的表达能力限制，它们不可能比 Weisfeiler-Lehman 检验更具表现力（Xu 等人，2019 年）。在实践中，k 值越小，性能越好，同时不会出现过度平滑或过度挤压的问题。

k *子图* **GNN 提取器** 更具表现力的提取器是使用 GNN 直接计算以 u 为中心的整个 k 跳子图的表示，而不仅仅是节点表示 u。最近的工作探索了在 GNN 中使用子图而不是节点周围的子树的想法，并取得了积极的实验结果（Zhang & Li, 2021; Wijesinghe & Wang, 2022），而且严格来说，GNN 还具有更强的提取能力。

比 1-WL 测试更强大（Zhang & Li，2021 年）。我们采用与 Zhang & Li（2021）相同的设置，并调整我们的 GNN 提取器，以利用整个 *k* 跳子图。*k 子图* GNN 提取器使用求和等池化函数聚合 *k 跳*邻域内所有节点的更新节点表示。形式上，如果我们用$_{N_k}(u)$ 表示节点 u 的 *k 跳*邻域（包括其自身），那么节点 *u* 的表示就是：

$$\phi(u, G) = \sum_{v \in N_k(u)} \text{GNN}^{(k)}(v)_{G}。 \quad (8)$$

我们发现，在使用池化函数之前，*k*-子图 GNN 提取器相当于在每个 *k 跳子图中*使用 *k-子树* GNN 提取器。为了确保归属相似性和结构相似性，我们通过连接将原始节点特征与 k 子图 GNN 提取器的节点表示进行了增强。虽然这种提取器比 *k* 子树提取器提供了更富表现力的子图表示，但它需要枚举合并所有 *k 跳子*图，因此不能像 *k 子树*提取器那样扩展到大型数据集。

其他结构提取器 最后，我们列出了用于不同目的的其他潜在结构提取器。一种可能的选择是直接学习一些"隐藏图 "作为 "锚子图"，通过使用 Nikolentzos & Vazirgiannis（2020）中介绍的概念来表示子图，以提高模型的可解释性。Niko- lentzos & Vazirgiannis（2020）通过计算整个图和每个隐藏图之间的匹配行走次数，获得了输入图的向量表示，而我们可以通过比较隐藏图和以每个节点为中心的 *k* 跳子图，将其扩展到节点级别。隐藏图 的邻接矩阵是网络中的一个可训练参数，因此可以通过端到端的训练来确定哪些子图结构具有预测性。然后，对于一个训练有素的模型，可视化学习到的隐藏图可以为数据集中的结构主题提供有用的见解。

此外，特定领域的 GNN 也可用于提取更具表现力的子图再现。例如，Bodnar 等人（2021 年）最近提出了一种在常规细胞复合体上运行的新型信息传递方案，该方案得益于为分子提供更强的表达能力。我们的自我关注机制可以充分受益于更具领域针对性和表现力自我关注和内核方法之间的联系。

4.2. 结构感知变压器

在定义了结构感知自我注意功能之后，结构感知转换器的其他组件也将遵循 3.1 节中描述的转换器架构；可视化概览见图 2。具体来说，自我关注功能之后是跳转连接、
一个 FFN 和 FFN 前后的两个归一化层。此外，我们还在跳转连接中加入了阶数因子，这对减少阶数因子的影响非常有用。
的 GNN 的发展。

最后，另一种可能的结构提取方法是在以每个节点为中心的 k 跳子图上使用非参数图核（*例如* Weisfeiler-Lehman 图核）。这为图核与深度学习的结合提供了一种灵活的方式，或许能提供新的理论见解

高度连接的图形成分的压倒性影响（Mialon 等人，2021 年），即.....、

$$x_v' = x_v + 1/\sqrt{d_v}\,\text{SA-attn}(v)、 \quad (9)$$

其中，d_v 表示节点 v 的度数。经过转换器层后，我们会得到一个结构相同但节点特征不同的新图 $G' = (V, E, \mathbf{X}')$，其中 $\mathbf{X}'$ 与转换器层的输出相对应。

最后，对于图属性预测，有多种方法可以将节点级表征汇总到图表征中，如取平均值或总和。另外，我们还可以使用虚拟 [CLS] 节点的嵌入（Jain 等人，2021 年），该节点附着在输入图上，与其他节点没有任何连接。我们将在第 5 节中对这些方法进行比较。

4.3. 与绝对编码相结合

公式 (5) 中的自我关注是结构感知的，而大多数绝对编码技术只是位置感知的，因此可以提供补充信息。实际上，我们发现这种组合能进一步提高性能，这一点我们将在第 5 节中展示。我们选择使用 RWPE（Dwivedi 等人，2022 年），不过也可以使用任何其他绝对位置表示法，包括可学习的表示法。

我们进一步认为，仅使用变换器的绝对位置编码会表现出过于宽松的结构归纳偏差，即使两个节点具有相似的局部结构，也不能保证生成相似的节点表示。这是因为基于距离或拉普拉斯的位置表示通常作为结构或位置特征，但并不能提供节点间结构相似性的度量，尤其是在两个节点来自不同图的归纳情况下。第 5 节中的经验也证实了这一点，即在不使用我们的结构编码的情况下，它们的性能相对较差。与此相反，结构感知注意力中使用的子图表示可以量身定制，以确保节点之间的结构相似性，从而在节点具有相似性的情况下生成相似的节点级表示。

属性和周围结构。我们可以用下面的定理来正式说明这一点：

定理 1. *假设 f 是一个 Lipschitz 映射，其 Lipschitz 常量用 $Lip(f)$ 表示，且结构提取器 ϕ 在子图空间上以常数 C_φ 为界。对于具有相同节点数 $|V| = |V'|$ 的两个图 $G = (V, E, \mathbf{X})$ 和 $G' = (V', E', \mathbf{X}')$ 中的任意一对节点 v 和 v'，它们的*代表图◆◆◆间的距离为 $|V| = |V|$。

结构感知注意力后的发送量是有边界的：

$$\|SA\text{-}attn(v) - SA\text{-}attn(v')\| \le C\,[1\,\|h_v - h_{v'}\| + D(\mathbf{H}, \mathbf{H}')] + c_2\, d(\mathbf{x}, \mathbf{x}') \quad (10)$$

、

其中，$C_1, C_2 > 0$ 是常数，取决于 V，$Lip(f)$，C_φ 和 SA-attn 中参数的谱规范，其表达式见附录；$h_w := \phi(w, G)$ 表示节点 w 处的子图表示法，对于任何 $w \in V$ 和 $h'_w := \phi(w', G')$ 同样，$\mathbf{H} = (h_w)_{w \in V}$ 和 $\mathbf{H}' = (h'_{w'})_{w' \in V'}$ 分别表示 G 和 G' 中子图表示的多集。用 $\Pi(V, V')$ 表示从 V 到 V' 的置换集，D 是两个具有相同心率的表示多集之间的最优匹配度量，定义为

$$D(\mathbf{X}, \mathbf{X}') := \inf_{\pi \in \Pi(V, V')} \sup_{w \in V} \|x_w - x'_{\pi(w)}\|$$

证明见附录。度量 D 是两个多集合之间的最优匹配度量，用于衡量它们之间的差异程度。该定理表明，如果两个节点所属的图的节点特征和子图表示的多集总体相似，同时这两个节点上的子图表示也相似，那么来自 SA-attn 的两个节点表示就是相似的。具体来说，如果两个节点属于相同的图，*即 $G = G'$*，那么公式（10）右侧的第二项和最后一项等于零，因此它们的表示之间的距离受限于它们对应的子图表示之间的距离。然而，对于具有绝对位置编码的变换器来说，两个节点表示之间的距离并不受其结构相似性的限制，因为两个位置表示之间的距离并不一定代表两个节点在结构上的相似程度。尽管存在更强的归纳偏差，我们仍将在下一节证明我们的模型具有足够的表现力。

4.4. 表现力分析

与经典的 GNN 相比，图变换器的表达能力几乎没有

表 1：SAT 与 SOTA 方法在图回归和分类任务上的比较。ZINC 结果在适用情况下使用边权重，否则不使用边权重。

^表示我们通过改编原论文提供的代码自己获得的结果。† 表示性能指标越高越好；↓ 表示越低越好。

	锌 ↓	集群 †	模式 †
# GRAPHS	12,000	12,000	14,000
平均值# 节点数	23.2	117.2	118.9
平均值# 边数	49.8	4,303.9	6,098.9
公制	MAE	准确性	准确性
GIN	0.*387*±0.015	64.*716*±*1*.553	85.*590*±0.011
GAT	0.*384*±0.007	70.*587*±0.447	78.*271*±0.186
PNA	0.*188*±0.004	67.*077*±0.977[s]	86.*567*±0.075
变压器+RWPE	0.*310*±0.005	29.*622*±0.176	86.*183*±0.019
图形转换器	0.*226*±0.014	73.*169*±0.622	84.*808*±0.068
SAN	0.*139*±0.006	76.*691*±0.650	86.*581*±0.037
GRAPHORMER	0.*122*±0.006	-	-
K 子树 SAT	0.*102*±0.005	77.*751*±0.121	**86.*865*±0.043**
K 子图 SAT	**0.*094*±0.008**	**77.*856*±0.104**	86.*848*±0.037

这样我们就有可能研究输出表征的表现力。更具体地说，我们正式证明了来自结构感知注意力层的节点表征至少与其子图表征一样具有表现力
的可注入性，由结构提取器给出。

与查询相关的注意力函数：

人研究过，因为绝对编码中引入的软结构归纳偏差通常很难描述。由于我们的 SAT 依赖于子图结构提取器的独特设计，它可以

定理 2. *假设节点属性空间是可数的。对于两个图 $G = (V, E, \mathbf{X})$ 和 $G' = (V', E', \mathbf{X}')$ 中的任意一对节点 v 和 v'，假设 V 中存在一个节点 u_1，使得对于任意 $w \in V$，$x_{u1} = x_w$；V 中存在一个节点 u_2，使得对于任意 $w \in V$，其子图表示 $j(u_2, G) = j(w, G)$。那么，存在一组参数和一个映射 f：* $\mathbb{R}^d \to \mathbb{R}^{d_{out}}$，*使得它们在结构感知注意力后的表示不同，即 $SA\text{-}attn(v) \neq SA\text{-}attn(v')$，如果它们的子图表示不同，即 $\phi(v, G) \neq \phi(v', G')$。*

请注意，定理中的假设是温和的，因为我们总是可以添加一些绝对编码或随机噪音，使一个节点的属性与其他所有节点不同，子图表示也是如此。在对 GNN 进行表达式分析时，一般会采用可数假设（*如* Xu 等人 (2019)）。我们假设 f 是任意映射，而不仅仅是自我关注函数定义中的线性函数，因为通过通用近似定理（Hornik，1991 年），在多层变换器中，它实际上可以被 FFN 近似。定理 2 表明，如果结构提取器具有足够的表现力，那么得到的 SAT 模型也至少具有同样的表现力。此外，表现力更强的提取器可以产生表现力更强的 SAT 模型，从而获得更好的预测性能，这一点在第 5 节中也得到了经验证实。

表 2：OGB 数据集上 SAT 与 SOTA 方法的比较。

	OGBG-PPA †	OGBG-CODE2 †
# GRAPHS	158,100	452,741
平均值# 节点数	243.4	125.2
平均值# 边数	2,266.1	124.2
公制	准确性	F1 分数
GCN	0.*6839±0*.0084	0.*1507±0*.0018
GCN 虚拟节点	0.*6857±0*.0061	0.*1595±0*.0018
GIN	0.*6892±0*.0100	0.*1495±0*.0023
GIN 虚拟节点	0.*7037±0*.0107	0.*1581±0*.0026
DEEPERGCN	0.*7712±0*.0071	-
EXPC	**0.*7976±0*.0072**	-
变压器	0.*6454±0*.0033	0.*1670±0*.0015
GRAPHTRANS	-	0.*1830±0*.0024
K 子树 SAT	0.*7522±0*.0056	**0.*1937±0*.0028**

5. 实验

在本节中，我们将评估 SAT 模型与用于图表示学习的几种 SOTA 方法（包括 GNN 和 Transformers）在五个图和节点预测任务上的对比情况，并分析我们架构的不同组件，以确定是什么驱动了性能的提升。总之，我们发现了 SAT 的以下几个方面：

- 结构感知框架在图和节点分类任务上实现了 SOTA 性能，超过了 SOTA 图变换器和稀疏 GNN。
- SAT 的两个实例，即 *k*-子树 SAT 和 *k*-子图 SAT，*总是*比建立在基础 GNN 上的 SAT 有所改进，这突出表明我们的结构感知方法提高了表达能力。
- 我们的研究表明，通过我们的结构感知注意力将结构纳入其中，相对于仅使用节点属性相似性而非结构相似性的 RWPE vanilla Transformer，带来了显著的改进。我们还证明，*k* 值越小，性能越好，同时不会出现过度平滑或过度扭曲的问题。
- 我们的研究表明，选择适当的绝对位置编码和读出方法可以提高性能，但其效果远不如将结构纳入方法。

CLUSTER（Dwivedi 等人，2020 年）、PATTERN（Dwivedi 等人，2020 年）、OGBG-PPA（Hu 等人，2020a）和 OGBG- CODE2（Hu 等人，2020a）。

我们将我们的方法与下列 GNN 进行了比较：GCN (Kipf & Welling, 2017)、GraphSAGE (Hamilton et al., 2017)、
GAT（Velic˘kovic´等人，2018年）、GIN（Xu等人，2019年）、

此外，我们注意到，SAT 只考虑了很小的超参数搜索空间，就实现了 SOTA 性能。通过调整更多的超参数，性能可能会进一步提高。

5.1. 数据集和实验设置

我们用五个中大型节点和图属性预测基准数据集评估了我们方法的性能，其中包括 ZINC（Dwivedi 等人，2020 年）、

PNA (Corso 等人，2020)、DeeperGCN (Li 等人，2020a) 和 ExpC (Yang 等人，2022)。我们的比较伙伴还包括最近提出的几种图变换器，包括带有 RWPE 的原始变换器（Dwivedi 等人，2022 年）、Graph Transformer（Dwivedi 和 Bresson，2021 年）、SAN（Kreuzer 等人，2021 年）、Graphormer（Ying 等人，2021 年）和 GraphTrans（Jain 等人，2021 年），后者是一种在 GNN 上使用普通变换器的模型。

所有比较方法的结果均来自原始论文，如果没有，则来自 Dwivedi 等人（2020 年）。我们考虑了配备不同 GNN 提取器的 k *子树*和 k *子图* SAT，包括 GCN、GIN、GraphSAGE 和 PNA。对于 OGBG-PPA 和 OGBG- CODE2，由于需要大量内存，我们没有对 k *子图* SAT 模型进行实验。有关数据集、实验设置和超参数的全部细节见附录。

5.2. 与最新方法的比较

我们在表 1 和表 2 中展示了 SAT 与其他 GNN 和 Transformers 的性能比较。在这些数据集上，SAT 模型的性能始终优于 SOTA 方法，这表明它有能力将 GNN 和变换器的优势结合起来。特别是在 CODE2 数据集上，我们的 SAT 模型在参数数量相对较少和超参数调整最小的情况下，以较大的优势超过了 SOTA 方法，这将使其在 OGB 排行榜上名列前茅。

5.3. SAT 模型与稀疏 GNN 比较

表 3 总结了 SAT 在不同 GNN 中相对于用于提取子图表示的稀疏 GNN 的性能。我们注意到，SAT 的两个变体都持续为其基础 GNN 带来了巨大的性能提升，使其成为任何 GNN 模型的系统增强器。此外，PNA 是我们考虑过的最具表现力的 GNN，在与 SAT 配合使用时始终表现最佳，这从经验上验证了我们在第 4.4 节中的理论发现。

5.4. 超参数研究

表 3 展示了 SAT 相对于稀疏 GNN 的附加值，而我们现在要剖析的是 SAT 的各个组成部分。

表 3：由于 SAT 使用 GNN 来提取结构，我们将原始稀疏 GNN 的性能与使用该 GNN（"基础 GNN"）的 SAT 的性能进行了比较。在不同的 GNN 选择中，我们发现 *k 子树*和 *k 子图* SAT 的性能总是优于其使用的原始稀疏 GNN。评估指标与表 1 相同。

		锌↓		CLUSTER†	模式†
		W/ EDGE ATTR.	W/O EDGE ATTR.	全部	全部
GCN	基础 GNN	0.192±0.015	0.367±0.01168.498±0.976	71.892±0.334	
	K-SUBTREE SAT	0.127±0.010	**0.174±0.009**	77.247±0.094	86.749±0.065
	K-SUBGRAPH SAT	0.114±**0.005**	0.184±0.002	**77.682±0.098**	**86.816±0.028**
GIN	基础 GNN	0.209±0.009	0.387±0.01564	.716±1	.553 85.590±0.011
	K-SUBTREE SAT	0.115±0.005	0.166±0.00777.255±0.085		**86.759±0.022**
	K-SUBGRAPH SAT	0.095±**0.002**	**0.162±0.01377**	.502±0.**282** 86	.746±0.014
图形	base gnn-0.398±0.002 63.844±0.110 50.516±0.001 k-subtree sat-			**0.164±0.004**	
	77.592±0.074 86.818±0.043 K-subgraph Sat-			0.168±0.005	**77.657±0.185**
	8 6 . 838±0.010				
PNA	基础 GNN	0.188±0.004	0.320±0.03267	.077±0.977 86.567±0.075	
	K-SUBTREE SAT	0.102±0.005	0.147±0.00177.751±0.121		**86.865±0.043**
	K-SUBGRAPH SAT	0.094±**0.008**	**0.131±0.00277.856±0.104**	86.848±0.037	

对 ZINC 数据集进行 SAT 分析，以确定架构的哪些方面能带来最大的性能提升。

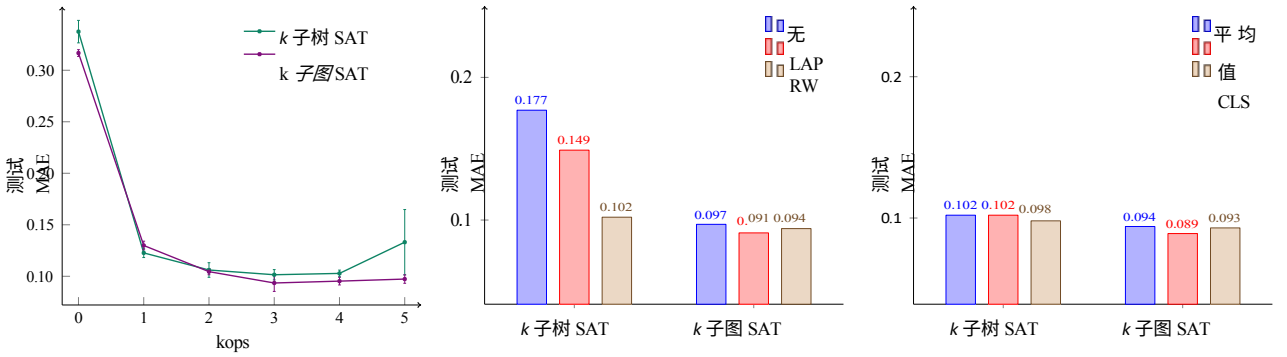
SAT **中** *k 的***影响** SAT 的主要贡献在于它能够明确地将结构信息纳入自我注意。在这里，我们试图证明这种信息提供了关键的预测信息，并研究 *k 的*选择如何影响结果。图 3a 显示了在 ZINC 数据集上使用 PNA 的 *k 子树*提取器和 *k* 子图提取器在不同 k 的情况下对测试 MAE 的影响。$k = 0$ 对应于 vanilla Transformer 仅使用绝对位置编码，*即*不使用结构。我们发现，采用结构化方法可大幅提高性能，*k 子树*和 *k 子图*提取器的最佳性能都在 $k = 3$ 左右。当 k 增加到 $k = 4$ 以上时，*k*-子树提取器的性能下降，这与观察到的现象一致，即 GNN 在较浅的网络中效果最佳（Kipf & Welling，2017 年）。我们观察到，*k-子图*并没有受到这个问题的影响，这凸显了其有用性的一个新方面。另一方面，*k*-子树提取器的计算效率更高，可扩展至更大的 OGB 数据集。

绝对编码的效果 我们在此评估绝对编码是否为 SAT 带来了补充信息。在图 3b 中，我们进行了一项消融研究，显示了带有和不带有绝对位置编码（包括 RWPE 和拉普拉斯 PE）的 SAT 的结果（Dwivedi 等人，2020 年）。带有位置编码的 SAT 性能优于不带位置编码的 SAT，这证实了两种编码的互补性。不过，我们也注意到，绝对编码带来的性能增益远小于使用结构感知注意力所带来的增益，如图 3a 所示（比较 $k = 0$ 与 $k > 0$ 的实例），这强调了结构感知注意力是该模型更重要的方面。

读出方法比较 最后，我们在图 3c 中比较了在 ZINC 数据集上使用不同读出方法聚合节点级表示的 SAT 模型的性能，包括第 4.2 节中讨论的 CLS 池化。与 GNNs 中读出方法的显著影响不同（Xu 等人，2019 年），我们观察到 SAT 模型的影响很小。

5.5. 模型解读

除了性能提升之外，我们还发现，与只有绝对位置编码的经典变换器相比，SAT 提供了更好的模型可解释性。我们在突变性数据集上分别训练了一个 SAT 模型和一个带有 CLS 读出的 Transformer，并在图 4 中展示了 [CLS] 节点与 SAT 和 Transformer 学习到的其他节点之间的关注度得分。这两个模型的显著区别在于 SAT 具有结构感知节点嵌入，因此我们可以将以下可解释性的提高归功于此。虽然两种模型都能识别出一些已知的致突变化学主题，如 NO_2 和 NH_2，但 SAT 学习到的关注度分数更稀疏、信息量更大，这意味着 SAT 对这些已知致突变主题的关注度权重要高于带有 RWPE 的 Transformer。普通的 Transformer 甚至没有关注一些重要的原子，如 NH_2 基团中的 H 原子。SAT 突出显示的唯一 H 原子是 NH_2 组中的 H 原子，这表明我们的 SAT 确实考虑到了结构。由于更加关注这些具有鉴别作用的图案，因此 SAT 模型受数据集中常见的其他化学图案（如苯）的影响较小，从而全面提高了性能。更多结果见附录。



(a) k 的影响　(b) 绝对编码的效果　(c) 读数方法的影响

图 3：我们对 ZINC 数据集 SAT 性能的不同驱动因素进行了分析（越低越好）。在图 3a 中，我们展示了 k 的大小变化对性能的影响（k=0 相当于不具备结构感知能力的普通变换器）。图 3b 显示了不同绝对编码方法的影响，图 3c 显示了不同读出方法的影响。

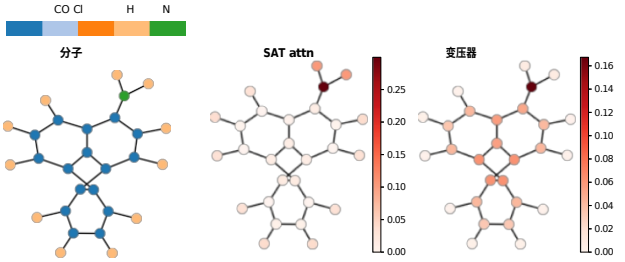


图 4：SAT 和类变换器的注意力可视化。中间一列显示的是 SAT 模型学习到的 [CLS] 节点的注意力权重，右边一列显示的是采用随机走位编码 (RWPE) 的类变换器学习到的注意力权重。

6. 讨论

我们引入了 SAT 模型，该模型成功地将结构信息纳入了 Transformer 架构，克服了绝对编码的局限性。除了以最小的超参数调整获得 SOTA 经验性能外，SAT 还提供了比 Transformer 更好的可解释性。

局限性 如上所述，*k-子图*SAT 对内存的要求高于 *k-子树*SAT，如果访问大内存 GPU 的能力受到限制，其适用性就会受到限制。我们看到 SAT 的主要局限性在于它与 Transformer 存在相同的缺点，即自我关注计算的二次复杂性。

例如 Horn 等人（2021 年）最近推出的拓扑 GNN。此外，SAT 框架非常灵活，可以纳入任何能生成结构感知节点表示的结构提取器，甚至可以扩展到使用 GNN 之外，如可微分图核。

未来工作的另一个重要领域是集中精力降低高内存成本和时间复杂性。

未来工作 由于 SAT 可以与 y GNN 相结合，我们工作的一个自然延伸就是将 SAT 与结构提取器相结合，而结构提取器已被证明是严格意义上的

正如最近开发所谓线性变压器的努力所做的那样，这种变压器在时间和空间要求上都具有线性复杂性（Tay 等人，2020；Wang 等人，2020；Qin 等人，2022）。

致谢

这项工作部分得到了阿尔弗雷德-克虏伯-冯-博伦和哈尔巴赫基金会的阿尔弗雷德-克虏伯大学青年教师奖（K.B.）的支持。作者还要感谢 Bastian Rieck 博士和 Carlos Oliver 博士对手稿提出的宝贵意见，这些意见极大地改进了手稿。

参考资料

Abbe, E. 群落检测与随机块模式：Recent developments.*机器学习研究期刊》（JMLR）*，18（177）：1-86，2018.(Cited on 17)

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications.*国际学习表征会议（ICLR）*，2021 年。(引用 1)

Alsentzer, E., Finlayson, S. G., Li, M. M., and Zitnik, M. Subgraph neural networks.子图*神经网络。*

信息处理系统》，NeurIPS，2020 年。(引用于 21)

Bodnar, C., Frasca, F., Otter, N., Wang, Y. G., Lio`, P., Mont- ufar, G. F., and Bronstein, M. Weisfeiler and lehman go cellular：Cw 网络。In *Advances in Neural Infor- mation Processing Systems (NeurIPS)*, 2021.(被引用 5 次)

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view.2020 年美国*人工智能学会（AAAI）人工 智能会议论文集。*(引用 1)

Corso, G., Cavalleri, L., Beaini, D., Lio`, P., and Velic˘kovic´、

P.图 网的主邻域聚合。*神经信息处理系统进展（NeurIPS）*，2020 年。(引用 2, 7)

Dosovitskiy, A., Beyer, L., Kolesnikov, A., W eissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words：规模图像识别变换器。*国际学习表征会议（ICLR）*，2020 年。(Cited on 1)

Dwivedi, V. P. and Bresson, X. A generalization of trans- former networks to graphs.In *AAAI Workshop on Deep Learning on Graphs：方法与应用》*，2021 年。(引用于 2, 3, 7)

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *ArXiv preprint arXiv:2003.00982*, 2020.(引用于 7, 8, 16, 17)

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bres- son, X. Graph neural networks with learnable structur al and positional representations.In *International Confer- ence on Learning Representations*, 2022.(引用于 2, 3、5, 7, 17)

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., and Y in、

D.用于社交推荐的图神经网络在*万维网大会*，2019 年。(引用 1)

Gao, B. and Pavel, L. On the properties of the softmax func- tion with application in game theory and reinforcement learning. *ArXiv preprint arXiv:1704.00805*, 2017.(Cited on 14)

Gao, H. and Ji, S. Graph u-nets.*国际机器学习大会》*，第 2083-2092 页，2019 年。(被引用 21 次)

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Re gep、C., Liu, G., Hayter, J. B., Vickers, R., Roberts, C., Tang, J., et al.

发现和开发。*生物信息学简报》*，22（6）：bbab159，2021。(引用 1)

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chem- istry.*国际机器学习会议（ICML）*，2017 年。(引用 1, 2）

Hamilton, W. L., Ying, R., and Leskovec, J. I n ductive representation learning on large graphs.In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.(引用 2, 7）

Horn, M., De Brouwer, E., Moor, M., Moreau, Y., Rieck, B., and Borgwardt, K. Topological graph neural networks.2021.(被引用9次)

Hornik, K. 多层前馈网络的逼近能力。*神经网络*，4（2）：251-257，1991.

(引自 6、15）

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark：图上机器学习的数据集。*神经信息处理系统进展（NeurIPS）*，2020a。(7, 16, 17引用）

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neu- ral networks.*国际学习表征会议（ICLR）*，2020b。(引用 2)

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Gener- ative models for graph-based protein design.In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.(Cited on 1)

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. Zinc: A free tool to discover chem- istry for biology.*化学信息与建模期刊》*，52（7）：1757-1768，2012 年。(引用 16 次）

Jain, P., Wu, Z., Wright, M., Mirhoseini, A., Gonzalez, J. E., and Stoica, I. Representing longrange context for graph neural networks with global attention.*神经信息处理系统进展（NeurIPS）*，2021 年。(引用 3、5、7）

Kersting, K., Kriege, N. M., Morris, C., Mutzel, P., and Neumann, M. Benchmark data sets for graph kernels, 2016. http://graphkernels.cs. tu-dortmund.de.(引用次数 20)

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks.In *International Conference on Learning Representations (ICLR)*, 2017.(引用于 2、7、8）

Kreuzer, D., Beaini, D., Hamilton, W. L., Le´tourneau, V ., and Tossou, P. Rethinking graph transformers with spec- tral attention.*神经信息处理系统进展（NeurIPS ）*，2021 年。(引用于 2, 7)

Li，G.、Muller，M.、Thabet，A. 和 Ghanem，B. Deepgcns：全球计算机网络能否像全球网络一样深入？*计算机视觉国际会议（ICCV）论文集*，2019 年。(Cited on 1)

Li, G., Xiong, C., Thabet, A., and Ghanem, B. Deepergcn ：2020a.(引用于 2, 7)

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding：为图形表示学习设计更强大的神经网络。*神经信息处理系统进展（NeurIPS）*， 2020b.(引用 2)

Li, Q., Han, Z., and Wu, X. Deeper insights into graph convolutional networks for semi-supervised learning.2018年*AAAI人工 Intelli-gence大会论文集*。(引用次数 1)

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts.*国际学习表征会议（ICLR ）*，2016 年。(引用次数 17)

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization.*国际学习表征会议（ICLR）*，2018年。(引用次数：17)

Mesquita, D., Souza, A. H., and Kaski, S. 图神经网络中的池化反思。*神经信息处理系统进展（NeurIPS）*， 2020 年。(引用次数 21)

Mialon, G., Chen, D., Selosse, M., and Mairal, J. Graphit ：变压器中的图形结构编码，2021 年。(引用 3、5)

Micchelli, C. A., Xu, Y., and Zhang, H. Universal k *ernels.机器学习研究期刊》（JMLR）*，7（12），2006 年。(引用次数 16)

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen 、
J.E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks.In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.(Cited on 1)

Nikolentzos, G. and Vazirgiannis, M. Random walk graph neural networks.*神经信息处理系统进展（NeurIPS）*，2020 年。(引用次数 5)

Oono, K. and Suzuki, T. Graph neural networks e xponentially lose expressive power for node classification.*国际学习表征会议（ICLR）*，2020 年。(引用 1)

Qin, Z., Sun, W., Deng, H., Li, D., Wei, Y., Lv, B., Yan, J.. 、

Kong，L. 和 Zhong，Y. cosformer：重新思考注意力中的软最大值。*学习表征国际会议*，2022 年。(被引用9次)

Rives, A.、Meier, J.、Sercu, T.、Goyal, S.、Lin, Z.、Liu, J.、Guo, D.、Ott, M.、Zitnick, C. L.、Ma, J.等人. 将无监督学习扩展到 2.5 亿个蛋白质序列所产生的生物结构和功能。*美国国家科学院院刊》*，118（15），2021。(引用 1)

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., and Huang, J. Self-supervised graph transformer on large scale molecular data.*神经信息处理系统进展（NeurIPS）*，2020 年。(被引用 3 次）

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representation.*计算语言学协会北美分会论文集》（NAACL）*，2018 年。(引用 2)

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. Masked label prediction：用于半监督分类的统一信息传递模型。In Zhou, Z.-H. (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pp.国际人工智能联合会议组织，2021 年 8 月。(Cited on 3)

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. Efficient transformers：*arXiv preprint arXiv:2009.06732*, 2020.(Cited on 9)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Atten- tion is all you need.In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.(引用 1、2、3）、
17)

Velic˘kovic´, P., Cucurull, G., Casanova, A., Romero, A., Lio`, P., and Bengio, Y. Graph Attention Networks.*国际学习表征会议（ICLR）*，2018。(引用于 2, 7)

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer：具有线性复杂性的自我关注，2020 年。(引用 9）

Wijesinghe, A. and Wang, Q. A new perspective on "how graph neural networks go beyond weisfeiler-lehman?".*学习表征国际会议*，2022 年。(被引用 4 次）

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks?*国际学习表征会议（ICLR）*，2019 年。(引自 1、2、4、6、7、8、13、15）

Yang, M., Wang, R., Shen, Y., Qi, H., and Yin, B. Breaking the expression bottleneck of graph neural networks.doi: 10.1109/TKDE.2022.3168070.(引用于 2, 7)

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y.变换器在图表示方面真的表现糟糕吗？*神经信息处理系统进展（NeurIPS）*，2021 年。(引用 3, 7）

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling.*神经信息处理系统进展》*，31，2018.(引用次数 21)

You, J., Ying, R., and Leskovec, J. Position-aware graph neural networks.*国际机器学习大会（ICML）*，2019 年。(被引用2次)

Zhang, J., Zhang, H., Xia, C., and Sun, L. Graph-bert: Only attention is needed for learning graph representations. *ArXiv preprint arXiv:2001.05140*, 2020.(被引用2次)

嵌套图神经网络。*第 35 届神经信息处理系统（NeurIPS）大会论文集*，2021 年。(引用于 4, 5）

附录

本附录提供理论和实验材料，其组织结构如下：A 节详细介绍了图神经网络的背景。B 节介绍定理 1 和 2 的证明。C 节提供实验细节和其他结果。D 节提供模型解释细节和其他可视化结果。

A. 图神经网络的背景

图神经网络的总体思路是，通过整合节点邻居发送的信息，迭代更新节点的嵌入。Xu 等人（2019）通过将不同的框架概括为 AGGREGATE、COMBINE 和 READOUT 步骤，为这一过程所包含的步骤提供了一个总体框架。各种不同的 GNN 通常可以理解为这三种功能的变化。对于给定的层 l，AGGREGATE 步骤汇总（例如使用总和或平均值）给定节点的邻居表示，然后在 COMBINE 步骤中将其与上一层中给定节点的表示相结合。然后再使用非线性函数（如 ReLU），将更新后的节点表征传递到下一层。网络有多少层，这两个步骤就重复进行多少次。值得注意的是，这两个步骤的输出所提供的节点表示法只考虑了大小增加 1 的局部子结构，因此需要一个非常深的网络来捕捉给定节点与所有其他节点之间的交互（深度不应小于图的直径）。在网络结束时，READOUT 函数会提供一个汇集函数，将表示转换为适当的输出级粒度（如节点级或图级）。AGGREGATE 和 READOUT 两个步骤都必须不受节点排列的影响。

B. 理论分析

B.1. 结构感知注意力表征的可控性

定理 1. *假设 f 是一个 Lipschitz 映射，其 Lipschitz 常量用 $Lip(f)$ 表示，且结构提取器 ϕ 在子图空间上以常数 C_φ 为界。对于节点数相同的两个图 $G = (V, E, \mathbf{X})$ 和 $G' = (V', E', \mathbf{X}')$ 中的任意一对节点 v 和 v'，在经过结构感知关注后，它们的表示之间的距离有以下约束：*

$$\|SA\text{-}attn(v) - SA\text{-}attn(v')\| \leq C_1[\|h_v - h_{v'}\| + D(\mathbf{H}, \mathbf{H}')] + C_2 D(\mathbf{X}, \mathbf{X}'), \tag{11}$$

其中 $h_w := \phi(w, G)$ 表示任意 $w \in V$ 的节点 w 处的子图表示，$h_{w'} := \phi(w', G')$ 同样。$\mathbf{H} = (h_w)_{w \in V}$ 和 $\mathbf{H}' = (h_{w'})_{w' \in V'}$ 分别表示 G 和 G' 中子图表示的多集。用 $\Pi(V, V')$ 表示 V 和 V' 之间的置换集，D 是两个具有相同心数的表示多集之间的匹配度量，定义为

$$D(\mathbf{X}, \mathbf{X}') := \inf_{\pi \in \Pi(V, V')} \sup_{w \in V} \|x_w - x_{\pi(w)}\|$$

C_1 和 C_2 是由以下公式给出的常数：

$$C_1 = \frac{2}{n_{out}} Lip(f) C_\varphi \|\mathbf{W}_Q\|_\infty \|\mathbf{W}_K\|_\infty, C_2 = Lip(f)$$

证明 让我们用

$$z_v = (\langle \mathbf{W} h_{Q_v}, \mathbf{W} h_{K_w} \rangle)_{w \in V} \in \mathbb{R}^n, z_{v'} = (\langle \mathbf{W} h_{Q_{v'}}, \mathbf{W} h_{K_{w'}} \rangle)_{w' \in V'}$$

∈ ℝ 、 ⁿ

对于任意 $z \in \mathbb{R}^n$，其第 i 个系数为 $\mathrm{softmax}(z)_i \in \mathbb{R}$。

$$\mathrm{softmax}(z)_i = \frac{\exp(z_i / \sqrt{d_{\text{out}}})}{\sum_{j=1}^{n} \exp(z_j / \sqrt{d_{\text{出}}})}.$$

那么，我们有

$$
\begin{aligned}
&\|\text{SA-Attn}(v) - \text{SA-Attn}(v')\| \\
&= \Big\| \sum_{w \in V} \text{softmax}(z_v)_w f(x_w) - \sum_{w' \in V'} \text{softmax}(z'_v)_{w'} f(x'_{w'}) \Big\| \\
&= \Big\| \sum_{w \in V} (\text{softmax}(z_v)_w - \text{softmax}(z'_v)_{\pi(w)}) f(x_w) + \sum_{w \in V} \text{softmax}(z'_v)_{\pi(w)} f(x_w) - \sum_{w' \in V'} \text{softmax}(z'_v)_{w'} f(x'_{w'}) \Big\| \\
&\le \sum_{w \in V} |\text{softmax}(z_v)_w - \text{softmax}(z'_v)_{\pi(w)}| \|f(x_w)\| + \sum_{w \in V} \text{softmax}(z'_v)_{\pi(w)} \|f(x_w) - f(x'_{\pi(w)})\| \\
&\le \sum_{w \in V} |\text{softmax}(z_v)_w - \text{softmax}(z'_v)_{\pi(w)}| \|f(x_w)\| + \sum_{w \in V} \text{softmax}(z'_v)_{\pi(w)} \text{Lip}(f) \|x_w - x'_{\pi(w)}\|
\end{aligned}
$$

其中 $\pi : V \to V'$ 是任意排列，我们使用了三角不等式。现在我们需要分别约束这两个项。我们首先约束第二项：

$$
\begin{aligned}
\sum_{w' \in V'} \text{softmax}(z'_v)_{w'} \|f(x_{\pi^{-1}(w')}) - f(x'_{w'})\| &\le \sum_{w' \in V'} \text{softmax}(z'_v)_{w'} \|f(x_{\pi^{-1}(w')}) - f(x'_{w'})\| \\
&\le \sum_{w' \in V'} \text{softmax}(z'_v)_{w'} \text{Lip}(f) \|x_{\pi^{-1}(w')} - x'_{w'}\| \\
&= \text{Lip}(f) \sum_{w' \in V'} \text{softmax}(z'_v)_{w'} \|x_{\pi^{-1}(w')} - x'_{w'}\| \\
&\le \text{Lip}(f) \sup_{w' \in V'} \|x_{\pi^{-1}(w')} - x'_{w'}\| \\
&= \text{Lip}(f) \sup_{w \in V} \|x_w - x'_{\pi(w)}\|
\end{aligned}
$$

其中第一个不等式是三角不等式，第二个不等式利用了 f 的 Lipschitzness。对于第一项，我们可以通过以下方法对其进行上界计算

$$
\begin{aligned}
&\sum_{w \in V} |\text{softmax}(z_v)_w - \text{softmax}(z'_v)_{\pi(w)}| \|f(x_w)\| \\
&\le \sup_{w \in V} \|f(x_w)\| \cdot \|\text{softmax}(z_v) - \text{softmax}((z'_v)_\pi)\|_1 \\
&\le \sup_{w \in V} \|f(x_w)\| \cdot \sqrt{d_{\text{out}}} \|\text{softmax}(z_v) - \text{softmax}((z'_v)_\pi)\|_2 \\
&\le \sup_{w \in V} \|f(x_w)\| \cdot \sqrt{d_{\text{out}}} \cdot \frac{1}{2} \|z_v - (z'_v)_\pi\|_2 \\
&\le \sup_{w \in V} \|f(x_w)\| \cdot \sqrt{d_{\text{out}}} \cdot \frac{1}{2} \|z_v - (z'_v)_\pi\|_2 \cdot \text{Lip}(f)
\end{aligned}
$$

其中，滥用符号，$(z)_\pi \in \mathbb{R}^n$ 表示对于任意 $z \in \mathbb{R}^n$，*w-th* 项为 $z_{\pi(w)}$ 的向量。第一个不等式来自简单的矩阵规范不等式，第二个不等式*使用了* softmax 函数为 $1/d_{\text{out}}$ - Lipschitz 的事实（参见 Gao & Pavel (2017)）。那么，我们有

$$
\begin{aligned}
\|z_v - (z'_v)_\pi\|^2 &= \sum_{w \in V} \langle \mathbf{W_Q} h_v, \mathbf{W_K} h_w \rangle - \langle \mathbf{W_Q} h'_v, \mathbf{W_K} h'_{\pi(w)} \rangle^2 \\
&= \sum_{w \in V} \langle \mathbf{W_Q} h_v, \mathbf{W_K} (h_w - h'_{\pi(w)}) \rangle + \langle \mathbf{W_Q} (h_v - h'_v), \mathbf{W_K} h'_{\pi(w)} \rangle^2 \\
&\le 2 \sum_{w \in V} \langle \mathbf{W_Q} h_v, \mathbf{W_K} (h_w - h'_{\pi(w)}) \rangle^2 + \langle \mathbf{W_Q} (h_v - h'_v), \mathbf{W_K} h'_{\pi(w)} \rangle^2 \\
&\le 2 \sum_{w \in V} \|\mathbf{W_Q}\|^2 \|\mathbf{W_K} (h_w - h'_{\pi(w)})\|^2 + \|\mathbf{W_Q} (h_v - h'_v)\|^2 \|\mathbf{W_K} h'_{\pi(w)}\|^2
\end{aligned}
$$

$$\leq 2 \sum_{w \in V} C_{\varphi}^{2} \|W_Q\|^2 \cdot \|W_K\|^2 \cdot \|h_w - h'_{\pi(w)}\|^2 + \|W_Q\|^2 \cdot \|h_v - h'_{\pi(w)}\|^2 \leq 2C_{\varphi}^{2} \|W_K\|^2 \cdot$$

$$\leq 2nC_{\varphi}^{2} \|W_Q\|^2 \cdot \|W_K\|^2 \cdot \|h_v - h'_v\|^2 + \sup_{w \in V} \|h_w - h'_{\pi(w)}\|^2,$$

其中第一个不等式来自 $(a+b)^2 \le 2(a^2+b^2)$，第二个不等式使用了考希-施瓦茨不等式，第三个不等式使用了谱规范的定义和结构提取函数的约束。然后，我们得到以下不等式

$$\begin{aligned}&\Big\| \sum_{w \in V} (\mathrm{softmax}(z)_{vw} - \mathrm{softmax}(z')_{\pi(w)}) f(x_w) \Big\| \\ &\le \sqrt{2}\, n \mathrm{Lip}(f) C_{\text{输出}} \Big(\|W_Q\|_\infty \|W_K\|_\infty \|h_v - h'_{v'}\| + \sup_{w \in V} \|h_w - h'_{\pi(w)}\| \Big)\end{aligned}$$

综合第一项和第二项的上限，我们就得到了结构感知注意力表征之间距离的上限：

$$\|\text{SA-attn}(v) - \text{SA-attn}(v')\| \le C_1 \|h_v - h'_{v'}\| + \sup_{w \in V} \|h_w - h'_{\pi(w)}\| + C_2 \sup_{w \in V} \|x_w - x'_{\pi(w)}\|,$$

对于任意排列 $\pi \in \Pi(V, V')$，其中

$$\begin{aligned}C_1 &= \sqrt{2}\, n \mathrm{Lip}(f) C_{\text{输出}} \|W_Q\|_\infty \|W_K\|_\infty \\ C_2 &= \mathrm{Lip}(f).\end{aligned}$$

最后，通过取排列集合的下极值，我们就得到了定理中的不等式。 □

B.2. 表现力分析

在此，我们假定 f 可以是任何连续映射，在实际应用中，可以通过通用近似定理（Hornik，1991 年）用 MLP 网络对其进行近似。

定理 2. *假设节点属性空间 $\mathcal{X}$ 是可数的。对于两个图中的任意一对节点 v 和 v' $G = (V, E, \mathbf{X})$，且 $G' = (V', E', \mathbf{X}')$，假设 V 中存在一个节点 u_1，对于任意 $w \in V$ 而言，$x_u \neq x_w$，且 V 中的节点 u_2，对于任意 $w \in V$，其子图表示 $\phi(u_2, G) \neq \phi(w, G)$。那么，存在一组参数和一个映射 $f : \mathcal{X} \to \mathbb{R}^d$ 出，使得它们在结构感知注意力后的表示是不同的、即 $\text{SA-attn}(v) \neq \text{SA-attn}(v')$，如果它们的子图表示不同，即 $\phi(v, G) \neq \phi(v', G')$。*

证明 本定理相当于证明了原始点积注意关于查询的注入性，即证明了

$$\mathrm{Attn}(h, x, G) = \sum_{u \in V} \frac{\kappa_{\exp}(h_v, h_u)}{\sum_{w \in V} \kappa_{\exp}(h_v, h_w)} f(x_u)$$

在 h_v 中是注入的，其中

$$\kappa_{\exp}(h, h') := \exp\langle W_Q h + b_Q, W_K h' + b_K \rangle / \sqrt{d_{\text{out}}}。 \tag{12}$$

这里我们考虑公式 (1) 中省略的偏移项。让我们来证明定理的反证。我们假设对于任意参数集和任意映射 f，$\mathrm{Attn}(h_v, x_v, G) = \mathrm{Attn}(h'_{v'}, x'_{v'}, G')$，并想证明 $h_v = h'_{v'}$。

在不失一般性的前提下，我们假设 G 和 G' 的节点数相同，即 $V = V' = n$。否则，我们可以很容易地在较小的图中添加一些虚拟孤立节点。现在，如果我们取 $W_Q = W_K = 0$，那么所有的软最大系数都将相同，我们有

$$\sum_{w \in V} f(x_w) = \sum_{w \in V} f(x'_w).$$

因此，根据 Xu 等人（2019）的 Lemma 5，存在一个映射 f，使得多集 X 和 X' 完全相同。

因此，我们可以用一个序列 V 来重新枚举两个图中的节点（滥用符号，我们继续使用 V 这里），使得对于任意 $u \in V$ 而言，$x_u = x'_u$。那么，我们可以将等式 $\mathrm{Attn}(h_v, x_v, G) = \mathrm{Attn}(h'_v, x'_v, G'_v)$ 改写为

$$\sum_{u \in V} \left(\frac{\kappa_{\exp}(h_v, h_u)}{\sum_{w \in V} \kappa_{\exp}(h_v, h_w)} - \frac{\kappa_{\exp}(h'_v, h'_u)}{\sum_{w \in V} \kappa_{\exp}(h'_v, h'_w)} \right) f(x_u) = 0.$$

现在，由于 V 中存在一个节点 u_1，其属性与其他所有节点都不同，*即*对于任意 w，$x_u 1 \neq x_w \in V$ *中*，我们可以找到一个映射 f，使得 $f(x_u 1)$ 不在 $(f(x))_{w\in V, w\neq u} 1$ 的跨度中。那么，根据它们的独立性，我们有

$$\sum_{w\in V} \frac{\kappa_{\exp}(h_v, h_u)}{\kappa_{\exp}(h_v, h_w)} = \sum_{w\in V} \frac{\kappa_{\exp}(h'_v, h'_{u1})}{\kappa_{\exp}(h'_v, h'_w)},$$

对于任意 $\mathbf{W_Q}$, $\mathbf{W_K}$, b_Q 和 b_K。

一方面，如果我们取 $\mathbf{W_Q} = 0$，那么对于任意 $\mathbf{W_K}$、b_Q 和 b_K，我们可以得到

$$\frac{\exp(\langle b_Q, \mathbf{W}\, \mathbf{h}_{Ku} + b_K \rangle / \sqrt{d_{out}})}{\sum_{w\in V} \exp(\langle b_Q, \mathbf{W}\, \mathbf{h}_{Kw} + b_K \rangle / \sqrt{d_{out}})} = \frac{\exp(\langle b_Q, \mathbf{W}\, \mathbf{h}'_{u1} + b_K \rangle / \sqrt{d_{out}})}{\sum_{w\in V} \exp(\langle b_Q, \mathbf{W}_K\, h'_w + b_K \rangle / \sqrt{d_{out}})}.$$

另一方面，如果我们取 $b_Q = 0$，那么对于任意 $\mathbf{W_Q}$, $\mathbf{W_K}$ 和 b_K，我们可以得出

$$\frac{\exp(\langle \mathbf{W}\, \mathbf{h}_{Qv}, \mathbf{W}\, \mathbf{h}_{Ku} + b_K \rangle / \sqrt{d_{out}})}{\sum_{w\in V} \exp(\langle \mathbf{W_Q} h_v, \mathbf{W}\, \mathbf{h}_{Kw} + b_K \rangle / \sqrt{d_{out}})} = \frac{\exp(\langle \mathbf{W}\, \mathbf{h}'_{Qv}, \mathbf{W}\, \mathbf{h}_K \rangle / \sqrt{d_{out}})}{\sum_{w\in V} \exp(\langle \mathbf{W_Q} h'_v, \mathbf{W_K}\, h'_{u1} + b_K \rangle / \sqrt{d_{out}})}$$
$$= \frac{\exp(\langle \mathbf{W}\, \mathbf{h}'_{Qv}, \mathbf{W}\, \mathbf{h}_{Ku} + b_K \rangle / \sqrt{d_{out}})}{\sum_{w\in V} \exp(\langle \mathbf{W}\, \mathbf{h}'_{Qv}, \mathbf{W}\, \mathbf{h}_{Kw} + b_K \rangle / \sqrt{d_{out}})},$$

其中，第二个等式是用 $\mathbf{W}\, \mathbf{h}'_{Qv}$ 代替上述等式中的 b_Q 而得到的。然后，我们可以重写上面的如下所示：

$$\sum_{w\in V} \exp(\langle \mathbf{W}\, \mathbf{h}_{Qv}, \mathbf{W_K}(h_w - h_{u1}) \rangle / \sqrt{d_{out}}) = \sum_{w\in V} \exp(\langle \mathbf{W}\, \mathbf{h}'_{Qv}, \mathbf{W_K}(h_w - h_{u1}) \rangle / \sqrt{d_{out}}).$$

如果我们用 φ 表示 $\mathbb{R}^{d_{out}} \to H$ 中与点积核 $\kappa_{\exp}(t, t') = \exp(\langle t, t' \rangle / \sqrt{d_{out}})$ 相关的特征映射。而 H 是对应的重现核希尔伯特空间，那么对于任意 $\mathbf{W_Q}$ 和 $\mathbf{W_K}$，我们可以得到

$$\Big\langle \varphi(\mathbf{W}\, \mathbf{h}_{Qv}) - \varphi(\mathbf{W}\, \mathbf{h}'_{Qv}), \sum_{w\in V} \varphi(\mathbf{W_K}(h_w - h_{u1})) \Big\rangle_H = 0.$$

根据假设，存在一个 $u_2 \in V$，使得 $h_{u2} - h_{u1} \neq 0$，并且 $\kappa_{\exp}$ 是一个通用内核（Micchelli 等人，2006 年），$\mathbf{W_K} \to \varphi(\mathbf{W_K}(h_u - h_{u2}))$ 在 H 中是致密的，我们有 $\varphi(\mathbf{W}\, \mathbf{h}_{Qv}) = \varphi(\mathbf{W}\, \mathbf{h}'_{Qv})$。根据注入性，我们可以得出结论 φ，即

$$\mathbf{W}\, \mathbf{h}_{Qv} = \mathbf{W}\, \mathbf{h}'_{Qv},$$

对于任意 $\mathbf{W_Q}$，因此 $h_v = h'_v$。现在，将 $h_v = j(v, G)$ 和 $h'_v = j(v', G')$ 取值，我们就得到了定理。 □

C. 实验细节和其他结果

在本节中，我们将提供实施细节和其他实验结果。

C.1. 计算细节

所有实验均在配备有 GTX1080、GTX1080TI、GTX2080TI 和 TITAN RTX 的共享 GPU 集群上进行。其中约 20 个 GPU 同时使用，本研究项目的总计算成本约为 1k GPU 小时。

C.2. 数据集 描述

我们提供了实验中使用的数据集的详细信息，包括 ZINC（Irwin 等人，2012 年）、CLUSTER（Dwivedi 等人，2020 年）、PATTERN（Dwivedi 等人，2020 年）、OGBG-PPA（Hu 等人，2020a）和 OGBG-CODE2 （Hu 等人，2020a）。对于每个数据集，我们都遵循其各自的训练协议，并使用标准的训练/验证/测试分割和评估指标。

ZINC 数据集ZINC 数据集是一个由分子组成的图回归数据集，任务是预测受限溶解度。与 Dwivedi 等人（2020 年）一样，我们也使用了 12K 个分子的子集，并采用了相同的拆分方法。

表 4：在不同数据集上训练的 SAT 模型的超参数。RWPE-p 表示在随机游走位置编码中使用 p *个*步骤，从而产生一个 p *维*向量作为每个节点的位置表示。

超参数	锌	集群	图案	OGBG-PPA	OGBG-CODE2
#Layers	6	16	6	3	4
隐藏的尺寸	64	48	64	128	256
FFN 隐藏的尺寸 #注意	8	8	2×隐藏尺寸 8	离子 8	8
头辍学者			{0.0, 0.1, 0.2, 0.3, 0.4}		
子图大小 k 读出			{1, 2, 3, 4}		
方法 绝对 PE	平均 RWPE-20	无 RWPE-3	无 RWPE-7	无	无
学习率	0.001	0.0005	0.0003	0.0003	0.0001
批量大小	128	32	32	32	32
#Epochs	2000	200	200	200	30
热身步骤	5000	5000	5000	10 个纪元	2 个纪元
重量衰减	1e-5	1e-4	1e-4	1e-4	1e-6

PATTERN 和 CLUSTER。PATTERN 和 CLUSTER Dwivedi 等人（2020 年）是使用随机块模型（Abbe，2018 年）创建的合成数据集。这两个数据集的目标都是节点分类，PATTERN 侧重于检测数据集中的给定模式，而 CLUSTER 侧重于识别图中的群落。对于 PATTERN，二元类标签对应于节点是否属于预定义模式的一部分；而对于 CLUSTER，多类标签表示是否属于某个社区。我们使用 Dwivedi 等人（2020 年）所使用的拆分方法。

OGBG-PPA。PPA（Hu 等人，2020a）由蛋白质-蛋白质关联网络组成，其目标是将网络正确归入 37 个类别之一，这些类别代表了网络的物种类别。节点代表蛋白质，边代表蛋白质之间的关联。边缘属性代表与关联相关的信息，如共同表达。我们使用 Hu 等人（2020a）提供的标准拆分方法。

OGBG-CODE2.CODE2 （Hu 等人，2020a）是一个包含 Python 编程语言源代码的数据集。该数据集由抽象语法树（Abstract Syntax Trees）组成，任务是对组成方法名称的子符号进行正确分类。我们使用 Hu 等人（2020a）提供的标准拆分。

C.3. 超参数选择与可重复性

超参数选择。一般来说，我们只进行非常有限的超参数搜索，就能得出表 1 和表 2 中的结果。表 4 总结了在不同数据集上训练 SAT 模型的超参数，其中只调整了丢弃率和子图 k 的大小（k *1* 、2、3、4）。我们在 ZINC、PATTERN 和 CLUSTER 上使用固定的 RWPE（Dwivedi 等人，2022 年）和 SAT。在所有实验中，我们使用验证集来选择丢弃率和子树或子图的大小 k *1* 、2、3、4。为简单起见，所有其他超参数都是固定的，包括将读出方法设置为均值集合。我们没有在 OGBG-PPA 和 OGBG-CODE2 上使用 RWPE，因为我们观察到其性能改善甚微。请注意，我们只在 CLUSTER 和 PATTERN 上的 k *子图*SAT 模型中使用了 $k = 1$，因为它需要大量内存，与使用更大 k 的 k 子树 SAT 相比，这已经带来了性能提升。报告的结果是 ZINC、PATTERN 和 CLUSTER 上 4 粒种子的平均值，与 Dwivedi 等人（2020 年）所做的一样；是 OGBG-PPA 和 OGBG-CODE2 上 10 粒种子的平均值。

优化。我们的所有模型都是使用 AdamW 优化器（Loshchilov & Hutter，2018 年）和 Vaswani 等人（2017 年）为 Transformers 建议的标准热身策略进行训练的。我们使用 L1 损失或交叉熵损失，这取决于任务是回归还是分类。在 ZINC、PATTERN 和 CLUSTER 数据集上使用了 Transformer 中提出的学习率调度器，在较大的 OGBG-PPA 和 OGBG-CODE2 数据集上使用了余弦调度器（Loshchilov & Hutter，2016 年）。

表 5：使用表 4 中超参数的 k-subtree SAT 模型的参数数和每个 epoch 的训练时间。在 SAT 中使用了各种 GNN 作为基础 GNN。

	锌	集群	图案	OGBG-PPA	OGBG-CODE2
基地 GNN			#参数		
GCN	421k	571k	380k	766k	14,030k
GIN	495k	684k	455k	866k	14,554k
PNA	523k	741k	493k	1,088k	15,734k
单个 TITAN RTX/epoch 的基本 GNN			GPU 时间		
GCN	6s	142s	40s	308s	40 分钟
GIN	6s	144s	62s	310s	40 分钟
PNA	9s	178s	90s	660s	55 分钟

表 6：在 ZINC 数据集上使用不同结构提取器和读出方法的 SAT 模型的测试 MAE。

		无边缘属性			带边缘属性		
	基准 GNN	平均值	总和	CLS	平均值	总和	CLS
K 子树 SAT	GCN	0.174±0.009	0.170±0.010	0.167±9.005	0.127±0.010	0.117±0.008	0.115±0.007
	GIN	0.166±0.007	0.162±0.010	0.157±0.002	0. 0.005	0. 0.008	0. 0.003
	图形	0.164±0.004	0.165±0.008	0.156±0.005	115±-	112±-	104±-
	PNA	0.147±0.001	0.142±0.008	**0.135±0.004**	0.102±0.005	0.102±0.003	**0.098±0.008**
K 子图 SAT	GCN	0.184±0.002	0.186±0.007	0.184±0.007	0.114±0.005	0.103±0.002	0.103±0.008
	GIN	0.162±0.013	0.158±0.007	0.162±0.005	0. 0.002	0. 0.002	0. 0.010
	图形	0.168±0.005	0.165±0.005	0.169±0.005	095±-	097±-	098±-
	PNA	0.131±0.002	0.129±0.003	**0.128±0.004**	0.094±0.008	**0.089±0.002**	0.093±0.009

参数数和计算时间在表 5 中，我们报告了使用表 4 所选超参数的 k *子树*GNN 提取器 SAT 的参数数和每个 epoch 的训练时间。请注意，我们在 OGB 数据集上的 SAT 所使用的参数数少于大多数先进方法。

C.4. 其他结果

我们还提供了 ZINC、OGBG-PPA 和 OGBG-CODE2 的其他实验结果。

C.4.1. 有关锌的其他结果

我们在表 6 中对使用不同结构提取器和不同读出方法的 SAT 实例进行了更全面的比较。我们发现，使用 PNA 的 SAT 模型始终优于其他 GNN。此外，读出方法对预测性能的影响很小。

C.4.2. 关于 OGBG-PPA 的其他结果

表 7 总结了在 OGBG-PPA 上使用不同 GNN 的 k-subtree SAT 与最先进方法的比较结果。所有结果都是使用不同随机种子运行 10 次计算得出的。

C.4.3. 关于 OGBG-CODE2 的其他结果

表 8 总结了在 OGBG- CODE2 上使用不同 GNN 的 *k-subtree* SAT 与最先进方法的比较结果。所有结果都是使用不同随机种子运行 10 次计算得出的。

表 7：SAT 和 SOTA 方法在 OGBG-PPA 数据集上的比较。所有结果均通过 10 次不同的运行计算得出。

	OGBG-PPA	
	方法测试精度	验证精度
GCN	0.6839±0.0084	0.6497±0.0034
gcn 虚拟节点	0.6857±0.0061	0.6511±0.0048
GIN	0.6892±0.0100	0.6562±0.0107
琴弦虚拟节点	0.7037±0.0107	0.6678±0.0105
变压器	0.6454±0.0033	0.6221±0.0039
K-子树 sat-gcn	0.7483±0.0048	**0.7072±0.0030**
k-subtree sat-gin	0.7306±0.0076	0.6928±0.0058
K 子树 sat-pna	0.7522±**0.0056**	0.7025±0.0064

表 8：SAT 和 SOTA 方法在 OGBG-CODE2 数据集上的比较。所有结果均通过 10 次不同的运行计算得出。

	OGBG-CODE2	
	方法测试 F1 分数	验证 F1 分数
GCN	0.1507±0.0018	0.1399±0.0017
虚拟节点	0.1581±0.0026	0.1461±0.0013
GIN	0.1495±0.0023	0.1376±0.0016
虚拟节点	0.1581±0.0026	0.1439±0.0020
变压器	0.1670±0.0015	0.1546±0.0018
GRAPHTRANS	0.1830±0.0024	0.1661±0.0012
K-子树 sat-gcn	0.1934±0.0020	**0.1777±0.0011**
k-subtree sat-gin	0.1910±0.0023	0.1748±0.0016
K 子树 sat-pna	0.1937±**0.0028**	0.1773±0.0023

D. 模型解读

在本节中，我们将提供有关模型可视化的实施细节。

D.1. 数据集和训练细节

我们使用突变性数据集（Kersting 等人，2016 年），该数据集由 4337 个分子图组成，根据其突变效应进行标记。我们以 80/10/10 的比例将数据集随机分成训练集/评估集/测试集。我们首先使用 RWPE 训练一个双层 vanilla Transformer 模型。隐藏维度和头部数量分别固定为 64 和 8。出于可视化目的，我们选择了第 4.2 节所述的 CLS 池化作为读出方法。我们还使用完全相同的超参数设置来训练 k *子树* SAT，但*它不使用任何绝对位置编码*。我们训练足够多的历时，直到两个模型都收敛为止。带有 RWPE 的经典变换器的测试准确率为 78%，而 k *子树* SAT 的测试准确率为 82%。

D.2. 其他结果

注意力分数的可视化。在此，我们将提供更多可视化示例，说明通过 SAT 和 vanilla Transformer 学习到的突变性数据集 [CLS] 节点的注意力分数。图 5 提供了注意力学习权重的几个示例。如中间一行左侧面板所示，即使对于非常大的图形，SAT 通常也能学习到更稀疏、信息量更大的权重。

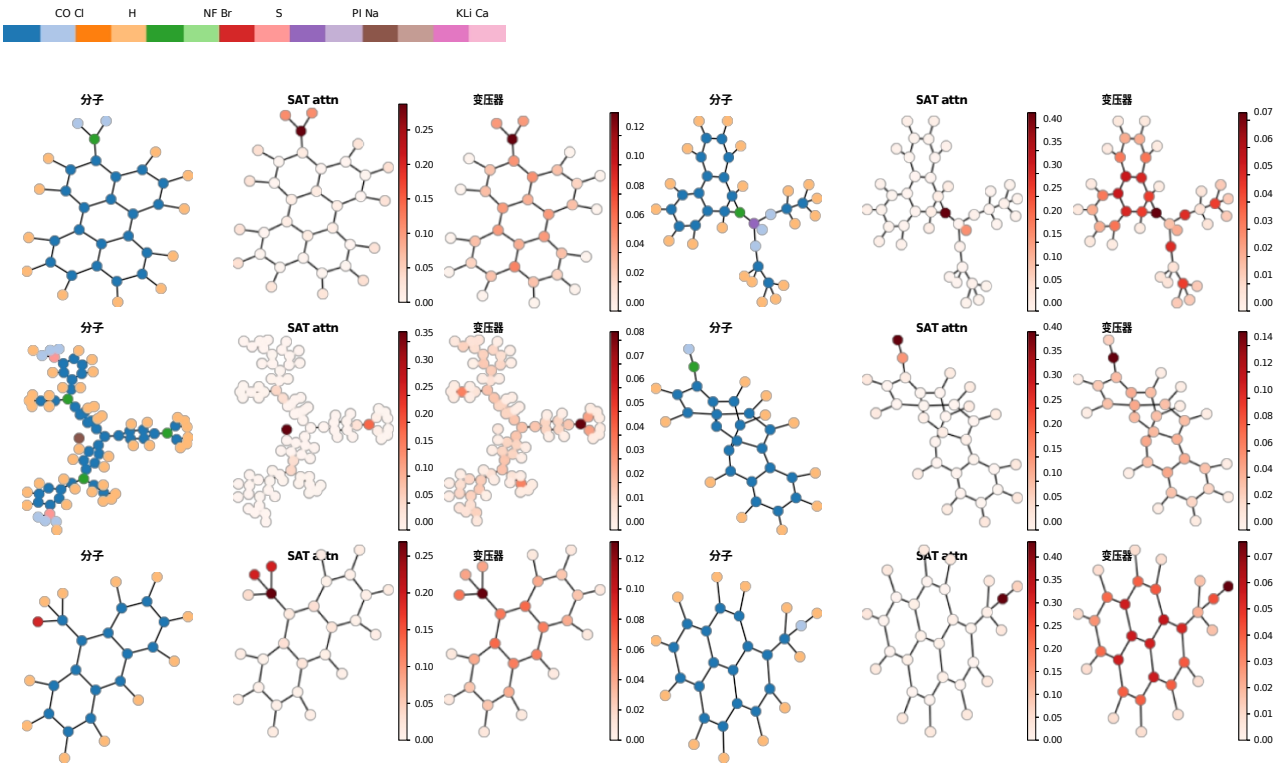


图 5：SAT 和 Transformer 的注意力可视化。中间一列显示的是 SAT 模型学习到的 [CLS] 节点的注意力权重，右边一列显示的是带有 RWPE 的经典转换器学习到的注意力权重。

E. 与子图神经网络和图池化的关系

下面我们将阐明 SAT 与子图神经网络（Alsentzer 等人，2020 年）以及图集合这一一般主题的关系（和区别）。

E.1. 与子图神经网络的区别

子图神经网络（SNN）（Alsentzer 等人，2020 年）探讨了明确纳入位置、邻域和结构信息，以解决子图预测问题。SNN 可生成子图（而非节点）级别的表示。而 SAT 则是通过生成具有结构感知能力的节点表示来模拟 Transformer 架构中节点之间的*结构交互*（通过点积注意）。这种结构感知是通过结构提取器实现的，结构提取器可以是任何提取给定节点局部结构信息的函数，并不一定需要明确提取子图。例如，*k-子树* GNN 提取器并不明确提取子图，而只是使用从 GNN 生成的节点表示。这也使得 k-subtree SAT 具有很好的可扩展性。与 SNN 不同的是，生成的 SAT 的输入不是子图，而是原始图，而结构感知节点表示是作为每层点积注意的查询和关键来计算的。

E.2. 与图表池的关系

在 GNN 中，结构信息传统上是通过邻域聚合过程纳入节点嵌入的。纳入结构信息的另一种补充方式是通过一种称为*局部池化的*过程，该过程通常基于图聚类（Ying 等人，2018 年）。局部池化可以粗化网络中*第 l* 层的邻接矩阵，例如，将聚类算法应用于聚类节点，然后将邻接矩阵替换为聚类分配矩阵，在聚类分配矩阵中，聚类内的所有节点都由一条边连接。池层的另一种方法是基于节点采样（Gao & Ji，2019）。虽然 Mesquita 等人（2020 年）发现，与更简单的操作相比，这些局部池化操作目前并不能提高性能，但局部池化理论上可以纳入现

有 SAT 的 *k 子树*和 *k 子图*GNN 提取器中，因为它是 GNN 中的另一层。