•计算机科学与技术•

**DOI:**10.12454/j.jsuese.202400688

本刊网刊

# 一种高效的对称注意力图神经网络模型

袁　军[1],周丽君[1,2],张俊然[1*]

(1.四川大学 电气工程学院,四川 成都 610065;2.西北民族大学 电气工程学院,甘肃 兰州 730030)

**摘　要:**图神经网络(GNN)已成为处理图结构数据的重要方法;同时,注意力作为GNN领域广泛使用的机制,在各种图优化任务中取得了较好的使用效果。然而,静态注意力(SA)和动态注意力(DA)的计算和数据复杂度较高。因此,本文提出一种融入对称注意力(SDA)的鲁棒图对称注意力网络(graph symmetric attention network,GSDAT)模型。首先,提出一种无须学习参数的SDA机制,利用特征空间的几何关系计算目标节点与邻居节点间的注意力系数,并计算邻居节点的贡献度;其次,发现ReLU函数可以弥补SA和DA中softmax函数缺少识别负相关节点的能力,提取贡献显著的邻居节点用于节点的特征更新,缓解类重叠现象并降低数据的复杂度。在13个公开基准数据集上的实验结果表明,GSDAT在节点分类和链接预测两个下游任务上实现了与目前6种主流GNN有竞争力的准确度。通过消融实验发现,SDA对性能提升的效果更为显著,与图注意力网络模型GAT、GATv2、GTN相比,GSDAT占用的计算资源较少;GSDAT的F1、F2、F3、ONB 4个指标的值分别为1.109、0.627、0.160、0.473,较基线算法有极大改善,表明本文方法的数据复杂度更低。结果表明,本文模型利用无参数化的SDA有效降低了SA等模块的计算复杂度,并通过ReLU函数有效降低了数据复杂度。

**关键词:**图神经网络;对称注意力;ReLU;数据复杂度

**中图分类号:**TP391.9　　**文献标志码:**A　　**文章编号:**2096-3246(2026)06-0375-10

图神经网络(GNN)旨在以端到端的方式解决图相关的任务[1],目前在各领域均得到广泛应用,如交通速度预测[2]、驾驶员机动预期[3]、动作识别[4]、点云分类[5–6]以及场景图生成等[7–10]。

近年来,Kipf等[11]提出在图上直接操作图卷积的图卷积神经网络(GCN)来解决半监督分类任务,通过光谱图卷积的一阶近似来激发卷积体系结构的选择。Hamilton等[12]从节点的局部邻域采样和聚合特征来生成节点嵌入,提出GraphSAGE模型。Xu等[13]提出一个理论框架来分析GNN的表达能力,捕获不同的图结构,并提出图同构网络(GIN)。Kim等[14]认为大多数GNN在聚合和传播信息时,平等对待邻居节点,而不考虑节点的贡献。然而,现实世界的图通常具有噪声,不相关节点之间的连接将导致GNN学习到次优表示。Sun等[15]提出将在自然语言处理和计算机视觉领域有广泛应用前景的注意力机制引入GNN,用于自适应选择特征和自动过滤噪声信息。Velickovic等[16]首先将注意力机制引入GNN,提出图注意力网络(GAT),利用自注意力层来解决图卷积或其他类似方法赋予所有邻居节点相同重要性的问题。此后,注意力成为被广泛应用于图深度神经网络的机制[14]。Brody等[17]指出,GAT计算的注意力是一种静态注意力(SA),其计算的注意分数排名在查询节点上是无条件的;通过修改SA的操作顺序提出基于动态注意力(DA)的GAT模型GATv2。Transformer处理序列中单词间的所有连接时,是在完全图上运行的。受此启发,Dwivedi等[18]提出一种针对任意图的Transformer图注意力网络(GTN)模型,该结构可以扩展到边缘特征表示以及用拉普拉斯特征向量来表示位置编码。与局部注意力方法不同的是,Rong等[19]认为Graph

**收稿日期:**2024 – 08 – 29　　**修回日期:**2025 – 03 – 20　　**网络出版日期:**2025 – 11 – 24

**基金项目:**国家自然科学基金联合重点项目(U2333209);国家自然科学基金"数学与医疗健康交叉"重点专项项目(12126606);琶洲实验室(黄埔)研发项目(2023K0605);四川省科技计划项目(23DYF2913)

**作者简介:**袁　军(2000—),男,硕士生.研究方向:图神经网络.E-mail:yuanjun202204@163.com

* **通信作者:**张俊然,教授,E-mail: zhangjunran@126.com

Transformer可以直接学习高阶图的属性。综上所述，图注意力机制的选择对GNN的指标提升、鲁棒性、计算效率等有较大影响。

现有大多数注意力机制通过引入过多的训练参数来计算邻居节点的贡献，以获得邻居节点的有效信息，存在诸多缺陷。首先，SA和DA忽视了网络参数量增加将导致计算复杂度过高的问题，且未能考虑某些邻居节点与目标节点在特征空间的几何关系。其次，Graph Transformer中全局连接的注意力机制使得端到端模型在训练过程中需不断更新整个网络的权重[20−21]，不能充分利用图拓扑中的稀疏性，导致过高的计算复杂度。最后，SA和DA采用softmax函数对注意力系数进行归一化操作，而softmax函数不能对某些注意力系数为负的邻居节点进行判别选择，导致属于不同类别的节点在特征空间中发生严重的类重叠现象，使变换后的特征空间更不可分、数据复杂度更高，这对节点分类任务来说是不利因素[22−23]。

基于以上分析：首先，本文提出一种无须学习参数的对称注意力（symmetric attention，SDA）机制，在特征空间中将邻居节点的特征向量投影到目标节点的特征向量方向，以此获取节点间的几何关系；通过投影长度计算节点间的注意力系数，避免使用过多的网络参数，降低计算复杂度。其次，与softmax函数对邻居节点注意力系数的归一化操作相比，本文使用ReLU函数对注意力系数进行判别选择，缓解在特征空间的类重叠现象，降低数据复杂度。最后，将SDA和ReLU函数结合提出图对称注意力网络（graph symmetric attention network，GSDAT）模型，为测试模型性能，在节点分类与链路预测两个任务上使用Cora−ML、CiteSeer等多个数据集进行相关指标测试。

## 1 对称注意力网络模型

### 1.1 线性变换模块

给定一个图$\boldsymbol{G}$，$\boldsymbol{G}=(V,E)$；其中，$V$和$E$分别为图中节点和边的集合。将集合$V$和$E$的长度，即节点和边的数量分别记作$t$和$e$。将图$\boldsymbol{G}$的节点特征矩阵记作$\boldsymbol{X}$，$\boldsymbol{X}\in\mathbb{R}^{t\times d}$；其中，$d$为特征维度。线性变换模块的主要功能是将图$\boldsymbol{G}$的特征维度从$d$变换到$p$，通常将$p$的选定范围设置为2的幂，且满足$p<d$，并利用网格搜索策略确定。因此，图$\boldsymbol{G}$的线性变换如式(1)所示。

$$\boldsymbol{H}=\boldsymbol{X}\times\boldsymbol{W} \tag{1}$$

式中：$\boldsymbol{H}$为图$\boldsymbol{G}$线性变换后的特征矩阵，$\boldsymbol{H}\in\mathbb{R}^{t\times p}$；$\boldsymbol{W}$为训练参数，$\boldsymbol{W}\in\mathbb{R}^{d\times p}$。

### 1.2 对称注意力模块

将注意力函数的数学表达式记为$f$:$\mathbb{R}^p\times\mathbb{R}^p\to\mathbb{R}$。$[n]$和$[m]$分别表示主键集合$\{1,2,\cdots,n\}\in\mathbb{N}$和查询键集合$\{1,2,\cdots,m\}\in\mathbb{N}$；其中，$n$和$m$分别为键向量数量与查询向量数量。主键向量和查询向量集合分别记作$K$和$Q$，$K=\{\boldsymbol{k}_1,\boldsymbol{k}_2,\cdots,\boldsymbol{k}_n\}\in\mathbb{R}^p$，$Q=\{\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m\}\in\mathbb{R}^p$；其中，$\boldsymbol{k}_1,\boldsymbol{k}_2,\cdots,\boldsymbol{k}_n$为主键向量，$\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_m$为查询向量。

Brody等[17]提出了SA和DA，具体的定义和区别如下：

SA：如果注意力函数$f_{\mathrm{a}}$存在一个最高得分主键$j_f\in[n]$，使得对于每个查询键$i\in[m]$和主键$j\in[n]$都有$f_{\mathrm{a}}(\boldsymbol{q}_i,\boldsymbol{k}_{j_f})\geqslant f_{\mathrm{a}}(\boldsymbol{q}_i,\boldsymbol{k}_j)$，则$f_{\mathrm{a}}$计算的是静态注意力。

DA：如果存在一个映射$\varphi^{\mathrm{D}}$:$[m]\to[n]$使得注意力函数$f_{\mathrm{b}}$对于任意查询键$i\in[m]$和主键$j\in[n]$，且$j\neq\varphi^{\mathrm{D}}(i)\in[n]$都有$f_{\mathrm{b}}(\boldsymbol{q}_i,\boldsymbol{k}_{\varphi^{\mathrm{D}}(i)})>f_{\mathrm{b}}(\boldsymbol{q}_i,\boldsymbol{k}_j)$，则$f_{\mathrm{b}}$计算的是动态注意力。

图1为注意力计算结果。图1(a)为SA计算的注意力图。显然，SA的能力非常有限，因为注意力函数总是选择$\boldsymbol{k}_4$作为最重要的主键向量，而不管查询键集合$[m]$。这样的注意力函数不能对不同的键与不同查询具有不同相关性的情况进行建模。图1(b)为DA计算的注意力图。DA可以用查询键$i$选择$\varphi(i)$作为最重要的主键。因此，如果某个注意力函数总是给某个键分配不小于任何其他键的权重，且不以查询为条件，这个注意力函数就被认为是静态的。

受到SA与DA的启发，本文旨在利用特征空间的几何关系来实现DA的设计。SDA的定义如下：

如果存在一个映射$\varphi^{\mathrm{S}}$:$[m]\to[n]$使得注意力函数$f_{\mathrm{S}}$对于任意查询键$i\in[m]$和主键$j\in[n]$，满足$j\neq\varphi^{\mathrm{S}}(i)\in[n]$都有$f_{\mathrm{S}}(\boldsymbol{q}_i,\boldsymbol{k}_{\varphi^{\mathrm{S}}(i)})>f_{\mathrm{S}}(\boldsymbol{q}_i,\boldsymbol{k}_j)$和$f_{\mathrm{S}}(\boldsymbol{q}_i,\boldsymbol{k}_j)=f_{\mathrm{S}}(\boldsymbol{q}_j,\boldsymbol{k}_i)$成立，则$f_{\mathrm{S}}$计算的注意力属于对称注意力。

在SDA的定义中，使$f_{\mathrm{S}}(\boldsymbol{q}_i,\boldsymbol{k}_{\varphi^{\mathrm{S}}(i)})>f_{\mathrm{S}}(\boldsymbol{q}_i,\boldsymbol{k}_j)$主要是为了保证所设计的注意力具有动态性，使得计算出的注意力系数排名在图中所有节点间并非共享，且不同查询节点依赖性最大的主键节点是变化的，即每个查询节点都有不同的关键节点注意力系数排名。$f_{\mathrm{S}}(\boldsymbol{q}_i,\boldsymbol{k}_j)=f_{\mathrm{S}}(\boldsymbol{q}_j,\boldsymbol{k}_i)$体现了所设计的注意力满足对称特性。图1(c)直观展示了SDA计算的注意力图矩阵是一个主对角对称矩阵，且计算的注意力系数排名满足动态注意力的性质。

由图1可见，以$\boldsymbol{k}_2$对于$\boldsymbol{q}_6$的注意力系数为例：SA计算的结果为−0.30，而SDA计算的结果为−0.97，表明$\boldsymbol{k}_2$与$\boldsymbol{q}_6$呈负相关；SA经过softmax函数输出后，计算的注意力系数为0.15(图1(d))，特征加权时将包含负相关的特征信息，不利于分类等任务；对于使用ReLU函数的SDA，注意力系数为0(图1(f))，特征加权时不会增加其负相关信息。

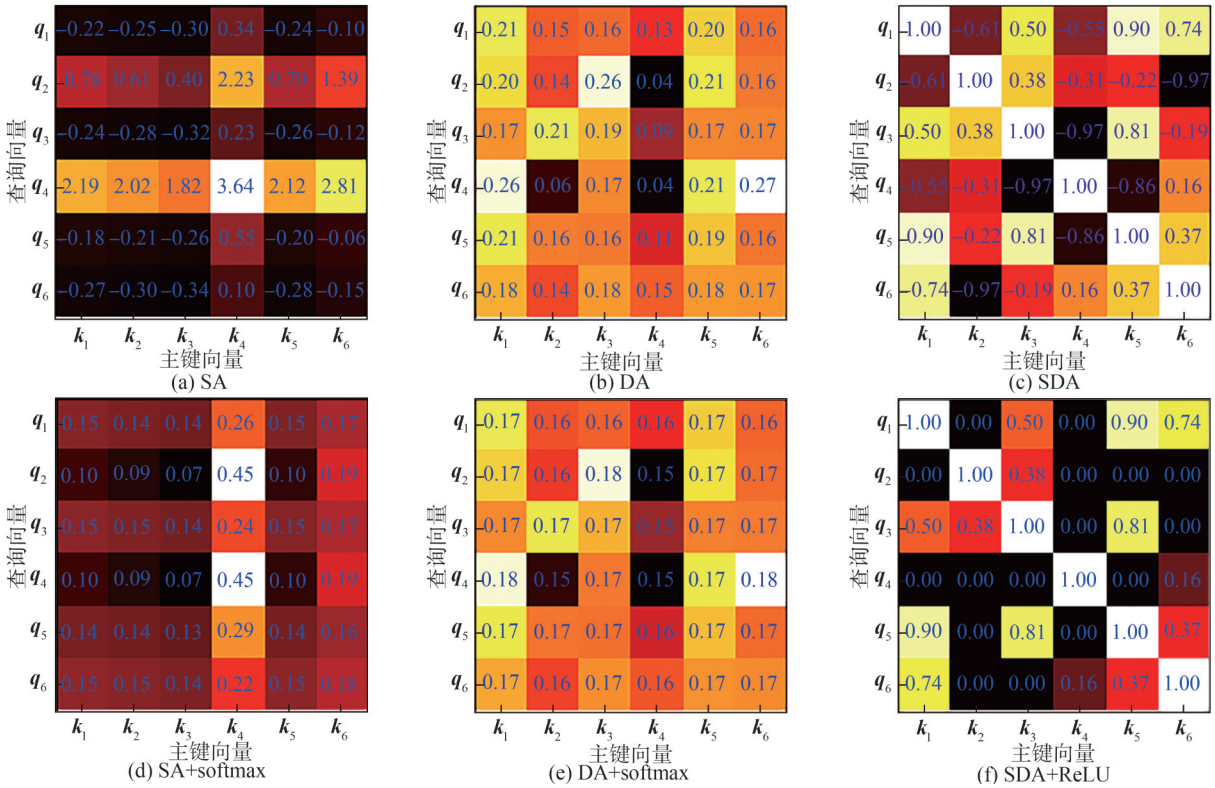



**图1　注意力计算结果**

**Fig. 1　Results of attention calculation**

SDA 在图结构数据执行如下操作:首先,将所有目标节点 $I=\{v|v\in V\}$ 记作查询键集合 $[m]$;然后,通过边集合 $E$ 搜索目标节点 $v$ 的邻居节点集合 $N_v$,将所有目标节点 $I$ 的邻居节点集合 $\{N_v|v\in I\}$ 作为主键集合 $[n]$。主键向量矩阵 $\boldsymbol{K}$ 和查询向量矩阵 $\boldsymbol{Q}$ 满足 $\boldsymbol{K}=\boldsymbol{Q}=\boldsymbol{H}$[17]。

目标节点 $u$ 和邻居节点 $v$ 经过线性变换的特征向量分别记作 $\boldsymbol{h}_u$ 和 $\boldsymbol{h}_v$,$\boldsymbol{h}_u,\boldsymbol{h}_v\in\boldsymbol{H}$。因此,邻居节点 $v$ 在目标节点 $u$ 上的投影 $P$ 和对称注意力 $\alpha_{uv}$ 的关系满足:

$$\alpha_{uv}\propto P=\cos(\boldsymbol{h}_u,\boldsymbol{h}_v)\cdot\|\boldsymbol{h}_v\|_l \tag{2}$$

式中:$\|\cdot\|_l$ 表示向量的 $l$ 范数;$\propto$ 表示正比运算操作;$\cos(\boldsymbol{h}_u,\boldsymbol{h}_v)$ 为 $\boldsymbol{h}_u$ 和 $\boldsymbol{h}_v$ 的夹角余弦,如式(3)所示。

$$\cos(\boldsymbol{h}_u,\boldsymbol{h}_v)=\frac{\boldsymbol{h}_u\odot\boldsymbol{h}_v}{\|\boldsymbol{h}_u\|_l\cdot\|\boldsymbol{h}_v\|_l} \tag{3}$$

式中,$\odot$ 表示逐元素乘法。

结合式(3)和(2)可得计算邻居节点 $v$ 对于目标节点 $u$ 的对称注意力的函数 $f_S$:

$$\alpha_{uv}=f_S(\boldsymbol{h}_u,\boldsymbol{h}_v)=\frac{\boldsymbol{h}_u\odot\boldsymbol{h}_v}{\|\boldsymbol{h}_u\|_l\cdot\|\boldsymbol{h}_v\|_l} \tag{4}$$

用 ReLU 函数处理式(4),实现对特定节点的判别选择:

$$\alpha_{uv}=\mathrm{ReLU}(f_S(\boldsymbol{h}_u,\boldsymbol{h}_v)) \tag{5}$$

综上所述,式(4)和(5)是对 SDA 的数学建模,利用特征向量在特征空间中的投影来计算注意力系数。当两个节点的特征向量异向时,投影为负(即注意力系数为负)。式(11)是 Brody 等[17]对原始 DA 的建模,其主要改进在于通过修改 SA 的操作顺序,并通过网络的不断迭代来学习动态的注意力。因此,SDA 与 DA 的区别在于前者只需用特征向量投影的几何关系来实现动态的注意力,而后者需要不断地学习。

### 1.3　消息聚合模块

利用式(5),计算节点 $u$ 所有邻居节点的注意力系数,通过注意力系数来计算邻居节点特征的加权平均作为节点 $u$ 新的特征向量 $\boldsymbol{h}_u^*$:

$$\boldsymbol{h}_u^*=\sigma(\boldsymbol{h}_u+\sum_{p\in N_u}\alpha_{uv}\boldsymbol{h}_v) \tag{6}$$

式中,$\sigma(\cdot)$ 为激励函数,$N_u$ 为节点 $u$ 所有邻居节点的集合。

### 1.4　特征融合模块

Vaswani 等[24]发现使用多头注意力机制可以增加注意力学习的稳定性:首先,使用 $M$ 个独立的 SDA 机制;然后,通过拼接或平均的方式进行特征融合。假设图 $\boldsymbol{G}$ 的节点特征矩阵 $\boldsymbol{X}$ 的第 $k$ 个线性变换如下:

$$\boldsymbol{H}^k=\boldsymbol{X}\times\boldsymbol{W}^k \tag{7}$$

式中,$\boldsymbol{H}^k$ 为第 $k$ 个变换特征矩阵,$\boldsymbol{W}^k$ 为第 $k$ 个变换矩阵。

如果采用拼接的方式进行特征融合，则输出的特征向量如式(8)所示。

$$\boldsymbol{h}_u^*=\mathop{\Big\|}_{k=1}^{M}\sigma(\boldsymbol{h}_u^k+\sum_{v\in N_u}\alpha_{uv}^k\boldsymbol{h}_v^k) \tag{8}$$

式中：$\boldsymbol{h}_u^k$、$\boldsymbol{h}_v^k$分别为节点$u$和$v$在第$k$个注意力头的特征向量，$\boldsymbol{h}_u^k,\boldsymbol{h}_v^k\in\boldsymbol{H}^k$；$\|$表示拼接操作；$\alpha_{uv}^k$为由第$k$个对称注意力机制计算的注意力。

如果采用平均的方式进行特征融合，则输出的特征向量如式(9)所示。

$$\boldsymbol{h}_u^*=\frac{1}{K}\sum_{k=1}^{M}\sigma(\boldsymbol{h}_u^k+\sum_{v\in N_u}\alpha_{uv}^k\boldsymbol{h}_v^k) \tag{9}$$

### 1.5 计算复杂度

在单层和单注意力头条件下，GAT、GATv2 和 GTN的注意力函数$f_a$、$f_b$、$f_c$分别如式(10)～(12)所示：

$$f_a(\boldsymbol{h}_i,\boldsymbol{h}_j)=\text{Leaky Re LU}(\boldsymbol{a}_1^{\mathrm{T}}[\boldsymbol{h}_i\|\boldsymbol{h}_j]) \tag{10}$$

式中，$\boldsymbol{a}_1$为长度为$2p$的向量。因此，GAT除了要学习参数矩阵$\boldsymbol{W}\in\mathbb{R}^{d\times p}$，还要学习向量$\boldsymbol{a}_1$。所以，GAT的学习参数量为$dp+2p$。

$$f_b(\boldsymbol{h}_i,\boldsymbol{h}_j)=\boldsymbol{a}_2^{\mathrm{T}}\text{Leaky Re LU}([\boldsymbol{h}_i\|\boldsymbol{h}_j]) \tag{11}$$

式中，$\boldsymbol{a}_2$为长度为$p$的向量。同理，GATv2 也要额外学习向量$\boldsymbol{a}_2$。所以，GATv2的学习参数量为$dp+p$。

$$f_b(\boldsymbol{h}_i,\boldsymbol{h}_j)=(\boldsymbol{h}_i^Q\cdot(\boldsymbol{h}_j^K)^{\mathrm{T}})\Big/\sqrt{p} \tag{12}$$

式中，$\boldsymbol{h}_i^Q$和$\boldsymbol{h}_j^K$分别为经过两个线性变换$\boldsymbol{W}^Q$和$\boldsymbol{W}^K$后的结果，$\boldsymbol{W}^Q\in\mathbb{R}^{d\times p}$、$\boldsymbol{W}^K\in\mathbb{R}^{d\times p}$。所以，加上训练参数$\boldsymbol{W}$后，GTN的学习参数量为$3dp$。

GSDAT的学习参数量为$dp$。对于$M$头注意力，空间复杂度为$O(Mdp)$。表1为$M$头注意力模型的空间复杂度。

**表1 $M$头注意力模型的空间复杂度**

**Tab. 1 Spatial complexity of the $M$-head attention models**

| 模型 | GSDAT | GAT | GATv2 | GTN |
|---|---|---|---|---|
| 空间复杂度 | $O(Mdp)$ | $O(Mdp+2Mp)$ | $O(Mdp+Mp)$ | $O(3Mdp)$ |

## 2 实验与结果

### 2.1 实验设置

本文在13个公开基准数据集上进行实验，表2为数据集详细信息，列出了节点数、边数、特征维度与类别数；数据集名称后的括注为本文使用的简称。

采用Adam[25]作为优化器，将dropout[26]应用到两层之间的输入，丢弃概率记为$\delta$，$\delta$=0.4。将式(2)中的$l$设置为2。为保证实验的公平性，使用网格选参(grid selection)的方法进行超参数选择。用节点分类实验和链接预测实验做性能测试。所有实验均使用Intel Core i3-12100和32 GB内存。

**表2 数据集详细信息**

**Tab. 2 Detailed information of datasets**

| 数据集名称 | 节点数 | 边数 | 特征维度 | 类别数 |
|---|---|---|---|---|
| Cora-ML[27](Cor) | 2 995 | 16 316 | 2 879 | 7 |
| CiteSeer[28](Cit) | 3 327 | 9 104 | 3 703 | 6 |
| Facebook[29](Fac) | 4 039 | 88 234 | 1 283 | 193 |
| Wiki[29](Wik) | 2 405 | 17 981 | 4 973 | 17 |
| CS[30] | 18 333 | 163 788 | 6 805 | 15 |
| Minesweeper[31](Min) | 10 000 | 39 402 | 7 | 2 |
| Tolokers[31](Tol) | 11 758 | 519 000 | 10 | 2 |
| Questions[31](Que) | 48 921 | 152 450 | 301 | 2 |
| reed98[32](ree) | 962 | 37 624 | 754 | 3 |
| amherst41[32](amh) | 2 235 | 181 908 | 1 193 | 3 |
| johnshopkins55[32](joh) | 5 180 | 373 172 | 2 406 | 3 |
| cornell5[32](cor) | 18 660 | 1 581 554 | 4 735 | 3 |
| penn94[32](pen) | 41 554 | 2 724 458 | 4 814 | 3 |

节点分类实验使用两层的图网络层：第一层计算的特征维度范围为{20, 64, 128}，注意力头数为{2, 4, 8}，学习率为{0.005, 0.050, 0.100}；第二层使用单头注意力计算$C$个特征($C$为类别数)，并通过softmax函数输出分类结果。此外，采用$L_2$正则化，其系数$\lambda$=5×10$^{-4}$，并使用负对数似然损失函数。数据集按7∶1∶2的比例划分训练集、验证集和测试集。

链接预测实验使用两层的图网络层和一个全连接层：第一层的参数设置与节点分类实验一致；第二层使用单头注意力，且计算特征维度范围为{64, 128}；全连接层用于输出链接存在的概率，并使用sigmoid函数输出，学习率设置为0.005，使用交叉熵损失函数。数据集按1∶1的比例设置负样本(节点间不存在边)和正样本(节点间存在边)，负样本和正样本的标签分别设置为0和1；同时，边按7∶1∶2的比例划分训练集、验证集和测试集。

### 2.2 性能测试

在节点分类实验中，将GSDAT模型与6种主流模型进行比较，即GCN、GraphSAGE、GTN、GAT、GATv2、RUM[33]。采用平均准确度作为分类性能的主要评价指标，平均准确度的值越大分类效果越好。表3为6个数据集上的节点分类平均准确度。

由表3可见，GSDAT比GAT、GTN、GATv2模型的分类结果更准确。例如，在数据集CS上，GSDAT的平均准确度比GAT、GTN、GATv2模型分别提升了2.63、7.94、5.36个百分点；在数据集Min和Tol上，最优模型为GSDAT，其平均准确度比次优模型RUM分别高1.91和16.11个百分点。在数据集Cit上，GSDAT的平均准确度比最优模型GraphSAGE低0.13个百分点；在

数据集CS上,GSDAT的平均准确度比最优模型RUM低3.39个百分点,比次优模型GraphSAGE低0.10个百分点。GSDAT在所比较的6个数据集中的4个上取得最高平均准确度。

**表3　6个数据集上的节点分类平均准确度**

**Tab. 3　Average accuracy of node classification on 6 datasets**

| 数据集 | 平均准确度/% | | | | | | |
|---|---|---|---|---|---|---|---|
| | GCN | GraphSAGE | GTN | GAT | GATv2 | RUM | GSDAT |
| Cor | 89.27±0.23 | 88.56±0.24 | 87.03±0.50 | 89.18±0.19 | 88.97±0.30 | 84.10±0.90 | 89.50±0.34* |
| Cit | 77.93±0.18 | 78.71±0.43* | 77.73±0.28 | 77.58±0.64 | 77.65±0.34 | 75.50±0.50 | 78.58±0.34 |
| Fac | 57.21±0.35 | 57.16±0.23 | 56.07±0.49 | 50.52±0.65 | 47.92±1.17 | 56.89±0.97 | 57.31±0.22* |
| CS | 88.91±0.06 | 89.91±0.25 | 81.87±0.81 | 87.18±0.36 | 84.45±0.70 | 93.20±0.00* | 89.81±0.13 |
| Min | 51.10±1.03 | 64.74±0.71 | 62.75±2.32 | 51.11±1.67 | 54.09±2.81 | 70.68±1.43 | 72.59±2.98* |
| Tol | 54.59±1.67 | 52.27±1.24 | 52.62±0.85 | 48.88±1.19 | 42.52±6.73 | 62.89±1.28 | 79.00±1.68* |

注:*为最优结果。

在链接预测实验中,将GSDAT与DC–GCN[34]等模型做对比。采用平均ROC–AUC(AUC,即ROC曲线下面积(area under ROC curve),用于衡量分类模型的整体表现)作为链接预测准确性的主要评价指标,平均ROC–AUC的值越大链接预测效果越好。表4为6个数据集上的链接预测平均ROC–AUC。由表4可见,GSDAT在Wik、ree、amh和joh这4个数据集上取得最高平均ROC–AUC,分别为86.33%、86.37%、84.49%和86.15%。在数据集ree、amh和joh上,次优模型分别为GCN、DC–GNN和GCN,GSDAT的平均ROC–AUC比次优模型分别高3.89、1.55和3.03个百分点。在数据集cor和pen上,GSDAT的平均ROC–AUC分别为81.54%和81.76%,低于最优模型DC–GNN的84.68%和86.69%。

因此,GSDAT模型在节点分类和链接预测两个场景中均具有良好的性能。

**表4　6个数据集上的链接预测平均ROC–AUC**

**Tab. 4　Average ROC–AUC of link prediction on 6 datasets**

| 数据集 | 平均ROC−AUC/% | | | | | | |
|---|---|---|---|---|---|---|---|
| | GCN | GIN | GTN | GAT | GATv2 | DC−GNN | GSDAT |
| Wik | 79.24±0.13 | 84.88±2.31 | 85.65±1.06 | 81.96±1.74 | 81.61±1.35 | 86.04±0.65 | 86.33±0.49* |
| ree | 82.48±0.20 | 57.55±16.66 | 80.00±0.22 | 74.62±0.99 | 76.72±0.14 | 78.13±1.18 | 86.37±0.27* |
| amh | 81.82±0.06 | 55.53±14.76 | 79.71±0.36 | 74.89±0.44 | 76.27±0.42 | 82.94±1.59 | 84.49±1.15* |
| joh | 83.12±0.44 | 49.81±19.00 | 82.21±0.52 | 78.52±0.62 | 80.03±0.50 | 81.93±1.82 | 86.15±1.64* |
| cor | 81.61±0.35 | 42.56±10.31 | 78.79±0.60 | 77.67±0.94 | 77.34±0.36 | 84.68±0.24* | 81.54±4.53 |
| pen | 80.57±0.02 | 50.59±1.31 | 79.57±0.95 | 76.74±1.49 | 76.93±0.77 | 86.69±0.22* | 81.76±2.84 |

注:*为最优结果。

### 2.3　消融实验

为研究各改进点对模型的有效性,在数据集Min、Que、Tol上对ReLU函数和SDA进行消融实验。GSDAT的SDA替换为SA的模型命名为SA+ReLU;GSDAT的ReLU函数替换为softmax函数的模型命名为SDA+softmax。表5为消融实验的平均准确度。由表5可见,在数据集Min、Que、Tol上,使用SDA比SA的平均准确度分别提升了10.17、30.38、17.65个百分点,而使用ReLU函数比softmax函数的平均准确度分别提升9.77、22.18、9.83个百分点。因此,SDA和ReLU函数能够有效提升分类性能。此外,可以发现SDA对性能提升的效果比ReLU函数更明显。

**表5　消融实验的平均准确度**

**Tab. 5　Average accuracy in ablation experiments**

| 注意力设置 | 平均准确度/% | | |
|---|---|---|---|
| | Min | Que | Tol |
| SDA+ReLU | 72.59±2.98 | 91.67±3.64 | 79.00±1.68 |
| SA+ReLU | 62.42±18.94 | 61.29±18.94 | 61.35±13.52 |
| SDA+softmax | 62.82±0.39 | 69.49±0.65 | 69.17±9.48 |

### 2.4　噪声实验

在数据集Cit、Cor上进行噪声实验,验证GSDAT的抗噪声能力。按照Brody等[17]提出的规则进行噪声图设计,设计的结构噪声如下:对于图$\boldsymbol{G}(\boldsymbol{G}=(V,E))$和噪声率$\rho(0\le\rho\le1)$,随机从边集合$V\times V$中选择$|E|\times\rho$条不存在的边$E^*$加入图$\boldsymbol{G}$,生成噪声图$\boldsymbol{G}^*(\boldsymbol{G}^*=(V,E\cup E^*))$。在噪声图$\boldsymbol{G}^*$上训练GSDAT、GTN、GAT、GATv2等模型,超参数设置与节点分类实验相同。图2为不同GNN模型的抗噪能力。

由图2可见,随噪声率的增加,GSDAT的平均准

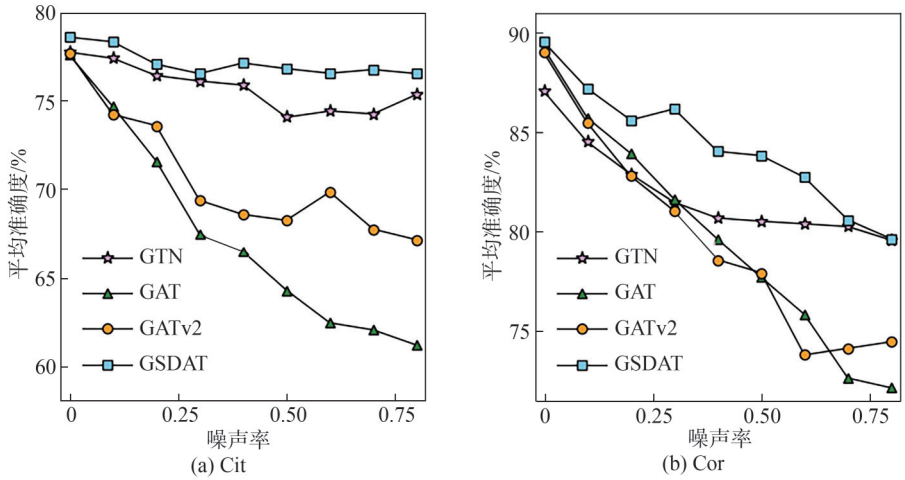



**图 2　不同GNN模型的抗噪能力**

**Fig. 2　Noise tolerance of different GNN models**

确度缓慢下降，而GAT和GATv2呈陡峭的下降趋势，且在数据集Cor上几乎直线下滑。这表明，GSDAT的平均准确度不易受结构噪声影响，而GAT和GATv2易受结构噪声影响，导致平均准确度迅速下降。因此，GSDAT对结构噪声具备良好的抗噪能力，对于处理具有结构噪声的图有一定应用潜力。

### 2.5　类重叠实验

为比较SDA、SA、DA的数据复杂度，以及研究ReLU与softmax函数的区别，在数据集Que上对SDA、SA、DA进行类重叠实验。类重叠实验的评价指标采用文献[35]提出的F1（fishers discriminant ratio）、F2（volume of overlap region）、F3（feature efficiency）和文献[36]提出的ONB（overlap number of balls）。其中，F1和F3的值越大，F2和ONB的值越小，则类重叠程度越低，即数据复杂度越低。表6为类重叠实验结果。

由表6可见，SDA在4个指标上均优于SA和DA。

**表 6　类重叠测试结果**

**Tab. 6　Results of class overlap test**

| 注意力 | F1 | F2 | F3 | ONB |
|---|---|---|---|---|
| SA | 0.191 | 0.821 | 0.010 | 0.558 |
| DA | 0.010 | 0.880 | 0.003 | 0.608 |
| SDA | 1.109* | 0.627* | 0.160* | 0.473* |

注：*为最优结果。

特别是在指标F1和F3上，SDA比DA提升约两个数量级，比SA提升了约一个数量级。此外，在指标F2和ONB上，SDA比SA分别降低0.194、0.085。因此，SA和DA使用softmax函数对注意力系数进行归一化会导致数据复杂度较高，而采用ReLU函数的SDA能够有效降低数据复杂度。

利用T–SNE算法对GSDAT等模型提取到的节点特征进行处理，图3为数据分布图，展示了SA、DA和SDA提取到的数据分布。

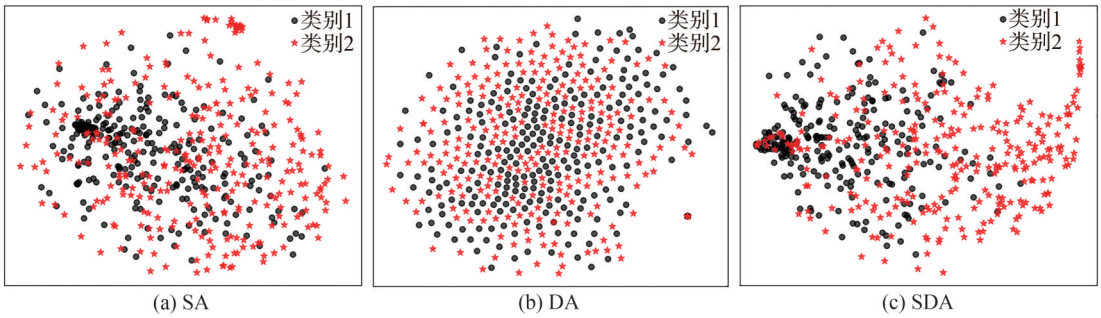



**图 3　数据分布图**

**Fig. 3　Maps of data distribution**

由图3（a）和（b）可见，使用softmax函数会使提取的特征分布混乱，不同类别的点重叠；由图3（c）可见，ReLU函数可以在空间分布上有效分离绝大多数的点，减少重叠点的数量。因此，图3可以定性地说明softmax和ReLU函数在特征提取上存在一定的差别，使用了ReLU函数的SDA在很大程度上缓解了类重叠现象，具有数据复杂度较低的优势。

### 2.6　参数效率

为验证SDA在计算复杂度方面的优势，比较GSDAT与GAT等GNN模型在实验中实际使用的参数量和推理时间。

图4为4种GNN模型在12个数据集上的空间开销。

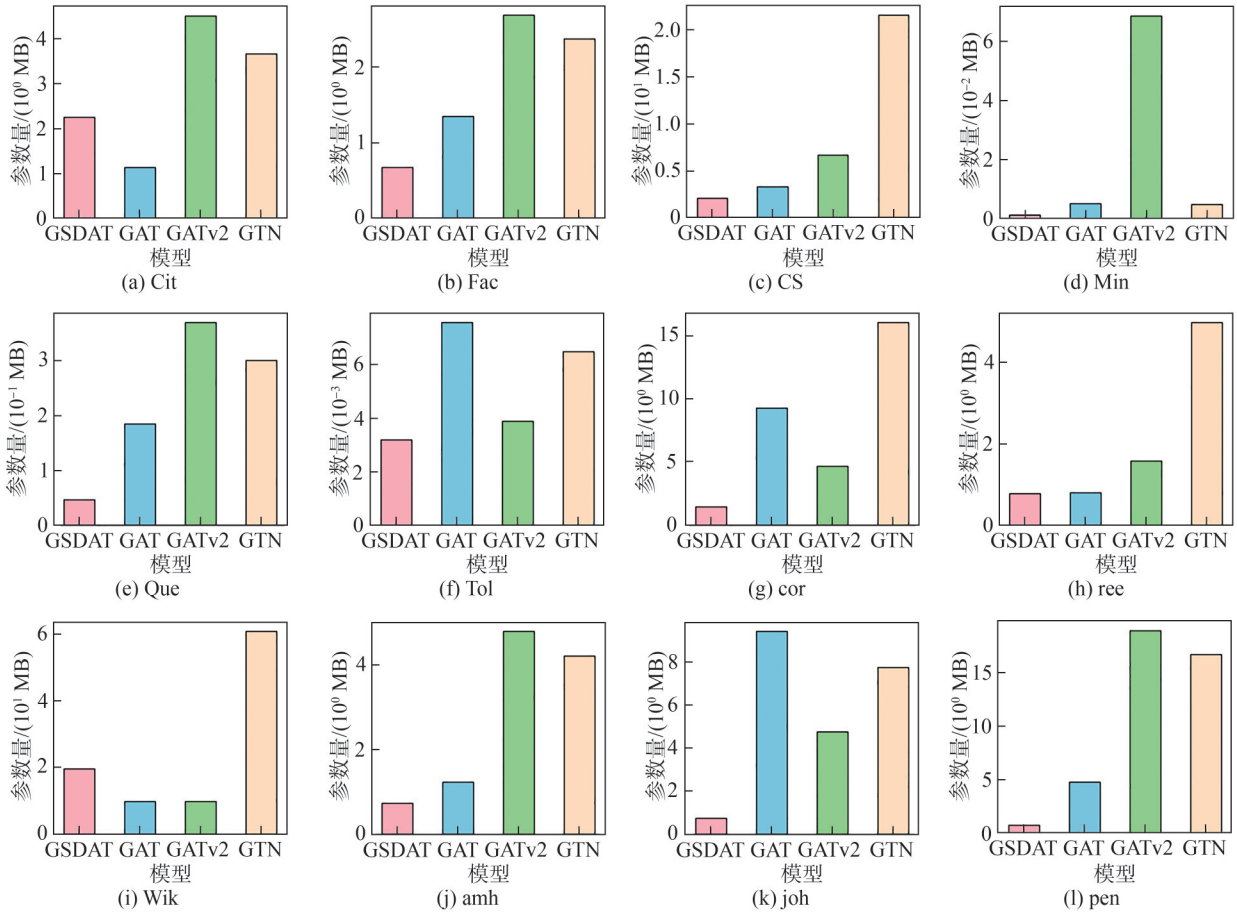



**图 4　4种GNN模型在12个数据集的空间开销**

**Fig. 4　Spatial overhead of 4 GNN models on 12 datasets**

由图4可见,GSDAT在Fac等10个数据集上具有最低的参数存储开销。在数据集Min和pen上,其开销小于GATv2的1/10;在cor和joh上,其开销小于GAT的1/10,并在所有所测数据集上小于GTN的开销。这表明,在相同参数精度与存储方式下,GSDAT的参数量较少。

表7为4种GNN模型在6个数据集上的推理时间。

**表7　4种GNN模型在6个数据集上的推理时间**

**Tab. 7　Inference time of 4 GNN models on 6 datasets**

| 数据集 | 推理时间/s | | | |
|---|---|---|---|---|
| | GTN | GAT | GATv2 | GSDAT |
| Cor | 5.79 | 3.22 | 1.84 | 0.96* |
| Cit | 8.19 | 2.66* | 5.88 | 3.51 |
| Fac | 10.35 | 6.75 | 13.89 | 4.69* |
| joh | 5.26 | 8.33 | 4.50 | 2.25* |
| cor | 32.09 | 21.35 | 17.81 | 8.79* |
| pen | 58.90 | 72.10 | 35.68 | 13.28* |

注:*为最优结果。

由表7可见:GSDAT在Cor、Fac、joh、cor和pen这5个数据集上取得最短推理时间。在数据集Cit上,GAT的推理时间为2.66 s,短于GSDAT的3.51 s;在数据集Cor上,GSDAT的推理时间为0.96 s;在数据集cor和pen上,GSDAT的推理时间比GTN分别缩短23.30 s和45.62 s;在数据集pen上,GSDAT的推理时间比GAT缩短58.82 s。

## 3　结　论

本文提出一种计算和数据复杂度低的GNN模型GSDAT。首先,提出SDA机制,通过向量投影进行实例化,用于计算邻居节点对目标节点的贡献。SDA使GSDAT的内存占用明显低于GAT等模型,可有效解决SA、DA、Graph Transformer计算复杂度过高的问题。其次,利用ReLU函数在负半轴恒为0的特性,替代softmax函数,能够自动判别并弃用注意力系数为负的邻居节点,有效缓解特征空间中严重的类重叠现象,使GSDAT具有数据复杂度低的特性。实验结果表明,GSDAT模型不仅在节点分类和链接预测两个下游任务上具有良好性能,且在噪声图上具有良好的鲁棒性。同时,通过消融实验发现SDA和ReLU函数都能有效提升模型在节点分类任务中的性能。

然而,GSDAT在具有193个类别的数据集Fac上,

未达到较高的节点分类平均准确度，在未来的工作中，将重点探索如何有效提高 GSDAT 在多类别节点分类任务中的性能。

**参考文献：**


[1] Wu Zonghan, Pan Shirui, Chen Fengwen, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4–24.

[2] Li Yaguang, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting [PP/OL]. V1. (2017–06–06)[2024–08–29]. https://arxiv.org/abs/1707.01926.

[3] Jain A, Zamir A R, Savarese S, et al. Structural-RNN: Deep learning on spatio-temporal graphs[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Las Vegas: IEEE, 2016: 5308–5317.

[4] Yan Sijie, Xiong Yuanjun, Lin Dahua. Spatial temporal graph convolutional networks for skeleton-based action recognition[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1): 7444–7452.

[5] Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs[C]//Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 4558–4567.

[6] Wang Yue, Sun Yongbin, Liu Ziwei, et al. Dynamic graph CNN for learning on point clouds[J]. ACM Transactions on Graphics, 2019, 38(5): 1–12.

[7] Xu Danfei, Zhu Yuke, Choy C B, et al. Scene graph generation by iterative message passing[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). Honolulu: IEEE, 2017: 3097–3106.

[8] Johnson J, Gupta A, Li Feifei. Image generation from scene graphs[C]//Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 1219–1228.

[9] Khademi M, Schulte O. Deep generative probabilistic graph neural networks for scene graph generation[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 11237–11245.

[10] Li Yikang, Ouyang Wanli, Zhou Bolei, et al. Factorizable net: An efficient subgraph-based framework for scene graph generation[C]//Computer Vision–ECCV 2018. Cham: Springer, 2018: 346–363.

[11] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[PP/OL]. V4. (2017–02–22)[2024–08–29]. https://arxiv.org/abs/1609.02907.

[12] Hamilton W L, Ying R, Leskovec J. Inductive Representation Learning on Large Graphs[C]//Procedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017). Long Beach: NIPS, 2017: 1024–1034.

[13] Xu Keyulu, Hu Weihua, Leskovec J, et al. How Powerful are Graph Neural Networks[C]//International Conference on Learning Representations(ICLR). Louisiana: OpenReview.net, 2019: 1–17.

[14] Kim D, Oh A. How to find your friendly neighborhood: Graph attention design with self-supervision[PP/OL]. V1. (2022–04–11)[2024–08–29]. https://arxiv.org/abs/2204.04879v1.

[15] Sun Chengcheng, Li Chenhao, Lin Xiang, et al. Attention-based graph neural networks: A survey[J]. Artificial Intelligence Review, 2023, 56(2): 2263–2310.

[16] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[C]//International Conference on Learning Representations. Vancouver: ICLR, 2018: 1–12.

[17] Brody S, Uri A, Yahav E. How Attentive are Graph Attention Networks[C/OL]//International Conference on Learning Representations, 2022: 1–26. https://openreview.net/pdf id=F72ximsx7C1.

[18] Dwivedi V P, Bresson X. A generalization of transformer networks to graphs[PP/OL]. V2. (2021–01–24)[2024–08–29]. https://arxiv.org/abs/2012.09699v2.

[19] Rong Yu, Bian Yatao, Xu Tingyang, et al. Self-supervised graph transformer on large-scale molecular data[PP/OL]. V1. (2020 – 06 – 18) [2024 – 08 – 29]. https://arxiv. org/abs/2007.02835.

[20] Ying Chengxuan, Cai Tianle, Luo Shengjie, et al. Do transformers really perform bad for graph representation [PP/OL]. V1. (2021–06–09)[2024–08–29]. https://arxiv.org/abs/2106.05234.

[21] Kim J, Oh S, Hong S. Transformers generalize deepsets and can be extended to graphs & hypergraphs[J]. Advance neural information processing systems, 2021, 34: 28016–28028.

[22] Charte D, Charte F, Herrera F. Reducing data complexity using autoencoders with class-informed loss functions[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(12): 9549–9560.

[23] Lancho C, Martín De Diego I, Cuesta M, et al. Hostility measure for multi-level study of data complexity[J]. Applied Intelligence, 2023, 53(7): 8073–8096.

[24] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[PP/OL]. V7. (2023–08–02)[2024–08–29]. https://arxiv.org/abs/1706.03762.

[25] Kingma D P, Ba J. Adam: A method for stochastic optimization[PP/OL]. V9. (2017–01–30)[2024–08–29]. https://arxiv.org/abs/1412.6980v9.

[26] Srivastava N, Hinton G, Krizhevsky A, etal. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal

of Machine Learning Research,2014,15(56):1929–1958.

[27] Yang Zhilin, Cohen W W, Salakhutdinov R. Revisiting semi-supervised learning with graph embeddings[C]//Proceedings of the 33rd International Conference on International Conference on Machine Learning.New York:ICML,2016:40–48.

[28] Bojchevski A,Günnemann S.Deep Gaussian embedding of graphs: Unsupervised inductive learning *via* ranking[C]// Proceedings of the 6th International Conference on Learning Representations.Vancouver:ICLR,2018.

[29] Yang Renchi, Shi Jieming, Xiao Xiaokui, et al. Scaling attributed network embedding to massive graphs[J].Proceedings of the VLDB Endowment,2020,14(1):37–49.

[30] Shchur O, Mumme M, Bojchevski A, et al. Pitfalls of graph neural network evaluation[PP/OL].V2.(2019–06–18)[2024–08–29].https://arxiv.org/abs/1811.05868v2.

[31] Platonov O,Kuznedelev D,Diskin M,et al.A critical look at the evaluation of GNNs under heterophily: Are we really making progress [PP/OL].V2.(2024–03–02)[2024–08–29]. https://arxiv.org/abs/2302.11640v2.

[32] Lim D, Hohne F, Li Xiuyu, et al. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods[PP/OL].V1.(2021–10–27)[2024–08–29]. https://arxiv.org/abs/2110.14446v1.

[33] Wang Yuanqing, Cho K. Non-convolutional graph neural networks[PP/OL].V3.(2024–09–29)[2024–08–29].https://arxiv.org/abs/2408.00165v3.

[34] Dong Yanfei,Dupty M H,Deng L,et al.Differentiable cluster graph neural network[PP/OL].V1.(2024–05–25)[2024–08–29].https://arxiv.org/abs/2405.16185v1.

[35] Ho T K,Basu M.Complexity measures of supervised classification problems[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2002,24(3):289–300.

[36] Pascual–Triana J D,Charte D,Andrés Arroyo M,et al.Revisiting data complexity metrics based on morphology for overlap and imbalance: Snapshot, new overlap number of balls metrics and singular problems prospect[J]. Knowledge and Information Systems,2021,63(7):1961–1989.

## An Efficient Symmetric Attention Graph Neural Network Model

*YUAN Jun*[1], *ZHOU Lijun*[1,2], *ZHANG Junran*[1*]

(1.School of Electrical Engineering, Sichuan University, Chengdu 610065, China;
2.School of Electrical Engineering, Northwest Minzu University, Lanzhou 730030, China)

**Abstract:**

**Objective** Graph neural networks (GNNs) have become an important approach for processing graph-structured data, whereas attention, a popular mechanism for GNNs, has achieved improved performance in various graph optimization tasks. However, both static attention (SA) and dynamic attention (DA) can result in high computational costs and greater data complexity. In particular, when dealing with large-scale graph data with numerous nodes and edges, the computational effort can increase substantially. Therefore, reducing computational complexity can significantly improve the training and inference speed of models, enabling GNNs to be deployed across a wider range of applications.

**Methods** First, in this paper, we propose a robust graph neural network model (GSDAT) based on symmetric attention (SAD). The model significantly reduces the number of parameters for training and the inference time. Second, to address the high computational complexity of SA and DA, we propose a SAD mechanism without learning parameters that uses geometric relationships in the feature space to calculate the attention coefficients between a target node and its neighboring nodes, thereby determining the contributions of neighboring nodes. This mechanism avoids introducing additional training parameters to compute the attention graph, effectively reducing the high computational costs. Finally, SA and DA use the softmax function to map the computed attention coefficients, which can increase data complexity during feature weighting in the message aggregation phase. We therefore use the ReLU function to address the limitations of softmax in handling negatively correlated nodes in SA and DA by extracting neighboring nodes with significant contributions for node feature updates. This approach helps alleviate class overlap and reduce data complexity.

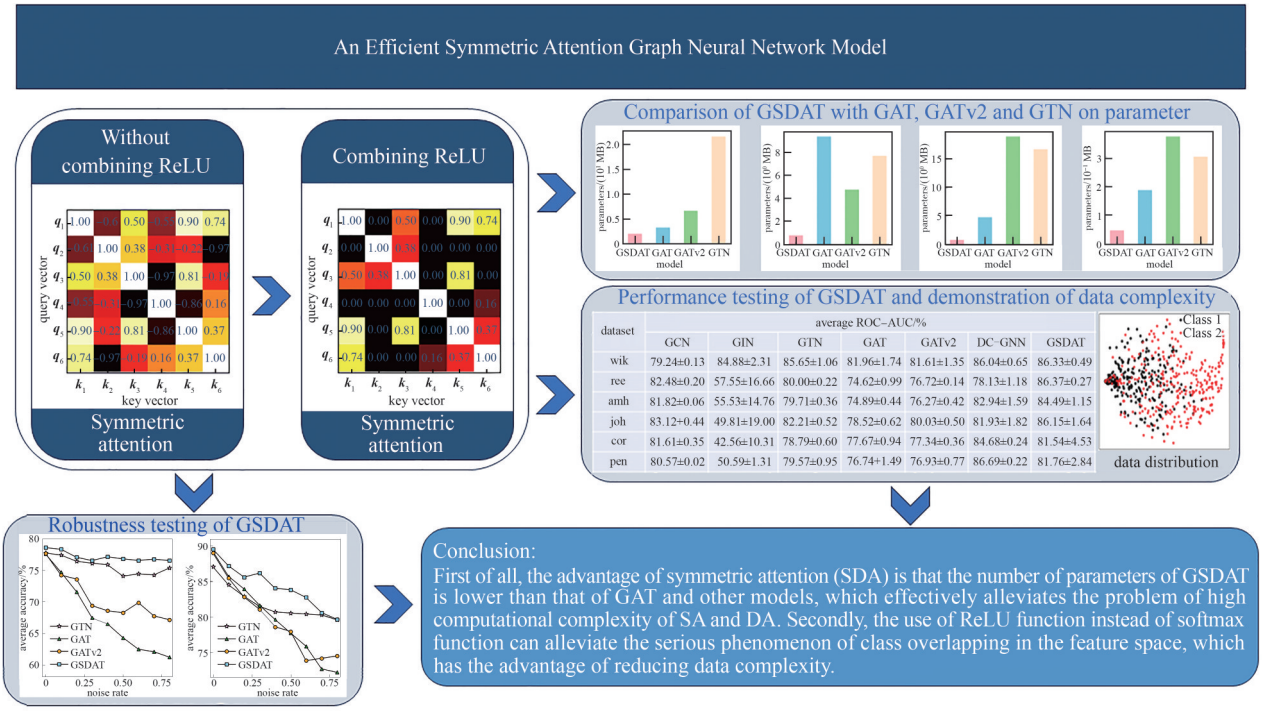
**Results and Discussions** Experiments on 13 publicly available benchmark datasets show that GSDAT achieves accuracy comparable to or higher than that of six mainstream GNNs across two downstream tasks: node classification and link prediction. For node classification, GSDAT improves accuracy on the CS dataset by 2.63, 7.94, and 5.36 percentage points relative to the respective comparison models. On the Min and Tol datasets, it achieves accuracy improvements of 7.85 and 24.41 percentage points, respectively, over the sub-optimal models, GraphSAGE and GCN. However, its average accuracy on the Cit and CS datasets is slightly lower than that of the optimal model, GraphSAGE, by 0.13 and 0.10 percentage point, respectively, although it outperforms most of the other comparison algorithms. For link prediction, GSDAT achieves the best receiver operating curve (ROC) area under the curve (AUC) scores on six datasets, including Wik, with values of 86.33%, 86.37%, 84.49%, 86.15%, 81.54%, and 81.76%. On the Ree, Amh, and Joh datasets, it improves the ROC–AUC score by 6.37, 4.78, and 3.49 percentage points, re-

spectively, over the sub-optimal model, GTN. These results demonstrate GSDAT's effectiveness in node classification and link prediction. Ablation experiments further demonstrate the contribution of SAD to performance improvements. Noise experiments show that GSDAT is robust to structural noise, indicating its potential for applications involving graphs with structural noise. Compared with the graph attention network models GAT, GATv2, and GTN, GSDAT also requires fewer computational resources. Furthermore, its F1, F2, F3, and ONB metric values are 1.109, 0.627, 0.160, and 0.473, respectively, representing substantial improvements over the baseline algorithm and indicating reduced data complexity.

**Conclusions** In this study, we propose GSDAT, a GNN model with low computational and data complexity. First, this paper proposes the SAD mechanism, which uses vector projection to calculate the contributions of neighboring nodes to a target node. This mechanism significantly reduces GSDAT's memory requirements compared with models such as GAT and helps address the high computational complexity of SA, DA, and graph transformers. Second, the ReLU function is used to replace the softmax function. Because ReLU outputs 0 for the negative half-axis, it can automatically identify and suppress neighboring nodes with negative attention coefficients, thereby helping alleviate class overlap in the feature space and reducing data complexity.

**Key words**: graph neural networks; symmetric attention; ReLU; data complexity

**Graphical abstract**:




（编辑　陈　雪）

引用格式: Yuan Jun,Zhou Lijun,Zhang Junran.An efficient symmetric attention graph neural network model[J].Advanced Engineering Sciences,2026,58(6):375–384.[袁军,周丽君,张俊然.一种高效的对称注意力图神经网络模型[J].工程科学与技术,2026,58(6):375–384.]