

# 基于注意力机制的自适应图卷积三维点云识别算法

马原, 余黎煌, 李佳蔚, 鲍喜荣

(东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

**摘要:** 为了更好地捕捉三维点云的局部几何结构信息, 提出了一种基于注意力机制的自适应图卷积三维点云识别算法. 为了解决固定卷积核忽略特征的缺点, 首先通过图结构特征动态学习自适应卷积核; 其次为了提高模型对局部几何结构的建模能力, 通过向量注意力机制自适应地调整卷积核的权重分配; 而后使用点云的位置特征构建图, 并利用自适应卷积核来对新构建的图结构特征进行卷积操作; 最后通过池化得到新的点云特征. 实验结果表明, 相较于之前的点云卷积算法, 所提算法在采样点较少时仍可以很好地提取局部几何结构信息并在分类任务上取得较高精度. 所提算法在 ModelNet40, ScanObjectNN 和 ShapeNetPart 数据集上的效果对比目前的点云分类和分割方法具有一定的优势.

**关键词:** 三维点云; 注意力机制; 自适应; 图卷积; 动态学习

中图分类号: TP 391.4 文献标志码: A 文章编号: 1005-3026(2024)06-0786-07

## Adaptive Graph Convolutional 3D Point Cloud Recognition Algorithm Based on Attention Mechanism

MA Yuan, SHE Li-huang, LI Jia-wei, BAO Xi-rong

(School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China. Corresponding author: SHE Li-huang, E-mail: shelihuang@ise.neu.edu.cn)

**Abstract:** To better capture the local geometric structural information of 3D point clouds, an adaptive graph convolutional 3D point cloud recognition algorithm is proposed based on attention mechanism. To address the drawback of fixed convolutional kernels ignoring features, the algorithm first dynamically learns adaptive convolutional kernels based on graph structural features. Furthermore, to enhance the modeling capability of the model for local geometric structures, the weight distribution of the convolutional kernels using a vector attention mechanism is adjusted adaptively. Subsequently, a graph is constructed using the position features of the point cloud and perform convolution operations on the newly constructed graph structural features using the adaptive convolutional kernels. Finally, new point cloud features through pooling is obtained. Experimental results demonstrate that the proposed algorithm effectively extracts local geometric structural information and achieves higher accuracy in classification tasks even with a limited number of sampled points, outperforming previous point cloud convolutional algorithms. The proposed algorithm also exhibits certain advantages compared to existing methods for point cloud classification and segmentation, as evidenced by the performance evaluation on the ModelNet40, ScanObjectNN, and ShapeNetPart datasets.

**Key words:** 3D point clouds; attention mechanism; self-adaptation; graph convolution; dynamically learn

点云分析已成为三维理解中的热门主题, 引起了学术界和工业界的广泛关注. 点云被认为是

三维对象的最简单但最有效的表示形状. OctNet<sup>[1]</sup>和Kd-Net<sup>[2]</sup>试图通过使用更有效的数据结构 and 体素的计算来缓解这些问题,但考虑到从单点视角覆盖整个场景的难度,将这些方法应用于大规模扫描数据从根本上来讲依旧是困难的. PointNet<sup>[3]</sup>首先提出在每个点上独立地应用多层感知机来直接对点进行操作,其提供的基于点的处理思路和轻量级的网络结构在当时极具创造性,但由于输入点的顺序是不变的,故不能对局部几何信息进行编码. Shen等<sup>[4]</sup>通过局部点之间的欧氏距离和几何仿射来定义卷积核,但该卷积核是固定的,不能灵活运用于网络. DGCNN (dynamic graph convolution neural network)<sup>[5]</sup>是考虑局部邻域内点信息的PointNet,在局部图结构上收集特征信息并使用特殊边缘卷积进行特征提取,由此来获得局部几何信息.但其构建的卷积核无法自适应地调整以适应不同的图像尺寸或点云密度,从而仍损失一部分局部细节或全局信息.图卷积根据点之间的空间特征相似度将点云表示为图数据,并将二维卷积推广到三维数据.为了处理具有不同邻域大小的无序点集,标准图卷积利用每对点上的共享权重函数来提取相应的边缘特征.但这导致了固定的各向同性卷积核,而此后的卷积会忽略不同点云的特征对应.对于来自点云的不同语义部分的点,卷积核要区分这些点并分配给它们不同的权重.

由于注意力机制在自然语言处理<sup>[6]</sup>和图像处理<sup>[7]</sup>上的成功应用,Hu等<sup>[7]</sup>将标量点积自注意力应用于局部图像块中,Zhao等<sup>[8]</sup>开发了一系列向量自注意力操作符;Zhao等<sup>[9]</sup>提出将点云的位置属性视为点集,并应用自注意力机制实现了局部自适应特征学习.这些方法表明,注意力机制能够根据不同点之间的关系和重要性,自适应地分配权重,并通过点之间的关系,捕捉全局和局部上下文信息,更好地理解点云数据中的语义和结构信息,从而提高模型的性能和准确度.

为了解决固定核卷积被相同地应用于所有点对从而忽略不同特征对应,以及自适应核通过点的特征构造从而忽略了局部几何结构信息并将自适应核直接用于点特征变换造成的信息冗余等缺点,本文基于注意力机制提出一种新型自适应图卷积算子,选择采样点与邻域点之间的结构特征作为输入,由此更好地学习点之间的局部几何结构特征.由于仅使用几何结构特征,所以算子对点排列的刚性变换具有鲁棒性.

本文工作的主要贡献:

1) 本文为点云处理设计了一个通过点转换器生成的自适应卷积核,这种适应性不依赖于预定义的权重,而是代表了每对点之间所特有的多样性.

2) 本文的自适应核使用采样点与邻域点之间的学习特征构建,同时对点本身的特征进行约束,在减少特征信息冗余的同时增加对点云局部结构特征的提取.

3) 本文基于自适应卷积构建高效的网络用于点云分类和分割任务,并在多个基准数据集上取得优于之前工作的结果.

## 1 方 法

本文借鉴规则卷积网络(CNN)以及RS-Conv<sup>[10]</sup>中的配置,同时通过学习Transformer和自注意力机制在点云处理中的应用,并以图神经网络对点云分析中的局部几何特征提取提出了新的自适应图卷积(adaptive attention graph convolution, A-GConv).

### 1.1 A-GConv

先把一般的卷积运算公式化为<sup>[9]</sup>

$$f_{ij} = \sigma \left( A \left( \left\{ T(f_{x_j}), \forall x_j \right\} \right) \right). \quad (1)$$

其中: $x$ 是点; $f$ 为该点的特征向量;通过半径为 $r$ 的球体采样函数 $T$ 来变换所有采样点的特征,然后通过函数 $A$ 和非线性激活函数 $\sigma$ 来聚合这些特征从而得到中心点的特征,函数 $A$ 是对所有变换后的特征进行求和操作,保证点集的置换不变性. $T(f_{x_j}) = w_j \cdot f_{x_j}$ ,其中 $w_j$ 为可学习权重.由于无法处理不规则的采样,所以这种方法存在着一定的局限性.虽然后续一些研究人员对权重的选择进行了优化(如RS-Conv),但本文认为仍存在进步的空间.

本文的点变换器基于向量自注意力机制<sup>[11]</sup>.在向量自注意力机制中,注意力权重的计算是特别的,同时注意力权重可以调制单个特征通道的向量<sup>[9]</sup>:

$$Y_i = \sum_{x_j \in X} \rho \left( \gamma \left( \beta \left( \varphi(x_i), \psi(x_j) \right) + \delta \right) \right) \odot \alpha \left( (x_j) \right). \quad (2)$$

其中: $Y_i$ 是输出特征; $\varphi, \psi$ 和 $\alpha$ 是逐点特征变换,如线性投影或多层感知机(MLP); $\delta$ 是位置编码函数; $\rho$ 是诸如Softmax之类的归一化函数; $\beta$ 是一个关系函数,这里本文采用减法; $\gamma$ 是一个映射函数(例如MLP);标量注意力层计算由 $\varphi$ 和 $\psi$ 转换的特征之间的标量积,并将输出用作聚集由 $\alpha$ 转

换的特征的注意力权重;⊙是逐项点积.关系函数中的两项分别对应于点*x*与其局部邻域(本文采用*K*个最近邻)中的一组点.本文在每个数据点周围的局部邻域内应用自注意力,其中映射函数采用具有两个线性层和一个RELU非线性层的MLP.本文利用向量自注意力机制的原理,进而设计了新的特征提取模块Trans block,如图1所示,其中*F1*为原始特征向量,*F2*为经过注意力机制调整后的向量.图2为在目标点附近A-GConv处理的图解.

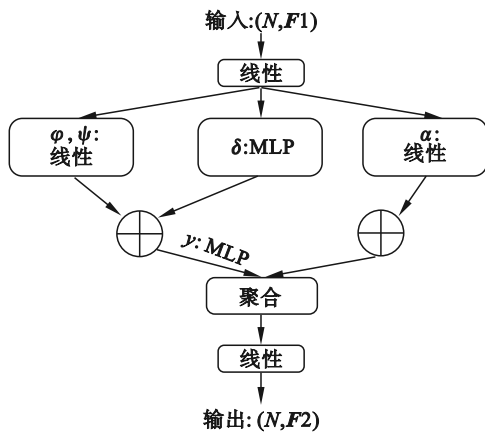


图1 转换模块

Fig. 1 Transformation block

将输入点云表示为 $X=\{x_i|i=1,2,\dots,N\}\in\mathbf{R}^{N\times 3}$ ,相应的点的特征定义为 $F=\{f_i|i=1,2,\dots,N\}\in\mathbf{R}^{N\times D}$ .在这里 $x_i$ 定义为第*i*个点的三维坐标( $x,y,z$ ),而在其他情况下可以与其他属性的向量组合(例如颜色和法向量).然后本文从给定的点云计算有向图 $G(V,E)$ , $V=\{1,2,\dots,N\}$ 和 $E\subseteq V\times V$ 分别表示点和边的集合.本文通过使用*K*最近邻采样(KNN)的方法来构建图.在给定*D*维特征的输入时,A-GConv层可以输出具有相同点数的*M*维

新特征,用于更准确地反映局部几何特征.

为解决固定卷积核无法精确提取特征的缺点,本文使用自注意力机制设计自适应内核来捕捉每对点之间的关系,对于输出的*M*维特征,本文的A-GConv使用自注意力机制对每个点的点特征( $f_i, f_j$ )生成动态卷积核:

$$e_{ijm} = T_m([f_i, f_j - f_i]). \quad (3)$$

式中:*m*表示本文的A-GConv中定义的单个滤波器对应的*M*个输出维度之一,同时为了结合局部邻域中捕获的全局结构和特征差异,本文将 $[f_i, f_j - f_i]$ 作为构建自适应卷积核的输入特征,其中 $[\cdot, \cdot]$ 是串联运算; $T_m$ 为特征映射函数,本文通过向量自注意力机制将其定义为

$$T_m(\Delta f_{ij}) = \rho(\gamma(\varphi(\Delta f_{ij}) - \psi(\Delta f_{ij}) + \delta)) \odot \alpha(\Delta f_{ij}). \quad (4)$$

本文只对点的空间关系进行卷积操作来避免可能带来的特征信息冗余,得到公式:

$$F_{\text{OUT}} = \sigma(A(\{e_{ijm} \cdot \Delta x_{ij}\})). \quad (5)$$

其中: $\Delta x_{ij}$ 表示每个点的坐标和其邻域内相对坐标的串联,*A*代表着对应点的空间关系.本文通过将*M*维度的自适应内核与相应的空间关系相结合,这样使得输入空间中的空间位置可以有效地合并到每一层中,并与从自适应内核提取的特征相结合,最后通过对每个通道进行叠加进而得到点的边缘特征 $E_{ij}=\{E_{ij1}, E_{ij2}, \dots, E_{ijM}\}$ ,然后通过聚合函数来处理邻域中所有的边特征进而来定义中心点的输出特征.由于在卷积核内对点的输入顺序是不变的,因此在聚集函数*A*对称的情况下,本文的A-GConv满足置换不变性.同时在以图特征为特征输入的前提下,本文的模型对刚性变换(例如平移,旋转)也具有一定的鲁棒性.

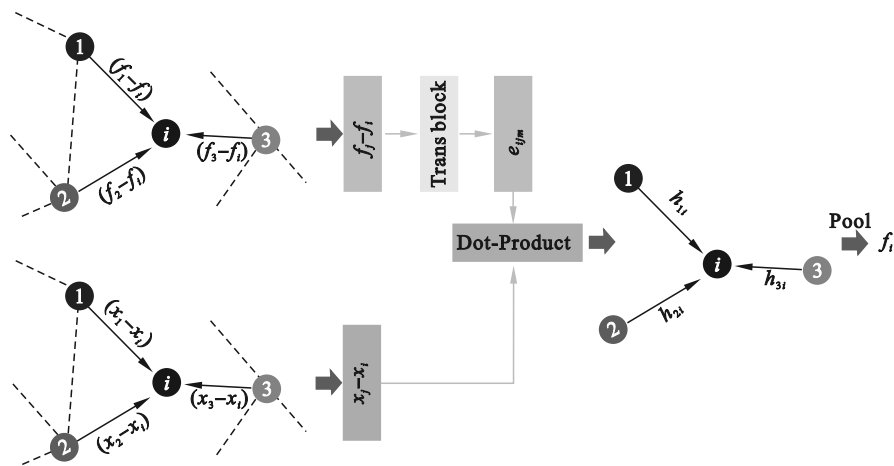


图2 在目标点附近A-GConv处理的图解

Fig. 2 Diagram of A-GConv module processing near the target point

### 1.2 特征选择

A-GConv 通过根据中心点与各个邻域点的单独特征  $[f_i, f_j]$  生成自适应内核,然后将该内核应用于点对  $[x_i, x_j]$  以描述点在输入空间中的空间关系.输入也可以包括表示其他有价值属性的附加维度,例如颜色、法线、深度等.此时本文通过修改输入的维度,也可以去捕捉来自不同特征维度和空间坐标之间的关系,这是 A-GConv 设计中的另一选项.本文默认只使用空间位置作为输入.

本文选择空间位置作为输入而不是特征鉴于以下原因:①学习自适应核已经将点特征包含在内,再与特征卷积会导致特征信息的冗余;②MLP 对于空间关系的学习更加容易;③通过特征计算会导致内存成本的增加.

### 1.3 网络结构

#### 1.3.1 分类网络

分类网络结构如图 3 所示,本文直接应用了动态图结构同时舍弃了池化和插值操作.分类网络首先使用两个 A-GConv 模块来提取几何特征,并将点云特征维度由 3 维升至 64 维.使用自定义数量的 ResG Block 模块得到更高维(128, 256, 512 或 1 024)点云,随后使用全局池化来获得点云的全局特征,最后通过全连接层变换全局特征进行分类.其中 ResG Block 模块为残差连接的图卷积集成模块,可以用不同数量以适应不同的网络需求(权衡精度与网络复杂度).具体来

说,每一层图结构都是通过计算各个点之间的特征相似度来进行更新的,而非直接使用固定的空间位置.在每一层中,本文通过对边集合的重新计算来使得邻域内点的特征最接近中心点,同时鼓励网络从语义上构建图结构,并通过将特征空间中的相似点分组在一起来扩展中心点的感受野.

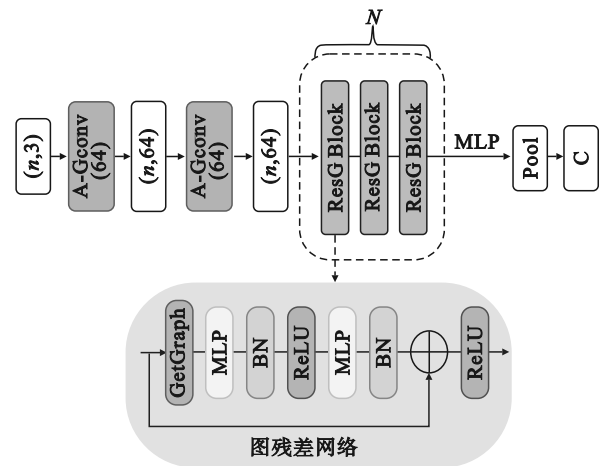


图 3 分类网络结构

Fig. 3 Classification network

#### 1.3.2 分割网络

对点云进行降采样至点集基数  $(N, N/4, N/16, N/64)$ ,根据采样点构建新的图并汇集图结构特征(见图 4).在每个图池化层后进行升维(128, 256, 512),然后通过 MLP 获得全局特征,即图中最右侧方块.

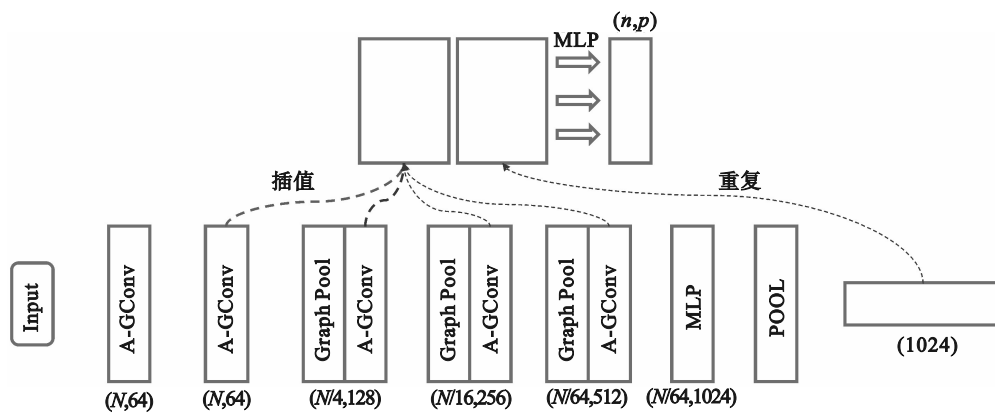


图 4 用于分割任务的网络结构

Fig. 4 Part segmentation network

将深层次的特征从子采样层插值到原始点,图中插值采用最邻近插值后将全局特征复制维度至  $(N, 1 024)$  与完成插值的局部特征连接为最终点特征,最后使用多个共享的全连接层来转换逐点特征.整体来看,本文的 A-GConv 分割网络

通过逐步子采样和池化点云来捕捉较粗粒度的全局特征,然后将这些特征插值到较细粒度的尺度上,以获得点级别的准确分割预测,从而在实现了对大型点云的高效处理的同时,仍可以捕捉局部细节.

## 2 实验结果

### 2.1 ModelNet40 上的形状分类

本文遵循 PointNet 的实验设置,对每个模型对象统一采样 1 024 个点,同时只使用采样点的位置坐标( $x, y, z$ )作为输入。

本文根据每一层的特征相似度重新计算图,对于每一层的邻域大小都设置为 20,然后使用正常图卷积,最后使用一个共享的全连接层(1 024)来聚合多尺度特征。所有层中都使用批量规范化处理,同时使用 LeakyReLU 进行非线性处理,使用 MaxPooling 函数作为提取全局特征的方法。对于优化器,本文使用 SGD 优化器,动量设置为 0.9,初始学习率设置为 0.1,并使用余弦退火算法<sup>[12]</sup>将学习率降至 0.001,所有训练模型的批量大小设置为 32。本文使用 PyTorch 实现代码,并在 RTX 4090 GPU 上部署并训练网络。上述超参数以类似的方式用于其他任务。

评估指标是平均类精度(mAcc)和总体精度(OA)。同时为了更加清晰地比较,表 1 中列出了输入的数据类型以及每个方法对应的输入点数<sup>[13-17]</sup>。本文只使用 1 024 个点的坐标作为输入,不进行任何数据增强操作,这已经优于使用较大输入的其他方法。

表 1 基于 ModelNet40 数据集的分类结果

Table 1 Classification results based on ModelNet40 dataset

方法	输入	点数量	mAcc/%	OA/%
VoxNet	voxel	—	83.0	85.9
PointNet	xyz	1 024	86.0	89.2
PointNet++	xyz	1 024	—	90.7
PointNet	xyz	5 120	—	91.9
PointCNN	xyz	1 024	88.1	92.5
KPCConv	xyz	7 168	—	92.9
DGCNN	xyz	1 024	—	92.9
Point Trans	xyz	1 024	—	92.8
PCT	xyz	1 024	—	93.2
PointNext	xyz	1 024	—	93.2
本文	xyz	1 024	90.7	93.5

### 2.2 ScanObjectNN 上的形状分类

ModelNet40 数据集是评估点云分类任务的典型基准,但由于该数据集的综合性质以及目前点云分析的快速发展,ModelNet40 数据集可能无法满足现代方法的要求。本文还选择在

ScanObjectNN 数据集<sup>[18]</sup>上进行了分类实验。

ScanObjectNN 数据集是近期发布的一个点云数据集,该数据集中包括一共 15 000 个对象,这些对象被分为 15 个类,同时在现实世界中存在 2 902 个独特的对象实例。由于遮挡、噪音和背景的存在,该数据集对现有的点云分析方法提出了新的挑战。在本文的实验中,考虑了最难的扰动变体。本文使用了 SGD 优化器训练模型 200 个 epoch,批量大小仍设置为 32,分类结果见表 2。

表 2 在 ScanObjectNN 数据集上的分类结果  
Table 2 Classification results on the ScanObjectNN dataset

方法	mAcc	OA
PointNet	63.4	68.2
PointNet++	75.4	77.9
PointCNN	75.1	78.5
DGCNN	73.6	78.1
PRANet <sup>[19]</sup>	79.1	82.1
本文	79.3	82.7

由于数据集中的数据多存在遮挡、噪音等干扰,所以对于局部特征的提取更加重要。本文的模型着重于在局部几何特征的提取效果,同时在这个数据集上不仅得到了更高的准确率,同时平均准确率也稳定在较高的数值,验证了本文提出模型的有效性。

### 2.3 部件分割

在 ShapeNetPart 数据集上本文遵循 PointNet<sup>[3]</sup>中的实验设置。表 3 列出了平均类别交并比(mcIoU)和平均交并比(mIoU)。遵循 PointNet 中的评估方案,每个形状的 IoU 是通过每个部分的 IoU 取平均得到的。平均 IoU(mIoU)是通过对所有测试实例的 IoU 取平均计算得到的。平均类别 IoU(mcIoU)是所有形状类别的平均 IoU。与其他方法相比,本文模型取得了更为优秀的结果。

表 3 在 ShapeNetPart 数据集上的分割结果  
Table 3 Segmentation results on ShapeNetPart dataset

算法	mcIoU	mIoU
PointNet	80.4	83.7
PointNet++	81.9	85.1
PointCNN	84.6	86.1
DGCNN	82.3	85.2
PRANet	85.1	86.4
本文	84.8	86.7

## 2.4 消融实验

### 2.4.1 网络深度

本文通过在分类网络中设置不同数量的 ResG Block 来改变整理网络的深度,然后在 ModelNet40 数据集上进行评估.实验结果如表 4 所示,可以看出,虽然增加深度可以提升网络性能,但当到达一定深度后,增加深度并不总是能带来更好的性能.在使用 3 个 Block 层时取得了最优的效果.

表 4 不同数量的 ResG Block 在 ModelNet40 数据集上的分类精度  
Table 4 Classification accuracy of different numbers of ResG Block on ModelNet40 dataset %

Block	mAcc	OA
1	87.9	90.1
2	89.3	92.3
3	90.7	93.3
4	89.5	92.7

### 2.4.2 邻域点数量

本文研究了邻域点数量  $K$  的设置,结果如表 5 所示.随着邻域点数量的变化,当邻域点过少时,模型缺少足够的信息进行特征提取.而当邻域点过多时,许多被选中的数据点可能距离过远并不能提供有效的局部信息,同时也可能引入噪声进而降低网络的准确度.经过测试,将  $K$  设置为 20 时获得最优性能.

表 5 使用不同邻域点数量  $K$  的分类精度  
Table 5 Classification accuracy using the number of different neighborhood points  $K$  %

$K$	mAcc	OA
10	89.9	92.9
20	90.7	93.3
30	89.5	92.1
40	86.8	89.7

### 2.4.3 注意力类型

本文设计了 4 种类型算子进行比较,如表 6 所示.一是 MLP,它在模型中替代了点变换层,而没有引入自注意力机制;二是 MLP 和池化的组合,这样每个点可以与其邻域进行信息交换,但不使用自注意力机制;三是使用标量自注意力;四是本文的方法.实验表明,使用标量自注意力相比不使用自注意力效果更好,但仍不如本文使用的向量自注意力,由此证明使用向量自注意力在点云处理任务中更有利.

表 6 使用不同算子的分类精度

Table 6 Classification accuracy using different operators %

算子	mAcc	OA
MLP	86.5	88.3
MLP+Pooling	87.9	90.5
Scalar	89.5	92.2
本文	90.7	93.3

### 2.4.4 简易集成性

分别针对 PointNet<sup>[3]</sup>模型和 DGCNN<sup>[5]</sup>模型,在不修改两者网络结构的前提下,用 A-GConv 模块对原始点云数据进行特征提取,随后将提取出的特征数据输入两个网络,如表 7 所示.在使用 A-GConv 模块后,两个网络的性能都得到提升,由此证明本文提出的 A-GConv 模块可以很容易地集成到别的卷积网络中,进而提高模型性能.

表 7 集成 A-GConv 模块后网络的分类精度  
Table 7 Classification accuracy of network after integration of A-GConv module %

算子	mAcc	OA
PointNet	86.0	89.2
A-GConv+PointNet	87.5	90.5
DGCNN	—	92.9
A-GConv+DGCNN	—	93.1

## 3 结 论

1) 本文提出的 A-GConv 模块没有使用固定的动态内核来捕捉点之间的关系,而是通过自注意力机制去规范点自身特征,进而更灵活地塑造三维点云的几何结构特征.

2) 本文的模型在几个公共数据集上取得较优的表现.

3) A-GConv 模块可以集成到别的卷积网络中,进而提高其模型的原始性能.

### 参考文献:

- [1] Riegler G, Ulusoy A O, Geiger A. OctNet: learning deep 3D representations at high resolutions [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).Honolulu,2017:6620-6629.
- [2] Klokov R, Lempitsky V. Escape from cells: deep KD-networks for the recognition of 3D point cloud models[C]//2017 IEEE International Conference on Computer Vision (ICCV).Venice,2017:863-872.
- [3] Charles R Q, Hao S, Mo K C, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).Honolulu,2017:77-85.

