融合可变形核和自注意力的点云分类分割边卷积网络

陈思帆[1]，杨家志[1,2*]，黄琳[3]，吕志玮[1]，沈露[1]

（1.桂林理工大学计算机科学与工程学院，广西 桂林 541006；2.桂林航天工业学院机电工程学院，广西 桂林 541004；
3.桂林理工大学物理与电子信息工程学院，广西 桂林 541006）

摘　要：点云数据具有无序性和离散分布的特点，传统动态图卷积方法在处理点云数据时仍充满挑战，无法准确地表示三维点间的特征对应关系。为此，提出一种融合可变形核和自注意力的点云分类分割边卷积网络 DKSA-Net，该网络由 DKConv（Deformable Kernels Edge Convolution）模块和 SAConv（Self-Attention Edge Convolution）模块组成。通过融合可变形核与边卷积构建 DKConv 模块，能够动态学习点的特征，生成可变形核，不会忽略不同特征之间的对应关系，从而更好地处理不同特征之间的对应关系。引入自注意力机制，并与边卷积结合构建 SAConv 模块，能够对特征进行更细粒度的特征提取，充分捕捉点云的重要特征，增强模型的判别能力。实验结果表明，DKSA-Net 在 ModelNet40 和 ShapeNet 数据集上取得出色性能，分别达到 93.4%的总体精度（OA）、90.7%的平均精度（mAcc）和 86.1%的平均并交比（mIoU），且有着较低模型复杂度和较好鲁棒性，具有优秀的点云数据处理能力。

关键词：可变形核；自注意力；点云分类；点云分割；深度学习

中图分类号：TP391　　**文献标志码：**A　　**DOI：**10.19678/j.issn.1000-3428.0069284

Edge Convolutional Network for Point Cloud Classification and Segmentation Incorporated Deformable Kernel and Self-Attention

CHEN Sifan[1], YANG Jiazhi[1,2*], HUANG Lin[3], LÜ Zhiwei[1], SHEN Lu[1]

(1. School of Computer Science and Engineering, Guilin University of Technology, Guilin 541006, Guangxi, China;
2. School of Mechanical and Electrical Engineering, Guilin University of Aerospace Technology, Guilin 541004, Guangxi, China;
3. School of Physics and Electronic Information Engineering, Guilin University of Technology, Guilin 541006, Guangxi, China)

【Abstract】 Owing to the unordered and discrete nature of point cloud data, traditional dynamic graph convolution method faces significant challenges in processing this data, making it difficult to accurately represent feature correspondences between 3D points. To address this issue, a network called DKSA-Net is proposed, which incorporates deformable kernels and self-attention. This network consists of two main modules: Deformable Kernels edge Convolution (DKConv) and Self-Attention edge Convolution (SAConv). By integrating deformable kernels with edge convolution to construct the DKConv module, the network can dynamically learn point features, generate deformable kernels, and maintain feature correspondences, thereby improving the handling of feature correspondences. In addition, by introducing the self-attention mechanism and combining it with edge convolution to construct the SAConv module, the network can perform finer-grained feature extraction, fully capture important point cloud features, and enhance the discriminative ability of the model. The experimental results show that DKSA-Net achieves excellent performance on the ModelNet40 and ShapeNet datasets, with an Overall Accuracy (OA) of 93.4%, an average Accuracy (mAcc) of 90.7%, and an average Intersection-over-Union (mIoU) of 86.1%. Furthermore, it demonstrates relatively low model complexity and high robustness, showcasing exceptional capabilities in processing point cloud data.

【Key words】 deformable kernel; self-attention; point cloud classification; point cloud segmentation; deep learning

0　引言

随着三维数据采集技术的快速发展，三维点云数据在各领域被广泛应用，涵盖了自动驾驶[1]、机器人[2]、虚拟现实[3]、文物保护[4]等多个领域，点云数据处理是实现三维视觉、环境建模和自动驾驶等先进技术的关键步骤。然而，由于三维点云数据具有离散分布、非结构化、无序性的特点，对其进行有效处理一直是一个巨大的挑战。尽管基于深度学习的三维点云处理方法在分类和分割等任务上取得了进展，但面对这项具有挑战性的任务，仍需要持续不断地创新和研究，以更好地满足未来三维点云数据分

收稿日期：2024-01-22　　**修回日期：**2024-03-17

基金项目：国家自然科学基金（62166012）。

通信作者 E-mail：* 29848260@qq.com

计算机工程 www.ecice06.com

析和应用的需求。

现有点云数据处理方法包括基于体素的方法、基于多视图的方法以及基于点的方法。在MATURANA等[5]提出的基于体素的VoxNet和SU等[6]提出的基于多视图的方法MVCNN(Multi-View Convolutional Neural Network)中，点云投影操作会导致一些真实点云对象的相对位置信息丢失，致使几何结构信息的损失。QI等[7]提出了基于点的方法——PointNet网络，直接使用不规则的点云数据作为输入，保持了点云的完整结构，克服了前2种方法的局限性。然而该方法忽略了点云的局部特征，仅关注其全局特征。而后QI等[8]提出PointNet++网络，通过将三维点云分组为多个邻域，并逐层提取局部信息，解决了PointNet网络无法提取局部特征的问题。相关学者根据PointNet++的设计思路，进行优化后提出了PointASNL[9]与So-Net[10]网络。但由于这2种方法都只处理局部点集中的每个点，而没有考虑点与其邻点之间的关系，导致局部特征提取不充分。为了更好地捕捉局部特征，WANG等[11]提出了DGCNN（Dynamic Graph CNN)模型，其核心模块为EdgeConv，通过聚合每个点及其k个邻域点的特征来提取局部特征。LI等[12]提出PointCNN，通过X-Conv将点转换成代表点的局部坐标系，并在该坐标系上进行卷积操作，从而实现了对点云数据的特征提取和学习。然而，标准的卷积操作无法在三维点间有区分地表示特征对应关系，对于不规则和非结构化的点云数据，无法应用具有固定大小或模式的内核进行学习。因此，LIN等[13]设计了3D-GCN模型，提出了可变形核(Deformable Kernels)的概念，其中，每个核的形状和权重在训练期间是可学习的。然而其卷积过程中舍弃了邻域中的许多特征，过于依靠核中点进行匹配，无法充分捕捉点云的重要特征，影响了模型性能。

针对传统动态图卷积方法无法准确地表示三维点间的特征对应关系的问题，提出了一种深度学习点云网络DKSA-Net，设计了DKConv（Deformable Kernels Edge Convolution）模块和SAConv（Self-Attention Edge Convolution）模块。DKConv模块由DGCNN[11]的EdgeConv模块与3D-GCN[13]的Deformable Kernels融合而成，模块中的核的形状和权重是可学习的，且保留了邻域的重要特征，能够有效地得到不同语义部分间的多种关系。SAConv模块由DGCNN[11]的EdgeConv模块与Transformer[14]中的Self-Attention模块组合而成，该模块能够对上一神经网络层的特征进行更细粒度的特征提取。DKSA-Net通过融合可变形核和边卷积弥补了传统动态图卷积的问题，能够更好地处理不同特征之间的对应关系，并且使用自注意力机制增强捕获特征的能力，从而更好地捕捉了整个点云的全局上下文信息特征。

1 DKSA-Net 网络组成

本文提出了一个新的点云网络DKSA-Net来执行点云分类分割任务，图1为DKSA-Net网络模

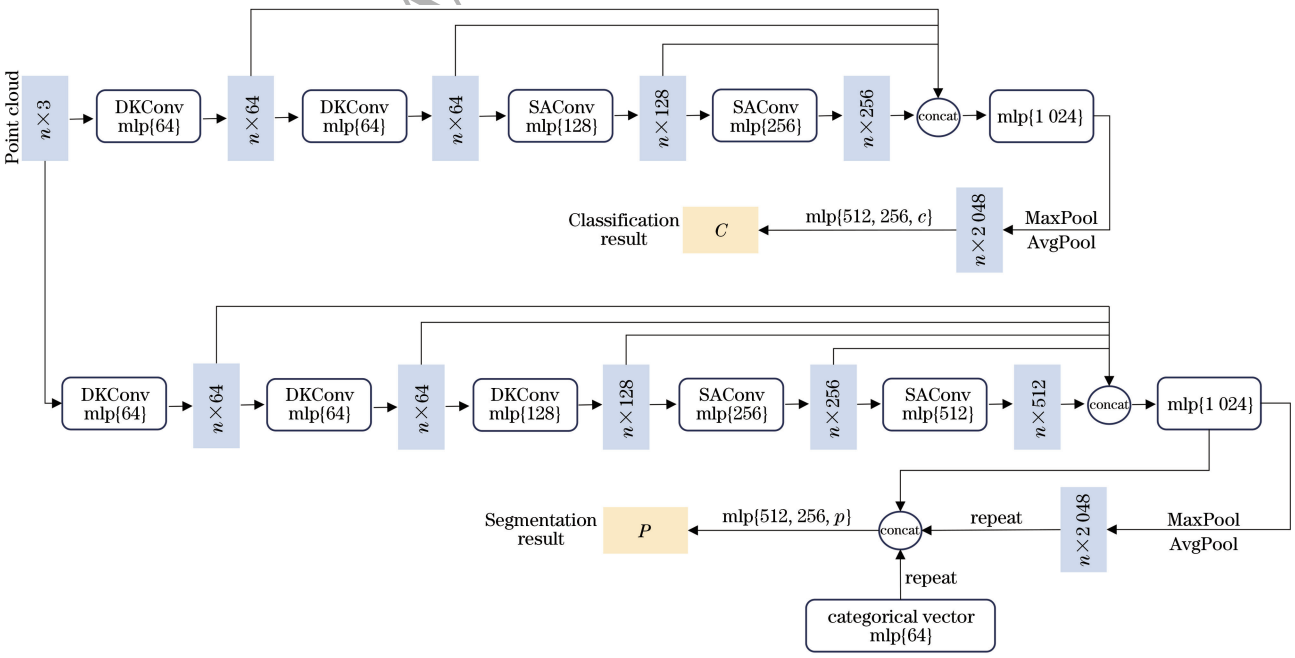

图 1 DKSA-Net 网络模型架构

Fig. 1 DKSA-Net network model structure

型架构(彩色效果见《计算机工程》官网 HTML 版,下同),该图上半部分为模型的分类网络,下半部分为模型的分割网络。DKSA-Net 网络模型主要由 DKConv 模块和 SAConv 模块组成,其中 DKConv 模块由 Deformable Kernels 和 EdgeConv 模块融合而成,SAConv 模块由 EdgeConv 模块和 Self-Attention 模块组成。接下来将对该模型的分类网络和分割网络以及各模块进行介绍。

1.1 分类网络

DKSA-Net 分类网络接收大小为 $n\times3$ 的输入数据,经过 2 个低维 DKConv 模块,分别产生 $n\times64$ 的特征向量。接着,这些特征向量经过 2 个相对高维的 SAConv 模块,得到 $n\times128$ 和 $n\times256$ 的特征向量。随后,将这 4 个特征向量合并得到全局特征,全局特征分别经过平均池化和最大池化层得到 2 个高维的特征向量再进行 concat 操作得到最后的 $n\times2\,048$ 特征向量。最终,将特征输入到 3 层全连接网络中,用于分类预测。在全连接网络的前 2 层中,采用随机丢弃神经节点技术,以一定的概率 p 暂时丢弃神经元,设置丢弃概率 $p=0.5$。最后一层全连接层输出最终的分类预测结果 C。

1.2 分割网络

DKSA-Net 分割网络与分类网络很相似,其由 3 个 DKConv 模块和 2 个 SAConv 模块构成以捕捉点云的全局特征,接着使用多层感知机对这些特征进行聚合,然后将全局特征经过最大池化层和平均池化层,通过 concat 操作形成高维特征向量。再将高维特征向量、全局特征向量与部件特征向量进行 concat 操作拼接在一起,形成最终的用于部件分割所需的特征表示。接下来,将这个特征表示输入到全连接网络中,获得最终的部件分割分数 P。图 1 中的 categorical vector 表示点云类别的 one-hot 向量,repeat 表示复制操作。

1.3 DKConv 模块和 SAConv 模块

1.3.1 DKConv 模块

DKConv 模块由 Deformable Kernels 和 EdgeConv 模块融合而成,模块结构如图 2 所示,该模块可以有效地捕捉不同语义之间的多种关联,并更好地处理不同特征之间的相互关系,从而更全面地捕获整个点云的全局上下文信息特征。

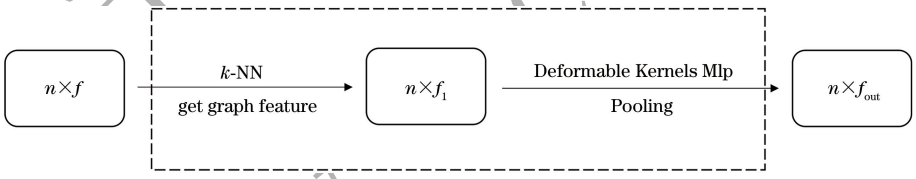

图 2 DKConv 模块

Fig. 2 DKConv module

k-近邻(k-NN)函数会找到与节点最接近的 k 个邻居节点,用于构建图的邻接关系,确定节点之间的连接关系和邻居节点;get graph feature 函数用于将节点的邻居节点特征聚合到当前节点以生成新的特征表示;Deformable Kernels Mlp 是可变形核在点云中的应用,卷积核根据点与点之间的关系,可以在点云中动态地变形,以适应不同的局部特征,如式(1)所示:

$$D_{ijm}=\mathrm{ReLU}((x_i,x_j-x_i)\cdot[x_i,x_j-x_i]) \tag{1}$$

式中:(x_i,x_j-x_i)为捕捉的特征;$[x_i,x_j-x_i]$为点之间的相关性。

该模块的输入为 $n\times f$,经过 k-NN 函数和 get graph feature 函数,得到 $n\times f_1$,用于获取图中节点的特征表示,之后经过 Deformable Kernels Mlp,允许卷积核根据局部特征自适应地调整形状和位置,有助于更好地捕获点云数据中的局部结构,最后输出为 $n\times f_{\mathrm{out}}$。

Deformable Kernels 结构如图 3 所示,通过进行关键点定位选择关键点,这些点通常根据点云的几何性质或标志性特征进行选择,如图中点云(飞机)选择了一个点,通过方向向量找到特征,然后为每个关键点引入可学习参数,如权重,使卷积核能够自适应地调整其形状,最终生成可变形核,应用调整

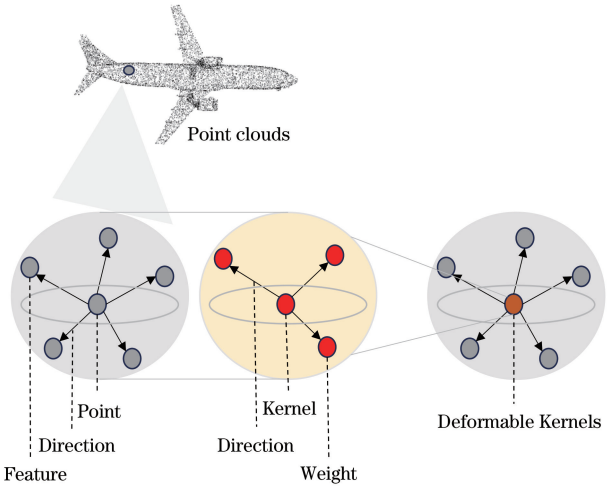

图 3 可变形核

Fig. 3 Deformable Kernels

后的卷积核进行卷积操作，从而更好地捕捉点云中的局部特征。

1.3.2 SAConv 模块

SAConv 模块由 EdgeConv 模块和 Self-Attention 模块组成，模块结构如图 4 所示。EdgeConv 模块的主要目标是从点云中学习局部特征和结构信息，通过建立邻接关系、聚合局部特征来实现，而 Self-Attention 模块能够在捕捉局部特征时，动态地分配不同点之间的关注权重，这有助于模型更好地理解点云中的重要关系和结构。EdgeConv 模块如图 5 所示。

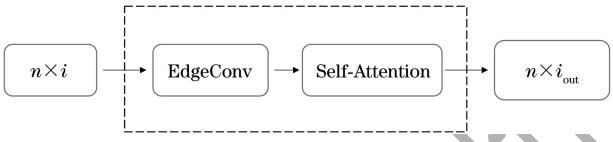

图 4 SAConv 模块

Fig. 4 SAConv module

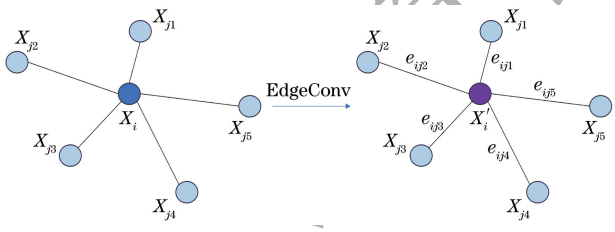

图 5 EdgeConv 模块

Fig. 5 EdgeConv module

首先，为了捕捉点云中的局部结构，EdgeConv 模块使用 k-NN 算法来构建每个点与其 k 个最近邻点之间的连接关系，形成一个 k-NN 图，所构建的图结构 G 的定义为：

$$G=(V,E) \tag{2}$$

式中：V 表示 N 个局部顶点集合；E 表示顶点之间的边的集合。V 和 E 的表达式如下：

$$V=\{p_i \mid i=1,2,\cdots,N\} \tag{3}$$

$$E=\{e_i=(e_{i_1},e_{i_2},\cdots,e_{i_j}) \mid i=1,2,\cdots,N\} \tag{4}$$

对于每个点，EdgeConv 模块将考虑其 k 个最近邻点的特征，并将这些特征聚合以生成边缘特征[其中边缘特征结合了全局形状结构 x_i 和局部邻域信息 x_j-x_i，如式(5)所示]，使用最大池化聚合边缘特征，这个特征聚合过程涉及到权重的计算，以确定每个邻居点对中心点的贡献。

$$e_{ij}=h_\Theta(x_i,x_j-x_i) \tag{5}$$

$$e'_{ijm}=\mathrm{ReLU}(\theta_m\cdot(x_j-x_i)+\phi_m\cdot x_i) \tag{6}$$

通过聚合邻居点的特征，EdgeConv 模块会生成更新的特征表示，反映点云中的局部关系，如式(6)、式(7)所示：

$$x'_{im}=\max_{j:(i,j)\in\varepsilon} e'_{ijm} \tag{7}$$

注意力机制在图像处理、自然语言学习等领域被广泛应用，它使得网络关注关键特征并忽略次要特征[15]。本文引入 Self-Attention 模块，其结构如图 6 所示。

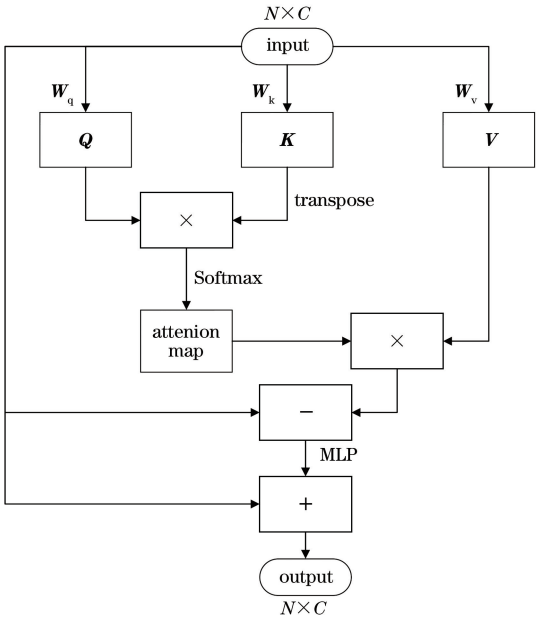

图 6 自注意力模块

Fig. 6 Self-Attention module

Self-Attention 模块接收了上一层边卷积模块的输出作为输入，使用可学习的权重矩阵 $\boldsymbol{W}_q$、$\boldsymbol{W}_k$ 和 $\boldsymbol{W}_v$ 进行线性变换，分别生成查询矩阵 $\boldsymbol{Q}$、关键字矩阵 $\boldsymbol{K}$ 和数值矩阵 $\boldsymbol{V}$：

$$(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=(\boldsymbol{W}_q,\boldsymbol{W}_k,\boldsymbol{W}_v)\times N \tag{8}$$

式中：N 为模块输入。

接着，利用得到的 $\boldsymbol{Q}$ 和 $\boldsymbol{K}$ 进行矩阵乘法计算它们之间的相关性：

$$\alpha=\boldsymbol{Q}\times\boldsymbol{K}^{\mathrm{T}} \tag{9}$$

再进行归一化得到注意力矩阵 $\boldsymbol{A}$，与 $\boldsymbol{V}$ 进行加权求和，得到特征 $\boldsymbol{O}$：

$$\boldsymbol{O}=\boldsymbol{V}\times\boldsymbol{A} \tag{10}$$

最后将其与 N 进行作差，再将值经过多层感知机(MLP)并与 N 求和，得到最终的结果。其中，“×”为矩阵乘法，“−”为矩阵减法，“+”为矩阵加法。

2 分类与部件分割实验

2.1 实验环境与实验设置

实验均在 PyTorch 深度学习框架上进行，版本为 1.10.0，CUDA 版本为 11.3，Python 版本为 3.8.10，所用 GPU 为 NVIDIA RTX 3090，软件平

台为 Windows 11 下 Pycharm 专业版。采用 ModelNet40 数据集和 ShapeNet 数据集分别对分类任务和部件分割任务进行性能测试。在数据增强方面，采用了随机旋转、移动和缩放方法。分类任务训练周期为 300 轮，批量大小为 32，采用余弦退火(Cosine Annealing)[16]方法动态调整学习率，最大迭代次数 $T_{max}=250$，达到 T_{max} 之后，学习率调整为 0.000 1，采用随机梯度下降(SGD)优化器进行优化。为了保证实验的公平性，实验中所需要的硬件环境在云 GPU 平台进行配置，软件环境则根据论文数据进行配置。

2.2 分类任务结果分析

为了验证 DKSA-Net 模型在分类任务上的有效性，使用 ModelNet40[17]公开数据集进行训练与测试。ModelNet40 数据集有 40 个类别分类，共有 12 311 个模型，其中 9 843 个模型用于训练，2 468 个模型用于测试。本文采用平均精度(mAcc)和总体精度(OA)作为点云分类的评价指标。实验结果如表 1 所示，表中标粗数值为最优结果，其中，voxel 表示体素输入，xyz 表示三维点坐标输入，xyz，normal 表示坐标＋点云法线向量输入。

由表 1 可知，DKSA-Net 在 ModelNet40 数据集上的 OA 和 mAcc 分别达到了 93.4%和 90.7%，取得了优异的表现。相比于经典算法 PointNet 和 PointNet＋＋，以及近几年基于深度学习的方法 DGCNN、SpiderCNN、3D-GCN、PointASNL、PRA-Net、MANet，在 OA 上分别高出 4.2、1.5、0.5、1.0、1.3、0.2、0.2、0.9 百分点。通过实验对比表明，本文所设计的 DKConv 模块和 SAConv 模块的堆叠使用使得 DKSA-Net 在分类任务上具有较强的性能，在输入仅为 1 024 个点的三维坐标中也能够保证分类任务过程中稳定且高质量地提取特征。

表 1 ModelNet40 数据集上的分类实验结果对比

Table 1 Comparison of classification experiment results on ModelNet40 dataset %

模型	输入	mAcc	OA
PointNet[7]	xyz	86.2	89.2
PointNet＋＋[8]	xyz，normal	—	91.9
VoxNet[5]	voxel	—	85.9
DGCNN[11]	xyz	90.2	92.9
SO-Net[10]	xyz，normal	87.3	90.9
SpiderCNN[18]	xyz，normal	—	92.4
3D-GCN[13]	xyz	—	92.1
PointASNL[9]	xyz，normal	—	93.2
PointCNN[12]	xyz	88.1	92.2
PCNN[19]	xyz	—	92.3
PRA-Net[20]	xyz	90.6	93.2
LFT-Net[21]	xyz，normal	89.7	93.2
RMFP-DNN[22]	xyz	88.9	92.6
PointStack[23]	xyz	89.6	93.3
AGCN[24]	xyz，normal	**90.7**	92.6
DTNet[25]	xyz	—	92.9
Ours	xyz	**90.7**	**93.4**

2.3 部件分割任务结果分析

为了验证 DKSA-Net 模型在部件分割任务上的有效性，使用 ShapeNet[26]公开数据集进行训练与测试。ShapeNet 数据集共有 16 个大的类别和 5 个小的部件类别，共有 16 881 个数据模型。本文采用类交并比(IoU)和平均交并比(mIoU)作为评价指标。实验结果如表 2 所示，表中标粗体数值为最优结果。部分物品分割可视化结果如图 7 所示。

表 2 ShapeNet 数据集上的部件分割实验结果对比

Table 2 Comparison of experiment results of component segmentation on ShapeNet dataset %

模型	mIoU	IoU															
		飞机	包	帽子	车	椅子	耳机	吉他	刀	灯	电脑	摩托	杯子	手枪	火箭	滑板	桌子
PointNet[7]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.3
PointNet＋＋[8]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
3D-GCN[13]	85.1	83.1	**84.0**	86.6	77.5	90.3	74.1	90.9	86.4	83.8	95.6	66.8	94.8	81.3	59.6	75.7	82.8
DGCNN[11]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
AGCN[24]	85.4	83.3	79.3	87.5	78.5	90.7	**76.5**	**91.7**	**87.8**	84.7	95.7	72.4	93.2	**84.0**	**63.7**	76.4	82.5
DTNet[25]	85.6	83.0	81.4	84.3	78.4	90.9	74.3	91.0	87.3	84.7	95.6	69.0	94.4	82.5	59.0	76.4	**83.5**
PCNN[19]	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
PointASNL[9]	**86.1**	84.1	84.7	**87.9**	**79.7**	**92.2**	73.7	91.0	87.2	84.2	95.8	**74.4**	**95.2**	81.0	63.0	76.3	83.2
Ours	**86.1**	**85.2**	81.6	86.9	79.5	91.4	75.5	**91.7**	87.6	**85.3**	**96.1**	73.5	95.0	82.4	60.1	**77.1**	82.7

计算机工程 www.ecice06.com

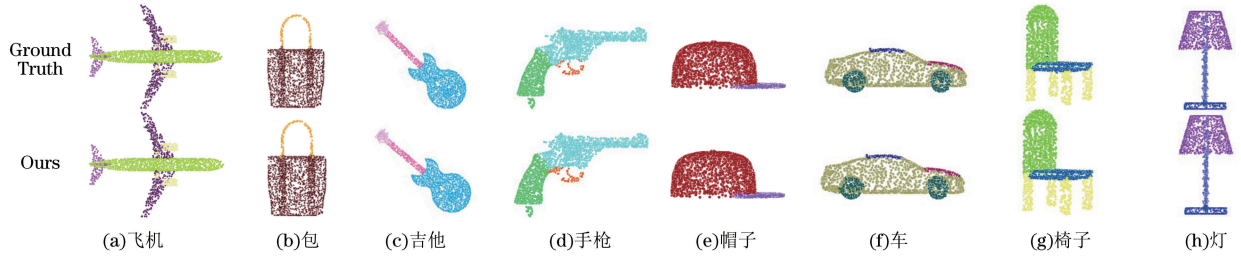

图 7 部件分割可视化结果

Fig. 7 Component segmentation visualization results

由表 2 可知，DKSA-Net 在 ShapeNet 数据集上的 mIoU 达到了 86.1%，其中飞机、吉他、灯、电脑、滑板类 IoU 分别达到了 85.2%、91.7%、85.3%、96.1%、77.1%，取得了优异的表现，相比于其他模型，这些物体部件分割效果最好。在 mIoU 上相比于 PointNet、PointNet++和 DGCNN，分别高出了 2.4%、1.0%、0.9%，与 PointASNL 并列。如图 7 所示，第 1 行为原始点云分割可视化结果，第 2 行为 DKSA-Net 分割可视化结果。从图中可以看出，模型的部件分割效果与原始数据十分接近，更加接近真实。表 2 和图 7 结果表明，DKSA-Net 在部件分割任务上取得优异的表现，验证了 DKSA-Net 的有效性，其可适应广泛的应用场景。

3 验证性实验

3.1 消融实验

本文采用 DKConv 模块和 SAConv 模块提取点云的全局特征与局部特征。为了验证这 2 个模块的有效性，本文对点云分类任务与部件分割任务进行消融实验。

3.1.1 点云分类消融实验

DKSA-Net 以 DGCNN 为框架，分类网络由 2 层 DKConv 模块和 2 层 SAConv 模块组成。实验通过逐层减少模块来进行验证 DKSA-Net 分类网络的有效性，为此设置了 5 组对比实验，实验设置如表 3 所示，其中 DKConv(64)表示 1 层特征维度为 64 的网络，DKConv(64,64)为 2 层网络，下同。

由表 3 实验数据可得，以 DGCNN 模型进行的第 1 组实验，得到了 mAcc 为 90.2%和 OA 为 92.9%的实验结果。在第 2 组实验中，添加了 1 层 DKConv 模块后，mAcc 和 OA 相比第 1 组实验下降了 0.2 和 0.1 百分点，这是由于在第 1 组中多个 EdgeConv 模块的不断堆叠获得的特征比起单个 DKConv 模块与 EdgeConv 模块组合更加丰富。在第 3 组实验中，通过再添加 1 层 DKConv 模块后，mAcc 和 OA 分别达到了 90.3%和 93.1%，相比第 1 组实验精度得到了提升，这是因为 2 个 DKConv 模块的组合使得特征的提取能力得到了提高，展示了可变形核在无序三维点云中描述局部结构信息的能力。在第 4 组实验中，在第 3 组的基础上再添加 1 层较高维的 SAConv 模块，在自注意力模块的作用下，对上一层网络的特征进行更细粒度的提取，使得模型性能得到了提高，mAcc 为 90.4%，OA 为 93.3%。在第 5 组实验中，使用与前面相同的思路，添加 1 层高维的 SAConv 模块，形成了本文的 DKSA-Net 分类网络模型，mAcc 和 OA 分别为 90.7%和 93.4%。实验表明 DKConv 模块和 SAConv 模块能够显著提升点云模型分类性能。

表 3 点云分类消融实验结果

Table 3 Result of point cloud classification ablation experiment %

Groups	Methods	mAcc	OA
1	DGCNN[11]	90.2	92.9
2	+DKConv(64)	90.0	92.8
3	+DKConv(64,64)	90.3	93.1
4	+DKConv(64,64)+SAConv(128)	90.4	93.3
5	+DKConv(64,64)+SAConv(128,256)(Ours)	**90.7**	**93.4**

3.1.2 点云部件分割消融实验

DKSA-Net 以 DGCNN 为框架，分割网络由 3 层 DKConv 模块和 2 层 SAConv 模块组成。实验通过逐层减少模块来进行验证 DKSA-Net 分割网络的有效性，为此设置了 6 组对比实验，实验结果如表 4 所示。

表 4 点云部件分割消融实验结果

Table 4 Result of point cloud component segmentation ablation experiment %

Groups	Methods	mIoU
1	DGCNN[11]	85.2
2	+DKConv(64)	85.4
3	+DKConv(64,64)	85.5
4	+DKConv(64,64,128)	85.8
5	+DKConv(64,64,128)+SAConv(256)	86.0
6	+DKConv(64,64,128)+SAConv(256,512)(Ours)	**86.1**

www.ecice06.com 计算机工程

由表 4 实验数据可以看出，DKSA-Net 分割网络模型的 mIoU 达到了 86.1%，部件分割消融实验设置与分类消融实验类似。实验验证了 DKSA-Net 分割网络中的 DKConv 模块和 SAConv 模块对模型的有效性，表明其对分割任务效果有着显著提升。

3.2 鲁棒性实验

为了验证 DKSA-Net 网络具有较好的鲁棒性，与经典网络 PointNet++和 DGCNN 进行对比，分别对分类模型和分割模型进行鲁棒性实验。对于分类模型，分别以 256、512、768、1 024 个点作为输入，在 ModelNet40 数据集上进行实验，实验结果如图 8 所示。对于分割模型，分别以 256、512、1 024、2 048 个点作为输入，在 ShapeNet 数据集上进行实验，实验结果如表 5 所示。

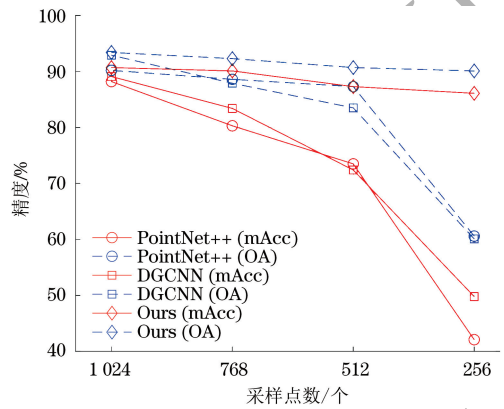

图 8 不同点云数量下的 ModelNet40 数据集分类精度

Fig. 8 Classification accuracy on ModelNet40 dataset with different number of point clouds

表 5 不同点云数量下的 ShapeNet 数据集部件分割精度

Table 5 Component segmentation accuracy on ShapeNet dataset with different number of point clouds %

输入点数/个	mIoU		
	PointNet++[8]	DGCNN[11]	Ours
256	41.7	43.2	76.5
512	63.5	68.3	80.6
1 024	80.7	81.1	84.2
2 048	85.1	85.2	86.1

由图 8 折线图可知，在分类网络中，当输入点数从 1 024 个开始减少时，PointNet++和 DGCNN 折线下降趋势大，不稳定，而本文网络下降趋势较为平缓，即使在输入仅有 256 个点时，mAcc 和 OA 依旧能保持在 86.5%和 90.1%。由表 5 实验结果可知，在分割网络中，当输入点数仅为 256 个时，mIoU 仍能保持在 76.5%。实验表明，DKSA-Net 模型在应对不同数量的点云输入时表现出色，具有很高的适应性。DKSA-Net 能够保持相对稳定的精度，这反映出了其在处理不同数据密度和复杂度下的出色鲁棒性。

3.3 超参数实验

为了验证 k-NN 算法中 k 值选取对模型性能的影响，对 DKSA-Net 分类网络在 ModelNet40 数据集上迭代 300 轮，输入 1 024 个点的情况下，进行超参数实验，实验结果如图 9 所示。

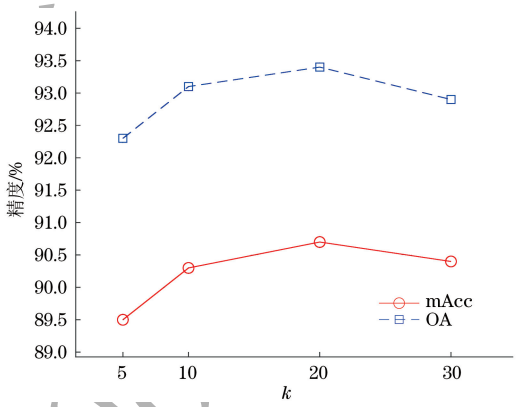

图 9 不同 k 值对网络模型的影响

Fig. 9 Effect of different k values on the network model

由图 9 实验结果可得，k 的不同取值影响了模型对于周围环境的感知范围和特征捕捉能力，从而影响了模型的性能。当 k 值较小时，感受野也小，模型将只考虑少数几个最近的邻居点，这虽然能够更好地捕获局部特征，但使得模型对于局部特征变化十分敏感，容易受到干扰。当 k 值过大时，模型将考虑更多的邻居点，能够更好地捕捉全局特征，然而较大的感受野容易导致模型失去对局部特征的敏感性，从而影响模型性能。因此本文选取 $k=20$ 来进行模型训练。

3.4 基于梯度的方法

为了测试不同的梯度下降法对模型性能的影响，本节使用 SGD、适应性矩估计(Adam)、自适应梯度算法(AdaGrad)在 ModelNet40 数据集上进行测试。

不同的梯度下降法会影响模型更新权重的能力，从而影响模型的性能。由表 6 可知，使用 ModelNet40 数据集进行点云分类时，SGD 表现效果最好，这是由于SGD适用于大型数据集，能够更快地

表 6 不同梯度下降法对网络模型的影响

Table 6 Effect of different gradient descent methods on the network model %

梯度下降法	mAcc	OA
SGD	90.7	93.4
Adam	88.3	91.7
AdaGrad	88.9	91.9

www.ciceu

处理每个小批量样本并更新权重。此外，SGD 相对于 Adam 和 AdaGrad 具有较小的内存占用，对于大规模数据集和资源受限的情况更具优势。

3.5 模型复杂度实验

在此节，通过测试不同模型在 ModelNet40 数据集上的一轮训练耗时和模型大小，以此为指标验证模型的复杂度。为确保实验结果的公平性，将所有实验模型参数设置为：训练周期为 300 轮，批量大小为 32，采用 Cosine Annealing 方法动态调整学习率，最大迭代次数 $T_{max}=250$，达到 T_{max} 之后，学习率调整为 0.000 1，采用 SGD 优化器进行优化。

由表 7 所示，DKSA-Net 在 ModelNet40 数据集上训练一轮耗时约为 94 s，相比于其他网络模型在一轮训练上所消耗的时间大幅度减少。训练所生成的模型大小为 8.46 MB，虽然比 DGCNN 的模型大小略大一点（这是由于 DKSA-Net 是以 DGCNN 为框架进行改进的），但相较于其他模型，仍然相对较小，复杂度更低。实验结果表明，在保持较短训练时间和较小模型规模的同时，DKSA-Net 能够保持优秀的性能。

表 7 不同模型在 ModelNet40 数据集上的性能对比

Table 7 Performance comparison of different models on ModelNet40 dataset

模型	一轮耗时/s	模型大小/MB	OA/%
PointNet[7]	214	22.30	89.2
PointNet++[8]	634	20.10	90.7
DGCNN[11]	183	6.90	92.9
3D-GCN[13]	146	15.70	92.1
Ours	94	8.46	93.4

4 结束语

为了更好地处理点云无序数据点上的信息，针对传统动态图卷积方法无法在三维点间准确地表示特征对应关系的问题及局部特征提取不充分的问题，本文提出了一种新的点云网络 DKSA-Net。DKSA-Net 通过将 DKConv 模块和 SAConv 模块结合起来，高效地获取不同语义部分间的关系及提取更细粒度的特征，能够有效地处理在三维点间表示特征之间的对应关系，其结合了邻域中的特征以及核中点间的匹配特征，增强了处理点云数据的能力，为点云产生更关键的全局上下文信息，提供了更稳健准确的预测。在分类数据集 ModelNet40 上，DKSA-Net 取得了 93.4% 的 OA 和 90.7% 的 mAcc；在分割数据集 ShapeNet 上，DKSA-Net 取得了 86.1% 的 mIoU。同时，DKSA-Net 展现出较高的鲁棒性，在逐渐减少采样点数的情况下，能保持一个稳定的精度，并且有着较低的模型复杂度。然而，本文对于模块的设计尚存在不足之处，未来将探索模型模块设置及参数设置，使其轻量化，进一步优化网络，并探索在语义分割任务上的应用。

参考文献

[1] LI Y, IBANEZ-GUZMAN J. Lidar for autonomous driving: the principles, challenges, and trends for automotive lidar and perception systems [J]. IEEE Signal Processing Magazine, 2020, 37(4): 50-61.

[2] AHMED S M, TAN Y Z, CHEW C M, et al. Edge and corner detection for unorganized 3D point clouds with application to robotic welding[C]//Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). New York, USA: ACM Press, 2018: 7350-7355.

[3] RAHMAN M M, TAN Y H, XUE J, et al. Notice of violation of IEEE publication principles: recent advances in 3D object detection in the era of deep neural networks: a survey[J]. IEEE Transactions on Image Processing, 2020, 29: 2947-2962.

[4] 张瑞菊，周欣，赵江洪，等. 一种古建筑点云数据的语义分割算法[J]. 武汉大学学报(信息科学版)，2020，45(5)：753-759.
ZHANG R J, ZHOU X, ZHAO J H, et al. A semantic segmentation algorithm of ancient building's point cloud data [J]. Geomatics and Information Science of Wuhan University, 2020, 45(5): 753-759. (in Chinese)

[5] MATURANA D, SCHERER S, MATURANA D, et al. VoxNet: a 3D convolutional neural network for real-time object recognition[C]//Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). New York, USA: ACM Press, 2015: 922-928.

[6] SU H, MAJI S, KALOGERAKIS E, et al. Multi-view convolutional neural networks for 3D shape recognition[C]// Proceedings of the IEEE International Conference on Computer Vision (ICCV). Washington D. C., USA: IEEE Press, 2015: 945-953.

[7] QI C R, SU H, MO K, et al. PointNet: deep learning on point sets for 3D classification and segmentation [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Washington D. C., USA: IEEE Press, 2017: 652-660.

[8] QI C R, YI L, SU H, et al. PointNet++: deep hierarchical feature learning on point sets in a metric space[EB/OL]. [2023-10-12]. http://arxiv.org/abs/1706.02413.

[9] YAN X, ZHENG C D, LI Z, et al. PointASNL: robust point clouds processing using nonlocal neural networks with adaptive sampling [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Washington D. C., USA: IEEE Press, 2020: 1-10.

[10] LI J X, CHEN B M, LEE G H. SO-Net: self-organizing network for point cloud analysis [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2018: 9397-9406.

[11] WANG Y, SUN Y B, LIU Z W, et al. Dynamic graph CNN for learning on point clouds [J]. ACM Transactions on Graphics, 2019, 38(5): 1-12.

[12] LI Y, BU R, SUN M, et al. PointCNN: convolution on X-transformed points[EB/OL]. [2023-10-12]. http://arxiv.

计算机工程 www.ecice.com

org/abs/1801.07791.
[13] LIN Z H, HUANG S Y, WANG Y F. Convolution in the cloud: learning deformable kernels in 3D graph convolution networks for point cloud analysis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Washington D. C., USA: IEEE Press, 2020: 1800-1809.
[14] VASWANI A, SHAZEER N, PARMAR N. Attention is all you need[EB/OL]. [2023-10-12]. https://arxiv.org/abs/1706.03762? context=cs.LG.
[15] 任欢, 王旭光. 注意力机制综述[J]. 计算机应用, 2021, 41(S1): 1-6.
REN H, WANG X G. Review of attention mechanism[J]. Journal of Computer Applications, 2021, 41(S1): 1-6. (in Chinese)
[16] LOSHCHILOV I, HUTTER F. SGDR: stochastic gradient descent with warm restarts[EB/OL]. [2023-10-12]. https://arxiv.org/abs/1608.03983v5.
[17] WU Z R, SONG S R, KHOSLA A, et al. 3D ShapeNets: a deep representation for volumetric shapes[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Washington D. C., USA: IEEE Press, 2015: 1912-1920.
[18] XU Y F, FAN T Q, XU M Y, et al. SpiderCNN: deep learning on point sets with parameterized convolutional filters[C]// Proceedings of European Conference on Computer Vision (ECCV). New York, USA: ACM Press, 2018: 90-105.
[19] ATZMON M, MARON H, LIPMAN Y. Point convolutional neural networks by extension operators[EB/OL]. [2023-10-12]. https://arxiv.org/abs/1803.10091v1.
[20] CHENG S L, CHEN X W, HE X W, et al. PRA-Net: point relation-aware network for 3D point cloud analysis[J]. IEEE Transactions on Image Processing, 2021, 30: 4436-4448.
[21] GAO Y B, LIU X B, LI J, et al. LFT-net: local feature transformer network for point clouds analysis[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(2): 1-11.
[22] 田钰杰, 管有庆, 龚锐. 一种鲁棒的多特征点云分类分割深度神经网络[J]. 计算机工程, 2021, 47(11): 234-240.
TIAN Y J, GUAN Y Q, GONG R. A robust deep neural network for multi-feature point cloud classification and segmentation[J]. Computer Engineering, 2021, 47(11): 234-240. (in Chinese)
[23] WIJAYA K T, PAEK D H, KONG S H. Advanced feature learning on point clouds using multi-resolution features and learnable pooling[J]. Remote Sensing, 2024, 16(11): 1835.
[24] XIE Z Y, CHEN J Z, PENG B. Point clouds learning with attention-based graph convolution networks[J]. Neurocomputing, 2020, 402: 245-255.
[25] HAN X F, JIN Y F, CHENG H X, et al. Dual transformer for point cloud analysis[EB/OL]. [2023-10-12]. https://arxiv.org/abs/2104.13044.
[26] YI L, KIM V G, CEYLAN D, et al. A scalable active framework for region annotation in 3D shape collections[J]. ACM Transactions on Graphics, 2016, 35(6): 1-12.

编辑 金胡考

计算机工程 www.ecice06.com