# 集多头点注意力与边卷积的点云分类分割模型

熊伟[1,2)], 娄政浩[1)*], 徐敏夫[1)], 袁和金[1,2)]

[1)] (华北电力大学计算机系 保定 071000)
[2)] (河北省能源电力知识计算重点实验室 保定 071000)
(louzh@ncepu.edu.cn)

**摘 要:** 针对动态图卷积模型只在局部尺度上独立提取点特征, 未将局部点互相关联的问题, 提出了一种集多头点注意力与边卷积的点云分类和分割模型. 首先, 设计单头点注意力模块分别计算点云的注意力特征与邻域注意力特征, 学习点云的旋转不变性, 使用多头机制将单头点注意力模块进行聚合, 构建多头点注意力模块, 赋予邻域内不同点相应的注意力系数; 其次, 设计加权金字塔池化模块进行特征融合, 获得更加丰富的特征信息; 最后, 提出结合交叉熵损失和焦点损失的联合损失函数, 解决数据集中存在的难分类样本和类别不平衡问题. 在 ModelNet40 数据集和 ShapeNet 数据集上分别进行了点云分类与分割实验, 在 ModelNet40 数据集上, 所提模型的总体精度提升到了 94.1%; 在 ShapeNet 数据集上的平均交并比提升到了 86.3%, 有效地提升了模型的分类和分割性能.

**关键词:** 点云数据; 分类与分割; 多头点注意力; 边卷积; 特征融合; 损失函数
**中图法分类号:** TP391.41 **DOI:** 10.3724/SP.J.1089.2023-00126

# Edge Convolution with Multi-Head Point Attention for Point Cloud Classification and Segmentation

Xiong Wei[1,2)], Lou Zhenghao[1)*], Xu Minfu[1)] , and Yuan Hejin[1,2)]

[1)] (*Department of Computer, North China Electric Power University, Baoding* 071000)
[2)] (*Hebei Key Laboratory of Knowledge Computing for Energy & Power, Baoding* 071000)

**Abstract:** A point cloud classification and segmentation model MHPAEC, which combines multi-head point attention and edge convolution, is proposed to address the problem of DGCNN only extracting point features independently at local scales without correlating local points. Firstly, design a single-head point attention module to calculate the attention features of the point cloud and the neighborhood attention features, learn the rotation invariance of the point cloud, use a multi-head mechanism to aggregate the single-head point attention modules, construct a multi-head point attention module, and assign corresponding attention coefficients to different points in the neighborhood. Secondly, design a weighted pyramid pooling module for feature fusion to obtain richer feature information. Finally, aiming at the problem of difficult to classify samples and category imbalance in the task of point cloud classification and segmentation, a joint loss function combining cross entropy loss and focal loss is proposed to make samples fully trained point cloud classification and segmentation experiments were conducted on the ModelNet40 dataset and ShapeNet dataset, respectively. On the ModelNet40 dataset, the overall accuracy of the MHPAEC network reached 94.1%; on the ShapeNet dataset, the mean intersection over union of the MHPAEC network reached 86.3%, effectively

收稿日期: 2023-05-23; 修回日期: 2023-10-09. **熊伟**(1976—), 男, 博士, 高级工程师, 硕士生导师, 主要研究方向为计算机网络、元学习; **娄政浩**(1998—), 男, 硕士研究生, 论文通信作者, 主要研究方向为计算机视觉、点云分类与分割; **徐敏夫**(1997—), 男, 硕士, 助理工程师, 主要研究方向为计算机视觉; **袁和金**(1977—), 男, 博士, 副教授, 硕士生导师, 主要研究方向为模式识别与计算机视觉.

improving the classification and segmentation performance of the network model.

**Key words:** point cloud data; classification and segmentation; multi-head point attention; edge convolution; feature fusion; loss function

点云数据作为三维空间中最重要的数据形式之一，广泛应用于自动驾驶[1]、机器人导航[2]、三维建模[3]中. 点云分类和分割是点云数据处理中的核心问题，其主要目标是将点云数据划分为不同的类别或对象，以实现对点云的理解和分析. 然而，点云数据的无序性和复杂性给点云分类和分割带来了巨大挑战，第一，点云数据体现出非结构化特性，这种非欧几里得数据没有明显的拓扑结构和规则网格，因此难以处理和表示；第二，点云数据中的点分布不均匀，有些区域可能非常密集，而其他区域则会稀疏，这可能导致模型难以学习区域间的特征信息. 所以有必要在考虑点云数据特性的前提下，学习更深层次的点特征信息与邻域特征信息，有助于提升点云分类和分割任务的性能.

传统的机器学习方法往往是将点云数据转换为其他形式的数据，如体素[4]或图像[5]，但是要对数据进行预处理和配准，需要更多的计算资源和时间. 为了解决这些问题，一些直接处理原始点云数据的方法被提出. PointNet[6]使用空间转换网络对每个点的特征向量进行变换，通过聚合整个点云数据提取特征，但是其需要对每个点进行独立的操作，计算量较大且耗时较长. PointNet++[7]将点云数据分层处理，逐步聚合局部信息，更加精确地捕捉点云数据的局部特征；缺点是会丢失一些全局上下文信息，例如，点云的整体形状和结构. 3D-GCN[8]模型采用图卷积神经网络处理点云数据，更准确地捕捉点云数据的结构信息，但是有许多超参数需要手动调节，且这些超参数对模型的性能影响较大. MSP-Net[9]提出多尺度局部区域划分算法并进行多尺度特征融合，但是未考虑单一尺度内局部区域之间的关系. DGCNN[10]采用动态图卷积网络进行局部邻域处理，在每个点周围建立邻域，并聚合邻域内的点来提取特征，但是忽略了邻域间的特征关系. LDGCNN[11]在 DGCNN 的基础上进行改进，将特征分层处理，在训练结束后，分类器重新进行训练，这样使得训练时间增加. 基于 DGCNN，RSMix[12]通过随机选择 2 个点云并在作刚性变换后进行混合来生成新的训练样本，但如果数据分布过于单一或数量不足，则可能会限制数据增强的效果.

近年来，注意力机制在点云深度学习模型上取得了很好的效果[13]，GAPNet[14]模型使用全局平均池化操作，将点云特征转化为固定大小向量，在各个数据集上取得了较好的分类性能. PointASNL[15]通过引入自适应下采样(adaptive sampling, AS)模块和局部-非局部(local-nonLocal, L-NL)模块，将全局信息与局部信息相结合，对点云数据进行分类和分割. FGP-Net[16]使用偏置注意力机制关注上下文差异，可在点云分类和分割任务中更好地捕捉细粒度属性. LFT-Net[17]采用自注意力机制来捕捉点云中的局部特征，并使用 Transformer 网络对点云进行处理，它的设计旨在提高点云分析任务的准确性和效率. DTNet[18]使用了双 Transformer 模块来处理点云的全局和局部特征，使得模型具有更好的可扩展性和表现力，能够更好地处理数量、大小和形状不同的点云数据. Point-BERT[19]利用 Transformer 的自注意机制捕捉点云数据中的全局上下文信息，使用掩码点建模来提高点云特征的鲁棒性和鉴别性，从而提高分类和分割性能.

本文设计了一种集多头点注意力与边卷积的点云分类和分割(multi-head point attention edge convolutional neural network, MHPAEC)模型，主要工作如下:

(1) 使用单头点注意力(single-head point attention, SHPA)模块提取注意力特征与邻域注意力特征，替换 DGCNN 中使用的空间转换网络，学习点云的旋转不变性.

(2) 使用多头机制聚合来自多个单头点注意力模块的不同特征，构建多头点注意力(multi-head point attention, MHPA)模块，捕获不同邻域点对中心点的重要程度.

(3) 设计加权金字塔池化(weighted pyramid pooling, WPP)模块，对特征进行不同维度的池化操作，然后动态地调整不同池化结果的权重，更好地提取点云数据的全局和局部特征.

(4) 将交叉熵损失函数与焦点损失函数[20]结合，提出联合损失函数，使得模型在解决部分样本

难分类和类别不平衡问题的情况下，能够更好地对点云数据进行分类和分割.

## 1 本文模型

DGCNN模型结构如图1所示. DGCNN首先将点云数据输入到空间转换网络(spatial transform network, STN)，使用 $K$ 最近邻($k$-nearest neighbor, $K$NN)算法构造动态有向图，然后将其乘以旋转矩阵学习点云的旋转不变性，输出图形数据. 边卷积(edge convolution, EdgeConv)层对图形数据进行卷积操作，在卷积层后使用最大池化层对卷积结果进行下采样. 最后，将经过多个动态图卷积层和最大池化层处理后的特征图进行展平，并输入到全连接层中进行分类. 图1中 $N$ 表示点的数量，$m$ 表示点云分类的类别数量，$s$ 表示部分分割类别标签的个数.

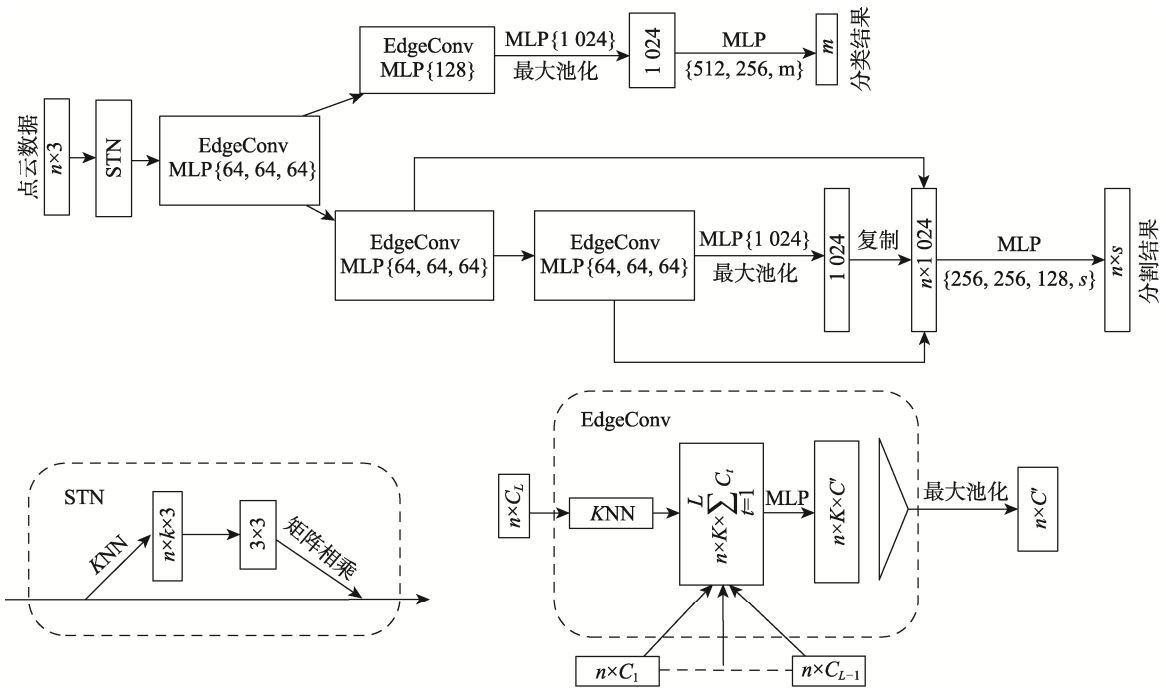



图 1 DGCNN 模型结构图

基于DGCNN模型，本文模型结构如图2所示. 模型的输入是 $N$ 个点的 $F$ 维点云，表示为 $\{\boldsymbol{p}_i \in \boldsymbol{R}^F, i=1,2,\cdots,N\}$. 其中，$\boldsymbol{p}_i$ 表示点云中的一个点，$i$ 为点云中点的索引；本文设 $F$=3，每个点包含三维坐标 $\boldsymbol{p}_i=(x_i,y_i,z_i)$. 首先将点云输送到MHPA模块进行2个分支的计算，注意力与边卷积分支用于计算点云的注意力，并输出 $N\times128$ 维特征；将其与原始点云数据拼接，送入6层边卷积模块中生成6个高维特征. 邻域注意力分支计算点云的邻域注意力，并经过最大池化操作生成 $N\times128$ 维的邻域特征；将2个分支的输出送入WPP模块进行特征融合，在点云分类网络中通过最大池化操作转变为 1×1 024 的全局特征；最后经过多层感知机(multilayer perceptron, MLP)模块，即MLP(512, 256, $m$)构成的分类器得到分类结果，其中 $m$ 表示点云分类的类别数量. 在点云分割网络中将特征融合后的特征与全局特征拼接，并利用MLP(256, 256, 128, $s$)计算每个点的分割分数，其中，$s$ 表示部分分割类别标签的个数.

### 1.1 STN

在DGCNN模型中，STN用来提取点云数据特征，以减少点云数据旋转变化对模型性能的影响，但是其使得整个模型大小变为原来的2倍. 因此本文考虑提出更轻量的多头注意力机制模块来对点云数据特征进行提取，MLP 是 MHPA 模块的核心组成部分.

对于 STN，可以用一个3×3的旋转矩阵 $\boldsymbol{R}$ 来表示，将原始点云数据与旋转矩阵相乘，可以让模型学习到点云的旋转不变性，如

$$\boldsymbol{P}_{\text{offset}}=\begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix}\times\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{1}$$

其中，$r_{ij}$ 是第 $i$ 行第 $j$ 列上旋转矩阵 $\boldsymbol{R}$ 的值. 另外，

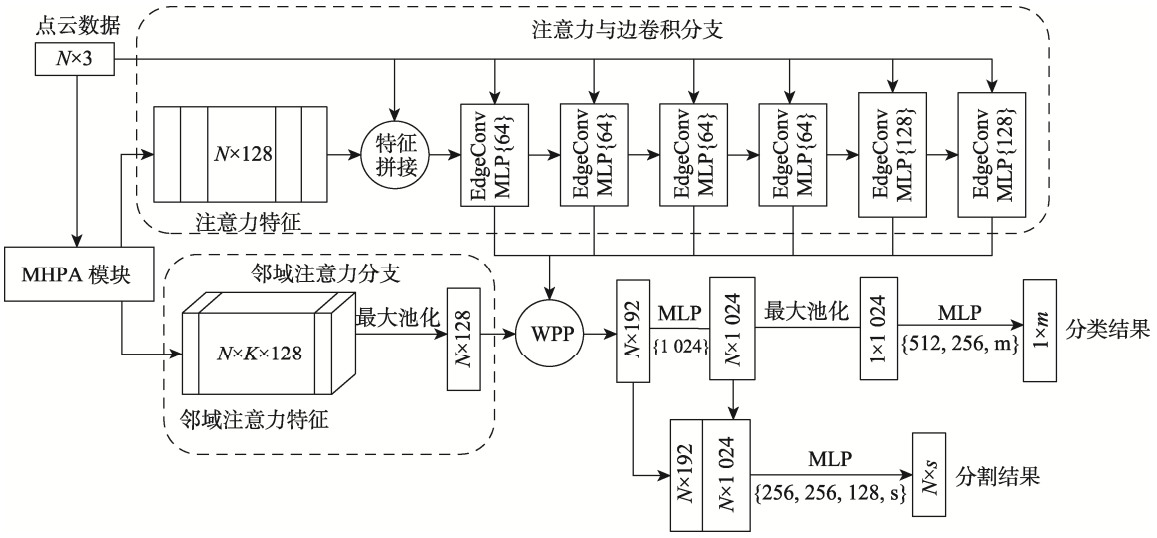



图 2　MHPAEC 模型结构图

$x_i$, $y_i$ 和 $z_i$ 是点云 $\boldsymbol{P}$ 中单点 $p_i$ 的坐标.

对于 MLP 模块，如果将 MLP 转换为矩阵形式，其隐藏层的特征向量表示为

$$\boldsymbol{P}_{\text{offset}}=\begin{pmatrix}x_1 & y_1 & z_1\\ x_2 & y_2 & z_2\\ \vdots & \vdots & \vdots\\ x_n & y_n & z_n\end{pmatrix}\times\begin{pmatrix}w_{11} & w_{12} & \cdots & w_{1C'}\\ w_{21} & w_{22} & \cdots & w_{2C'}\\ w_{31} & w_{32} & \cdots & w_{3C'}\end{pmatrix}+\begin{pmatrix}b_1 & b_2 & \cdots & b_{C'}\end{pmatrix} \tag{2}$$

其中，$w_{iC'}$ 和 $b_{C'}$ 表示 MLP 中的可训练参数；$C'$ 表示输出通道数.

对比式(1)(2)可以发现，STN 的操作相当于点云数据和一个特定的旋转矩阵进行了相乘计算；而 MLP 对于点云数据有多参性，不同参数通道的 MLP 可以使点云进行一定程度的旋转与平移，式(2)也对点云数据进行了和式(1)类似的计算操作，所以使用 MLP 替代 STN，理论上可以和 STN 一样提取出点云数据的特征信息.

## 1.2　MHPA 模块

### 1.2.1　SHPA 模块

MHPA 模块由多个 SHPA 模块组成，SHPA 模块结构如图 3 所示. 其中 tile 表示对张量维度进行复制; T 表示转置操作.

由图 3 可知，SHPA 模块并行计算注意力和邻域注意力. 首先，模型会对点云中的点以及邻域内的边进行编码，通过 MLP 来获取点和边更高维度的特征. 点 $i$ 更高维度的特征为

$$\boldsymbol{v}_i'=\text{MLP}(\boldsymbol{v}_i,\theta) \tag{3}$$

邻域注意力特征为

$$\boldsymbol{e}_{ij}'=\text{MLP}(\boldsymbol{e}_{ij},\theta) \tag{4}$$

其中，$\theta$ 表示一组可学习参数; $\boldsymbol{v}_i$ 表示第 $i$ 个点的特征；$\boldsymbol{e}_{ij}$ 表示点 $i$ 与点 $j$ 的边的特征.

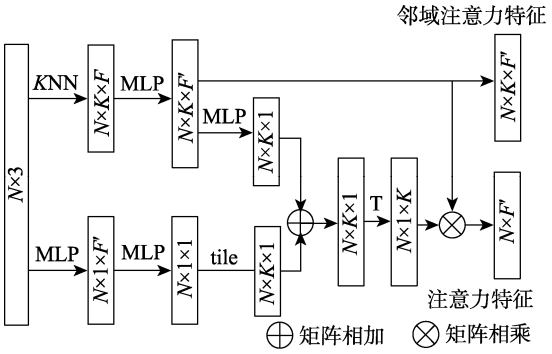



图 3　SHPA 模块结构图

将 $\boldsymbol{v}_i'$ 和 $\boldsymbol{e}_{ij}'$ 分别送入 MLP 即可获得自注意力系数和邻域注意力系数，对它们相加即得到注意力系数；然后使用非线性激活函数 LeakyReLU 进行激活，输出的注意力系数如

$$\boldsymbol{c}_{ij}=\text{LeakyReLU}\left(\text{MLP}(\boldsymbol{v}_i',\theta)+\text{MLP}(\boldsymbol{e}_{ij}',\theta)\right) \tag{5}$$

其中，$\text{MLP}(\boldsymbol{v}_i',\theta)$ 表示自注意力系数，$\text{MLP}(\boldsymbol{e}_{ij}',\theta)$ 表示邻域注意力系数.

使用 Softmax 函数对所有相邻点的系数进行归一化，使得邻域点的特征聚合到中心点，生成中心点的注意力系数

$$\boldsymbol{\alpha}_{ij}=\exp(\boldsymbol{c}_{ij})\Big/\sum_{k\in N_i}\exp(\boldsymbol{c}_{ik}) \tag{6}$$

其中，$k$ 表示邻域内点的索引；$N_i$ 表示邻域内所有点的集合.

最后使用计算所得的归一化系数与边特征相乘，获得每个点的注意力特征

$$\hat{\boldsymbol{v}}_i=\text{ReLU}\left(\sum_{j\in N_i}\boldsymbol{\alpha}_{ij}\boldsymbol{e}_{ij}'\right) \tag{7}$$

1.2.2 多头机制

多头机制支持并行计算，能够对不同特征子空间进行分离或者加权处理. 本文将多个 SHPA 模块进行拼接，提出 MHPA 模块，其模型结构如图 4 所示.

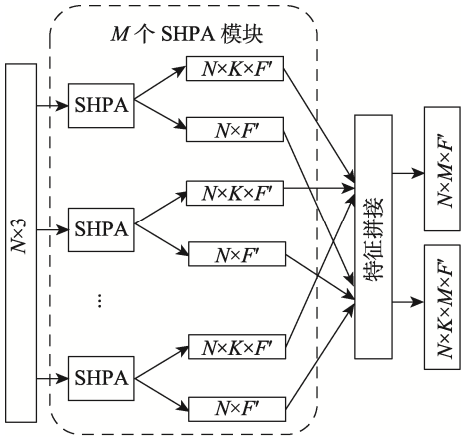



图 4 MHPA 模块结构图

点的注意力系数由点的自注意力系数和邻域注意力系数相加得到，故输出的注意力特征包含了点自身和点周围丰富的几何信息，对多个注意力特征进行聚合能够使得特征维度的层次变得丰富. 邻域注意力赋予邻域内 $K$ 个点不同的注意力系数，模型能够学习到邻域点对中心点的相关性，使用多头机制有助于提取更充分的点相关特征信息. 由图 4 可知，多头注意力特征和多头邻域特征由 $M$ 个独立的注意特征和邻域特征拼接获得. 多头注意力特征计算公式、多头邻域注意力特征计算公式分别为

$$\hat{\boldsymbol{v}}_i' = \mathop{\Vert}_{j=m}^{M} \hat{\boldsymbol{v}}_i^j \tag{8}$$

$$\hat{\boldsymbol{e}}_i' = \mathop{\Vert}_{j=m}^{M} \hat{\boldsymbol{e}}_i^j \tag{9}$$

其中，$\hat{\boldsymbol{v}}_i^j$ 和 $\hat{\boldsymbol{e}}_i^j$ 分别表示第 $i$ 个点第 $j$ 个头的注意力特征和邻域注意力特征；$\hat{\boldsymbol{v}}_i'$ 和 $\hat{\boldsymbol{e}}_i'$ 分别表示第 $i$ 个点多头注意力特征和多头邻域注意力特征；$M$ 表示多头机制中头的总数；$\Vert$ 表示对所有特征通道进行拼接.

### 1.3 WPP 模块

MHPA 模块在输出注意力特征之后会被送入边卷积模块中进行特征提取. 为有效地提高点云分类和分割精度，本文设置了 6 层边卷积模块，以提取更深层次的特征. 但是当卷积层数过多时，可能会导致一些低级几何特征保留在较浅层，在较深层中出现特征失真的情况，故在每一层边卷积之前，相应特征都会与原始的三维点云数据进行拼接，确保进行边卷积中的特征同时具有点云数据浅层与深层的特征信息，如图 2 所示.

对于注意力边卷积分支和邻域注意力分支输出的 7 层特征，本文提出动态 WPP 模块进行特征融合，如图 5 所示. 其中 $k$ 表示池化核的尺寸大小，$s$ 表示池化窗口的移动步幅，$C'$ 表示输出通道数.

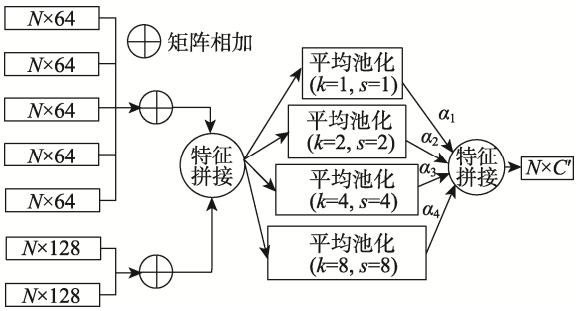



图 5 WPP 模块

为了减少特征中的噪声和信息冗余，对大小相同的特征进行相加操作，并对相加结果进行拼接，以提高模型的特征表达能力. 在点云分类和分割数据集中，有很多不同尺度的点云对象，将它们的拼接结果送入 4 个不同大小的平均池化窗口，对输入的特征图进行下采样操作，并生成多个尺度的特征图，从而对于不同尺度点云对象进行分类和分割. 考虑不同尺度的池化结果对分类和分割有不同程度的影响，将池化结果进行自适应加权，以提高模型对于多层特征的感知能力. 加权融合之后的结果为

$$F_{\text{fuse}} = \sum_{i=1}^{n} \alpha_i F_i \tag{10}$$

其中，$F_i$ 表示第 $i$ 层的池化结果；$\alpha_i$ 表示通过训练得到的第 $i$ 个池化结果对应的权重，池化结果对应的权重越大，表示对分类和分割任务越有利.

### 1.4 联合损失函数

在点云数据集中，存在难分类样本以及不同类别之间点云数据量差异较大的问题. 以往点云分类和分割模型大多数采用交叉熵计算损失度量模型预测和真实标签之间的差值. 但是如果数据集的标签分布不平衡，模型收敛速度会较慢，并且占比较小的类别得不到充分训练. 交叉熵损失函数公式为

$$L_{\text{CE}} = -\sum_{i=1}^{n} (y_i \ln p_i) \tag{11}$$

其中，$n$ 表示分类的类别数量；$y_i$ 表示第 $i$ 类对应的

真实标签；$p_i$ 表示对第 $i$ 类的预测值.

焦点损失函数是一种用于解决类别不平衡问题的损失函数，其主要思想是对占比较小的样本施加较大的权重，从而缓解类别不平衡的问题. 本文将交叉熵损失与焦点损失结合，设计了一种联合损失函数为

$$L_{\text{joint}} = \lambda L_{\text{CE}} + (1-\lambda)L_{\text{FL}} \tag{12}$$

其中，

$$L_{\text{FL}} = -\sum_{i=1}^{n} \frac{N-N_i}{N} \ln(p_i)(1-p_i)^{\gamma} \tag{13}$$

表示焦点损失函数；$N$ 表示点云数据中点的总数；$N_i$ 表示第 $i$ 类别点的个数；$\frac{N-N_i}{N}$ 表示当前类别权重，通过减少主要类样本数据的权重和增加次要类样本数据权重来平衡数据分布，其目的是控制当前类别在反向传播中的重要性，可以使模型缓解类别不均衡问题；$\gamma$ 系数用来控制难易样本对损失函数的贡献程度，当 $\gamma>0$ 时，难分类样本的权重会得到提高，从而使模型更加关注难分类的样本；$\lambda$ 参数用以调节 2 个损失的比重，当 $\lambda=1$ 时，该损失函数仅计算交叉熵损失.

## 2　实验与结果分析

### 2.1　数据集与实验参数

本文在 ModelNet40 数据集[21]上进行点云分类实验，ModelNet40 数据集包含 40 个类别分类，一共有 12 311 个三维模型，该数据集中的 9 843 个模型用于训练，2 468 个模型用于验证和测试. 分类实验数据集较小，故选择了在 Ubuntu 系统上进行单 GPU 训练. 对于部分分割任务，使用 ShapeNet 数据集[22]进行点云分割实验，该数据集总共包含 16 个对象类别、50 个部件类别以及 16 881 个三维模型，其中 12 137 个模型用于训练，1 870 个模型用于验证，2 874 个模型用于测试. 对于给定对象的三维模型，需要对类别进行判定并对对象的部件进行语义分割，例如，桌子包含桌面和桌腿等部件. 分割实验数据集较大，故在 Ubuntu 系统上进行多 GPU 训练. 这样可以并行计算，提高训练速度和效率. 本文模型实验环境与主要参数如表 1 所示.

**表 1　实验环境与主要参数**

| 实验环境与参数 | 分类 | 分割 |
|---|---|---|
| 操作系统 | Ubuntu 18.04 | Ubuntu 18.04 |
| GPU 数量 | 1 | 2 |
| GPU 型号 | NVIDIA Tesla K80 | NVIDIA Tesla K80 |
| 最大迭代轮次 | 250 | 180 |
| 批量大小 | 16 | 6 |
| 优化器 | Adam | Adam |
| 初始学习率 | 0.001 | 0.003 |
| 衰减步长 | 20 | 20 |
| 衰减率 | 0.7 | 0.5 |
| 动量系数 | 0.9 | 0.9 |
| 输入点数量 | 1 024 | 2 048 |

### 2.2　评价指标

点云分类实验采取平均精度(mean class accuracy, mAcc)与总体精度(overall accuracy, OA)作为评价指标，点云分割实验采用类别的平均交并比(mean intersection over union, mIoU)作为评价指标. 它们的计算公式分别为

$$\text{mAcc} = \frac{1}{s}\sum_{i=1}^{s}\frac{\text{TP}_i}{\text{TP}_i+\text{FN}_i}\times 100\% \tag{14}$$

$$\text{OA} = \frac{\text{TP}}{\text{TP+FN}}\times 100\% \tag{15}$$

$$\text{mIoU} = \frac{1}{s}\sum_{i=1}^{s}\frac{\sum_{j=1}^{p}\text{TP}_j}{\sum_{j=1}^{p}\text{TP}_j+\sum_{j=1}^{p}\text{FP}_j+\sum_{j=1}^{p}\text{FN}_j} \tag{16}$$

其中，$s$ 表示类别数量；$p$ 表示每个类别点的总数；TP 表示预测结果与真实标签都为真的点的数量；FP 表示预结果为真，真实标签为假的点的数量；TN 表示预测结果为假，真实标签为真的点的数量.

### 2.3　消融实验

本文模型中使用了 SHPA 模块、MHPA 模块、WPP 模块以及联合损失函数. 为了验证其有效性，在点云分类和分割上设计了消融实验，实验结果如表 2 所示.

**表 2　消融实验结果**

| 方法 | OA/% | mIoU/% |
|---|---|---|
| 基线模型 DGCNN[10] | 92.6 | 85.2 |
| +SHPA | 92.9(+0.3) | 85.4(+0.2) |
| +MHPA | 93.3(+0.7) | 85.7(+0.5) |
| +WPP | 93.9(+1.3) | 86.1(+0.9) |
| +Joint Loss | 94.1(+1.5) | 86.3(+1.1) |

由表 2 可知，当使用 SHPA 模块替换 DGCNN 中的 STN 时，网络能够计算注意力特征与邻域注意力特征，OA 与 mIoU 分别提升 0.3 个百分点和 0.2 个百分点；使用多头机制对 SHPA 模块进行聚合，MHPA 模块能够提取更为丰富的特征信息，模型

的分类精度增加了 0.7 个百分点，分割性能提升了 0.5 个百分点，证明了 MHPA 的有效性；在此基础上将特征送入 WPP 模块，先对输入特征进行不同尺度的池化操作，再赋予池化结果动态权重，提取更为全面的特征信息，使得 OA 与 mIoU 分别提升 1.3 个百分点和 0.9 个百分点；最后使用联合损失函数计算损失，解决样本分类不平衡问题和难分类问题，分类精度提升了 1.5 个百分点，分割精度提升了 1.1 个百分点. 综合上述实验结果，不难看出，本文所设计的 SHPA 模块、MHPA 模块、WPP 模块以及联合损失函数能够有效地提升模型的分类和分割精度.

## 2.4 超参数实验

### 2.4.1 头部数量

在 MHPAEC 中主要是通过控制头部数量来控制点云中 SHPA 模块的计算次数，头部数量越多，点的邻域特征与注意力特征也会越丰富，但同时模型的复杂度也会增大. 为了探究头部数量对模型分类效果的影响，本文在 ModelNet40, ShapeNet 数据集上进行对比实验，实验结果如图 6 所示. 在实验中，头部数量分别选择 2, 4, 8 和 16，当头部数量设置为 8 时，分类和分割性能最好；头部数量超过 8 时，分类和分割性能明显下降. 因此，本文将 MHPA 模块中的头部数量设置为 8.

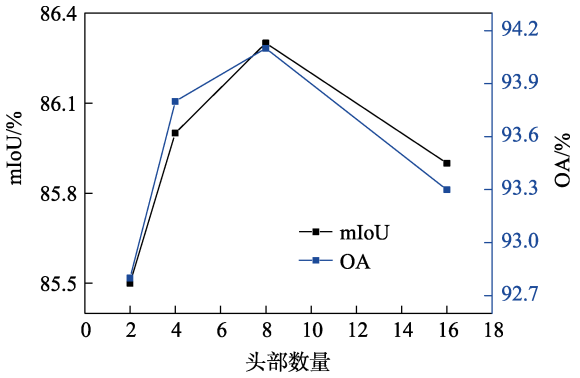



图 6 头部数量对模型精度影响

### 2.4.2 *K* 值

MHPA 模块和边卷积模块都使用了 *K*NN 算法，为了研究其中 *K* 的取值对分类和分割精度的影响，在 ModelNet40 和 ShapeNet 数据集上进行对比实验，实验结果如图 7 所示. 从实验结果中可以看出，当 *K*=20 时，分类和分割精度最高；当 *K* 值过小时，精度较低，边卷积模块无法充分提取边特征；当 *K* 值过大时，精度也会明显下降，这是由于此时欧几里得距离不能近似为测地距离，会破坏邻域的几何结构.

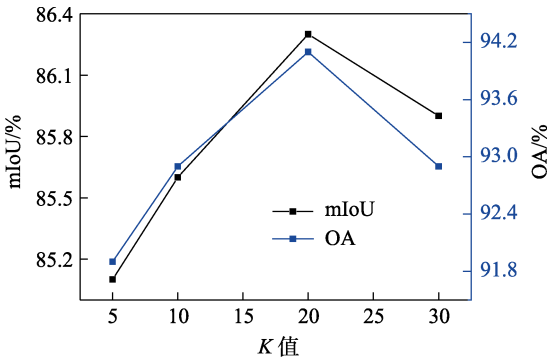



图 7 *K* 值对模型精度影响

综上所述，当选择头部数量为 8，*K*=20 时，MHPA 模块能够充分地提取点云数据的注意力特征与邻域注意力特征，边卷积模块也能够充分地提取到边特征，模型会达到最佳的分类和分割性能.

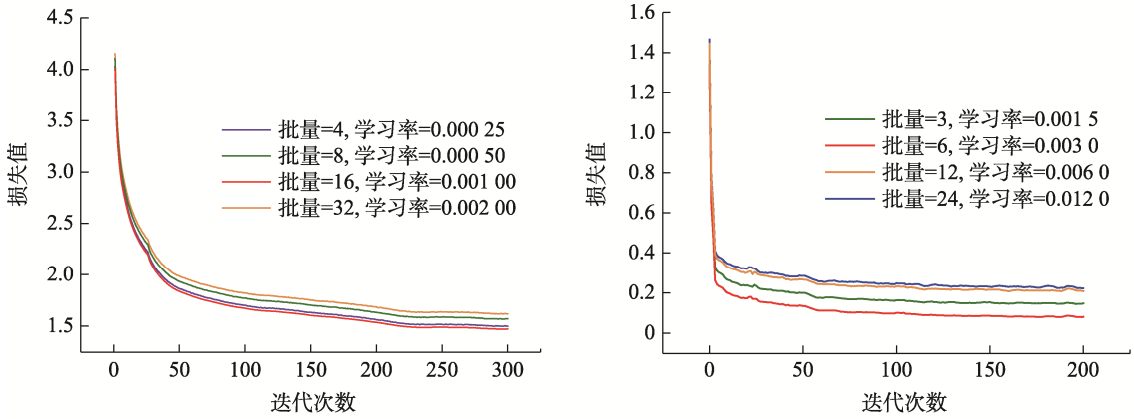
### 2.4.3 敏感性分析

本文实验中使用了一些参数，例如，批量大小、学习率和迭代次数，这些参数对于优化模型性能十分重要. 因此本文基于线性缩放规则[23]，对批量大小和学习率同时进行缩放，设计了一系列实验，以研究批量大小、学习率、迭代次数对分类和分割损失值的影响，实验结果如图 8 所示. 从图 8a 可以看出，当分类实验进行 250 次迭代，模型已经收敛；当批量大小和学习率分别设置为 16 和 0.001 时，模型的拟合效果最好，故将批量大小和学习率分别设置为 16 和 0.001. 由图 8b 可知，分割实验进行 180 次迭代时模型收敛，为避免过拟合，本文将分割实验的迭代次数设置为 180. 过大的批量大小和学习率可能会使模型在训练过程中学习到更多的噪声，从而降低拟合能力；而过小的批量大小和学习率会使模型无法充分地利用数据集的信息，导致收敛速度变慢、影响分类和分割性能. 因此，本文分割实验中设置批量大小为 6，学习率为 0.003.

## 2.5 点云分类实验

为验证本文的有效性，在 ModeNet40 数据集上进行实验，结果如表 3 所示. 为了进行不同维度的比较以及验证模型的分类性能，表 3 中列举了不同输入的算法，其中 Points 代表三维坐标输入，Normal 表示输入点云的法向量.

由表 3 可知，MHPAEC 在 ModelNet40 数据集上达到了 91.0%的 mAcc 与 94.1%的 OA，相较于原始点云数据作为输入的模型，MHPAEC 比 PointNet 和 PointNet++在 OA 上分别提升 4.9 个百分点和 3.4




a. 不同批量大小和学习率对分类损失影响　　b. 不同批量大小和学习率对分割损失影响

图 8　不同批量大小和学习率对损失值影响

**表 3　ModelNet40 分类实验结果**

| 模型 | 输入 | mAcc/% | OA/% |
|---|---|---|---|
| PointNet[6] | Points | 86.2 | 89.2 |
| PointNet++[7] | Points | 87.5 | 90.7 |
| 3D-GCN[8] | Points | | 92.1 |
| DGCNN[10] | Points | 90.2 | 92.6 |
| LDGCNN[11] | Points | 90.3 | 92.9 |
| RSMix[12] | Points | | 93.5 |
| GAPNet[14] | Points | 89.7 | 92.4 |
| PointASNL[15] | Points+Normal | | 93.2 |
| LFT-Net[17] | Points+Normal | 89.7 | 93.2 |
| DTNet[18] | Points | 90.4 | 92.9 |
| Point-BERT[19] | Points | | 93.8 |
| MHPAEC | Points | **91.0** | **94.1** |

注. 粗体表示最优值.

个百分点. 这主要是因为本文模型使用的边卷积模块采用 *K*NN 算法进行局部特征学习，保留了点云局部的边特征信息.

MHPA 模块计算点云数据的注意力特征，赋予邻域内点不同注意力系数，与同样使用图卷积处理点云数据的 3D-GCN，LDGCNN 和 RSMix 相比，OA 分别高 2 个百分点、1.2 个百分点和 0.6 个百分点.

DTNet 和 Point-BERT 都使用了 Transformer 结构对点云数据进行建模，捕捉点云中的局部和全局特征，但是对于注意力模块中输出的多个特征向量没有进行更好的特征融合. 本文的 WPP 模块能够对特征向量进行不同尺度的池化操作并赋予权重，提高了模型对不同大小对象的分类能力，故与 DTNet 和 Point-BERT 相比，OA 分别提升 1.2 个百分点和 0.3 个百分点.

与使用点云数据和法向量作为网络输入的 PointASNL 和 LFT-Net 相比，MHPAEC 在 OA 上都提高 0.9 个百分点. 这是因为法向量虽然提供了丰富的几何信息，但是同时会增加噪声. MHPAEC 只使用点云数据作为输入，有助于避免信息冗余，使得模型更加关注点云数据中的核心特征.

图 9 所示为 MHPAEC 模型在 ModelNet40 数据集上实验的精度曲线和损失曲线. 由图 9 可知，随着迭代次数的增加，精度不断提升，训练损失不断下降，参数不断优化，网络学习到的点云几何特征也不断增多. 当训练至 40 000 步时，分类精度和损失值处于平衡状态，网络呈现收敛趋势，模型达到最好的训练效果.

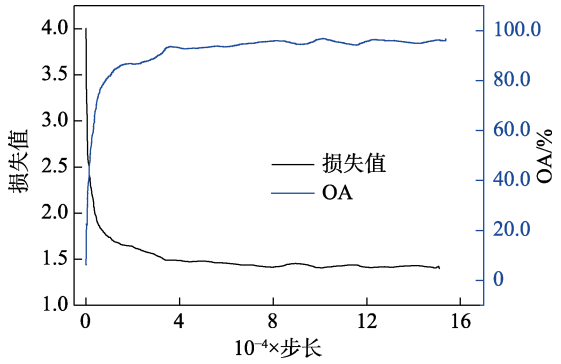



图 9　MHPAEC 在 ModelNet40 数据集上的实验结果

为了进一步直观地对比模型对不同类别的分类性能，将 MHPAEC 与 PointNet++，DGCNN 和 GAPNet 在 ModelNet40 数据集上进行 40 种类别的精度比较，如图 10 所示. 从图 10 中可以发现，airplane, bowl, car, guitar, keyboard, laptop, monitor, person, sofa 和 toilet 的 OA 均达到 100.0%. 这是因为动态图卷积结构能够有效地捕捉点云数据的结构信息，对于具有明显结构模式的类别(如 airplane 和 car 等)，有助于提高分类精度；使用 MHPA 模块可以更好地捕捉到关键区域的特征，提升 keyboard

和 monitor 这些有明显区域性质的类别的分类性能; bowl 和 guitar 等类别具有不同尺度, WPP 模块可以通过多个尺度的特征表示来捕捉不同层次的语义信息, 进而得到更好的分类结果; 联合损失函数中使用了焦点损失函数, 可以对错误分类的样本给予更大的惩罚, 有助于模型更加关注难分类的样本, 提高 person 和 sofa 这类难样本的分类精度.

### 2.6 点云分割实验

为了验证本文模型在部分分割任务的有效性, 在 ShapeNet 数据集上进行实验, 并与性能优秀的点云分割算法进行对比, 实验结果如表 4 所示.

从表 4 可以看出, MHPAEC 在 ShapeNet 数据集上的 mIoU 达到了 86.3%, 性能优于大多数主流点云分割模型. 模型对 pack, car, lamp, motorcycle 和 rocket 等类别进行分割时效果较好, 分别达到了最高的 86.7%, 79.9%, 85.8%, 75.7%和 63.9%. 这是因为这些 CAD 模型都比较复杂, 算法在 MHPA 模块中对点的邻域特征与自身特征进行学习, 然后进一步使用边卷积进行特征提取, 充分地学习到复杂模型更深层次的语义信息, 最后使用 WPP 模块进行特征融合, 在大量点云数据中聚合了注意力特征与边特征.

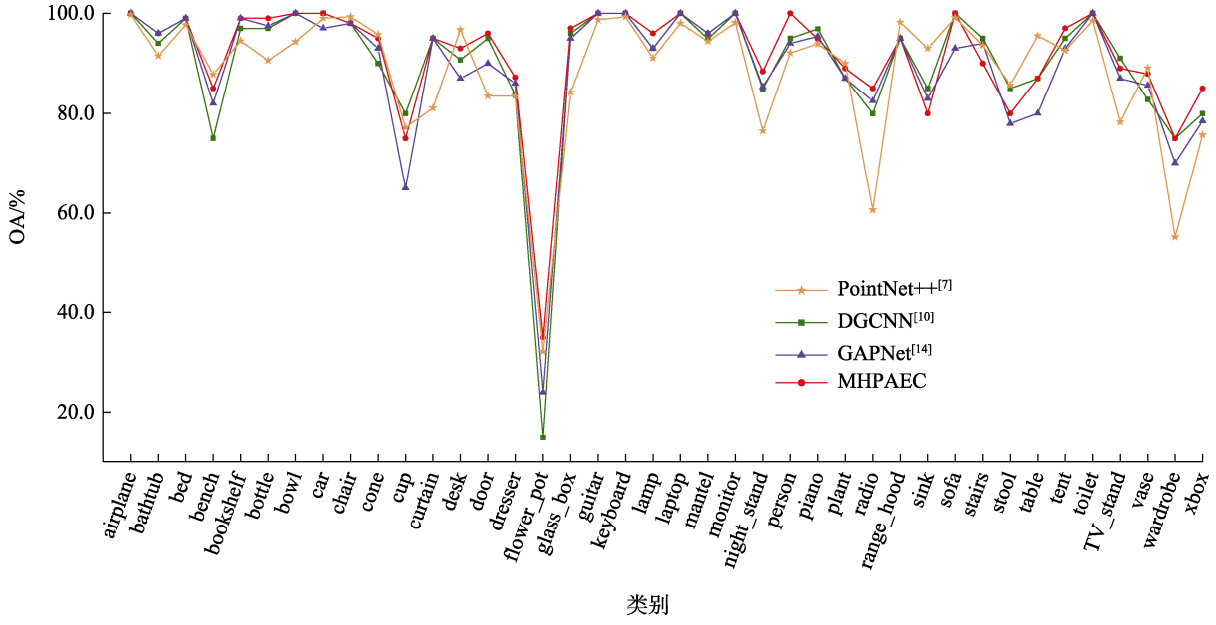



图 10 不同模型在 ModelNet40 上对各类别分类精度对比

**表 4 不同模型在 ShapeNet 数据集上部分分割实验结果**

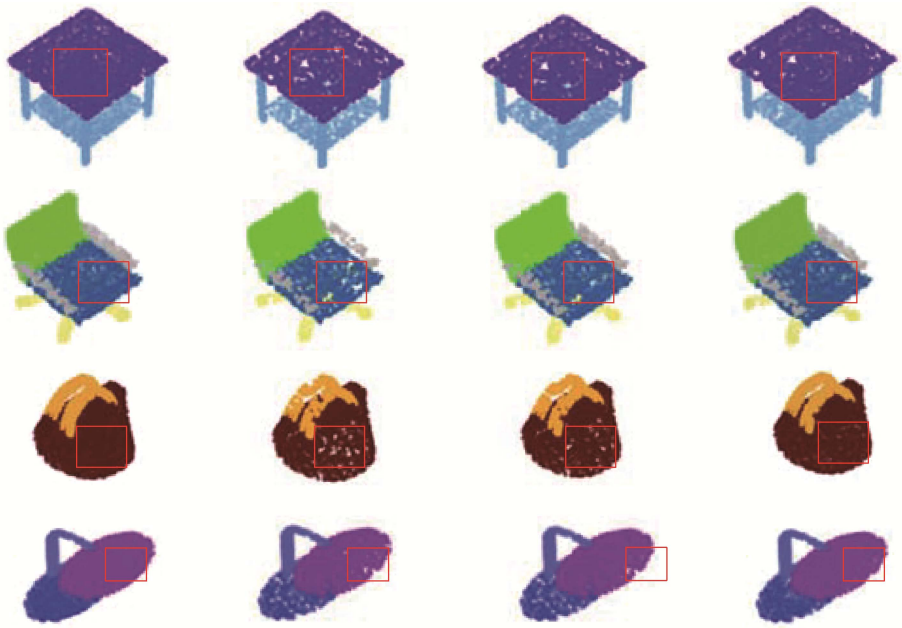
| 方法 | mIoU/% | IoU/% | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | air-plane | pack | cap | car | chair | head-set | guitar | knife | lamp | com-puter | mo-tor-cycle | bath-tub | pistol | rocket | skate board | desk |
| PointNet[6] | 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 | 80.6 |
| PointNet++[7] | 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[8] | 85.1 | 83.1 | 86.6 | 77.5 | 78.4 | 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[10] | 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 |
| LDGCNN[11] | 85.1 | 84.0 | 83.0 | 84.9 | 78.4 | 90.6 | 74.4 | 91.0 | **88.1** | 83.4 | 95.8 | 67.4 | 94.9 | 82.3 | 59.2 | 76.0 | 81.9 |
| PointASNL[15] | 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 |
| GAPNet[14] | 84.7 | 84.2 | 84.1 | **88.8** | 78.1 | 90.7 | 70.1 | 91.0 | 87.3 | 83.1 | **96.2** | 65.9 | 95.0 | 81.7 | 60.7 | 74.9 | 80.8 |
| DTNet[18] | 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** |
| Point-BERT[19] | 85.6 | **84.3** | 84.8 | 88.0 | 79.8 | 91.0 | **81.7** | **91.6** | 87.9 | 85.2 | 95.6 | 75.6 | 94.7 | **84.3** | 63.4 | 76.3 | 81.5 |
| MHPAEC | **86.3** | 83.6 | **86.7** | 82.8 | **79.9** | 91.3 | 76.6 | 91.2 | 86.5 | **85.8** | 94.7 | **75.7** | 94.3 | 82.9 | **63.9** | 73.6 | 82.4 |

注. 粗体表示最优值.

采用不同方法对部分分割结果可视化如图 11 所示，其中对于不同模型分割结果的不同之处，使用红色矩形框进行了标识. 可以看出，desk 和 chair 这 2 个模型部件的点云数据存在重叠；对于区域内不同类别的点云数据，MHPA 模块进行特征提取时，学习到了不同类别邻域点对中心点的重要程度，达到了较好的分割效果. 当对 pack 和 lamp 进行分割时，MHPAEC 在边界处实现了较好的分割性能，这是因为多层边卷积特征融合模块同时提取到了点云邻域的浅层特征与深层特征，所以 MHPAEC 分割结果更接近真值标签的分割结果.

### 2.7　模型复杂度实验

为了探究 MHPAEC 的实用性，对模型复杂度进行对比实验，使用参数量作为评价指标. 实验环境与参数设置均与第 2.1 节相同，在 ModelNet40 数据集上的实验结果如表 5 所示.



a. 真值图　　b. PointNet[6]　　c. DGCNN[10]　　d. MHPAEC

图 11　不同方法在 ShapeNet 数据集上分割结果可视化对比图

**表 5　模型性能对比**

| 模型 | 参数量 | OA/% |
|---|---|---|
| PointNet[6] | 3.48M | 89.2 |
| PointNet++[7] | 1.48M | 90.7 |
| DGCNN[10] | 1.84M | 92.6 |
| LDGCNN[11] | **1.08M** | 92.9 |
| MHPAEC | 2.02M | **94.1** |

注. 粗体表示最优值.

由表 5 可知，MHPAEC 的参数量为 2.02M，OA 为 94.1%. 与原始模型 DGCNN 和其他模型对比分析可知，MHPAEC 能够在以较小的模型参数量增加下有效地提高分类精度，表明本文模型有较好的实用性.

## 3　结　语

本文首先设计了一种集 MHPA 与边卷积的三维点云分类和分割模型. SHPA 模块能够分别提取注意力特征与邻域注意力特征，替代 STN 学习点云数据的旋转不变性. 使用多头机制将多个 SHPA 模块进行拼接，能够更好地提取点云数据特征. 其次，本文提出了 WPP 模块，对多维度特征进行融合，在提高模型对不同尺度特征感知能力的情况下，对池化结果进行动态加权处理，提高了模型的分类和分割性能. 最后，针对分类和分割任务中存在的样本不平衡和识别难问题，将交叉熵损失和焦点损失相结合，设计了联合损失函数，使样本得到充分训练. 在 ModelNet40 点云分类数据集和 ShapeNet 点云分割数据集上进行实验，分别达到了 94.1%的 OA 和 86.3%的 mIoU.

实验结果表明，MHPAEC 具有最好的分类性能和分割性能，然而对于大规模室内外场景的分类和分割效果还不是很理想，复杂的环境会对模型的分类和分割性能产生影响，存在较大的改进空间. 下一步的研究准备将点云数据与法向量或者多视图数据相结合，通过引入多维数据，学习模

型更复杂的几何细节，提升大规模点云数据的分类和分割精度；同时使用更轻量的特征提取模块，减少模型所需的训练时间和模型复杂度.

## 参考文献(References):


[1] Li Y, Ma L F, Zhong Z L, *et al*. Deep learning for LiDAR point clouds in autonomous driving: a review[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(8): 3412-3432

[2] Kim J, Chung D, Kim Y, *et al*. Deep learning-based 3D reconstruction of scaffolds using a robot dog[J]. Automation in Construction, 2022, 134: 104092

[3] Liu W P, Sun J, Li W Y, *et al*. Deep learning on point clouds and its application: a survey[J]. Sensors, 2019, 19(19): 4188

[4] Maturana D, Scherer S. VoxNet: a 3D convolutional neural network for real-time object recognition[C] //Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Los Alamitos: IEEE Computer Society Press, 2015: 922-928

[5] 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. Los Alamitos: IEEE Computer Society Press, 2015: 945-953

[6] Qi C R, Su H, Mo K C, *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. Los Alamitos: IEEE Computer Society Press, 2017: 77-85

[7] Qi C R, Yi L, Su H, *et al*. PointNet++: deep hierarchical feature learning on point sets in a metric space[C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2017: 5105- 5114

[8] Lin Z H, Huang S Y, Wang Y C 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. Los Alamitos: IEEE Computer Society Press, 2020: 1797-1806

[9] Bai Jing, Xu Haojun. MSP-Net: multi-scale point cloud classification network[J]. Journal of Computer-Aided Design & Computer Graphics, 2019, 31(11): 1917-1924(in Chinese)
(白静，徐浩钧. MSP-Net: 多尺度点云分类网络[J]. 计算机辅助设计与图形学学报, 2019, 31(11): 1917-1924)

[10] 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): Article No.146

[11] Zhang K G, Hao M, Wang J, *et al*. Linked dynamic graph CNN: learning through point cloud by linking hierarchical features[C] //Proceedings of the 27th International Conference on Mechatronics and Machine Vision in Practice. Los Alamitos: IEEE Computer Society Press, 2021: 7-12

[12] Lee D, Lee J, Lee J, *et al*. Regularization strategy for point cloud via rigidly mixed sample[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2021: 15895-15904

[13] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need[C] //Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2017: 6000-6010

[14] Chen C, Fragonara L Z, Tsourdos A. GAPointNet: graph attention based point neural network for exploiting local feature of point cloud[J]. Neurocomputing, 2021, 438: 122-132

[15] 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. Los Alamitos: IEEE Computer Society Press, 2020: 5588-5597

[16] Bai Jing, Shao Huihui, Ji Hui, *et al*. An end-to-end fine-grained classification network for 3D point clouds[J]. Journal of Computer-Aided Design & Computer Graphics, 2023, 35(1): 128-134(in Chinese)
(白静，邵会会，姬卉，等. 面向三维点云的端到端细粒度分类网络[J]. 计算机辅助设计与图形学学报，2023，35(1): 128-134)

[17] 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, 2023, 24(2): 2158-2168

[18] Han X F, Jin Y F, Cheng H X, *et al*. Dual transformer for point cloud analysis[J]. IEEE Transactions on Multimedia, 2023, 25: 5638-5648

[19] Yu X M, Tang L L, Rao Y M, *et al*. Point-BERT: pre-training 3D point cloud transformers with masked point modeling[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2022: 19291-19300

[20] Lin T Y, Goyal P, Girshick R, *et al*. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2020, 42(2): 318-327

[21] 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. Los Alamitos: IEEE Computer Society Press, 2015: 1912-1920

[22] 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): Article No.210

[23] Dean J, Corrado G S, Monga R, *et al*. Large scale distributed deep networks[C] //Proceedings of the 25th International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc, 2012: 1223-1231