北京交通大学

硕士专业学位论文

基于注意力机制的3D点云分类和语义分割算法研究

Research on 3D Point Cloud Classification and Semantic Segmentation Algorithm Based on Attention Mechanism

作者：宋文赞

导师：白慧慧

北京交通大学

2022年6月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：宋文赞　　导师签名：

签字日期：2022年6月1日　　签字日期：2022年6月1日

学校代码：10004 密级：公开

北京交通大学

硕士专业学位论文

基于注意力机制的 3D 点云分类和语义分割算法研究

Research on 3D Point Cloud Classification and Semantic Segmentation Algorithm Based on Attention Mechanism

作者姓名：宋文赞 学 号：20125226

导师姓名：白慧慧 职 称：教授

专业领域：人工智能 学位级别：硕士

北京交通大学

2022 年 6 月

致谢

时间悄无声息地流逝，已经来到了校园生活的尾声，还记得最初踏进校园的样子，每个人都意气风发，憧憬着北交大未来的日子，校园中充斥着大家的欢声笑语，科研生活令人向往而又恐惧；如今两年时间快要过去了，在这段日子中，学习、生活和工作上都有着或多或少的喜与悲，但更多的是自身的成长，每一步都是人生的磨练，亦是每个人需要珍惜的东西，在这里，我要感谢路过我两年研究生生涯中的每个人，在我求学路上的每一句话、每一个举动以及每一个微笑都让我受益匪浅。

首先最需要感谢的是我的导师，白慧慧老师，两年时间内在学习、生活中帮助了我许多，是我真正走进校园的领路人。研究生生活不同于本科，很多东西不再局限于课堂，而是需要自己不断地积累、挖掘。所以，第一步的方向要对，就是基本功的学习，还记得老师当初推荐学习的深度学习和机器学习的相关课程，在后面的科研生活中无疑起到了快速入门的作用，在一次次组会的讨论中不断加深了自己的理解；另外，要注意到研究不仅仅是为了毕业而研究，还要有自己的兴趣，走进三维领域的研究方向，直到现在我还感觉处处充满着惊喜；科研需要碰撞的火花，每次组会上老师的细心询问以及意见都会让我学习到很多，并且项目中共同学习的氛围也让我感受到了亦师亦友，再次感谢白慧慧老师这两年的悉心指导。

此外，感谢北交大给我提供了一个优秀的学习生活环境，无论是宿舍阿姨、门卫保安还是食堂的工作人员，都在勤勤恳恳地工作，一心一意地默默守护着交大学子；图书馆的各种书籍、文献资料以及学校举办的各种讲座为我的学习提供了莫大的帮助。感谢同届的张健翔、庞子微、孙夕越、黄淼、罗晓慧等同学，在日常的学习生活上对我的帮助，感谢每一次的实验和论文上的交流；感谢秦佳、刘曼、李锋、王召、魏喆、董想、许卓凡、何一帆等师兄师姐对我在模型画图、模型优化各方面的细心指导，以及在找工作时提供的一些机会；感谢宋召、孙志超等同学对我生活和工作上的帮助。感谢各位，时光不但没有冲淡反而坚定了我的初心，衷心祝愿大家鹏程万里。

最后，我要感谢我的家人，尤其是我的父母、姐姐，对我学习上的支持，是你们的刻苦工作才有了如今的我，在今后的日子里，我会继续努力，不会辜负大家的期望。

摘要

点云技术在最近几年越来越受到人们的重视，在计算机视觉、机器人技术以及自动驾驶等领域有着广泛的应用。深度学习作为人工智能领域的一项重要技术，已成功应用于解决各种二维视觉问题。近年来，很多研究工作开始致力于利用深度学习的技术来处理三维点云，主要涉及到点云的分类、分割、目标检测、三维重建、点云补全等应用。其中，点云的分类和分割作为点云技术研究的重要基础，可分别应用于城市的规划建设和自动驾驶等方面，但是目前的点云分类和分割因为缺乏点之间的交互，造成了信息获取的损失。由于注意力机制可以有效获取输入之间的长依赖关系，因此非常适合点云信息的交互处理。因此本文将注意力机制引入点云分类和分割任务中，主要研究工作如下：

（1）结合局部特征提取、自注意力机制和多尺度特征融合技术，提出了基于注意力机制的多尺度点云分类神经网络模型。该网络每一层的输入点数为上一层点数的一半，利用最远点采样（Farthest Point Sampling，FPS）来尽量的保持每一尺度下的点云形状；针对每一尺度，利用K最近邻（K Nearest Neighbor，KNN）算法来获取每个点的邻近点，实现局部特征的提取，从而丰富每个中心点的特征；之后在每一尺度下利用自注意力机制进行所有点之间的特征交互，实现全局特征的提取；最后设计了一种多尺度的融合模块，自动学习每个尺度下的权重，实现不同尺度下的特征融合，作为最后的特征。在公共数据集上进行实验，结果表明，本文提出的网络有效提高了点云分类的准确性。

（2）在点云分类模型的基础上，本文进一步改进注意力机制，结合上采样、多尺度融合等理论，提出了基于注意力机制的多尺度点云语义分割神经网络模型。具体的，本文从位置信息和语义信息两方面出发，分别提出了基于合成注意力机制的位置信息提取模块和基于自注意力机制的语义信息提取模块。不同于点云的分类，语义分割需要关注每个点的特征。因此，本文在网络的上采样过程中利用插值法来逐步恢复点的个数，并利用跳跃连接来丰富每个点的特征。最后通过通道注意力机制利用每个点不同维度的自适应性来得到最后的特征。在公共数据集上进行实验，结果表明，本文提出的网络有效提高了点云语义分割的准确性。

关键词： 点云分类；多尺度；注意力机制；点云语义分割

ABSTRACT

Point cloud technology has received more and more attention in recent years, and it has a wide range of applications in computer vision, robotics, and autonomous driving. As an important technology in the field of artificial intelligence, deep learning has been successfully applied to solve various two-dimensional vision problems. In recent years, many research works have begun to use deep learning technology to process 3D point clouds, mainly involving point cloud classification, segmentation, target detection, 3D reconstruction, point cloud completion and other applications. Among them, the classification and segmentation of point cloud, as an important basis for the research of point cloud technology, can be respectively applied to urban planning and construction and automatic driving, but the current point cloud classification and segmentation lack of interaction between points, resulting in the loss of information acquisition. Since the attention mechanism can effectively capture long dependencies between inputs, it is very suitable for interactive processing of point cloud information. Therefore, this thesis introduces the attention mechanism into the point cloud classification and segmentation tasks. The main research work is as follows:

(1) Combined with local feature extraction, self-attention mechanism and multi-scale feature fusion technology, a multi-scale point cloud classification neural network model based on attention mechanism is proposed. The number of input points in each layer of the network is half of the points in the previous layer, and the farthest point sampling (Farthest Point Sampling, FPS) is used to keep the point cloud shape at each scale as much as possible; for each scale, K nearest neighbor (K Nearest Neighbor, KNN) algorithm is used to obtain the adjacent points of each point to realize the extraction of local features, thereby enriching the features of each center point; then use the self-attention mechanism to perform feature interaction between all points at each scale to achieve global feature extraction; finally, a multi-scale fusion module is designed to automatically learn the weights at each scale to achieve feature fusion at different scales as the final feature. Experiments are conducted on public datasets, and the results show that the proposed network effectively improves the accuracy of point cloud classification.

(2) On the basis of the point cloud classification model, this thesis further improves the attention mechanism and proposes a multi-scale point cloud semantic segmentation neural network model based on the attention mechanism by combining the theories of

upsampling and multi-scale fusion. Specifically, the thesis proposes a position information extraction module based on synthetic attention mechanism and a semantic information extraction module based on self-attention mechanism from two aspects of position information and semantic information. Different from point cloud classification, semantic segmentation needs to pay attention to the features of each point. Therefore, the thesis uses interpolation to gradually recover the number of points in the upsampling process of the network, and uses skip connections to enrich the features of each point. Finally, the final features are obtained by utilizing the adaptability of different dimensions of each point through the channel attention mechanism. Experiments are conducted on public datasets, and the results show that the proposed network effectively improves the accuracy of point cloud semantic segmentation.

KEYWORDS： Point Cloud Classification; Multi-scale; Attention Mechanism; Point Cloud Semantic Segmentation

序言

由于点云扫描相关设备的快速发展以及点云在自动驾驶、机器人技术、AR 和 VR 等领域中的广泛应用，点云的可用性在不断的提高。因此，需要一种快速高效的点云处理算法来提高分类和分割的视觉感知能力。点云的分类识别可以用于三维模型的检索中，根据输入点云的类别进行相似数据集的查询；在铁路轨道中，可以用于对地面上的每一类物体进行分类；在地面植被中，可以按照植被到地面点的高度，将其划分为低矮植被、中等高度植被或者高植被。点云的语义分割同样可以用于在铁路场景检测中，识别铁路上的侵入异物，不但准确直观，还不易受到天气及环境的影响；并且可以应用在自动驾驶中，辨别出了行人、车辆、树木、建筑等物体；另外，采用 3D 点云语义分割技术，人们可以通过 AR 眼镜设备去感受虚拟的 3D 场景；最后，对真实场景进行语义分割可以利用机器人实现目标的抓取。点云分类和分割作为一个比较基础的工作，在很多应用上有着一定的作用和影响，成为了研究人员重点的研究方向。

本文针对点云分类和语义分割难点，提出了基于注意力机制的多尺度点云分类和点云语义分割神经网络模型，具体包括：为了融合局部特征和全局特征，提出了局部特征提取模块以及基于注意力机制的全局特征提取模块，进一步提出了多尺度融合模块来结合不同尺度下的特征来进行分类；基于合成注意力机制、自注意力机制分别获取位置信息和语义信息，通过上采样插值模块以及跳跃连接来恢复点的数量进行语义分割。实验论证，所提出的算法均具有科学可行性和有效性。

本论文获得国家自然科学基金项目（No.619720203）的支持。

目录

1 绪论

1.1 研究背景及意义

点云是一组具有三维坐标的点，是一种常用的三维表达形式。点云可以通过激光雷达、深度相机、双目相机等设备进行生成。以激光雷达为例，该设备利用激光测距的原理，通过发射单元发射激光束，遇到障碍物进行反射，然后利用接收单元进行捕获，之后计算发射和接收的时间来确定障碍物的距离，从而快速构建出被测目标的三维模型。其中除了可以获得目标的坐标信息外，通过对激光回波的处理，可以进一步获取到被测目标的颜色以及激光反射强度信息，其中强度信息主要与目标的表面材质、粗糙度有关。由于点云能保持三维空间中的基本几何信息，而无需进行离散化，所以在很多与场景理解有关的应用中，例如自动驾驶和机器人控制，其都是优先选择的表示方法。由于点云数目庞大，可以绘制出物体的三维轮廓，目前点云已经发展成为遥感、计算机视觉、生物医疗等多个领域常见数据源之一[1]。

点云的分类识别可用在地面植被的分类，按照点到地面的高度，可以将植被划分为“低矮植被、中等高度植被、高植被”，此处高度阈值可分别设置为 0.1-1 米，1-3 米，大于 3 米。目前，植物资源的常规普查大多是以手工方式进行，其结果具有较高的测量准确率，但会耗费大量的人力和物力，并且难以适应植物资源迅速更新的特点，而通过机载雷达获取点云，则具有客观和高效的特点，并且能够在较短的时间里获得大范围的土地信息，使其具有较好的分类和快速更新的能力；另外，可以通过机载雷达获取的点云对整个城市分类，其中主要包括建筑物、树木、草地、地面等 4 类物体，该做法对城市的规划有着巨大的意义，实现了对城区建设分区块的快速创建和展示，并能应用于环境调查、土地利用、城市规划以及城市数字化建设等方面[2]。

点云的语义分割可以用于自动驾驶领域，自动驾驶车辆在行驶过程中需要实时地检测周围的环境，主要包括行人和其它车辆。但是在安全行驶方面，除了行人和周围车辆外，自动驾驶车辆的驾驶行为会在不同程度上受着周围的建筑物、绿化植被和未知的障碍物的影响，而点云语义分割技术的实现目标，则是帮助自动驾驶车辆更准确地了解道路周边的环境状况，更智能地应对突发情况，来确保最大程度化的安全驾驶；另外，利用点云语义分割技术，在 AR 领域中，人们可以通过穿戴 AR 眼镜设备体验到各种虚拟的 3D 生活场景，也可以在设备上添加虚拟的信息内容，从而更加智能化地处理各种需求。

点云的分类是从全局出发，对所有点表示的物体进行整体的识别，而点云的语义分割则是从局部出发，细化到了每一个点上，然后根据点所处的位置环境对该点进行分类，可以认为点云的语义分割任务是对每个点进行分类，所以本文在实现上，首先研究点云分类任务，利用自注意力机制进行全局范围内点的特征交互，然后将其引入到点云语义分割任务中的局部邻域，通过获取丰富的局部特征来实现每个点的分类。

目前点云应用的领域十分广泛，点云分类和语义分割作为一个比较基础的工作，在很多应用上有着一定的作用和影响，但是基于原始点云本身的信息提取方法没有充分发掘出点云数据内部之间的自动化潜力，因此迫切需要研究出新的点云信息提取的相关理论和方法。所以，继续提高点云的分类和语义分割的准确度是具有重要的研究意义和价值的。

1.2 国内外研究现状

针对点云分类和语义分割的发展过程，本小节将从基于传统方法和基于深度学习方法两个方向分别介绍点云分类和语义分割的研究现状。

1.2.1 点云分类算法研究

传统的点云分类算法主要是基于局部邻域内所有点的几何结构来开发手工制作的点描述符，然后选择合适的分类器进行点云标签的预测。其中常见的分类器包括支持向量机（Support Vector Machines，SVM）[3]、随机森林（Random Forest，RF）[4]等。但是，传统的分类由于手工设计规则的主观性太强，当应用场景不同时，其泛化能力一般不理想，所以只能处理一些简单的分类任务。

近些年来基于深度学习的点云分类方法取得了一系列成果，因为深度学习能够实现自动化提取点云的特征，从而有效避免了传统算法中操作人员的主观因素的影响。如图 1-1 所示，基于深度学习的点云分类算法根据网络模型输入的点云数据类型以及点云处理方法的不同，现有的点云形状分类的相关方法主要分为基于多视图的方法、基于体积的方法、基于 MLP 的方法、基于 CNN 的方法、基于图卷积的方法和基于注意力机制的方法。

（1）基于多视图的方法首先将 3D 点云从多个视角投影到多个视图中并提取相关的视图特征，然后融合这些特征用于精确的形状分类。其中如何将多个视图特征整合起来进行全局表示是这类方法面临的主要挑战。MVCNN[5]是基于多视图的开创性方法，它在点云周围放置多个虚拟摄影机来生成多个视图的 2D 图片，然后

每个视图都通过卷积神经网络生成图像描述符，最后利用 Max-Pooling 操作提取多个视图的特征作为全局描述符。但是，Max-Pooling 操作仅保留了来自某一视图的最大特征值，明显会造成信息的丢失。MHBN[6]基于双线性池化与多项式核之间的关系，通过双线性池对局部卷积特征进行聚合，得到了一种有效的三维对象表示。Yang 等人[7]提出了一种关系网络，从不同的视角有效地连接相应的区域，来增强每个视图图像的信息。关系网络关注于一组视图之间的相互关系，通过整合这些视图以获得有区别的三维对象表示。Wei 等人[8]首先构建了以多个视图为图节点的“视图”图，然后在“视图”图上设计了图卷积神经网络，该设计考虑到了多个视图之间的关系，实现了分层次地学习具有判别性的形状描述符。

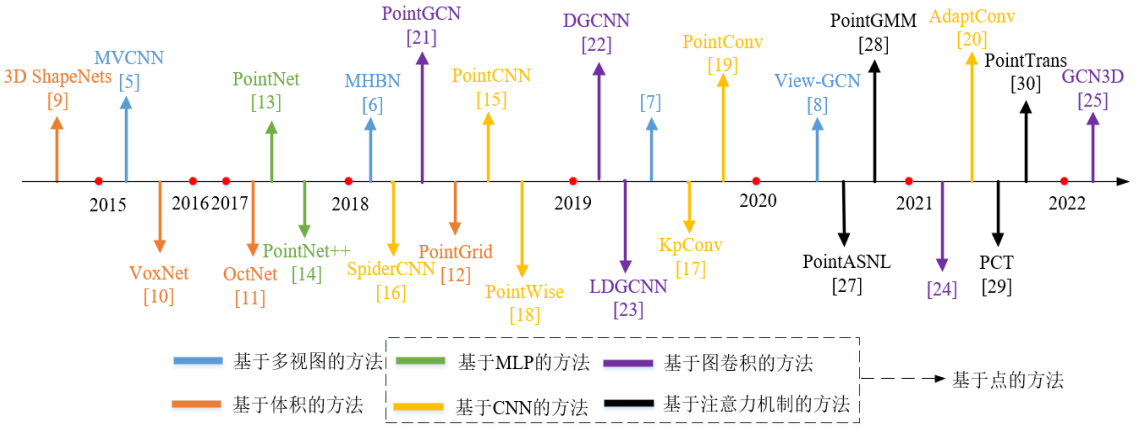

图 1-1 基于深度学习的点云分类方法发展历程

Figure 1-1 Development of point cloud classification method based on deep learning

（2）基于体积的方法是将点云划分为均匀的空间三维网格，然后使用三维的深度卷积网络 CNN 去提取特征进行识别，其中如何减少其计算量并提高其分类准确度是主要的挑战。Wu[9]使用卷积深度信念网络将几何三维形状表示为三维体素网格上的二进制变量的概率分布。VoxNet[10]是采用体素化方法的经典文章，其具有开创性的工作意义，它首先将原始点云转换为体积占用网格，然后利用有监督的三维卷积网络来实现点云物体的识别。虽然这些方法已经取得了不错的结果，但是计算量和内存的使用将会随着分辨率的增加而增加。为了解决这一问题，最近的研究已经提出了稀疏表示来减少对内存的需求。OctNet[11]使用浅层八叉树表示规则网格，并采用位字符串表示来提高编码效率，与密集输入网络模型相比，OctNet 降低了计算资源的消耗。PointGrid[12]提出了一种混合模型，通过点和网格的集合，可以更好的表现出局部几何形状的细节，其主要思想体现在数据的预处理上：将点云数据固定在一定大小的立方体中，然后在每个立方体中随机选取 k 个点，将这 k 的点的特征作为对应的立方体的特征，最后使用三维卷积来提取几何细节。

（3）基于 MLP 的方法不需要将点云进行转换，从而会保留原始点云的信息，其主要特征提取网络通过 MLP 实现。PointNet[13]作为第一篇直接处理点云的文章，有着巨大的贡献，为后面的工作指明了方向。其针对点云的置换不变性和旋转不变性，分别提出了对称函数、T-Net 网络。前者利用 Max-Pooling 来处理，因为所有点通过多层感知机来学习，拥有着相同的参数，所以当进行点的顺序变化的时候，不同的点在某一维度的值是不会改变的，而 Max-Pooling 就会一直提取到某一维度最大的特征值，从而不会受到点乱序的影响；后者通过学习一个$n\times n$的旋转矩阵，和输入点进行相乘，目的是将原始点云旋转到一个比较规整的方向，从而保持旋转不变的特性。但是明显的是 PointNet 网络在学习的过程，每个点是没有交互的，所以无法获得局部邻域的信息。PointNet++[14]针对此问题，进行了相应的修改，它在多尺度下进行半径查询来丰富每个点的特征，主要实现是以每个点本身为球心，获取指定 r 为半径内的所有点的信息，然后通过多层感知机以及 Max-Pooling 操作来更新自己的特征。邻域查询虽然丰富了每个点的特征，但是获取的特征主要是全局特征，从局部来说，点之间的相互作用还是相对比较少。

（4）基于 CNN 的方法在原始点云上进行处理，通过借鉴二维图片中的卷积思想来处理三维点云，主要区别是每个点和邻近点的权重以及特征的计算方法有所不同。PointCNN[15]的权重类似于图片自适应学习，而其特征为邻近点的特征以及两者之间的形状学习；SpiderCNN[16]的特征主要为邻近点的特征，但是其权重不再是自适应学习，而是利用阶跃函数和泰勒公式的相乘结果作为权重，丰富了学习参数；KPConv[17]的特征同样为邻近点的特征，但是其权重是从固定好的几个核点学习到的；PointWise[18]的权重为自适应学习，而特征为每个点累积特征的累加和；PointConv[19]的权重则利用了中心点和邻近点的相对坐标来学习，特征没有变化，但是根据点的位置进行了密度估计，自动地学习来适应点云的稀疏性变化；AdaptiveConv[20]的权重通过相对特征来学习，特征值为中心点和邻近点的坐标关系，因为其权重的学习具有自适应性并且在特征上融合了坐标信息，所以有着比较好的效果。

（5）基于图卷积的方法类似于基于 CNN 的处理方法，主要不同的是该类方法以图的思想来处理点云，通过选一个中心点，然后查找离其比较近的 k 个点建立图，每个点与中心点的欧式距离作为边。PointGCN[21]利用切比雪夫多项式构造图来对 3D 点云数据进行分类；DGCNN[22]以图的思想进行处理，在坐标以及特征的维度上，对每一个建成的局部图进行 Max-Pooling 操作来提取每个维度的最大特征来作为该局部图的中心点的特征；Linked-DGCNN[23]在 DGCNN 的基础上进一步通过跳跃连接来丰富每个点的特征；为了增大感受野，多尺度图卷积网络[24]在相同点云数量下，通过并行处理来提高网络的性能；在图卷积的基础上，GCN3D[25]

添加三维方向卷积使网络不仅在局部邻域学习到了其它点的特征信息，而且还增加了点对之间的方向信息。

（6）基于注意力机制的方法直接处理原始点云，因为考虑到点之间的长依赖性，往往有着更好的效果。随着 Attention[26]的兴起，越来越多的注意力机制相关的工作逐渐从自然语言处理向计算机视觉发展，并且三维点云方向也逐渐出现了与注意力机制相关的文章。Yan[27]提出了 PointASNL 来处理噪点，其中使用了自注意机制学习更新每个点的局部信息。Hertzet[28]提出了 PointGMM 网络，通过多层感知机分裂和注意力分裂来用于形状插值。PCT[29]提出了连续几个全局注意力机制的特征模块来学习点与点之间的关系，在分类中取得了比较好的效果。Point Transformer[30]受到图片应用上的向量自注意力机制[31]的启发，利用向量自注意力机制替代了标量自注意力机制，不仅减少了参数量，同时达到了更好的分类效果。

基于多视图的点云处理方法和基于体积的点云处理方法，在数据规则化后，可以很好地利用目前比较完备的卷积神经网络来处理，但是点云的多视图处理会导致某些重要的形状信息的丢失以及基于体积的处理会导致比较大的计算量，所以目前的很多方法都侧重于直接处理点云，如上面的（3）-（6）中所提到的方法。本文根据目前点云算法的效果，同样直接处理点云，通过引入注意力机制实现点之间的特征交互，保证了点云特征的长依赖性，利用 Mask 处理来进一步实现点云特征信息的选择。

1.2.2 点云语义分割算法研究

传统的点云分割算法主要是利用点云的位置和形状信息来分割出不同的区域边界。如可以通过三维物体边缘点的强度变化来得到不同分割区域的边界或者利用霍夫变换（Hough Transform，HT）[32]和随机采样一致性（Random Sample Consensus，RANSAC）[33]算法来检测平面、直线等。但是传统的分割方法得到的结果是不含有语义信息的，所以对于语义分割任务来说，需要进一步人工手动的对不同区域标注，效率极其低下。由此，实验人员开始逐渐转入到基于深度学习的点云语义分割算法的研究。

近些年来基于深度学习的点云语义分割方法取得了一系列成果，因为深度学习能够自动对三维空间中不同种类的物体标注不同的语义标签，实现了一个端到端的过程。基于深度学习的点云语义分割算法根据网络模型输入的点云数据类型以及点云处理方法的不同，现有的点云语义分割的相关方法主要分为基于多视图的方法、基于体积的方法和基于点的方法。而基于点的方法中，进一步可以分为基于 PointNet 框架的方法和基于 PointNet++框架的方法，本文分别简称为基于 PN 的

方法和基于 PN+的方法[34]。如图 1-2 所示，基于深度学习的点云语义分割方法发展历程如下。

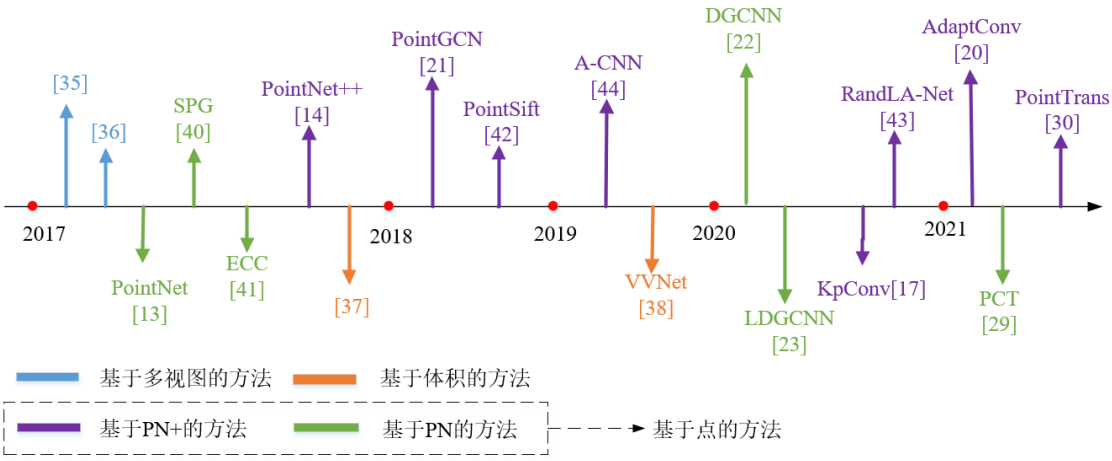

图 1-2 基于深度学习的点云语义分割方法发展历程

Figure 1-2 Development of point cloud semantic segmentation method based on deep learning

（1）基于多视图的方法中，Lawin 等人[35]首先将点云分别从多视角投影到一组 2D 图像上，而不是直接在三维空间中解决问题；然后将这些图像作为输入，利用卷积网络进行语义分割；最后，将得到的预测分数重新投影到原始点云上来得到分割结果。Boulch 等人[36]提出了一个将 CNN 应用于点云的多个二维图像视图的框架，主要包含视图的选择、完全卷积网络对 2D 视图的标记、使用有效的缓冲来标记每个 3D 点等步骤，最后在三维空间中执行标签预测的快速反投影来得到最终的结果。

（2）基于体积的方法中，Graham 等人[37]引入了一种稀疏的卷积运算，该运算专门处理稀疏数据，和以往的稀疏卷积网络不同的是它需要在子流形上运行，而不能将观测拓展到每一层；VV-NET[38]将非结构化的点云转化为规则化的体素网格，然后使用基于内核的插值变分自编码器（Variational Auto-Encoders，VAE）体系结构在每个体素中编码局部几何，最后为了处理点与点之间的稀疏分布，通过采用径向基函数（Radial Basis Function，RBF）来进一步的计算每个体素的局部连续表示。

（3）基于点的方法中，主要分为基于 PN 的方法和基于 PN+的方法。其中 PN 方法中的语义分割结构没有涉及到点云的下采样和上采样过程，而是直接对输入的所有点云直接处理，而 PN+方法中的语义分割的网络模型则是类似于 U-Net[39]的网络结构，包含了下采样和上采样的过程。其中基于 PN 的方法主要包括了 PointNet、DGCNN、LDGCNN、SPG[40]、ECC[41]、PCT 等网络，其主要思想是在网络的处理中，没有涉及到点云的下采样过程，最后学习到的特征主要是点云的全

局特征，但是点云的语义分割还会涉及到点云的细粒度特征，所以通常这些方法会进一步利用 Repeat 操作将全局特征分配到每个点上，并且利用跳跃连接将之前每个点的特征进行合并，来保证全局特征和局部特征的融合；而基于 PN+的方法中主要包括了 PointNet++、PointGCN、PointSift[42]、RandLA-Net[43]、A-CNN[44]、KPConv、AdaptiveConv、Point Transformer 等网络，其主要思想是如何设计点云的下采样以及上采样方法，使其通过下采样不断学习点云的更高维度的特征，通过上采样不断恢复原来每个点的特征，并利用跳跃连接来丰富每个点的特征。

基于多视图的点云处理方法和基于体积的点云处理方法，因为分别会导致形状信息的丢失和产生比较大的计算量，所以近些年的很多方法都侧重于直接处理点云，主要包括基于 PN 的方法和基于 PN+的方法。但是两者相比较而言，PN+中论文的实验有着更好的结果，因为这些网络的最初关注点就在每个点身上，而 PN 中论文提出的网络比较关注于整体，每个点在原来的特征上合并整体特征，对于局部的感知效果不如 PN+中的方法，所以目前很多的论文逐渐偏向于采用 PN+方法中的网络结构，而这些论文中的网络主要不同的是提取局部特征的方法设计。本文根据目前点云语义分割算法的发展历程，选择直接处理点云，并采样基于 PN+方法中的网络结构，设计了下采样和上采样的过程，其中下采样过程中引入了自注意力机制实现局部点之间的特征交互，并进一步利用位置信息丰富每个点的特征，上采样过程中使用 PointNet++中的插值方法来恢复点的数量以及获取更丰富的特征信息。

1.3 点云深度学习的挑战

在计算机视觉相关任务中，点云作为三维空间中的常用数据，发挥着重要的作用，但是因为其本身的一些特性，相比于图片来说，其无法很好的应用到卷积神经网络中，从而在特征提取中往往达不到满意的效果。接下来，本文详细介绍下点云本身的特性，并进一步分析目前点云分类、语义分割等任务的不足。

1.3.1 点云本身的特性

2D 图像相比于 3D 点云，明显缺少了现实世界中物体的深度信息以及多个物体之间的相对位置信息，从而不适用于需要深度信息以及定位信息的应用，如自动驾驶、虚拟现实（Virtual Reality，VR）、增强现实（Augmented Reality，AR）和机器人技术等。点云是三维空间中的一组数据点集，最简单的形式只包含 x、y、z 坐标，根据在统一的坐标原点下，可以构建出类似于现实世界的三维场景，但是点云

本身的一些特性也给计算机视觉，如分类、分割等应用带来了非常大的挑战，主要包括以下两点：

（1）点云的不规则性：点云数据不具有图片中像素点之间排列整齐的特性，而仅仅是一个三维空间中的集合，所以每个点周围的点数都是不一定的，并且相对位置和方向都有着出入，另外因为点的不规则性，导致了采集的点云数据可能由于遮挡等问题出现点的分布不均匀，其中一些区域的点密集，而一些区域的点比较稀疏。

（2）点云的无序性：一般某个场景下的点云是无序的，其主要由一组点形成，通常在文件中是以列表形式存储的。在 2D 图像中，图片中的每个像素点的位置固定，不可调换，但是点云作为一个集合，整体来说，点的存储顺序的不同不会影响物体的表示。因此，如果利用深度学习，尤其是卷积神经网络来处理点云，是非常有挑战性的。主要因为卷积神经网络是基于卷积操作运算的，而卷积运算需要在有序并且结构规则化的网格上进行的。从而早期的一些方法主要是将点云转换成规则的形式来使其容易被卷积神经网络处理，但是往往会涉及到信息的丢失和计算复杂度的提高。

1.3.2 点云分类和语义分割方法的不足

虽然目前研究人员在点云分类和语义分割领域的研究具备一定成效，并且在 ModelNet40[9]、ScanObjectNN[45]和 S3DIS[46]等数据集上取得了显著的效果。但是由于点云本身的一些特性的限制，基于深度学习的点云分类和语义分割算法的发展仍然面临着许多问题，其中主要有如下几点：

（1）基于深度学习的点云分类算法无法有效地获取到局部特征。常见的点云局部处理通常利用 KNN 进行分组，然后通过对邻近的点云进行处理来获取更新后的特征，但是往往局部分组内点的交互性不足，无法获得更丰富的特征。本文针对该问题提出了在每一分组内进行所有点的特征自适应学习，来获取更丰富的特征。

（2）基于深度学习的点云分类算法往往忽略了点云的全局信息。点云的分类是对整个物体的分类，所以最终学习的目标是最能表示物体的特征。但是由于点云的无序性，点云存储位置的变化会导致学习的困难。本文针对该问题，引入了自注意力机制，实现了全局范围内点对之间的交互，从而不会受到点云无序性的影响。

（3）基于深度学习的点云语义分割算法不能很好地利用到点云的位置信息。点云的语义分割是对每个点的分类，所以最终学习的目标是丰富每个点的特征。目前很多的网络只关注了点云的语义特性，而忽略了点云的位置信息，从而本文在保证点云语义特征的同时，引入了点云的位置信息，结合两者来丰富每个点的特征。

1.4 本文研究方法

在研究点云分类和语义分割算法的过程中，为了提升网络模型的准确度，部分研究者采用多视图转换以及体素化的方式，但这势必会导致点云信息的部分丢失和运算量的增加，从而大部分的研究者现在直接在原始点云上进行模型的训练，但是往往会在点信息交互的时候缺乏局部信息，以及只关注语义特征或者几何特征，不能很好的联系两者。本文针对点云分类和语义分割的问题，分别从基于神经网络的分类网络和语义分割网络的算法准确度两方面展开研究，实现更准确的点云分类和语义分割任务。主要内容如下：

（1）基于注意力机制的多尺度点云分类算法。在图像的很多工作中融合不同的尺度是提高分类性能的一个重要手段。低层特征分辨率很高，但是语义性较低；高层特征分辨率很低，但是语义性较高。所以将两者融合将会达到一个比较好的效果。为了降低计算量，本文直接在原始点云上进行操作，并且将图像的多尺度思想应用到点云分类网络，利用最远点采样（Farthest Point Sampling，FPS）进行下采样来实现多尺度，在每个尺度下，每个点首先通过空间特征变换模块在局部邻域内聚集特征；下一步为了进行长依赖关系的特征交互，在每个尺度下，分别利用自注意力机制来学习全局特征；最后为了更好地融合特征，提出了特征融合模块，对于每个尺度都要学习一个权重，然后进行多个尺度特征的自适应融合来提高整体的分类准确度。

（2）基于注意力机制的多尺度点云语义分割算法。为了提高分割的性能，本文的分割算法同样利用最远点采样（Farthest Point Sampling，FPS）方法进行下采样来实现多尺度，但是和分类不同的是，分类最后返回的是最能表征该物体的特征，而语义分割需要保留原始的点数，针对每一个点进行分类，所以经过下采样后，虽然语义特征增强了，但是其点数减少了，因此需要恢复到原来的点数，并且需要尽可能的保证每个点特征的丰富性，这里本文采用 PointNet++中提出的插值法，将其下采样前离得最近的几个点的特征平均处理后作为自身的特征，并利用残差结构保留原有的特征；为了增加每个点的区分度，本文的语义分割网络从位置信息和语义特征信息两方面出发，利用位置信息来丰富语义信息，其中位置信息利用合成注意力机制来学习每个位置的重要性，然后在每个局部邻域内利用自注意力机制进行局部信息内的交互，来丰富每个点的特征；最后不同于分类进行全局点之间的特征交互，而是使用了通道注意力机制来区分每个维度的重要性，从而增加每个点的独特性。

1.5 本文组织结构

本文一共分为五章，具体的组织结构如下：

第一章主要介绍点云分类和语义分割的研究背景，分析 3D 点云分类和语义分割在众多方面的应用以及价值，以此引出了本文内容作为基础性工作的重要研究意义。然后主要介绍了 3D 点云的分类和语义分割算法在国内外的发展现状，主要包括基于多视图、基于体素化、基于点等三个不同点云处理方向的发展。

第二章主要介绍关于 3D 点云分类和语义分割网络模型的技术内容。包括深度学习、自注意力机制等基本理论知识。然后介绍了在点云分类和语义分割方面常见的评价方式和标准的实验数据集。

第三章介绍了基于注意力机制的多尺度点云分类算法。为了提高网络的分类性能，该章分别从局部特征提取模块、全局特征提取模块、尺度融合模块出发，来一步步地验证网络的分类有效性。随后在 ModelNet40[9]合成数据集以及 ScanObjectNN[45]真实数据集上进行实验验证，从分类结果上来说明不同的模块对分类的影响。最后，在实验分析中，本章算法与当前主流点云分类算法进行对比，并取得了优异的结果。

第四章介绍了基于注意力机制的多尺度点云语义分割算法。因为语义分割和分类在任务上的不同，语义分割的侧重点不再是局部和全局信息的融合，而是局部信息的有效提取，以及位置信息的融合，本章分别提出了基于位置信息的合成注意力模块，以及基于语义信息的自注意力模块，来一步步地验证网络的语义分割性能。随后在 S3DIS[46]数据集以及 ShapeNetPart[47]补充数据集上进行实验验证，最后，在实验分析中，本章算法与当前主流点云语义分割算法进行对比，取得了优异的结果。

第五章主要是总结与展望，首先概括了本文的主要内容，然后分析了现有工作的不足之处，以及未来重点研究的方向。

2 相关理论技术

2.1 深度学习理论基础

人工神经网络（Artificial Neural Networks，ANN）是一个仿生学概念，其出现是受到了人类中枢神经系统的启发。ANN 的主要特点包括“具有一组可以被调节的权重”和“可以估计输入数据的非线性关系”。目前，神经网络由于任务的不同分为很多类别，如用于回归任务的反向传播（Back Propagation，BP）神经网络、用于图片处理的卷积神经网络（Convolutional Neural Network，CNN）和用于序列数据处理的时间递归神经网络（Long short term Memory Network，LSTM）等。但是对于本文研究的点云来说，主要涉及到的是多层感知机（Multilayer Perceptron，MLP），其也是最简单的神经网络。本文的模型操作除了涉及到多层感知机，还包含了一些基础的理论操作，如激活函数，本小节将对其分别进行简要阐述。

2.1.1 多层感知机

多层感知机（Multilayer Perceptron，MLP）又称前馈神经网络，其过程是定义一个映射，学习其中的参数，使其输出近似某个函数，其结构如图 2-1 所示：

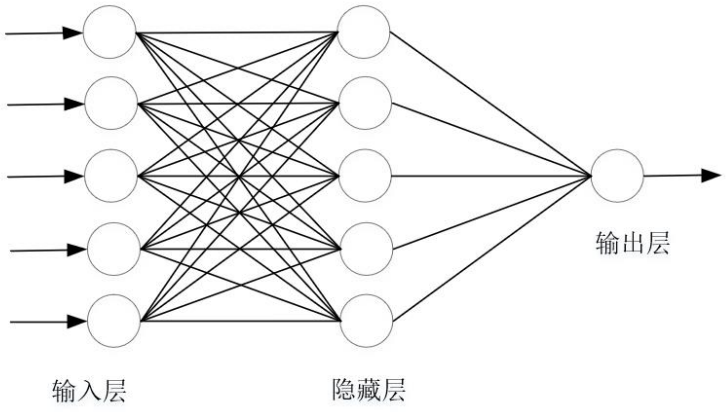

图 2-1 多层感知机操作示意图

Figure 2-1 The schematic diagram of multilayer perceptron operation

多层感知机（Multilayer Perceptron，MLP）非常适合点云的处理，因为点云本身是无序的，不具有图片整齐排列的特点，其表示形式是每个点对应的坐标和特征。为了实现更好的性能，本文需要针对点云进行局部信息的提取，所以很多方法提出

在每个点附近查找邻近点，来丰富每个点的特征，对于每个局部邻域内点的特征获取可以利用卷积操作，但是仅仅处理每个点周围点的特征，所以过滤器大小一般是 $1 \times N$，其中 N 为每个点的邻近点，从而该操作可以看成是多层感知机，每个点的所有邻近点共用一组参数，同样实现了局部共享。

目前，传统的多层感知机对特征信息的处理能力相比于卷积神经网络而言较差，所以在计算机视觉领域中的图片识别、分割等任务中依旧以卷积神经网络为主要的特征处理方式。但是最近随着注意力机制的兴起，引发了研究者对多层感知机的重新认识，虽然其结构简单，但是通过网络的设计在多个任务中也可以达到较好的效果，如 MLP-Mixer[48]网络，其主要结构为多层感知机，但是在图片的分类中达到了较好的效果，以及最近的网络 Point-Mixer[49]借鉴了 MLP-Mixer 网络的思想，在点云的分类、分割等任务中同样达到了较好的效果。

2.1.2 激活层

激活函数的使用主要是为了解决神经网络本身无法拟合非线性函数的问题。多层感知机操作中的权重映射过程是一个线性的学习过程，但是如果加入激活函数就可以使神经网络获取非线性特性，从而使其在训练过程中具有非线性映射的学习能力，提高网络的处理能力。本文的激活层主要使用以下几种激活函数。

（1）Sigmoid 激活函数

函数的数学公式如式(2-1)所示：

$$f(x)=\frac{1}{1+e^{-x}} \tag{2-1}$$

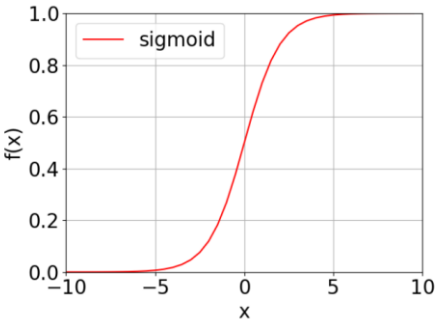

图 2-2 Sigmoid 函数

Figure 2-2 The Sigmoid function

Sigmoid 函数曲线图如图 2-2 所示，该函数一般作用于神经网络的隐藏层，它可以将输入的实数值转化为[0,1]范围之间的输出。从图中曲线可以看出，当输入的数值越小，输出越接近于 0；当输入的数值越大，输出越接近于 1，并且 Sigmoid

的导数只有在输入为 0 附近的值时，才有比较好的激活性，在输入值非常大以及非常小的时候会出现梯度消失等问题，所以一般不适合深层网络。

（2）线性整流函数（Rectified Linear Unit，ReLU）

函数的数学公式如式(2-2)所示：

$$f(x)=max(0,x)=\begin{cases}x, x>0\\0, x\le 0\end{cases} \tag{2-2}$$

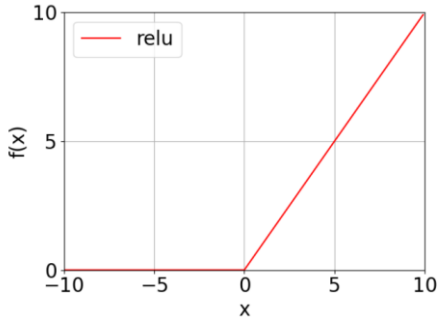

图 2-3 ReLU 函数

Figure 2-3 The ReLU function

ReLU 函数曲线图如图 2-3 所示，Sigmoid 函数通常会出现梯度消失的问题，而 ReLU 函数可以有效规避这个问题。另外，ReLU 函数因为运算简单，并且在负半区的导数为零，会造成网络的稀疏性，从而具有更快的计算速度和收敛速度。

（3）Softmax 激活函数

函数的数学公式如式(2-3)所示：

$$f(x)=\frac{e^{x_i}}{\sum_i^k e^{x_i}} \tag{2-3}$$

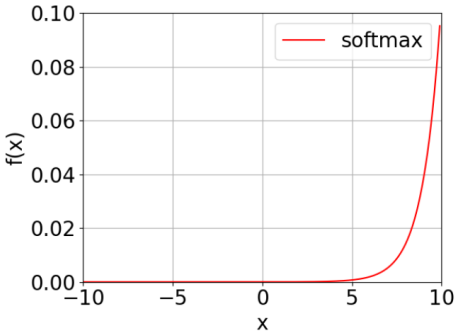

图 2-4 Softmax 函数

Figure 2-4 The Softmax function

Softmax 函数曲线图如图 2-4 所示，该函数一般用于多分类问题，假设某个分类网络为 k 分类，那么对于长度为 k 的任意实向量，其作用是将 k 个值分别压缩

到[0,1]之间，并且压缩后实向量中的 k 个元素值的和为 1，其结果作为最后的分类结果，哪个位置的值越大，那么分类结果则属于该类。

2.2 点云相关理论基础

在点云的分类和语义分割任务中，主要涉及到了最远点采样、K 最近邻算法、插值法等基础操作，本小节将对其分别进行简要阐述。

2.2.1 最远点采样

最远点采样（Farthest Point Sampling，FPS）作为点云相关任务中比较常见的下采样方法，尽可能保留了输入点云下采样前后的形状信息。其主要思想是在采样的时候，选择的下一个点与已采样的点集合的距离最远。主要可以分为四步：

（1）随机选取初始点加入采样点集合。

（2）计算其它所有点和初始点的欧式距离，选择和初始点距离最远的点加入采样点集合。

（3）计算采样点集合之外的每个点和采样点集合内每个点的欧式距离，其中采样点集合之外的每个点选择离采样点集合中每个点的最小距离作为采样距离，最后选择采样点集合外的最大采样距离的点加入采样点集合。

（4）重复步骤（3），直至采样点集合中点的个数达到指定要求。

2.2.2 K 最近邻

K 最近邻（K Nearest Neighbor，KNN）是一种经典的有监督的学习方法之一。因为其没有显示的学习过程，所以当数据分布很少或者没有先验知识的时候，其是一个不错的选择。KNN 常常用于解决分类问题，其原理为：当需要对测试样本进行分类时，首先通过扫描所有的训练样本，找到与该测试样本最相似的 K 个训练样本，根据这些样本的类别进行投票确定测试样本的类别。其中相似度量指标根据任务的不同会有所区别，常见的度量包括欧氏距离、曼哈顿距离等。

在点云任务中，KNN 主要用于局部邻域的分组，对于点云中的每一个点，通过 KNN 算法，找到其在三维空间中最相似的 K 个点构成一个分组，然后使用分组内的所有点来丰富中心点的特征。其中在点云分类和分割算法中，KNN 的相似度量指标主要包括点与点坐标的欧氏距离以及特征维度下的欧氏距离。坐标的欧氏距离作为相似度指标保证了点与点之间有着较近的位置关系，而特征维度下的欧

氏距离则保证了点与点之间有着相似的特征关系。

2.2.3 插值法

在进行点云下采样的过程中，点的数量在不断减少，而对于点云语义分割这类任务，本文需要恢复到最初的点数，针对每个点进行相应的特征表示，为了在恢复点数的过程中，尽可能的保留每个点学习的特征，本文采用 PointNet++论文中所使用的方法，进行点云的特征插值恢复。

函数的数学公式如式(2-4)所示：

$$f_i=\frac{\sum_{j=1}^{N_1}w_j(p_i)f_j}{\sum_{j=1}^{N_1}w_j(p_i)},w_j(p_i)=\begin{cases}\|p_i-p_j\|_2^{-1},p_j\in N_2(p_i)\\0,p_j\notin N_2(p_i)\end{cases} \tag{2-4}$$

其中，假设当前点的数量为 N_1，本文要恢复到点数 N_2，其中 N_2 为 N_1 的倍数，因为 N_1 是 N_2 进行下采样得到的，所以此时 N_1 个点的特征有着更大的感受野，并且其特征相对丰富，所以恢复的时候应该使用其特征 f_j。公式中的 p_i 为 N_2 个点中的某一个点，$N(p_i)$ 代表 N_1 中离 p_i 点最近的 k 个点，针对 N_2 个点中的每一个点，本文查找其在 N_1 个点中最近的 k 个点，然后对这 k 个点进行特征的加权处理，其中加权的特征是 $w_j(p_i)$，意义是 N_2 中的点和 N_1 中点的距离的倒数，也就说明离的越近，其特征的权重越大，为了不受到过多点的干扰，一般 k 取值较小，本文在实验中分别取了 1、3、5 进行实验来验证最佳的效果，结果发现 k 的值为 3 时有着最好的效果。

2.3 残差网络

在神经网络收敛的前提下，对于一个网络来说，其深度越深，网络的性能是逐渐增加至饱和，然后再迅速下降。因为神经网络本身来说是不容易拟合一个恒等映射的，所以随着深度的增加后面的输出可能变化太大，而残差网络[50]的出现就是为了解决神经网络本身的输入和输出维度的不一致问题，在一定程度上解决梯度弥散问题，然后使得网络可以在保持性能的基础上尽可能地增加深度，尝试达到更好的结果，传统的残差网络如图 2-5 所示，输入 X 经过中间卷积层的处理得到的结果和输入再次相加作为最后的结果。本文的实验结构中大多都利用了残差网络的结构来进行网络的训练，后面就不再过多赘述。

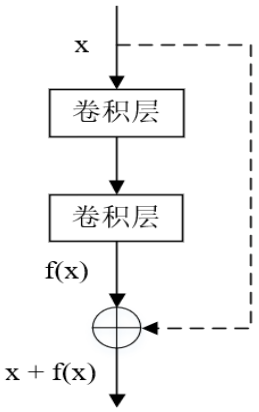

图 2-5 残差网络结构

Figure 2-5 The architecture of Residual Network

2.4 注意力机制网络

注意力机制网络的引入主要是为了解决计算能力以及优化算法的限制，在平常的网络中，神经网络的性能和模型的复杂情况有着密切的关系，所以要记住所有的信息一般会提高网络的性能，但是对于某些任务来说，获取所有的信息并不能达到最好的效果，因为一些背景信息可能会对最终的任务产生负面的影响，所以有时候需要对信息特征进行筛选提取，注意力机制的出发点就是这样，本文在网络中加入注意力机制，用来在同一层上进行关键特征的提取，同时抑制次要信息，使得网络训练过程中不断突出重要特征；另外在一些自然语言处理的任务中，需要在网络中保留之前的输入信息，但是对循环神经网络的长距离来说，信息保持的记忆能力并不高，所以注意力机制网络进一步改进为自注意力机制网络来解决长依赖问题，其主要保证了在网络的每一层中，每个输入都保持着与其它输入的关系，然后进行特征融合来更新本身的特征。

Attention 机制的本质是一个寻址处理的过程，给定一个从任务本身出发的查询量 Query，分别计算 Query 和对应的多个 Key 之间的关系，得到的关系作为相应的权重附加到对应的 Value 上，得到最后 Attention 的值。其主要的实现过程以及公式进一步说明如下：

（1）信息的输入

假设向量 $\overline{A}=[\overline{A_1},\overline{A_2},...,\overline{A_N}]$，表示含有 N 个输入信息点。

（2）注意力权重计算

令 $\overline{Key}=\overline{Value}=\overline{A}$，然后注意力权重计算如公式(2-5)所示：

$$a_i = Softmax(s(\overline{Key_i},\overline{Query})) \tag{2-5}$$

式中的 s 代表的 $\overline{Key_i}$ 和 $\overline{Query}$ 的打分机制，一般包括点积模型计算、缩放点积模型计算、双线性模型计算、加性模型计算等等，然后经过 Softmax 函数来归一化 $\overline{Query}$ 和每个 $\overline{Key_i}$ 的关系。

（3）信息加权平均

上面的注意力权重 a_i 可以认为是进行在上下文查询的时候，第 i 个信息 $\overline{Key_i}$ 受查询量 $\overline{Query}$ 关注的程度，然后根据加权平均的选择机制对所有的输入信息 $\overline{Value}$ 进行编码，得到最后的结果，如公式(2-6)所示：

$$y = \sum_{i=1}^{N} a_i \overline{Value_i} \tag{2-6}$$

以上就是 Attention 的处理机制，其主要原理是将信息输入到神经网络进行计算时不需要输入所有的 N 个信息，而是从输入中选择部分和任务相关性比较大的信息输入到神经网络。但是对于某些自然语言处理相关的任务，一般要关注长依赖问题，注意力机制中的自注意力机制解决了这个问题，它不同于全连接模型只能处理定长输入，而是根据不同的输入进行动态权重的生成。相比于普通的注意力机制，主要的变化是上面所提到的 $\overline{Query}$ 、 $\overline{Key}$ 、 $\overline{Value}$ 等向量都是由同一个输入 $\overline{A}$ 经过线性变化而来，即分别如公式(2-7)、公式(2-8)和公式(2-9)所示:

$$\overline{Query} = W_Q \overline{A} \tag{2-7}$$

$$\overline{Key} = W_K \overline{A} \tag{2-8}$$

$$\overline{Value} = W_V \overline{A} \tag{2-9}$$

除了输入的不同，自注意力机制和注意力机制的整体流程是一样的，而主要原理是输入为同一个点云特征的不同线性变化值，然后所学的权重是某一个节点和其它节点之间的相关性，之后进行加权求和，可以理解为一个节点的特征值是其它所有节点特征的加权和，从而在网络的训练过程中，保证了长依赖关系。

2.5 点云分类和分割评价指标

在对点云数据分别进行分类和分割处理后，点云的分类和分割网络的效果需要相应的指标来进行评判。目前，常用的评价方式是获取所有的测试结果，然后分别从整体效果、每一类别的效果等角度出发，构造出不同的计算公式，从而对网络模型的效果进行评价。

2.5.1 点云分类评价方式

对于点云分类的评价指标，本文使用的是所有类别的总体准确度（Overall Accuracy，OA）和每个类别的平均准确度（mean class Accuracy，mAcc）。

（1） 总体准确度（Overall Accuracy，OA）

OA 的计算公式如式(2-10)所示：

$$OA=\frac{\sum_{i=0}^{k}p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k}p_{ij}} \tag{2-10}$$

其中，$k+1$为共有的类别数量，p_{ii}代表每次分类的真实值为第i类，预测结果也为第i类的数量；p_{ij}代表每次分类的真实值为第i类，预测结果为第j类的数量。OA 的值越小，表示整体的分类准确度越低；反之，OA 的值越大，表示整体的分类准确度越高。

（2） 类平均准确度（mean class Accuracy，mAcc）

mAcc 的计算公式如式(2-11)所示：

$$mAcc=\frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}} \tag{2-11}$$

其中，$k+1$为共有的类别数量，p_{ii}以及p_{ij}的意义和 OA 中一样，但是 mAcc 的计算方式有了变化，其分别针对每一类计算预测的准确度，然后取平均，mAcc 的值和 OA 的值大小差别不大时，表明每一类的分类准确度比较平衡；反之，表明不同类别物体的分类准确度的差距较大。

2.5.2 点云分割评价方式

对于点云语义分割的评价指标，和点云分类相似，也使用了所有类别的总体准确度（Overall Accuracy，OA）和每个类别的平均准确度（mean class Accuracy，mAcc），这里不再赘述，但是因为点云语义分割的独特性，还有平均交并比（mean Intersection over Union，mIoU）作为标准度量，其分别计算每一类的真实值以及预测值两个集合的交集和并集，然后用交集和并集之比作为每一类的预测结果。

平均交并比（mean Intersection over Union，mIoU）评价指标的计算公式如式(2-12)所示：

$$mIoU=\frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{2-12}$$

其中，$k+1$为共有的类别数量。p_{ii}代表每次分割的真实值为第i类，预测结果也为第i类的数量；p_{ij}代表每次分割的真实值为第i类，预测结果为第j的数量；p_{ji}代表每次分割的真实值为第j类，预测结果为第i类的数量。mIoU 的计算过程主要是每一类预测正确的数量除以（预测属于此类的样本数量＋此类的样本总数量－此类预测正确的数量），然后对所有类取平均值。如图 2-6 所示，TP、FP、FN 分别代表真正例（预测值为 1，真实值为 1）、假正例（预测值为 1，真实值为 0）、假反

例（预测值为 0，真实值为 1），从而计算过程等价于 TP 比上 TP、FP、FN 三者的并集，然后在每个类上计算 IoU，最后取平均得到 mIoU。

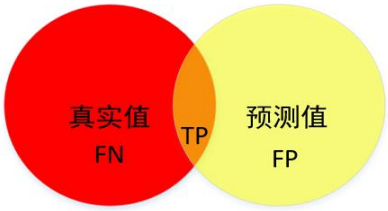

图 2-6 mIoU 计算

Figure 2-6 The mIoU Calculate

2.6 实验数据集

对于深度学习的相关任务来说，数据是不可缺少的，因此构建大规模且被世界所公认的数据集是有重大意义的。在三维领域内，根据任务的不同，学者们构造出了很多特定领域的数据集，如分类数据集 ModelNet40 和 ScanObjectNN，室内语义分割数据集 S3DIS，零件分割数据集 ShapeNetPart 等等。

为了验证本文所提算法的性能，本文分别在 ModelNet40、S3DIS 数据集上进行了相关的实验操作，并且为了验证网络模型的通用性，进一步在 ScanObjectNN 和 ShapeNetPart 数据集上做了对应的补充实验。下面重点介绍下以上四种数据集。

2.6.1 ModelNet40 数据集

ModelNet40 数据集是由普林斯顿大学提出的，其目标是为众多领域的研究者们提供一个全面及高质量的三维模型集合。其主要包含 12311 个 CAD 模型，一共属于 40 个对象类别，分别有飞机、桌子、椅子、花瓶、电脑、杯子等等。因为该数据集是合成的，所以不包含噪声点，如图 2-7 所示，这是其中部分训练样本可视化后的结果。

2.6.2 ScanObjectNN 数据集

点云分类的基准数据集一般是 ModelNet40 数据集，但是随着目前点云获取设备以及点云分析算法技术的快速发展，可能无法满足现代应用上的要求，主要因为

其是合成数据集，相对完美，没有噪点，所以本文也针对真实世界的 ScanObjectNN 数据集进行了补充实验。

ScanObjectNN 数据集是在 2019 年提出的点云基准，它包含了 15000 个对象，这些对象一共分为 15 个类别，如背包、沙发、床等，这些对象在现实世界中有 2902 个唯一的对象实例，和 ModelNet40 不同的是 ScanObjectNN 数据集中的每个对象都在一定程度上有着背景、噪声、遮挡的影响，如图 2-8 所示，这是其中部分训练样本可视化后的结果。

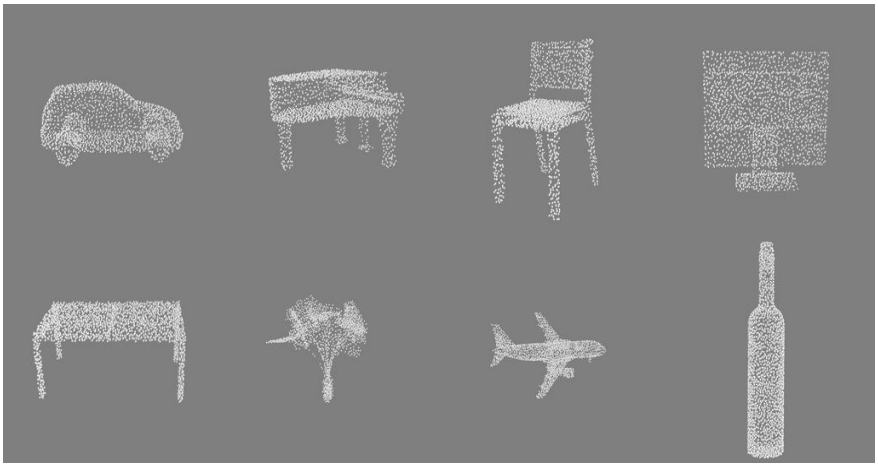

图 2-7 ModelNet40 数据集

Figure 2-7 The ModelNet40 dataset

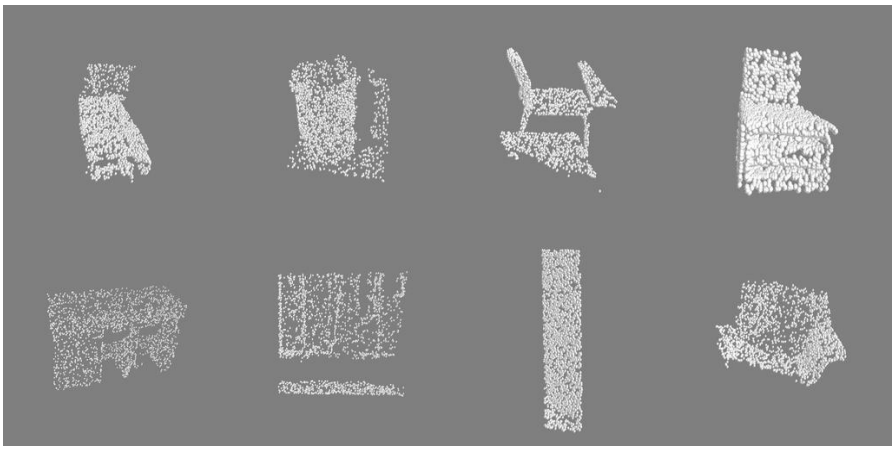

图 2-8 ScanObjectNN 数据集

Figure 2-8 The ScanObjectNN dataset

2.6.3 S3DIS 数据集

在点云的相关应用中，大规模场景分割是一项相对分类更具有挑战性的任务。

S3DIS 数据集是斯坦福提出的大场景的 3D 室内点云数据集，其主要是针对 6 个区域的 271 个房间，通过使用 Matterport 深度视觉相机进行扫描，生成并重构纹理网格数据，然后下采样来制作的点云数据。它一共从 3 个不同的建筑中扫描了 2.73 亿个点，然后用 13 个类别中的其中一个语义标签对每个点进行注释，主要包括椅子、桌子、板子、地板等对象。如图 2-9 所示，这是部分房间可视化后的结果。

图 2-9 S3DIS 数据集

Figure 2-9 The S3DIS dataset

2.6.4 ShapeNetPart 数据集

ShapeNetPart 数据集被标注用于 3D 对象的零件分割。该数据集一共包含 16880 个样本，主要由 16 种形状类别组成，其中每个类别的样本的零件数量在 2 到 6 之间不等，总共有 50 个不同的零件。所属类别主要包括飞机、背包、车、椅子等等，而飞机包含机身、机翼、引擎、机尾，背包包含主体和背带等等。虽然语义分割是对大场景的每一个物体的分类，而零件分割则针对物体的不同部位分类，但是零件分割和语义分割是类似的，都是对于每一个点进行分类，为了进一步验证本文提出的语义分割算法，本文将 ShapeNetPart 数据集作为补充数据集进行实验评估。如图 2-10 所示，这是其中部分训练样本可视化的结果。

2.7 本章小结

本章主要介绍了用于点云分类和语义分割方法的深度学习相关的方法理论技术，首先介绍了人工神经网络的基本概念，以及多层感知机和激活函数基本理论；

接着介绍了点云相关的理论基础，包括最远点采样、K 最近邻和插值法三个算法；然后介绍了残差网络的基本结构以及解决的问题；进一步详细介绍了本文后面使用的注意力机制以及其变形网络自注意力机制的基本原理；之后介绍了点云分类和语义分割算法常用的评价指标；最后分别介绍了点云分类任务和语义分割任务的常见数据集以及扩展数据集。

图 2-10 ShapeNetPart 数据集

Figure 2-10 The ShapeNetPart dataset

3 基于注意力机制的多尺度点云分类算法

3.1 引言

随着近几年自动驾驶和机器人领域的火热发展，工业上利用激光雷达获取点云数据的技术越来越发达，以及目前数据计算能力的快速提高，促进了点云处理相关算法在学术界上的快速发展，点云分类作为计算机视觉领域需求的基准有着很重要的意义，点云分类可以用于整个城市建设，实现城区建设分区块的快速创建和展示，该做法对城市的规划有着巨大的意义；利用雷达对铁路轨道上的障碍物进行扫描，对其生成的点云进行分类识别，可以有效地保证火车的安全行驶。

点云的分类不应该仅仅关注局部特征，同时应该获取全局特征，分类的任务是要获取更能表现物体的特征，虽然下采样的处理可以看成多尺度的特征提取，包含了更大的感受野，但是在每一尺度下，点云之间全局的依赖关系没有那么明显，所以针对这个问题，本章在提取局部特征的同时，进一步地利用全局特征提取模块来丰富每个点的特征。

本章提出了基于注意力机制的多尺度点云分类算法，所构造的模型包含三层不同尺度的特征提取，每一层首先经过分组，利用空间特征变换进行局部特征的提取，然后进一步通过融合了自注意力机制的残差网络，其中该注意力机制实现了全局信息的特征提取，每个点的特征受到了其它所有点的特征影响。两个模块保证了局部信息和全局信息的获取，在网络的分类中起到了一定的贡献。为了对每个尺度下的特征有一定区分度，本章进一步提出了特征融合模块，该模块从输入本身出发，学习不同尺度下的权重，最后将不同尺度下的特征相加作为最后的特征用来分类。经过调参和大量实验对比，本章所提算法在 ModelNet40 数据集上，不涉及到输入的转换，直接在原始点云上处理，对于 40 类物体的分类精度超过了一众的算法，有着更好的分类能力；为了验证本章所提算法的通用性，本章进一步在包含噪声的 ScanObjectNN 真实数据集上进行训练和测试，通过结果发现，本章所提算法同样在其上面达到了较好的效果。

3.2 整体网络结构

为了更好地挖掘点云的局部以及全局特征信息，本章设计的基于注意力机制的多尺度点云分类算法的网络框架包含 1 个特征融合模块以及 3 个不同尺度下的

特征提取层，不同尺度的每个层级具有相同的局部和全局结构。如图 3-1 所示，3 个不同尺度下的特征提取层的输入分别是原始输入和上一层经过最远点采样以及局部特征提取模块的输出，不同于相同点数下的多尺度构造网络[24]，每一尺度下的点数是不一样的，然后每一尺度下的输入经过空间特征变换的局部特征提取模块来学习到每个点的局部信息特征，之后经过基于自注意力机制的全局特征提取模块进行全局点的特征选择和交互，为了避免全局信息交互后的梯度消失问题，本文在每个全局特征模块下增加了残差连接，接下来每一尺度经过最大池化层来进行每一维度下的特征选择。最后，本章提出多尺度特征融合模块来融合不同尺度下的特征，将三个尺度下的特征经过加权后相加，作为最后的结果。特征融合模块在实验中利用了 detach 函数来保证每一尺度的特征以及特征融合模块的单独训练，所以在最后的结果中，损失函数 loss 等于 3 个尺度下的损失函数 loss 以及经过特征融合模块 loss 的总和，但是最后的预测结果是结合了特征融合模块的三个尺度结果的输出。接下来，本章分别一一介绍每个模块的详细结构。

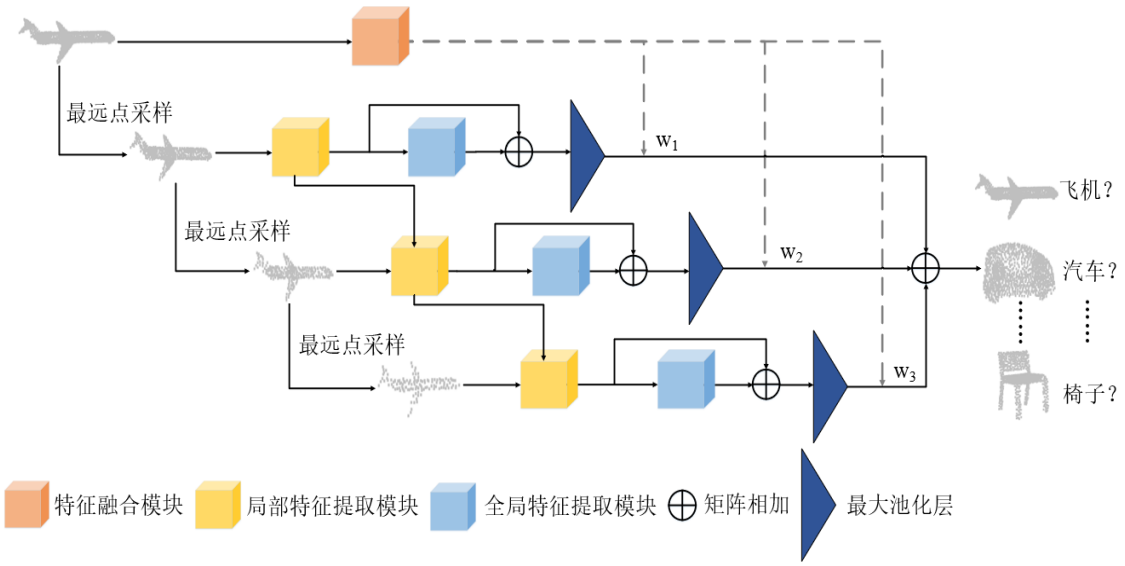

图 3-1 基于注意力机制的多尺度点云分类网络

Figure 3-1 Multi-scale point cloud classification network based on attention mechanism

3.3 基于空间特征变换的局部特征提取模块

在进行局部特征提取的时候，网络的输入首先基于 KNN 算法，进行相似特征点的获取，这里的 KNN 不是基于点的位置信息，而是借鉴 DGCNN 中的相似特征的 KNN 分组，是在空间特征维度下的分组查找，这样有利于在分类情形下进行关键特征的聚集，以及不同特征的分离；然后下一步是从每一个分组中提取有效特征。

之前一般的做法是直接利用对称函数，如最大池化或者平均池化，来进行每一维度特征的提取，但是因为在分组上有些点和中心点的特征差距较大，所以这时候可能会引入噪声点，因此本小节提出的网络在对局部信息进行对称函数提取之前，先对每一局部分组内所有点的特征自适应提取，来提升网络的学习能力。

如图 3-2 所示，空间特征变换的过程首先是学习同一分组内的每个点每个维度下的特征注意力值，然后和原来的特征进行相乘，此操作可以认为对于同一维度下每个点进行快速、有选择地关注到更有意义和价值的内容上，实现空间特征上的重新分布。另外为了使得网络可以更深，该模块针对每个局部特征提取都使用了残差网络的思想，将操作后的特征和原始特征相加，最后经过最大池化操作提取每个分组内的有效特征作为最后的输出。

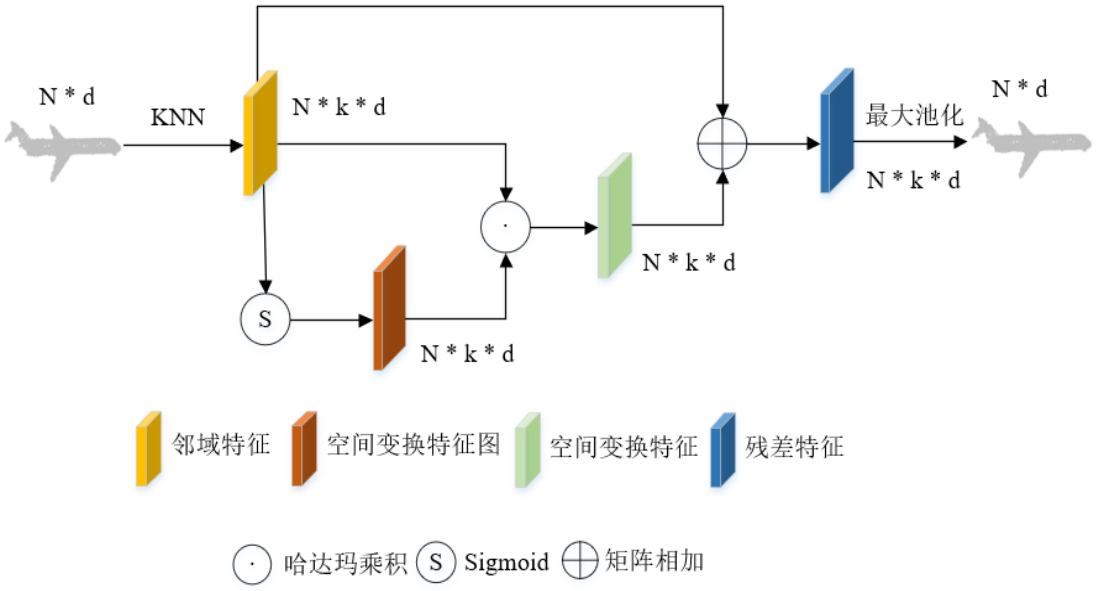

图 3-2 基于空间特征变换的局部特征提取模块

Figure 3-2 The Module of Local feature extraction based on Spatial Feature Transformation

3.4 基于自注意力机制的全局特征提取模块

对于点云的分类，本文不仅仅要关注局部特征，也要关注到全局特征，因为局部特征的内部提取主要是针对局部邻域的边缘形状特征，而很多种类其实在局部的特征是非常相似的，虽然多尺度的设置在网络的后半部分会有更大的感受野，但是往往由于相似性导致某些局部特征的区分度不是很大，从而可能会出现分类的错误，而全局特征进一步保证了不同尺度下特征的交互性，在不同尺度下有一个全

局的学习，来更好地对当前特征维度下的形状有着不同的区分。因此本小节引入自注意力机制，利用自注意力机制使所有点进行特征的交互，来学习其它点对当前点的影响，保证了每个点学习到所有点特征的长依赖性，其主要做法是用其它所有点和当前点的权重关系作为区分来提取所有点的特征，这样就实现了在全局特征维度下每个点结合有用点的信息而抑制无用点的信息。

如图 3-3 所示，同一个输入经过三个不同的线性层的输出分别作为自注意力模块的 Query、Key、Value，然后将 Query 和 Key 进行相乘操作，其结果经过 Softmax 操作作为权重，代表 Query 和 Key 的相似度，然后本小节接下来通过 Top_K 操作进行 Mask 处理，这里是因为在进行全局特征提取的时候，会引入部分噪点，所以全局下的点数是需要经过实验进一步确定的，经过 Top_K 操作后，每个点和其它所有点的关系权重会根据值的大小将一些点的权重值置为 0，之后再将处理后的权重和 Value 值相乘作为交互后的结果；最后通过残差结构，将该结果和原始的输入相加作为最后的输出。

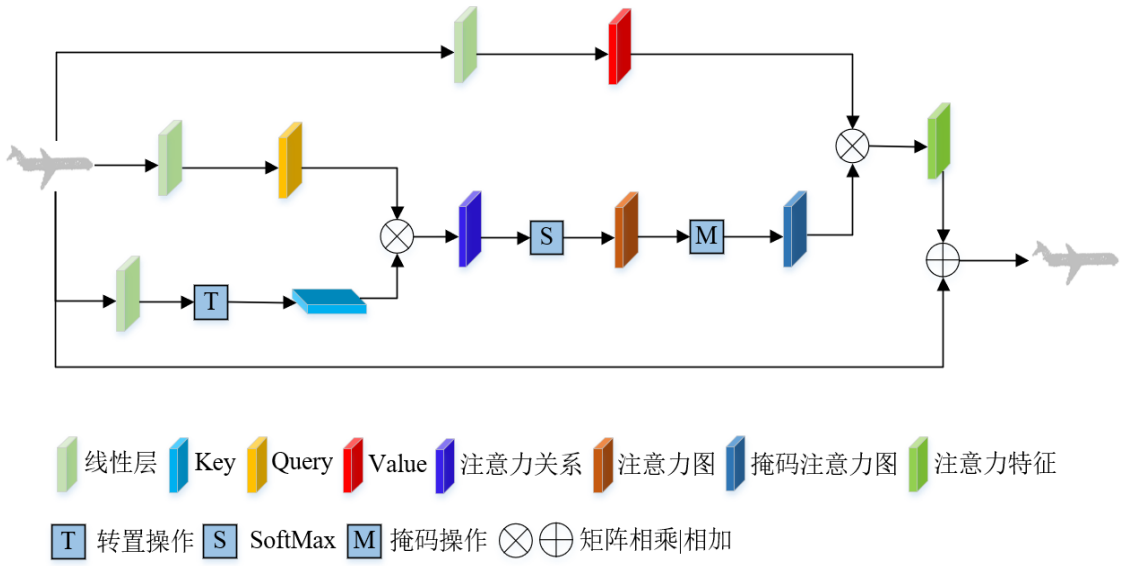

图 3-3 基于自注意力机制的全局特征提取模块

Figure 3-3 The Module of Global feature extraction based on Self-Attention Mechanism

3.5 多尺度特征融合模块

为了更好的利用每个尺度下学习到的特征，本小节借鉴了图片细粒度分类中的 MGE-CNN 网络[51]，提出了多尺度特征融合模块来进行不同尺度下的特征自适应融合，分别将学习到的参数作为不同尺度下特征的权重，然后分别相加作为最后的结果。如图 3-4 所示，在网络的设计中，特征融合模块主要在原始输入特征的基

础上进行了三组卷积层、ReLU 激活层、Batch Norm 归一化的自适应学习，然后通过最大池化操作以及 SoftMax 激活函数，针对主体网络的三个尺度分别学习三个权重参数作为多尺度融合模块的输出结果。

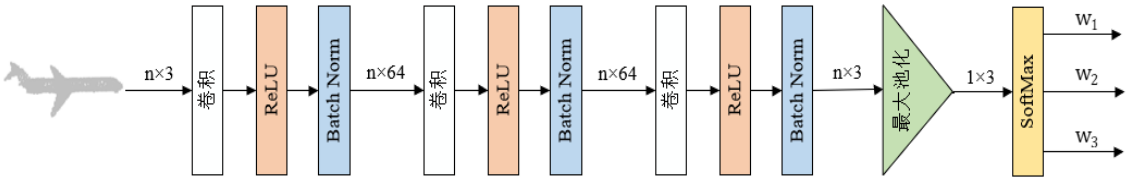

图 3-4 多尺度融合模块

Figure 3-4 The Multi-scale fusion module

3.6　损失函数

在点云分类领域的损失函数和在图像分类领域的损失函数几乎是一样的，通常为交叉熵损失函数。损失函数是指模型根据输入得到的输出和真值的差别，显而易见其值越小说明本文模型的效果越好，所以损失函数的合理设置是有助于提高分类精度的。通常网络模型由于损失函数的不同，性能也不尽相同，所以很多网络的改进都专注于损失函数的优化。因此，选择合适的损失函数对网络模型的训练能够起到出乎意料的作用。本小节将简单介绍在点云分类领域常用的两种损失函数：交叉熵损失函数、标签平滑损失函数。

（1）交叉熵损失函数

常见的交叉熵误差由于分类问题的数量不同，一般分为二分类和多分类，本文分别简称为 L_{o1} 损失函数和 L_{o2} 损失函数。如公式(3-1)所示， L_{o1} 损失函数计算的是模型最后的预测结果只有两种的情况。

$$L_{o1}=\frac{1}{B}\sum_{i=1}^{B}L_i=-\frac{1}{B}\sum_{i=1}^{B}(y_i*\log(p_i)+(1-y_i)*\log(1-p_i)) \tag{3-1}$$

其中，B 表示为输入点云的批次大小，y_i 表示样本 i 的标签，正类为 1，负类为 0，p_i 表示样本 i 预测为正类的概率。同理如公式(3-2)所示， L_{o2} 损失函数计算的是模型最后的预测结果为多种的情况。

$$L_{o2}=\frac{1}{B}\sum_{i=1}^{B}L_i=-\frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{M}y_{ic}\log(p_{ic}) \tag{3-2}$$

其中， B 表示为输入点云的批次大小， M 表示类别的数量， y_{ic} 表示样本 i 的真实类别等于 c 则取 1，否则取 0， p_{ic} 表示样本 i 预测为 c 类的概率。需要注意的是在

用了 Softmax 损失函数后，其最后多个结果 p_{ic} 的和是 1。因为本实验的数据集类别数量大于 2，所以为多分类，其实现首先需要通过网络模型的最后几个线性层得到每个类别相应的值，然后经过 Softmax 函数获得每个类别的概率输出，最后将其和真实类别的 one-hot 标签形式的值进行交叉熵损失函数的计算。

（2）标签平滑损失函数

标签平滑损失是对上面的交叉熵损失函数的进一步优化，公式(3-2)中的 p_{ic} 的计算如公式(3-3)所示：

$$p_{ic} = \frac{exp(z_{ic})}{\sum_{j=1}^{M} exp(z_{ij})} \tag{3-3}$$

其中，z_{ic} 为神经网络未经过 Softmax 的输出，在学习过程中要尽可能使各样本在正确类别上的输出概率为 1，从而需要对应的 z_{ic} 值为无穷大，这拉大了其与其它类别间的距离。但是这种情况下，当多分类任务本身的标签出现了问题，这对模型的伤害是非常大的，因为在训练过程中强行学习了一个非本类的样本，并且让其概率非常高，这会影响对后验概率的估计；并且有时候类与类之间并不是毫无关联，如果鼓励输出的概率间相差过大，这会导致一定程度上的过拟合，因此标签平滑损失的任务标签不再是 one-hot 标签，而是如公式(3-4)所示：

$$y_{ic} = \begin{cases} 1-\varepsilon, if\ i = c \\ \dfrac{\varepsilon}{M-1}, otherwise \end{cases} \tag{3-4}$$

其中 ε 是一个较小的常数，M 表示类别的数量，该公式说明 Softmax 函数中正确类别的概率优化目标不再为 1，同时公式(3-3)中的 z_{ic} 值不再是无穷大，而是一个具体的数值，这在一定程度上避免了过拟合，也缓解了错误标签带来的影响。

3.7 实验结果与分析

本章提出的基于注意力机制的多尺度点云分类网络分别在 ModelNet40 和 ScanObjectNN 数据集进行了模型的训练和测试。下面分别介绍下以上这两种数据集和相关的实验设置以及实验结果。

3.7.1 ModelNet40 数据集与实验设置

本章首先采用的 ModelNet40 数据集中提供的训练集来训练模型，并用该数据集中的测试集来测试模型的有效性。ModelNet40 包含了 40 个对象类别的 12311 个 CAD 模型，为了进行公平比较，本章采用官方数据的分类，将其中的 9843 个数据

用于训练，其余 2468 个数据作为测试，训练的时候采用和 PointNet 相同的策略，每个数据被统一均匀采样为 1024 个点以及采用了随机平移、随机异性缩放和随机输入衰减作为数据增强策略；在测试期间没有使用以上数据增强策略。

本章在 ModelNet40 数据集上对模型进行训练，其中模型采用的是深度学习框架 PyTorch。在深度学习模型训练中，不同的模型参数对实验结果有着很大的影响。本文通过大量实验并结合相关经验进行部分参数的调整，使模型达到最佳的效果。网络参数设置如下：（1）初始学习率为 0.0001，使用余弦退火计划调整每个时段的学习率；（2）优化方法为 Adam 求解器；（3）BatchSize 为 32；（4）训练 250 个 Epoch。其中 BatchSize 表示点云批处理的大小，Epoch 表示训练的次数。

3.7.2 ModelNet40 数据集实验结果分析

为了验证本章提出算法的有效性，本小节在 ModelNet40 数据集中的测试集上进行测试，并与点云分类相关的文章进行了对比，评价指标为类平均准确度（mean class Accuracy，mAcc) 以及总体准确度（Overall Accuracy，OA）。

表 3-1 不同算法在测试集上的质量评估结果

Table 3-1 Quality evaluation results of different algorithms on the test set

算法	输入	mAcc
3DshapeNets[9]	voxel	77.3
VoxNet[10]	voxel	83.0
PointNet[13]	point	86.2
A-SCN[52]	point	87.6
PointCNN[15]	point	88.1
PointWeb[56]	point	89.4
DGCNN[22]	point	90.2
PointMixer[49]	point	90.3
GCN3D [25]	point	90.3
Ours	point	**91.1**

表 3-1 给出了不同方法在 ModelNet40 测试集下 mAcc 的比较，其中 voxel 和 point 分别代表体素化的输入点云以及原始输入点云。从表 3-1 中可以看出，本章的模型取得了较好的结果，并且超过了最近的论文 PointMixer[49]、GCN3D[25]将近 1 个百分点。

表 3-2 不同算法在测试集上的质量评估结果

Table 3-2 Quality evaluation results of different algorithms on the test set

算法	输入	OA
3DshapeNets[9]	voxel	84.7
VoxNet[10]	voxel	85.9
MVCNN[5]	image	90.1
PointNet[13]	point	89.2
A-SCN[52]	point	90.0
Set Transformer[53]	point	90.4
PAT[54]	point	91.7
PointNet++[14]	point	91.9
SpecGCN[55]	point	92.1
PointCNN[15]	point	92.2
PointWeb[56]	point	92.3
SpiderCNN[16]	point	92.4
PointConv[19]	point	92.5
PointMixer[49]	point	92.7
LRConv [57]	point	92.8
DGCNN[22]	point	92.9
KPConv[17]	point	92.9
PointASNL[27]	point	92.9
GCN3D [25]	point	93.0
PointASNL[27]	point + nor	93.2
PCT[29]	point	93.2
AdaptiveConv[20]	point	93.4
Ours	point	**93.7**

表 3-2 给出了不同方法在 ModelNet40 测试集下 OA 的比较，其中 point + nor 代表输入不仅仅是原始点云，还包含了法向量信息。从表 3-2 中可以看出，本章的模型同样取得了较好的结果，并且超过了最近的论文 LRConv[57] 、GCN3D[25] 、AdaptiveConv[20]以及注意力机制相关的论文 PCT[29]，通过表 3-1 以及表 3-2 的结果综合比较，可以说明本章提出的基于注意力机制的分类算法整体来说具有有效性。

本章提出了基于空间特征变换的局部特征提取模块、基于自注意力机制的全

局特征提取模块以及特征融合模块，每个模块对网络模型的性能提升都起到了一定的作用。为了验证不同模块对网络性能的影响，本小节设置了相应的消融实验，分别将局部特征提取模块、全局特征提取模块以及特征融合模块命名为 A、B、C，进行每个模块以及多个模块的实验，结果如表 3-3 所示，可以发现 A、B、C 每个模块单独作用时，在 OA 评价指标上分别达到了 92.8%、93.0%、92.8%的准确度，在 mAcc 评价指标上分别达到了 89.8%、89.7%、89.9%的准确度，而当两个模块组合时，其效果都会在单独每个模块的基础上进一步的提高，如 A、B 模块共同作用时，其 OA 评价指标达到了 93.2%，其性能分别超过了 A、B 模块单独作用的效果，最后当三个模块共同作用的时候，其性能在 OA 以及 mAcc 评价指标上分别达到了 93.7%和 91.0%的最好效果。

表 3-3 不同模块性能测试

Table 3-3 Performance testing of different modules

A	B	C	mAcc	OA
√	✗	✗	89.8	92.8
✗	√	✗	89.7	93.0
✗	✗	√	89.9	92.8
√	√	✗	90.2	93.2
√	✗	√	90.1	93.1
✗	√	√	90.0	93.4
√	√	√	**91.0**	**93.7**

另外，对于网络的多尺度分支，本文也进行了相应的对比实验，如表 3-4，可以看出不同的分支个数的影响也是巨大的，因为点云分类本身来说，其数据集相对简单，并且输入点数相对较少，所以前期对于分支个数的增加，网络性能有一定的提高，但是当分支个数太多的时候，后面下采样点数会减少至几十个左右，对于网络性能的提升没有了作用，并且在特征融合的时候会降低网络的性能。

表 3-4 不同分支个数在测试集上的质量评估结果

Table 3-4 Quality evaluation results of different number of branches on the test set

分支个数	mAcc	OA
2	90.2	93.1
3	**91.1**	**93.7**
4	89.7	92.9

在全局特征提取模块中，本章提出了 Mask 操作，为了说明有效性，在这里，针对 Mask 操作中 k 的选择进行了对比实验，如表 3-5 所示，当 k 的值为 100 的时

候，结果最好。需要注意的是在三个分支下，点数是逐渐减少的，分别是 512、256、128，所以 k 的选择一般不超过 128。首先针对三个分支，当设置 k 的值相同时，分别取了 50、100、128 进行了对比实验；另外针对三个分支，当设置 k 的值不同时，分别取总点数的 50%、80%、100%进行对比实验。从实验结果可以分析得知，在点数过多的时候，k 选择越小越好，而点数比较少的时候，k 选择越大越好。

表 3-5 Mask 操作中不同的 k 在测试集上的质量评估结果

Table 3-5 Quality evaluation results of different k in the Mask Operation on the test set

k	mAcc	OA
50	90.3	93.0
100	**91.1**	**93.7**
128	90.2	93.5
50%	90.1	93.3
80%	90.0	93.2
100%	89.9	93.1

在局部特征提取模块中，通常在邻域的构造上有两种方法，分别是 KNN 以及 Ball Query，为了验证两者在本章算法中的性能，如表 3-6 所示，本章分别进行了基于 KNN 和基于 Ball Query 的邻域搜索，并使用了不同的 k 值和搜索半径 r 值，可以看到相比于 Ball Query，KNN 有着更好的性能，因为 Ball Query 的查找范围会局限在一个半径内，当指定半径内的邻近点不足的时候，会降低获取局部特征的能力，所以泛化能力相对不如 KNN；另外从表中可以发现适当的提高 k 值，有助于提高性能，同理，因为它捕获了更丰富的局部特征。

表 3-6 邻域查询方法在测试集上的质量评估结果

Table 3-6 Quality evaluation results of neighborhood query methods on the test set

评价指标	KNN(k=16)	KNN (k=32)	radius(r=0.15)	radius(r=0.25)
OA	92.0	**93.7**	93.2	93.1
mAcc	88.9	**91.1**	90.0	90.2

在网络对于局部特征的提取中，本小节分别选择了不同的对称函数来验证算法的性能，实验结果如表 3-7 所示，最大池化函数相比于平均池化函数、求和函数以及最大池化函数和平均池化函数的结合更能学习到利于网络正确分类的特征。

表 3-7 不同的局部聚合函数在测试集上的质量评估结果

Table 3-7 Quality evaluation results of different local aggregate function on the test set

评价指标	Max	Mean	Sum	Max + Mean
OA	**93.7**	92.5	93.3	92.8
mAcc	**91.1**	89.2	90.3	89.5

同理，在全局特征提取模块中，本小节同样分别选择了不同的对称函数来验证算法的性能，实验结果如表 3-8 所示，最大池化函数的性能还是比较优先于其它对称函数。

表 3-8 不同的全局聚合函数在测试集上的质量评估结果

Table 3-8 Quality evaluation results of different global aggregate function on the test set

评价指标	Max	Mean	Sum	Max + Mean
OA	**93.7**	92.9	92.9	93.6
mAcc	**91.1**	89.3	89.8	90.5

3.7.3 ScanObjectNN 数据集与实验设置

为了进一步验证算法的有效性，本章在 ScanObjectNN 真实数据集上进行了补充实验。ScanObjectNN 数据集不同于 ModelNet40 数据集，每个样本由坐标归一化后的 2048 个点表示，并且官方数据集提供了含旋转、缩放、随机变化等各种类型的数据集，为了进行公平比较，本章采用最有挑战性的数据集 PB_T50_RS，其中 PB 作为前缀，表示对原始点云数据集进行了扰动，T50 表示将原始点云本身的边界框分别沿着三个坐标系向质心随机移动，直到其最初大小的 50%，后缀 R 和 S 分别表示旋转和缩放操作，训练的时候采用和 PointMLP[58]相同的策略，每个数据被统一采样为 1024 个点，为了进一步提高网络的鲁棒性，采用了随机平移、随机打乱输入顺序作为数据增强策略；注意的是，在测试期间没有使用以上数据增强策略。

本章在 ScanObjectNN 数据集上对模型进行训练，其中模型采用的是深度学习框架 PyTorch。在深度学习模型训练中，不同的模型参数对实验结果有着很大的影响。本文通过大量实验并结合相关经验进行部分参数的调整，使模型达到最佳的效果。网络参数设置如下：（1）初始学习率为 0.001，使用余弦退火计划调整每个时段的学习率；（2）优化方法为 Adam 求解器；（3）BatchSize 为 24；（4）训练 200 个 Epoch。其中 BatchSize 表示点云批处理的大小，Epoch 表示训练的次数。

3.7.4 ScanObjectNN 数据集实验结果分析

本小节在 ScanObjectNN 数据集中的测试集上进行测试，并与点云分类相关的

文章进行了对比，来验证本章提出算法在真实数据集 ScanObjectNN 上的有效性，其中评价指标同样为类平均准确度（mean class Accuracy，mAcc）以及总体准确度（Overall Accuracy，OA）。

表 3-9 给出了不同方法在 ScanObjectNN 测试集下 mAcc 和 OA 的比较结果。从表 3-9 中可以看出，本章的模型在评价指标 mAcc 和 OA 上都取得了比较好的结果，并且在 OA 结果上，分别超过了最近的论文 DRNet[60]、GBNet[61]将近 0.7 和 0.5 个百分点，由此说明本章提出的注意力机制相关算法不仅仅在合成数据集 ModelNet40 有着较好的效果，而且可以很好的适应真实数据集 ScanObjectNN 的各种挑战，同样达到了一个比较好的结果，所以整体来说本章提出的基于注意力机制的多尺度点云分类网络具有有效性。

表 3-9 不同算法在测试集上的质量评估结果

Table 3-9 Quality evaluation results of different algorithms on the test set

算法	输入	mAcc	OA
3DmFV[59]	point	58.1	63.0
PointNet[13]	point	63.4	68.2
SpiderCNN[16]	point	69.8	73.7
PointConv[19]	point	74.1	77.4
PointNet++[14]	point	75.4	77.9
DGCNN[22]	point	73.6	78.1
PointCNN[15]	point	75.1	78.5
BGA-DGCNN[45]	point	75.7	79.7
BGA-PN++[45]	point	77.5	80.2
DRNet[60]	point	78.0	80.3
GBNet[61]	point	77.8	80.5
SimpleView[62]	point	—	80.5±0.3
Ours	point	**78.3**	**81.0**

3.8 本章小结

基于深度学习的点云分类算法构造的神经网络模型常常由于局部信息获取能力较弱以及缺乏全局信息等限制，在分类性能上往往达不到满意的效果。本章从这

两点出发，重新设计了一种基于空间特征变换的局部特征提取模块来丰富局部信息，并且在全局方面提出了基于自注意力机制的特征提取模块，利用 Mask 操作来优化注意力机制，达到了局部和全局信息的融合，使网络能够对远距离的点进行关系建模，保证了网络的全局识别能力。为了区分不同尺度下特征的重要性，本章进一步提出了特征融合模块进行自适应学习来达到最好的效果，相比于大多数的点云分类相关算法，本章提出的基于注意力机制的多尺度点云分类网络在 ModelNet40 合成数据集以及 ScanObjectNN 真实数据集上都取得了比较好的效果，为了证明每个模块的作用，本章分别做了对应每个模块的消融实验，并且针对网络结构中的分支个数、对称函数以及 Mask 操作 k 的取值等分别进行了对比实验，所有的结果充分证明了该算法具有一定的有效性。

4 基于注意力机制的多尺度点云语义分割算法

4.1 引言

点云语义分割同样作为计算机视觉领域中比较基础的任务，对于三维点云的各种应用有着一定的促进作用，其中不同于点云分类的是点云语义分割需要关注到每个点的信息，从而识别出每个点的特征作为分割结果。点云语义分割可以用于自动驾驶领域，自动驾驶车辆通过实时地检测周围的环境，来确保最大程度化的安全驾驶；在机器人领域中，通过对周围场景的语义分割，可以利用机器人完成一些简单的操作，如不同物体的抓取，摆放等，在某些危险的环境下，其能降低一些不可控的风险。

点云的语义分割是非常类似于点云分类的，主要体现在局部邻域的处理，很多算法能够很好的兼顾分类和分割两个任务，但是不同于分类的是全局信息的处理，分类是整体的，所以每个点学习其它所有点的信息是有意义的，而分割中每个点代表的种类是不同的，并且每个物体的位置是不一样的，所以分类中使用的长依赖关系在语义分割上则显得不合适，并且对于整体的分割效果来说甚至会有影响，所以针对这个方面，本章主要关注如何高效的提取局部特征，同时为了更进一步区分每一个点，本章在全局方面则利用了通道注意力机制来提高每个点的独特性。

本章提出了基于注意力机制的多尺度点云语义分割算法，所构造的网络类似于 U-Net 网络结构，分为下采样层和上采样层，其中下采样层利用最远点采样（Farthest Point Sampling，FPS）不断进行下采样，在局部处理中结合了位置信息以及语义信息来丰富局部特征，为了进一步提高表达能力，在全局方面利用通道注意力机制来提取关键的特征部分；上采样层则借鉴 PointNet++中的处理方法利用最近邻插值法来恢复原始点信息，最后进行每个点的语义分割处理。经过调参和大量实验对比，本章所提算法在 S3DIS 数据集上直接处理点云，达到了较好的语义分割效果。为了验证本章所提的算法的通用性，进一步在 ShapeNetPart 补充数据集上进行训练和测试，实验表明，本章所提算法在该数据集上同样取得了优异的结果。

4.2 整体网络结构

为了充分地突出点云中每个点的特性，本章设计的基于注意力机制的多尺度点云语义分割算法的网络框架类似于U-Net网络结构，主要分为下采样层和上采样层。

如图4-1所示，输入的点云经过连续的四个下采样层，点数逐渐减少，但是每个点的感受野逐渐增大，其包含的特征信息也逐渐全面；然后再经过四个上采样层，利用插值法不断恢复点的个数和特征；最后经过连续的几个线性层进行特征变换作为最后的输出。

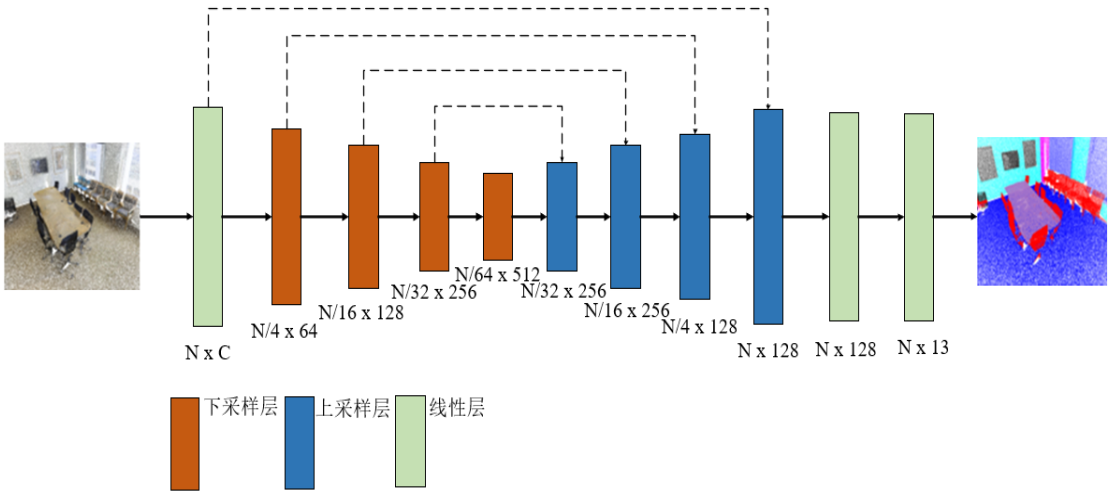

图 4-1 基于注意力机制的多尺度点云语义分割网络

Figure 4-1 Multi-scale point cloud semantic segmentation network based on attention mechanism

下采样层的体系结构如图 4-2 所示，下采样层其输入分别是位置信息 x、特征信息 f_1，首先通过最远点采样后得到相应的下采样后每个点的位置信息以及特征信息，然后利用 KNN 进行局部邻域的分组，之后分别经过本章提出的位置合成注意力机制模块（Position Synthesize Attention Block，PSAB）以及语义自注意力机制模块（Semantic Self-Attention Block，SSAB）丰富局部特征，下一步通过最大池化操作以及本章提出的全局通道注意力提取模块（Global Channel Attention Block，GCAB）得到最后的输出特征 f_2，同时将最远点采样后点的坐标 y 作为输出位置。

上采样层的体系结构如图 4-3 所示，上采样层其输入分别是当前层的位置输出信息 x_1、特征输出信息 f_1、上一层的位置输出信息 x_2 和特征输出信息 f_2，其中 $x_1 = 4x_2$，然后通过插值模块，其输出特征主要是从位置信息 x_2 中找到位置信息 x_1 最近的 k 个点，然后其对应特征的加权平均后的值作为最后的输出特征 $f_2^{`}$，并同时输出位置信息 x_1。接下来，本章分别一一介绍所提出的每个模块的详细结构。

4.3 基于合成注意力机制的位置信息提取模块

传统的自注意力机制往往是 token 对 token 的 Attention，在点云相关任务中，

本文可以将 token 理解为一个点，所以点云中自注意力机制是学习点与点之间的关系，主要的参数学习为一个$k\times k$的权重矩阵 B，最近的 Synthesizer[63]网络证明了 token 对 token 的交互中学习注意力的权重并不是那么重要，Synthesizer 中用合成的权重矩阵 B 在机器翻译、自动摘要、对话生成等任务上同样达到了基于自注意力机制网络的效果，某些任务甚至比自注意力机制的效果更好，本文将其提出的合成注意力机制引入到点云的语义分割中，在保证性能的同时，进一步的提高计算效率。Synthesizer 中的合成注意力机制主要分为两种，一种是 Dense 形式，另外一种是 Random 形式，接下来，本小节分别介绍下两者的原理，并说明如何将其应用到点云语义分割任务中。

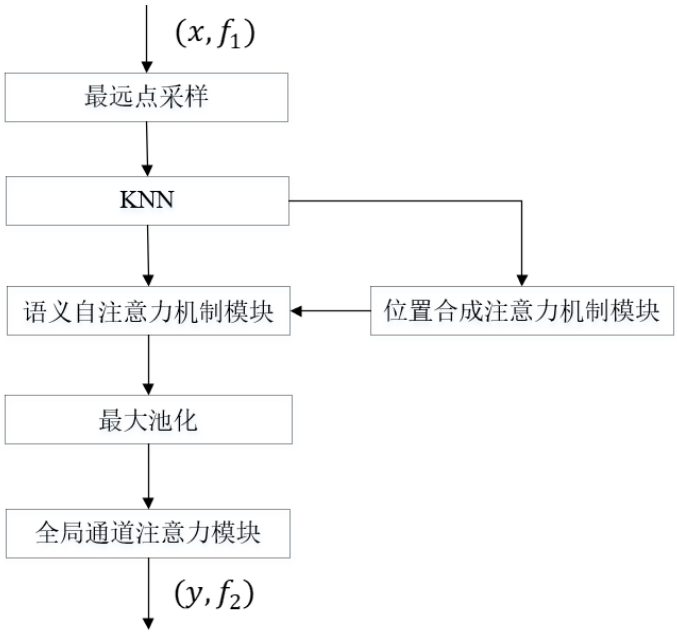

图 4-2 下采样层网络结构

Figure 4-2 Downsampling layer network structure

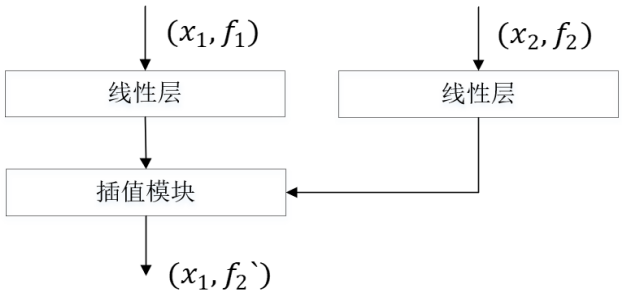

图 4-3 上采样层网络结构

Figure 4-3 Upsampling layer network structure

4.3.1　Dense 合成注意力机制

在 Dense 合成注意力机制中，权重矩阵 B 的生成，不再需要 Query 和 Key 的点积操作，而是直接通过简单的线性变换矩阵生成，假设输入的邻域点云集合为 $X \in R^{k \times D}$，其中 k 代表点云的个数，一般 $k=32$，D 代表每个点的特征维度，那么所需要的权重矩阵 B 的大小应该为 $k \times k$，所以注意力权重 B 的计算如公式(4-1)所示：

$$B = Relu(XW_1 + b_1)W_2 + b_2 \tag{4-1}$$

其中 $W_1 \in R^{D \times D_{mid}}$ 和 $W_2 \in R^{D_{mid} \times k}$ 为可学习的权重参数，b_1 和 b_2 为偏执。其点云的语义分割任务中的 Dense 合成注意力机制如图 4-4 所示，原始输入点云的个数为 n，经过最远点采样后，点数下降为 m，一般 m 取 $n/4$，然后经过 KNN 分组，得到每个点的邻域分布，网络上层分支经过 MLP 操作得到注意力权重矩阵 $B \in R^{k \times k}$，网络下层分支经过 MLP 操作设置最终输出维度的大小 d，接下来两个分支进行矩阵相乘实现局部邻域的点对之间交互，最后经过最大池化操作得到每个点的最终输出特征。

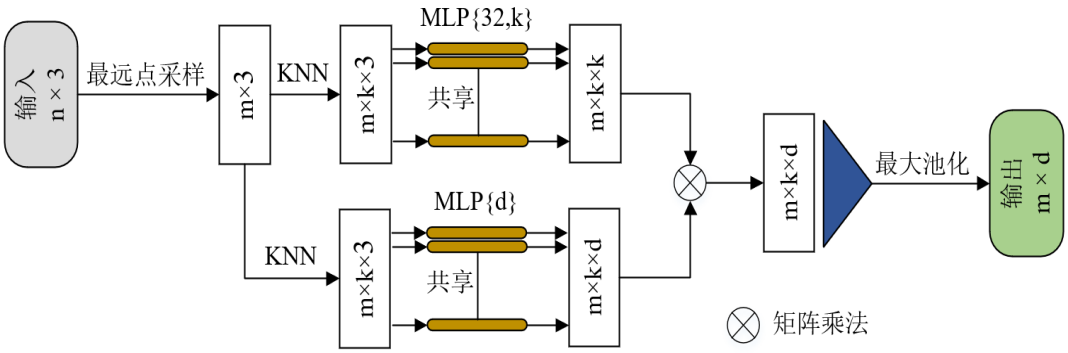

图 4-4 Dense 合成注意力机制

Figure 4-4 The Dense Synthesizes Attention Mechanism

为了进一步的降低参数的数量，Synthesizer 论文中针对权重矩阵 B 的生成进行了低秩分解，首先分别生成两个大小为 $k \times a$ 、 $k \times b$ 的矩阵 B_1 和 B_2，其中 $ab = k = 32$，然后针对矩阵 B_1 和矩阵 B_2 分别重复 b 次、a 次，最后将扩展后的 B_1 和 B_2 矩阵进行逐位相乘，从而实现了原有的参数量从 $k \times k$ 减少到 $k \times (a+b)$，将其应用到点云语义分割任务中的实现如图 4-5 所示。

4.3.2　Random 合成注意力机制

在 Random 合成注意力机制中，权重矩阵 B 的生成，不仅不需要 Query 和 Key

的点积操作，而且不依赖点云本身的输入，而是直接随机生成权重矩阵，所以将其应用到点云语义分割任务的 Random 合成注意力机制如图 4-6 所示，与 Dense 注意力机制相比较，可以看到在网络的上层，注意力权重矩阵的生成是直接随机生成的，和输入是没有关系的，这里的箭头只是为了方便展示，另外，在随机生成的时候，代码中可以设置其是否随着训练而不断地改变；网络下层分支同样经过 MLP 操作设置最终输出维度的大小 d。

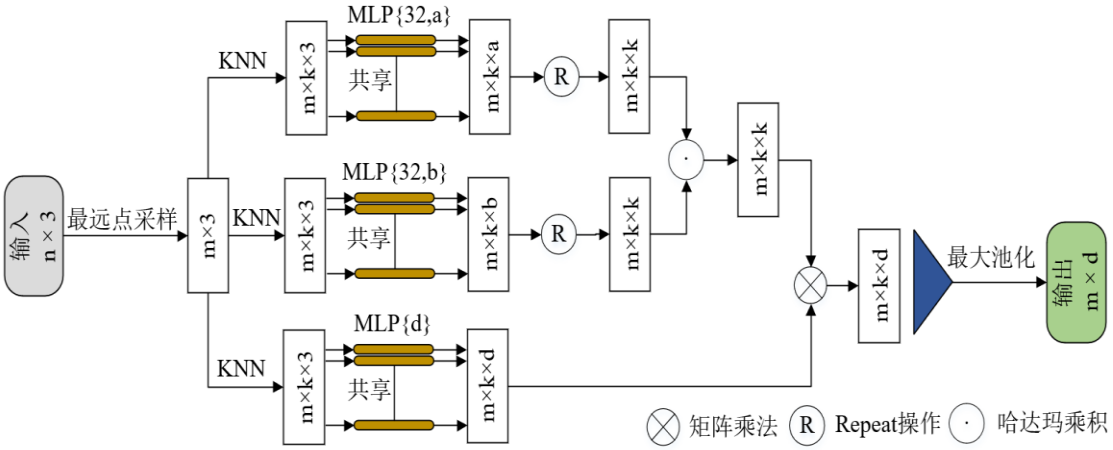

图 4-5 降秩 Dense 合成注意力机制

Figure 4-5 The Factorized Dense Synthesizes Attention Mechanism

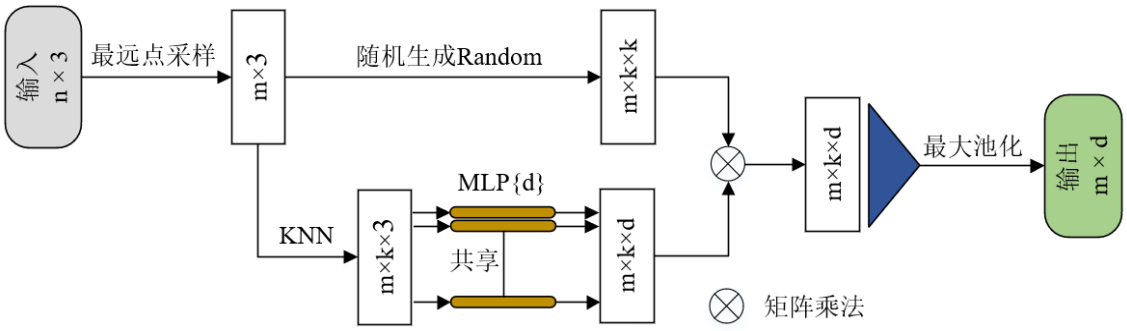

图 4-6 Random 合成注意力机制

Figure 4-6 The Random Synthesizes Attention Mechanism

同理，在 Random 合成注意力机制中，Synthesizer 论文同样针对随机生成的权重矩阵进行了进一步的参数减少，主要把原来的权重矩阵 $B \in R^{k \times k}$ 分成两个矩阵 $B_1 \in R^{k \times a}$ 和 $B_2 \in R^{k \times a}$，然后矩阵 B_1 进行转置后和矩阵 B_2 相乘，这样参数量就从原来的 $k \times k$ 减少到 $k \times 2a$，一般取 $2a < k$，将其应用到点云语义分割任务中的实现如图 4-7 所示。

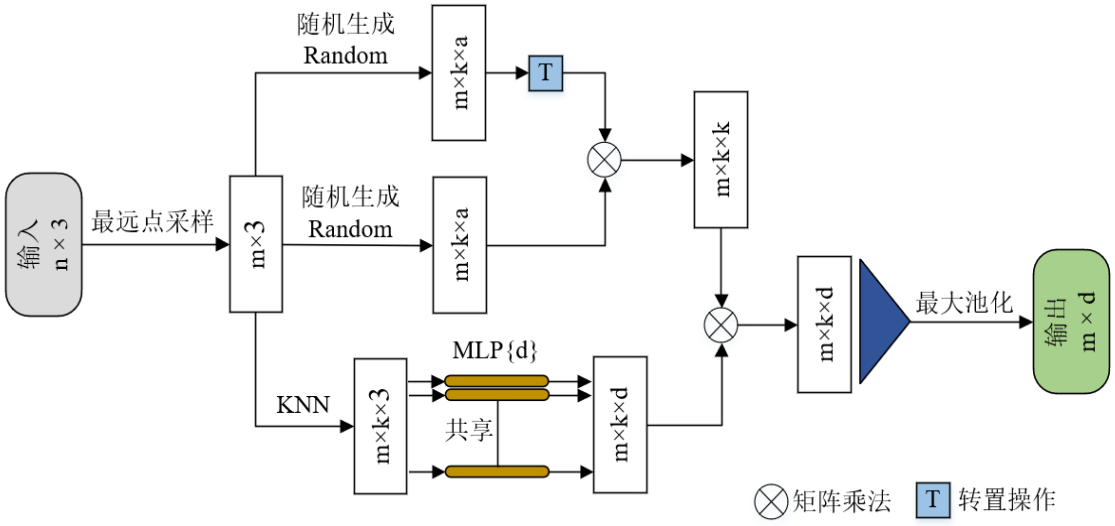

图 4-7 降秩 Random 合成注意力机制

Figure 4-7 The Factorized Random Synthesizes Attention Mechanism

本文在上面一共提出了四种合成注意力机制的方法，再加上标准的自注意力机制，一共有 5 种权重矩阵 B 的生成方法，另外，本文同样可以借鉴 Synthesizer 论文中的混合模式，将不同的权重矩阵 B 进行混合起来作为最后的权重矩阵，如公式(4-2)所示，其中 α_i 为可学习的参数，相加为 1。

$$B = \sum_{i=1}^{N} \alpha_i B_i \tag{4-2}$$

4.3.3 基于合成注意力机制的局部位置信息提取模块

在进行局部特征提取的时候，为了丰富局部特征，本小节引入了位置信息，提出了位置合成注意力机制模块（Position Synthesize Attention Block，PSAB），其网络示意图如图 4-8 所示，注意的是合成注意力（Synthetic Attention，SA）机制中没有利用到最大池化操作，所以输入输出的大小一样。

该模块的实现过程首先是根据 KNN 算法进行局部分组，这里的 KNN 不同于 DGCNN 的相似特征分组，而是基于每个点的位置信息处理的，因为相似特征分组对于每个点语义分割的特征处理是相似的，所以本文应该关注位置相近点的特征；然后通过每个局部邻域的点的相对位置以及距离来丰富每个点的位置信息，使其在每个局部邻域中所有点的区分性增加，所以在局部邻域内的其关系描述符包括中心点的坐标 x_i 、邻域点和中心点的相对坐标 $\Delta x_{i,jk}$ 以及邻域和中心点的欧式距离 $d(x_i, x_{jk})$，最终经过 Concat 操作形成 7 维的输入描述符；最后将丰富后的位置信息输入到合成注意力机制中，生成每个邻域点的相应特征 h_{jk} 。这里本章舍弃了用

自注意力机制来进行位置信息交互，而是借鉴了 Synthesizer 网络，用合成的注意力矩阵来代替通过点积生成的矩阵，使得局部邻域内所有点的位置信息进行特征自适应的提取，来最终提升网络的学习能力。

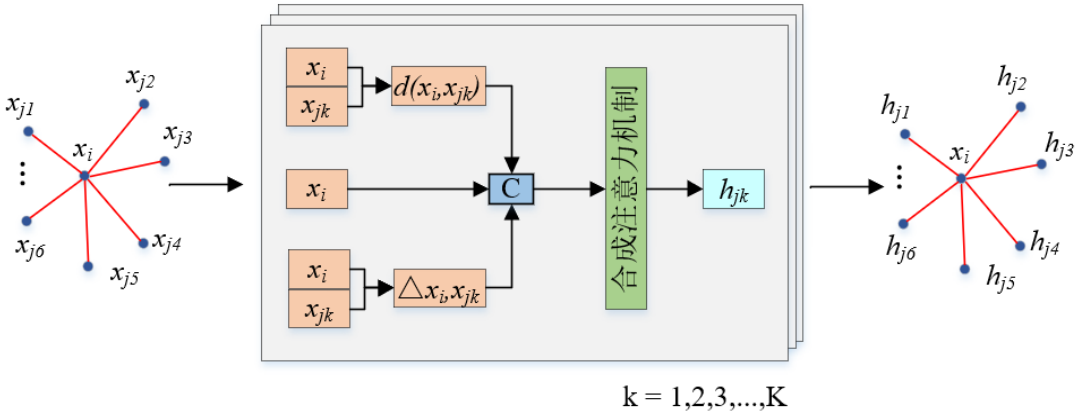

图 4-8 位置合成注意力机制模块

Figure 4-8 Position Synthesize Attention Block

4.4 基于自注意力机制的语义特征提取模块

基于合成注意力的信息提取模块，学习后的特征主要用于增强每个点的语义特征，所以在位置信息提取模块中，针对点对之间权重矩阵 B，本文选择的是合成注意力机制，因为其不过于依赖点与点之间的关系，而是突出每个点本身的特征。通过位置信息增强原有的语义特征后，本章接下来应该专注于局部邻域内所有点的特征交互，不同于点云分类任务中基于自注意力机制的全局特征提取模块，本小节提出的语义自注意力机制模块（Semantic Self-Attention Block，SSAB）是针对局部邻域的，因为点云语义分割涉及到每个点的语义信息，其主要受其局部邻域内其它点的影响，其网络示意图如图 4-9 所示，该模块的实现如下：

首先通过 KNN 进行局部分组，然后每个局部邻域的位置信息通过位置合成注意力机制模块（Position Synthesize Attention Block，PSAB）从位置关系中获取位置相关的特征，将其输出特征、中心点的特征 f_i 以及邻域点和中心点的相对特征 $\Delta f_{i,jk}$ 三者进行 Concat 操作，其结果作为语义自注意力机制模块（Semantic Self-Attention Block，SSAB）的关系描述符；之后将丰富后的特征信息输入到自注意力机制网络中，生成每个邻域点的相应特征 h_{jk}；最后对每一个局部邻域所有点的特征进行最大池化层的处理，获取中心点增强后的特征。需要注意的是自注意力机制的实现原理和点云分类中的基于自注意力机制的全局提取模块类似，如第三章的图 3-3 所

示，唯一不同的是，在语义分割任务中，自注意力机制主要用于局部邻域，其中每个局部邻域的大小一般为 32，然后 32 个点互相交互来获取局部邻域点的所有语义信息，从而不断提高每个点的感受野，进而有效的进行点云的语义分割。

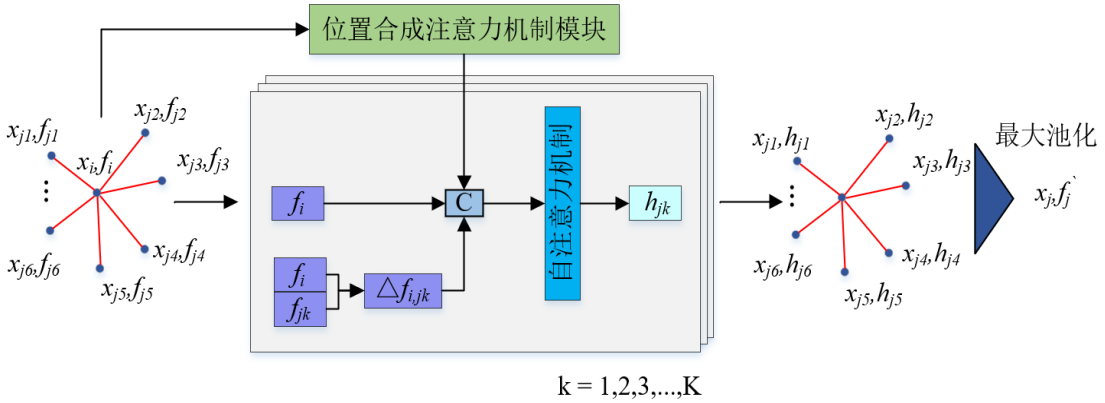

图 4-9 语义自注意力机制模块

Figure 4-9 Semantic Self-Attention Block

4.5 基于通道注意力机制的全局特征提取模块

为了进一步丰富点云的语义特征，本小节在全局特征上引入了通道注意力机制，主要针对每个点的语义特征进行自适应选择，从之前的分析中，在点云语义分割任务中，自注意力机制不适合用于提取全局特征，因为全局范围内点的特征交互对于点云的分割作用是有一定的负面影响的，所以本小节单独从每个点出发，进行每个点特征维度下的自适应学习，对不同维度的值进行注意力机制的筛选，进一步优化每个点的特征。

本章借鉴的是 CBAM[64]论文的思想，提出了全局通道注意力机制模块（Global Channel Attention Block，GCAB），如图 4-10 所示，该模块舍弃了空间注意力机制，仅仅使用了通道注意力机制，因为点云的不规则性，以及全局点之间的模糊性，空间注意力机制会涉及到全局范围内点之间的交互，是不适合点云语义分割任务的。

全局通道注意力机制模块的处理过程如下，首先输入的点云同时经过最大池化和平均池化，分别对于所有点保留每个维度下的最大值以及平均值，然后分别经过 3 组卷积、ReLU 激活、Batch Norm 操作，将输出结果对应相加；之后将相加后的结果经过 Sigmoid 激活函数作为通道注意力权重，这里的权重可以理解为每个维度的比重值；最后将通道注意力权重和原始输入对应相乘，将其结果和输入再次相加作为最后的输出。

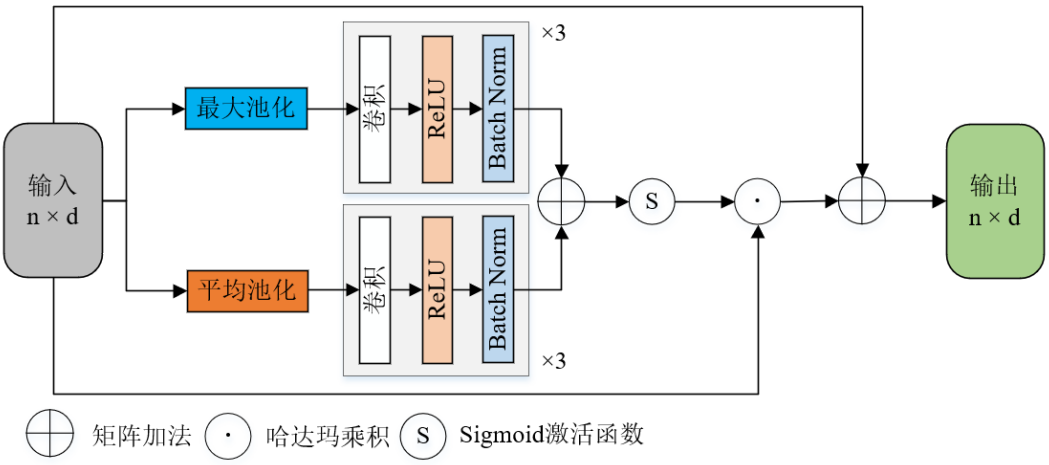

图 4-10 基于通道注意力机制的全局特征提取模块

Figure 4-10 The module of global feature extraction based on Channel Attention Mechanism

4.6 损失函数

点云语义分割常见的损失函数和点云分类任务中的损失函数是一样的，比较常用的是交叉熵损失函数，因为点云语义分割的主要任务是针对每个点来进行分类，所以可以将其每个点看成分类任务，所有点的损失函数的计算是一样的，网络整体的损失函数值则为每个点损失值的和。

如式(4-3)所示，语义分割的损失函数计算是输入点云批次下每个点对应的损失值的和。

$$L=\frac{1}{B}\frac{1}{N}\sum_{i=1}^{N}L_i=-\frac{1}{B}\frac{1}{N}\sum_{i=1}^{B\times N}\sum_{c=1}^{M}y_{ic}log(p_{ic}) \tag{4-3}$$

其中，B 表示为输入的点云的批次大小，M 表示类别的数量，N 表示为输入的每个点云的点数，y_{ic} 表示样本 i 的真实类别等于 c 则取 1，否则取 0，p_{ic} 表示样本 i 预测为 c 类的概率。需要注意的是在用了 Softmax 损失函数后，其最后多个结果 p_{ic} 的和是 1。因为本实验的数据集类别数量大于 2，所以为多分类，其实现首先需要通过网络模型的最后几个线性层得到每个点对应每个类别相应的值，然后经过 Softmax 函数获得每个类别的概率输出，之后将其和对应点真实类别的 one-hot 标签形式的值进行交叉熵损失函数的计算，最后对所有点的损失值取平均。

4.7 实验结果与分析

本章提出的基于注意力机制的多尺度点云语义分割网络分别在 S3DIS 和

ShapeNetPart 数据集进行了模型的训练和测试。下面分别介绍下以上这两种数据集和相关的实验设置以及实验结果。

4.7.1　S3DIS 数据集与实验设置

本章采用的是 S3DIS 数据集中提供的训练集来训练模型，并用该数据集中的测试集来测试模型的有效性。S3DIS 作为大规模场景分割数据集，一共包含 2.73 亿个点，每个点分别属于 13 个类别中的一个语义标签，如桌子、墙、椅子等等。对于数据的准备，一般常常分为两类，其一是将点均匀的采集到面积为$1m\times1m$的块中，其中高度不设限制；其二是使用网格下采样方法进行数据预处理，包含的点数比较多，虽然这会带来更规则的数据结构和更多的上下文信息，但它在训练期间内存使用率很高，从而训练所需要的显存也要足够大。本章数据集的训练采用和PointNet++相同的策略，使用第一种数据处理方法，为了公平比较，每个数据被统一均匀采样为 4096 个点以及采用了随机平移、随机异性缩放作为数据增强策略；在测试期间没有使用以上数据增强策略。

本章在 S3DIS 数据集上对模型进行训练，其中模型采用的是深度学习框架 PyTorch。在深度学习模型训练中，不同的模型参数对实验结果有着很大的影响。本文通过大量实验并结合相关经验进行部分参数的调整，使模型达到最佳的效果。网络参数设置如下：（1）初始学习率为 0.05，每 20 个 Epoch 等间隔调整学习率，调整倍数为 0.5；（2）优化方法为 Adam 求解器；（3）BatchSize 为 16；（4）训练 100 个 Epoch。其中 BatchSize 表示点云批处理的大小，Epoch 表示训练的次数。

4.7.2　S3DIS 数据集实验结果分析

为了验证本章提出的算法的有效性，本小节在 SIDIS 数据集中的测试集上进行测试，并与点云语义分割相关的文章进行了对比，评价指标为类平均准确度（mean class Accuracy，mAcc）、总体准确度（Overall Accuracy，OA）以及平均交并比（mean Intersection over Union，mIoU）。

表 4-1 给出了不同方法在 S3DIS 的 Area5 测试集下 mAcc、OA 和 mIoU 的比较。从表 4-1 中可以看出，本章提出的模型在这三种指标上都取得了较好的结果，并且在 mIoU 指标上超过了最近的 SSA-Pointnet++[68]、PCT[29]以及 SCF-Net[70]。由此说明本章提出的基于注意力机制的算法在点云的语义分割任务上，整体来说具有有效性。

表 4-1 S3DIS 数据集上的语义分割结果，在 Area5 上进行测试

Table 4-1 Semantic segmentation results on the S3DIS dataset, evaluated on Area5

Method	OA	mAcc	mIoU
PointNet[13]	—	49.0	41.1
SegCloud[65]	—	57.4	48.9
TangentConv[66]	—	62.2	52.6
PointCNN[15]	85.9	63.9	57.3
PointNet++[14]	—	—	57.3
SPG[40]	86.4	66.5	58.0
PCNN[67]	—	67.0	58.3
PAT[54]	—	70.8	60.1
PointWeb[56]	87.0	66.6	60.3
SSA-Pointnet++[68]	85.1	—	61.1
PCT[29]	—	67.6	61.3
HPEIN[69]	87.2	68.3	61.9
PointASNL[27]	87.7	68.5	62.6
SCF-Net[70]	—	—	63.8
Ours	**89.1**	**71.9**	**65.2**

表 4-2 给出了不同方法分别在 S3DIS 的 Area1、Area2、Area3、Area4、Area5、Area6 数据集下进行测试取得的 mAcc、OA 和 mIoU 的平均结果的比较。从表 4-2 中可以看出，本章提出的模型在这三种指标上都取得了较好的结果，并且超过了最近的论文 PAConv[74]。

本章提出了位置合成注意力机制模块（Position Synthesize Attention Block，PSAB）、语义自注意力机制模块（Semantic Self-Attention Block，SSAB）以及全局通道注意力机制模块（Global Channel Attention Block，GCAB），每个模块对网络模型的性能提升都起到了一定的作用。为了验证不同模块对网络性能的影响，本小节设置了相应的消融实验，结果如表 4-3 所示，可以发现每个模块单独作用时，在 mIoU 评价指标上分别达到了 61.6%、62.6%、62.2%的结果，在 mAcc 评价指标上分别达到了 68.6%、69.4%、69.1%的准确度，在 OA 评价指标上分别达到了 87.2%、87.5%、87.9%的准确度，而当其中两个模块组合时，其效果都会在单独每个模块的基础上进一步的提高，如 PSAB、SSAB 模块共同作用时，其 OA 评价指标达到了 89.0%，其性能分别超过了 A、B 模块单独作用的效果，最后当三个模块共同作用的时候，其性能在 mIoU、OA 以及 mAcc 评价指标上分别达到了 65.2%、89.1%和

71.9%的最好效果。

表 4-2 基于 S3DIS 数据集的语义分割结果，采用 6 折交叉验证进行测试

Table 4-2 Semantic segmentation results on the S3DIS dataset, evaluated with 6-fold cross validation

Method	OA	mAcc	mIoU
PointNet[13]	78.5	66.2	47.6
RSNet[71]	—	66.5	56.5
SPG[40]	85.5	73.0	62.1
PAT[54]	—	76.5	64.3
PointCNN[15]	88.1	75.6	65.4
PointWeb[56]	87.3	76.2	66.7
ShellNet[72]	87.1	—	66.8
HPEIN[69]	—	76.3	67.8
FPConv[73]	—	—	68.7
PAConv[74]	—	78.6	69.3
Ours	**88.2**	**79.3**	**69.6**

表 4-3 不同模块性能测试

Table 4-3 Performance testing of different modules

PSAB	SSAB	GCAB	OA	mAcc	mIoU
√	✗	✗	87.2	68.6	61.6
✗	√	✗	87.5	69.4	62.6
✗	✗	√	87.9	69.1	62.2
√	✗	√	87.2	69.6	63.6
✗	√	√	88.7	71.6	64.7
√	√	✗	89.0	71.2	64.6
√	√	√	**89.1**	**71.9**	**65.2**

针对位置合成注意力模块（Position Synthesize Attention Block，PSAB），本小节分别对其关系描述符的选择进行对比实验，结果如表 4-4 所示，其中 (x_j, y_j, z_j) 表示相邻点的坐标位置，(x_i, y_i, z_i) 表示中心点的坐标位置，e_{ij} 是指相邻点 j 和中心点 i 之间的欧氏距离，当关系描述符为中心点的坐标、中心点与近邻点的相对坐标、中心点与近邻点的距离三者的时候，效果更好。

在语义自注意力机制模块（Semantic Self-Attention Block，SSAB）中，本小节针对 Mask 操作中 k 的选择进行了对比实验，如表 4-5 所示，当 k 的值为 32 的时

候，结果最好。

表 4-4 位置合成注意力模块不同的关系描述符的测试结果

Table 4-4 Test results of different relationship descriptors for the Positional Synthesis Attention Block

Input	OA	mAcc	mIoU
$(x_j - x_i,\ y_j - y_i,\ z_j - z_i,\ x_j,\ y_j,\ z_j)$	88.8	70.7	64.2
$(x_j - x_i,\ y_j - y_i,\ z_j - z_i)$	88.8	69.6	63.6
$(x_j - x_i,\ y_j - y_i,\ z_j - z_i,\ x_i,\ y_i,\ z_i,\ e_{ij})$	**89.1**	**71.9**	**65.2**
$(x_j - x_i,\ y_j - y_i,\ z_j - z_i,\ x_i,\ y_i,\ z_i,\ e_{ij},\ x_j,\ y_j,\ z_j)$	88.5	71.5	64.9

表 4-5 Mask 操作中不同的 k 在测试集上的质量评估结果

Table 4-5 Quality evaluation results of different k in the Mask operation on the test set

k	OA	mAcc	mIoU
32	**89.1**	**71.9**	**65.2**
24	88.6	70.4	63.8
16	88.2	71.3	64.3

本小节进一步针对位置合成注意力模块（Position Synthesize Attention Block，PSAB）中的注意力机制进行了不同的尝试，结果如表 4-6 所示，其中 Dense 表示 Dense 合成注意力机制，Fac. Dense 表示降秩的 Dense 合成注意力机制，Random 表示随机合成注意力机制，Fac. Random 表示降秩的随机合成注意力机制，Dot Product 表示自注意力机制，可以发现 Dense 合成注意力机制相比于其它注意力机制有着更好的效果。

表 4-6 不同的注意力机制在测试集上的测试结果

Table 4-6 Test results of different attention mechanisms on the test set

注意力机制	OA	mAcc	mIoU
Dense	**89.1**	**71.9**	**65.2**
Fac. Dense	87.3	68.2	61.0
Random	88.2	70.2	64.3
Fac. Random	88.9	71.1	64.5
Dot Product	87.8	69.2	63.2

在网络每个尺度下对于局部特征的提取中，本小节分别选择了不同的对称函数来验证算法的性能，实验结果如表 4-7 所示。从下面的实验结果可以看出，最大池化函数相比于平均池化函数、求和函数以及最大池化函数和平均池化函数的结合有着更好的分割效果。

为了更好的查看网络的语义分割结果，本小节提供了在 S3DIS 数据集的 Area5 下测试的可视化结果，如图 4-11 所示，第一列是原始的输入场景，第二列是输入场景本身分割的效果，第三列是 PointNet++的分割效果，第四列是本章分割网络的

测试效果，其中从每一行红色框的对比来看，本文的网络相比于 PointNet++对于门、柱体、书架等物体有着更好的语义分割效果。

表 4-7 不同的局部聚合函数在测试集上的质量评估结果

Table 4-7 Quality evaluation results of different local aggregate function on the test set

评价指标	Max	Mean	Sum	Max + Mean
OA	**89.1**	87.4	88.1	85.2
mIoU	**65.2**	61.6	61.9	59.1
mAcc	**71.9**	67.9	68.2	66.2

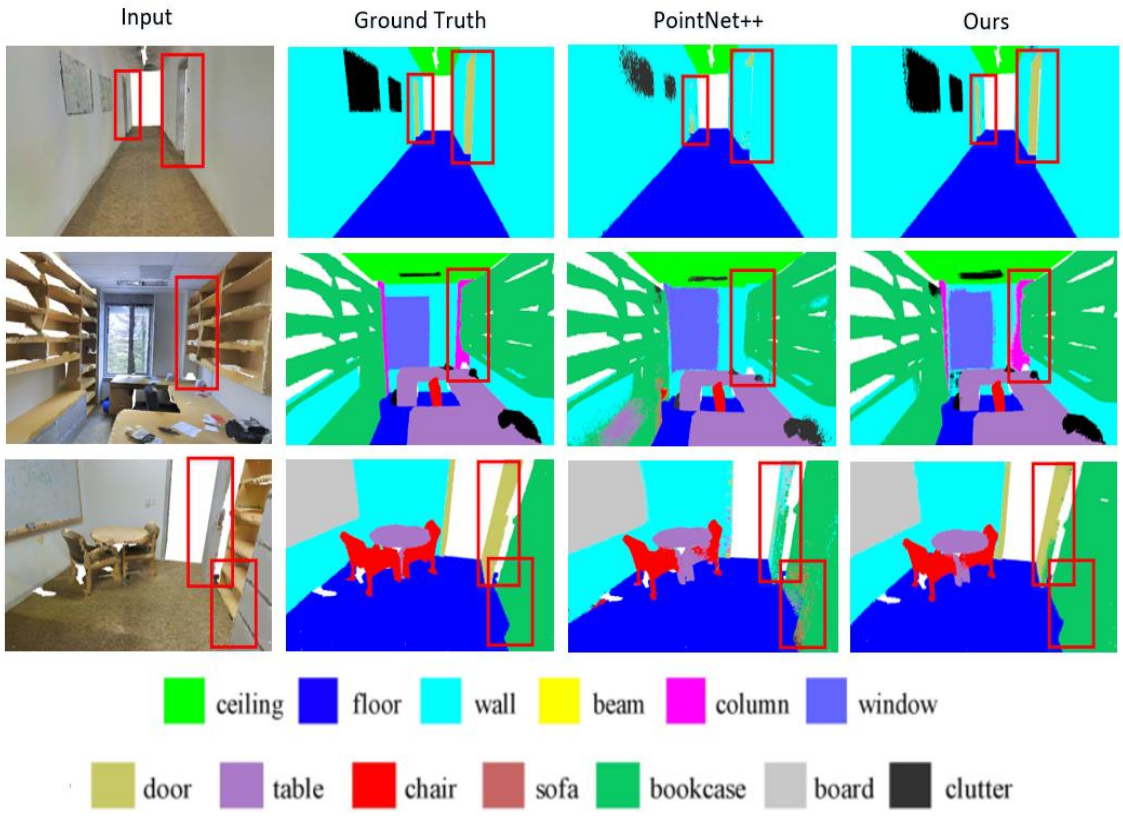

图 4-11 S3DIS 数据集 Area5 上的语义分割结果可视化

Figure 4-11 Visualization of semantic segmentation results on S3DIS dataset Area5

4.7.3 ShapeNetPart 数据集与实验设置

为了进一步验证算法的有效性，本小节在 ShapeNetPart 数据集上进行了补充实验。ShapeNetPart 数据集一共由 16880 的样本组成，其中 14006 个样本作为训练集，2874 个样本作为测试集。为了进行公平比较，训练的时候采用和 PointNet 相同的策略，每个数据被统一采样为 2048 个点以及采用了随机平移、随机异向缩放、

随机输入顺序作为数据增强策略；在测试期间没有使用以上数据增强策略。

本小节在 ShapeNetPart 数据集上对模型进行训练，其中模型采用的是深度学习框架 PyTorch。在深度学习模型训练中，不同的模型参数对实验结果有着很大的影响。本文通过大量实验并结合相关经验进行部分参数的调整，使模型达到最佳的效果。网络参数设置如下：（1）初始学习率为 0.003，每 40 个 Epoch 等间隔调整学习率，调整倍数为 0.5；（2）优化方法为 Adam 求解器；（3）BatchSize 为 32；（4）训练 400 个 Epoch。其中 BatchSize 表示点云批处理的大小，Epoch 表示训练的次数。

4.7.4　ShapeNetPart 数据集实验结果分析

为了验证本章提出算法的分割效果的通用性，本小节在 ShapeNetPart 数据集中的测试集上进行测试，并与点云零件分割相关的文章进行了对比，评价指标为整体类别平均交并比（Class mean Intersection over Union，Cls. mIoU），主要针对数据集中的 16 种整体类别取平均。

表 4-8 不同算法在测试集上的质量评估结果

Table 4-8 Quality evaluation results of different algorithms on the test set

算法	Cls. mIoU
PointNet[13]	80.4
SynSpec[75]	82.0
PCNN[67]	81.8
PointNet++[14]	81.9
DGCNN[22]	82.3
SpiderCNN[16]	82.4
PointConv[19]	82.8
Ours	**83.0**

表 4-8 给出了不同方法在 ShapeNetPart 测试集下整体类别平均交并比（Class mean Intersection over Union，Cls. mIoU）的比较。从表 4-8 中可以看出，本文的模型超过了大部分的网络，相对来说本文提出的基于注意力机制的多尺度点云语义分割网络在零件分割数据集上同样达到了一个比较好的结果，所以整体来说本文提出的基于注意力机制的多尺度点云语义分割网络具有一定的有效性。

4.8 本章小结

基于深度学习的点云语义分割算法构造的卷积神经网络由于局部特征提取时仅仅考虑了语义信息，而点云本身的位置信息没有很好的利用，并且很少利用到全局信息进行语义分割的增强，导致在语义分割方面的性能不能达到令人满意的效果。本章从三点出发，重新设计了一种结合位置信息和语义信息的局部特征提取算法来丰富局部信息，其中位置信息利用合成注意力机制模块来自适应学习，语义信息利用自注意力机制达到局部所有点之间的交互，然后在全局方面提出了基于通道注意力机制的特征提取模块，来进一步地增强全局下每个点的特征表达能力。网络整体结构类似于 U-Net 网络，分别经过了下采样特征提取、上采样特征恢复的过程，在其过程中不断的扩大感受野，尽可能的让每个点学习到局部邻域的关系，从而达到更好的分割效果。相比于很多点云语义分割的相关算法，本章提出的结合注意力机制的多尺度点云语义分割网络在 S3DIS 数据集的评价指标上取得了更好的效果，在 ShapeNetPart 补充数据集上同样有着较好的效果，从而在一定程度上证明了该算法的整体有效性。

5 结论

5.1 本文工作总结

本文针对点云的分类和语义分割问题，基于深度学习，从局部信息和全局信息两方面出发，结合自注意力机制，实现了点云分类和语义分割的准确度的有效提高。本文的具体工作可归纳如下：

本文首先介绍了目前点云分类和语义分割任务的算法不足点。基于深度学习的点云分类和语义分割算法，其主要问题是在网络特征提取的时候，局部特征提取的能力不足，无法充分地学习到局部邻域内点之间的相关性以及缺乏全局特征的有效提取。其中分类问题上，点与点全局范围内无法建立长依赖性；语义分割问题上，局部邻域往往只关注语义信息，忽略了位置信息。

根据上述问题，本文提出一种基于自注意力机制的多尺度点云分类网络，将空间特征变换机制应用在局部邻域中，降低不重要点的特征权重，使网络更加关注有用点的信息。利用自注意力机制来进行全局点之间的交互，建立点之间的长依赖性，使得网络在分类任务中受到所有点的关注。提出了特征融合模块来分别学习不同尺度下的全局信息，实现了不同感受野下的点的特征互补的效果。最后本文通过实验从常见的分类数据集以及分类指标上证明了本文提出的算法能够更好的实现点云的分类任务。

同理，在点云的分割问题上，本文提出一种基于注意力机制的多尺度点云语义分割网络，在局部信息提取的过程中，首先结合合成注意力模块学习点的位置信息，作为补充的特征信息，然后利用自注意力机制来实现点之间的交互，不同于点云分类任务的是，在语义分割中自注意力机制是用到局部邻域中的，而不是全局特征提取中，因为点云语义分割关注的是每个点的特征，其受到的影响主要是其局部邻域，全局上其它点的作用不是很大，接着在全局特征提取中引入了通道注意力机制，不会涉及到点之间的交互，而是针对每个点进行特征的自适应学习。最后通过实验从常见的语义分割数据集以及指标上证明了本文提出的算法能有效的提高点云的语义分割精度。

5.2 未来工作展望

在点云分类以及语义分割等问题上，利用深度学习进行三维的点云分类和分

割取得了重大的进展，然而这仅仅是一个开始，本文认为还有以下几点可以作为之后的研究方向，值得去探索与尝试：

（1） 基于点卷积的高效邻域搜索方法研究

目前基于点处理的分类和分割网络正逐渐成为研究最多的方向，因为不涉及到多视图和网格体素化的额外转换，从而避免了信息损失。这些方法致力于全面探索点特征之间的联系，一般所用的邻域搜索机制，如 KNN、Ball Query 等，但是这也很容易错过局部区域之间的低级特征，所以如何有效的进行邻域的搜索还是值得进一步研究的。

（2） 弱监督和无监督的 3D 分类和分割

深度学习在 3D 分割方面取得了巨大成功，但在很大程度上依赖于大规模的标记训练样本。弱监督和无监督学习被认为是大规模标记数据集无法实现的替代方法。所以未来如何针对少量的数据集就能实现点云的分类和分割可能会是热门的研究方向。

（3） 大规模场景的语义分割

目前很多方法在处理大场景的语义分割情况的时候，通常仅限于非常小的 3D 点云，如数据预处理成$1m \times 1m$的块，然后下采样到 4096 个点，而在没有数据预处理的情况下无法直接扩展到更大比例的点云（例如数百万个点或数百米）。尽管最近的相关论文可以直接处理 100 万个点，但是速度仍然不够，需要进一步研究大规模点云上的有效语义分割问题。

（4） 场景解析的多模态

对于点云的处理，通常有三种方法表示，分别是多视图、体素化和点云本身。前两种因为涉及到数据格式进一步的转换，在信息完整性分别会有一定程度的损失，例如多视图的几何信息较少，体素化的语义信息较少，所以目前点云分类和语义分割网络的很多网络都是直接在原始点云上处理的。但是多视图和体素化本身的关注点不同，如果能有效的结合多种表示方法，相互补充，可能会达到出乎预料的效果，所以多模态之后可能将是提高性能的另一种方法。

参考文献

[1] Jixian Z, Xiangguo L, Xinlian L. Advances and prospects of information extraction from point clouds[J]. Acta Geodaetica et Cartographica Sinica, 2017, 46(10): 1460.

[2] 李莹, 于海洋, 王燕, 等. 基于无人机重建点云与影像的城市植被分类[J]. 国土资源遥感, 2019, 31(1): 149-155.

[3] ZHANG J, LIN X, NING X. SVM-based classification of segmented airborne LiDAR point clouds in urban areas[J]. Remote Sensing, 2013, 5(8): 3749-3775.

[4] NI H, LIN X, ZHANG J. Classification of ALS point cloud with improved point cloud segmentation and random forests[J]. Remote Sensing, 2017, 9(3): 288.

[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, 2015: 945-953.

[6] Yu T, Meng J, Yuan J. Multi-view harmonized bilinear network for 3d object recognition[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 186-194.

[7] Yang Z, Wang L. Learning relationships for multi-view 3D object recognition[C]. //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 7505-7514.

[8] Wei X, Yu R, Sun J. View-gcn: View-based graph convolutional network for 3d shape analysis[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1850-1859.

[9] Wu Z, Song S, Khosla A, et al. 3D shapenets: A deep representation for volumetric shapes[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2015: 1912-1920.

[10] Maturana D, Scherer S. Voxnet: A 3D convolutional neural network for real-time object recognition[C]. //2015 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), 2015: 922-928.

[11] Riegler G, Osman Ulusoy A, Geiger A. Octnet: Learning deep 3D representations at high resolutions[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 3577-3586.

[12] Truc L, Ye D. PointGrid: A deep network for 3D shape understanding[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 9204-9214.

[13] 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, 2017: 652-660.

[14] 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 24th Annual Conference on Neural Information Processing Systems, 2017: 5099-5108.

[15] Li Y, Bu R, Sun M, et al. Pointcnn: Convolution on x -transformed points[J]. Advances in Neural Information Processing Systems, 2018, 31: 820-830.

[16] Xu Y F, Fan T Q, Xu M Y, et al. SpiderCNN: deep learning on point sets with parameterized convolutional filters[C]. //LNCS 11212: Proceedings of the 15th European Conference on Computer Vision, Munich, Sep 8- 14, 2018. Berlin, Heidelberg: Springer, 2018: 90-105.

[17] Thomas H, Qi C R, Deschaud J E, et al. Kpconv: Flexible and deformable convolution for point clouds[C]. //Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 6411-6420.

[18] Hua B S, Tran M K, Yeung S K. Pointwise convolutional neural networks[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 984-993.

[19] Wu W, Qi Z, Fuxin L. Pointconv: Deep convolutional networks on 3D point clouds[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 9621-9630.

[20] Zhou H, Feng Y, Fang M, et al. Adaptive graph convolution for point cloud analysis[C]. //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 4965-4974.

[21] Zhang Y, Rabbat M. A Graph-CNN for 3D Point Cloud Classification[C]. //2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, South Korea, April 22-27, 2018. IEEE, 2018:6279–6283.

[22] Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. ACM Transactions on Graphics, 2019, 38(5): 1-12.

[23] Zhang K, Hao M, Wang J, et al. Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features[J]. arXiv preprint arXiv:1904. 10014, 2019.

[24] 胡永东, 张正文, 李婕. 基于多尺度图卷积的点云分类网络[J]. 科技创新与应用, 2021(8): 99-101.

[25] 兰红, 陈浩, 张蒲芬. 集图卷积和三维方向卷积的点云分类分割模型[J/OL]. 计算机工程与应用: 1-15[2022-02-22]. http://kns.cnki.net/kcms/detail/11.2127.TP.20220125.1752.026.html.

[26] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017: 5998-6008.

[27] Yan X, Zheng C, 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, 2020: 5589-5598.

[28] Hertz A, Hanocka R, Giryes R, et al. Pointgmm: A neural gmm network for point clouds[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 12054-12063.

[29] Guo M H, Cai J X, Liu Z N, et al. PCT: Point cloud transformer [J]. Computational Visual Media, 2021, 7(2):187–199.

[30] Zhao H, Jiang L, Jia J, et al. Point transformer[C]. //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 16259-16268.

[31] Zhao H, Jia J, Koltun V. Exploring self-attention for image recognition[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10076-10085.

[32] Ballard D H. Generalizing the Hough transform to detect arbitrary shapes[J]. Pattern recognition, 1981, 13(2): 111-122.

[33] Bolles R C, Fischler M A. A RANSAC-based approach to model fitting and its application to finding cylinders in range data[C]. //IJCAI. 1981: 637-643.

[34] He Y, Yu H, Liu X, et al. Deep learning based 3D segmentation: A survey[J]. arXiv preprint arXiv:2103.05423, 2021.

[35] Lawin F J, Danelljan M, Tosteberg P, et al. Deep projective 3D semantic segmentation[C]. //International Conference on Computer Analysis of Images and Patterns. Springer, Cham, 2017: 95-107.

[36] Boulch A, Le Saux B, Audebert N. Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks[J]. 3DOR@ Eurographics, 2017: 17-24.

[37] Graham B, Engelcke M, Van Der Maaten L. 3d semantic segmentation with submanifold sparse convolutional networks[C]. //Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 9224-9232.

[38] Meng H, Gao L, Lai Y K, et al. VV-Net: Voxel vae net with group convolutions for point cloud segmentation[C]. //Proc of the 17th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 8499-8507.

[39] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]. //International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.

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

[41] Simonovsky M, Komodakis N. Dynamic edge-conditioned filters in convolutional neural networks on graphs[C]. //Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 3693-3702.

[42] Jiang M, Wu Y, Zhao T, et al. Pointsift: A sift-like network module for 3d point cloud semantic segmentation[J]. arXiv preprint arXiv:1807.00652, 2018.

[43] Hu Q Y, Yang B, Xie L H, et al. RandLA-Net: Efficient semantic segmentation of large-scale point clouds[C]. //Proc of the 33rd IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 11105-11114.

[44] Komarichev A, Zhong Z, Hua J. A-CNN: Annularly convolutional neural networks on point clouds[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 7421-7430.

[45] Uy M A, Pham Q, Nguyen T, et al. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data[C]. //Proc of the 17th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 1588–1597.

[46] Armeni I, Sener O, Zamir A R, et al. 3D semantic parsing of large-scale indoor spaces [C]. //Proc of the 29th IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 1534-1543.

[47] 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 (ToG), 2016, 35(6): 1-12.

[48] Tolstikhin I O, Houlsby N, Kolesnikov A, et al. Mlp-mixer: An all-mlp architecture for vision. arXiv preprint arXiv:2105.01601, 2021.

[49] Choe J, Park C, Rameau F, et al. PointMixer: MLP-Mixer for Point Cloud Understanding[J]. arXiv preprint arXiv:2111.11187, 2021.

[50] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[51] Zhang L, Huang S, Liu W, et al. Learning a mixture of granularity-specific experts for fine-grained categorization[C]. //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 8331-8340.

[52] Xie S, Liu S, Chen Z, et al. Attentional shapecontextnet for point cloud recognition[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4606-4615.

[53] Lee J, Lee Y, Kim J, et al. Set transformer: A framework for attention-based permutation-invariant neural networks[C]. //International Conference on Machine Learning. PMLR, 2019: 3744-3753.

[54] Yang J, Zhang Q, Ni B, et al. Modeling point clouds with self-attention and gumbel subset sampling[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 3323-3332.

[55] Wang C, Samari B, Siddiqi K. Local spectral graph convolution for point set feature learning[C]. //LNCS 11208: Proceedings of the 15th European Conference on Computer Vision, Munich, Sep 8-14, 2018. Berlin, Heidelberg: Springer, 2018: 56-71.

[56] Zhao H, Jiang L, Fu C W, et al. Pointweb: Enhancing local neighborhood features for point cloud processing[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 5565-5573.

[57] 高金金, 李潞洋. 基于局部关系卷积的点云分类与分割模型[J/OL]. 计算机工程与应用. 2021: 1-9.

[58] Ma X, Qin C, You H, et al. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework[J]. arXiv preprint arXiv:2202.07123, 2022.

[59] Ben-Shabat Y, Lindenbaum M, Fischer A. 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 3145-3152.

[60] Qiu S, Anwar S, Barnes N. Dense-resolution network for point cloud classification and segmentation[C]. //Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021: 3813-3822.

[61] Qiu S, Anwar S, Barnes N. Geometric back-projection network for point cloud classification[J]. IEEE Transactions on Multimedia, 2021: 1943-1955.

[62] Goyal A, Law H, Liu B, et al. Revisiting point cloud shape classification with a simple and effective baseline[C]. //International Conference on Machine Learning. PMLR, 2021: 3809-3820.

[63] Tay Y, Bahri D, Metzler D, et al. Synthesizer: Rethinking self-attention for transformer models[C]. //International Conference on Machine Learning. PMLR, 2021: 10183-10192.

[64] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]. //Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[65] Tchapmi L, Choy C, Armeni I, et al. SEGCloud: Semantic segmentation of 3D point clouds [C]. //Proc of the 5th Int Conf on 3D Vision. Piscataway, NJ: IEEE, 2017: 537-547.

[66] Tatarchenko M, Park J, Koltun V, et al. Tangent convolutions for dense prediction in 3D[C]. //Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Jun 18- 22, 2018. Washington: IEEE Computer Society, 2018: 3887-3896.

[67] Atzmon M, Maron H, Lipman Y. Point convolutional neural networks by extension operators[J]. arXiv preprint arXiv:1803.10091, 2018.

[68] 吴军, 崔玥, 赵雪梅, 陈睿星, 徐刚. SSA-PointNet++: 空间自注意力机制下的 3D 点云语义分割网络[J]. 计算机辅助设计与图形学学报, 2022, 34(03): 437-448.

[69] Jiang L, Zhao H, Liu S, et al. Hierarchical point-edge interaction network for point cloud semantic segmentation[C]. //Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 10433-10441.

[70] Fan S, Dong Q, Zhu F, et al. SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 14504-14513.

[71] Huang Q G, Wang W Y, Neumann U. Recurrent slice networks for 3D segmentation of point clouds[C]. //Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Jun 18-22, 2018. Washington: IEEE Computer Society, 2018: 2626-2635.

[72] Zhang Z Y, Hua B, Yeung S. ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics[C]. //Proc of the 17th IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2019: 1607-1616.

[73] Lin Y, Yan Z, Huang H, et al. Fpconv: Learning local flattening for point convolution[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 4293-4302.

[74] Xu M, Ding R, Zhao H, et al. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds[C]. //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 3173-3182.

[75] Yi L, Su H, Guo X, et al. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation[C]. //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 2282-2290.

作者简历及攻读硕士学位期间取得的研究成果

一、作者简历

2016 年 9 月——2020 年 7 月 燕山大学 信息科学与工程学院　软件工程

2020 年 9 月——2022 年 6 月 北京交通大学 计算机与信息技术学院 人工智能

二、软件著作权

[1]Web 目标检测系统（2022SR0578801）

[2]基于点云的轨面异物检测系统（2022R11L0443884）

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：宋文贤 签字日期：2022年6月1日

学位论文数据集

表 1.1： 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
点云分类；多尺度；注意力机制；点云语义分割	公开			
学位授予单位名称*		学位授予单位代码*	学位类别*	学位级别*
北京交通大学		10004	电子信息	硕士
论文题名*		并列题名		论文语种*
基于注意力机制的 3D 点云分类和语义分割算法研究				中文
作者姓名*	宋文赞		学号*	20125226
培养单位名称*		培养单位代码*	培养单位地址	邮编
北京交通大学		10004	北京市海淀区西直门外上园村 3 号	100044
专业领域*		研究方向*	学制*	学位授予年*
人工智能		点云分类和分割	两年	2022
论文提交日期*	2022 年 6 月			
导师姓名*	白慧慧		职称*	教授
评阅人	答辩委员会主席*		答辩委员会成员	
	杨唐文		章春娥 于双元	
电子版论文提交格式 文本（√） 图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ） 推荐格式：application/msword；application/pdf				
电子版论文出版（发布）者		电子版论文出版（发布）地		权限声明
论文总页数*	72			
共 33 项，其中带*为必填数据，为 21 项。				