基于深度学习的点云分割方法综述

孙广岩*，宋昕帅*，廖子睿*，李云飞*，林愈凯*，刘丁烨*，赵岫†，刘梦源†

中山大学 智能工程学院，深圳 518107

【摘　要】 近年来，深度传感器和三维扫描仪的普及导致了三维点云的快速发展。点云的语义分割作为理解三维场景的关键步骤，已经引起了研究人员的广泛关注。这一课题的最新进展主要由基于深度学习的方法主导。在本文中，我们总结了从间接分割到直接分割的各个方面的综述报告。首先，我们回顾了基于多视图和体素网格的间接分割方法，以及从不同角度的直接分割方法，包括基于点处理、优化 CNN、图卷积神经网络和时序的融合。最后，我们对点云语义分割技术的发展趋势进行了展望。本次分工如下：廖子睿与李云飞同学完成了 1 到 3.2 的内容，孙广岩与林愈凯同学完成了 3.3 到 3.5 的内容，宋昕帅与刘丁烨完成了 3.6 到 4 的内容。

【关键词】 深度学习，点云，点云分割，语义分割，综述

1　引 言

人们对美好生活的需求促进了人工智能、无人驾驶、机器人技术的不断发展，这些行业或技术都需要处理 3 维 (3 Dimensional，3D) 数据，并要实时地反馈处理结果。同时，3 维扫描设备的快速发展也使得人们越来越容易获取点云 (point cloud) 数据[1]，这些软硬件设备的发展让处理并分析三维点云成为了当前的一个热点。点云分割作为点云数据处理和分析的基础技术，在很多领域中都有着广泛的应用[2-3]。以自动驾驶为例，车辆行驶过程中，该项技术需要分割出路况信息，以及行人等信息，由此来实现目标检测[4-5]。类比 2 维图像分割，点云分割是一种将点云划分成若干个具有不同性质的区域并进行识别的技术。从点的角度出发，点云分割可看作一项对点云中每一点赋予相应意义标签的任务。

点云分割算法起源于 2 维图像算法，早期研究者将 3 维点云坐标投影至 2 维平面上，使用 2 维算法解决 3 维问题。然而投影转换过程中信息损失较大，限制了处理效果。同时期还有一些研究者将不规则点云转换成规则的体素网格。虽然它们有效地处理了不规则点云数据，但由于数据的间接表示和存储限制的问题丢失了大量的几何结构信息并造成数据稀疏的问题。随后，斯坦福大学 Qi 等人[6]提出的 PointNet 网络模型，该方法直接从点云数据入手，无需做任何中间转换操作，从而保留了点云的固有信息。自此直接对点云处理的方法逐渐发展起来。

2　传统的点云分割方法

鉴于点云分割技术的传统方法不是本文介绍的重点，以下简要概述几种传统方法进行点云分割的流程。

基于边缘的方法　边缘是描述点云物体形状的基本特征，这种方法检测点云一些区域的边界来获取分割区域，这些方法的原理是定位出边缘点的强度变化，Bir Bhanu 等人[7]提出了一种边缘检测技术，通过计算梯度，检测表面上单位法向量方向的变化来拟合线段。Xiaoyi Jiang 等人[8]是基于扫描线的分组进行快速分割，基于边缘的方法虽然分割速度比较快但是准确度不能保证，因为边缘对于噪声和不均匀的或稀疏的点云非常敏感。

基于区域分割方法　基于区域的方法使用邻域信息来将具有相似属性的附近点归类，以获得到分割区域，并区分出不同区域之间的差异性。基于区域的方法比基于边缘的方法更准确。但是他们在分割过度或不足以及在如何准确确定区域边界方面存在问题。研究者们将基于区域的方法分为两类：种子区域（或自下而上）方法和非种子区域（或自上而下）方法。

*组内评分：10 * 6

† 指导教师

基于模型的方法，圆锥，平面和圆柱形来对点云进行分组，那么根据这些几个形状，具有相同的数学表示的点将会被分割为同一组点，Martin A. Fischler 等人[9]引入了一种众所周知的算法 RANSAC（RANdom SAmple Consensus）,RANSAC 是强大的模型，用于检测直线，圆等数学特征，这种应用极为广泛且可以认为是模型拟合的最先进技术，在 3D 点云的分割中需要改进的方法都是继承了这种方法。基于模型的方法具有纯粹的数学原理，快速且强大，具有异值性，这种方法的主要局限性在于处理不同点云是的不准确性。这种方法在点云库中已经实现了基于线，平面，圆等各种模型。

3 基于深度学习的点云分割方法

3.1 基于多视图的方法

基于多视图的方法是将 3 维点云在多个视点下投影到 2 维平面，再对 2 维投影图像进行处理。其有如下缺点：一方面，由于 2 维多视图图像只是 3 维场景的近似值，并非是对 3 维场景真实且无损的呈现，这样就会造成几何结构的损失。例如在点云语义分割之类的复杂任务中可能会产生不太理想的结果。另一方面，由于点云呈现复杂的点信息和曲面信息，多视图图像无法全面覆盖到这些信息。以下介绍两篇较为经典的工作。

3.1.1 MVCNN

计算机视觉中一个长期存在的问题涉及用于识别的 3D 形状的表示：3D 形状应该用在其原生 3D 格式上运行的描述算子表示，例如体素网格或多边形网格，还是可以用基于视图的描述符有效地表示？Hang Su 等人[10]在学习从 2D 图像上的渲染视图集合中识别 3D 形状的背景下解决了这个问题。此外，他们提出了一种新颖的 CNN 架构，它将来自 3D 形状的多个视图的信息组合成单个紧凑的形状描述算子，从而提供更好的识别性能。可以应用相同的架构来准确识别人类手绘的形状草图。于是得出结论，2D 视图的集合可以为 3D 形状识别提供大量信息，并且适用于新兴的 CNN 架构及其衍生产品。

这篇文章的创新之处在于用物体的三维数据从不同“视角”所得到的二维渲染图，作为原始的训练数据。用经典、成熟的二维图像卷积网络进行训练，训练出的模型，对三维物体的识别、分类效果之好，甚至优于用三维数据直接训练出的模型。下图展示了 MVCNN 方法的大致框架，从物体的 12 个不同的视图渲染 3D 形状，并通过 CNN_1 来提取基于视图的特征。然后，这些数据跨多个视图合并，并通过 CNN_2 传递，以获得紧凑的形状描述算子（Shape descriptors）

将三维形状描述的数据应用在机器学习中有一些难度。首先，相比二维图像的数据库，三维模型的数据库颇为匮乏；另外，三维数据特征维度容易很高，造成过拟合和维数灾难。本文使用 2D 表示学习 3D 特征的优势在于

1. 相对效率更高。若使用体素表示的三维数据输入到神经网络中，要在合理、可接受的时间内训练出结果需要显著降低分辨率。例如，3D ShapeNet 使用形状的粗略表示，即由二进制体素组成的 30x30x30 网格。相反，相同输入尺寸的 3D 模型的单个投影只对应于 164×164 像素的图像，或者如果使用多个投影作为输入，则图像占用的空间较小。

2. 可以基于前人的工作。可以使用大量的图像数据库 (如 ImageNet) 来预训练 CNN 架构。因为图像是无处不在的，大型标记数据集是丰富的，可以学到很多关于 2D 图像分类的通用特征，然后微调到 3D 模型投影的细节。同时可以受益于图像描述算子（image descriptors）的进步与发展。

形状描述算子

形状描述算子可以大致分为两大类：一类是 3D shape descriptors，直接在原始的三维数据上进行描述表示，其中包括 a. polygon meshes（多边形网格），b. voxel-based discretizations（基于“体素”的离散化），c. point cloud（点云），d. implicit surfaces（隐式曲面）；还包括 viewbased descriptors，用二维投影的方法去描述三维物体。

Wu 等人[11]最近的工作 (通过 3D 卷积网络从基于体素的对象表示中学习形状描述算子) 解决了传统 3D 形状描述算子根据形状表面或体积的特定几何属性进行手工设计的繁琐与不便。例如形状可以用由表面法线和曲率[12]、在采样表面点处收集的距离、角度、三角形区域或四面体体等来表示。

另一方面，基于视图的描述算子具有许多理

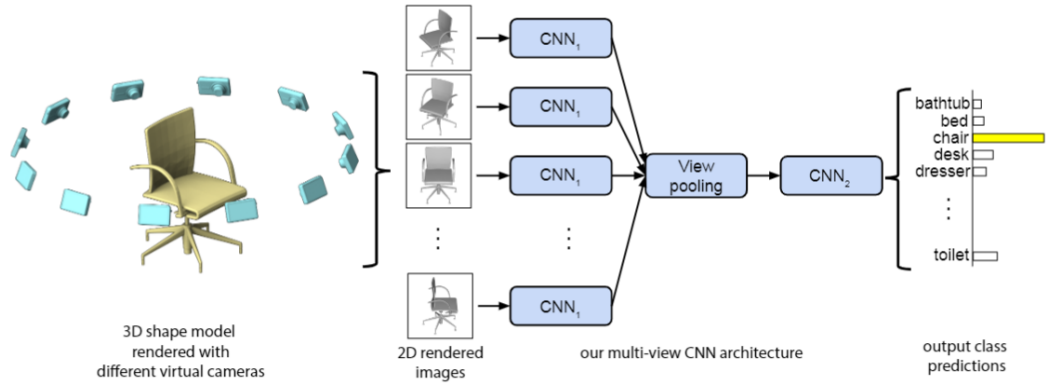

图 1 用于 3D 形状识别的多视图 CNN(使用第一个相机设置进行说明)

想的特性：它们相对低维，评估效率高，并且对一些复杂的 3D 形状表示表现较好.

网络方法 本文中的重点是开发基于视图的 3D 形状描述算子，这些描述符是可训练的，为识别和检索任务产生信息表示，并且计算高效。

3D 形状数据是以多边形网格的格式存储的。本文使用一种叫做“Phone reflection model”的方法，来由多边形网格产生渲染图。为了产生 3D 形状的多视角渲染图，需要设定一个“视角”（虚拟相机）来产生网格的渲染图。相机设置有两种方式，第一种类似图 1，围绕着 z 轴每个三十度取一张图片，虚拟相机向下倾斜三十度；第二种则是使用正二十面体包围三维模型，在每个面的中心放置虚拟相机，然后依次旋转 0°，90°，180°，270° 抓取四张图片，总共获得 80 张渲染的图片。

实验发现通过上述的不同的视角点获得的图像已经足够满足要求了。每一张视角图可以提取一次图像描述算子，每一个 3D 形状产生 12 张或者 80 张图，下一步要做的便是将所有的投影图的特征整合来描述三维特征，以便进行后续的识别任务。本文考虑了一下两类图像描述算子，首先是用 VLFeat 实现的 Fisher vector。对于每一张图像，先用 multi-scale SIFT 提取特征。再用 PCA 投影成 80 维，然后使用包含 64 个分量的高斯混合模型进行池化，并进行平方根和 l_2 正则化得到 Fisher 向量；另外是 CNN 特征，它包括 5 个卷积层，3 个全连接层，最后用 Softmax 分类。倒数第二层经过 ReLU 激活后用来描述图像特征，共 4096 维。网络是在 ImageNet 上预练过的。

两种方法都能获得比较好的特征描述算子。接着使用一对多线性 SVM 作为分类器，在测试时，作者把 12 个视图中置信度最高的类别返回效果较好，若使用所有试图置信度的平均值则效果不好，因为有些视图中并不包含三维模型的主要特征信息。

检索任务需要距离或相似性度量. 这里作者用 l_2 范数定义了两个三维形状之间距离的度量方式如下公式。

$$d(x,y)=\frac{\sum_j min_i\|(x_i-y_j)\|_2}{2n_y}+\frac{\sum_i min_j\|(x_i-y_j)\|_2}{2n_x}$$

为了解释这一定义，首先可以定义二维图像 xi 和三维形状 y 之间的距离，$d(xi,y)=min_j\parallel(x_i-y_j)\parallel_2$ 然后给定 x 的 2D 投影和 y 之间的所有 n_x 距离，这两个形状之间的距离通过简单的平均来计算。在公式 1 中，为了保证对称性，在两个方向上都采用了这种思想。

虽然对于 3D 形状，上面的多重的描述算子比现存的 3D 形状特征描述算子的效果更好，但是在多数情况下，这种算法是低效的。如公式 1 中所展示的，要衡量两个 3D 形状之间的距离，需要计算 $n_x\times n_y$ 个距离（在 3D 形状对应的视角 2D 图像下计算），这本身需要大量的计算。如前面所说，简单的求一个 3D 形状的多视角图像的特征描述算子的平均值，或者简单的将这些特征描述算子简单地串联起来，会导致不好的效果。所以，这一部分，作者集中于融合多视角 2D 图像产生的特征，以便综合这些信息，形成一个简单、高效的 3D 形状描算子。

因此，作者设计了 Multi-view CNN（MVCNN），放在基础的 2D 图像 CNN 之中。如图所示，同一个 3D 形状的每一张视角图像各自独立地经过第一

段的 CNN_1 卷积网络，在一个叫做 View-pooling 层进行“聚合”。之后，再送入剩下的 CNN_2 卷积网络。整张网络第一部分的所有分支，共享相同的 CNN_1 里的参数。在 View-pooling 层中，逐元素取最大值操作，另一种是求平均值操作，但在实验中，这并不有效。这个 View-pooling 层，可以放在网络中的任何位置。经过实验证明，这一层最好放在最后的卷积层（Conv5），以最优化的执行分类与检索的任务。View-pooling 优点类似于 max-pooling layer 与 maxout layer，不同点在于进行 max 操作时的维度不同，这里的 max 操作是“纵向的”，即在 12 个视角图像中，同一个位置的地方，进行 max 操作。而一般意义上所说的 max-pooling，通常是指在一个的领域像素单元内，取 max 像素值保留的操作。

在今后的工作中有许多方向仍待探索。一种是尝试不同的 2D 视图组合：哪些视图提供的信息最丰富？达到给定的精确度需要多少个视图？可以即时选择信息丰富的视图吗？另一个方向是：这种视图聚合技术是否可以用于从多个视图为真实世界的 3D 对象构建紧凑和区分的描述符，或者从视频自动构建，而不仅仅是用于 3D 多边形网格模型。这样的研究可以立即适用于被广泛研究的问题，如物体识别和人脸识别。

3.1.2 PVRNet

在本文中，Haoxuan You 等人[13]介绍了基于点云视图关系 (相关性) 的深度神经网络（PVRNet），这是一种有效的融合视图特征和点云特征的网络，并提出了关系评分模块。在关系评分模块的基础上，首先将点云特征与每个单视图特征融合为点云单视图融合特征，然后将点云特征与不同视图数特征融合为点云多视图融合特征。最后，进一步将点单视图融合特征和点多视图融合特征结合起来，实现对三维形状的统一表示。本方法的提出同样是为了解决处理 3d 数据的高成本问题的。同时，在 MVCNN 中，由于多视图特征被同等对待，这种融合不能充分利用多视图特征和点云特征之间的关系，本文提出，挖掘点云与不同视图之间的关系是实现更强大的形状描述算子的必要条件。

PVRNet 的输入是三维形状的两种模态数据: 多视图数据和点云数据。数据采集过程中为每个形状获得 12 个视图和 1024 个点。首先分别将点云和多视图数据送入对应的特征提取器，得到整个点云的全局特征和每个视图的视图特征。然后，提出一个关系评分模块来学习点云和各视图之间的相关性，在此基础上，进一步利用点单视图融合和点多视图融合，构建统一的三维形状特征。然后将统一的特征用于形状分类和检索的应用。

关系评分模块 受 VQA 任务的关系推理网络的启发 (Santoro 等人 2017)，作者将点云与第 i 个视图的关系分数定义为:

$$RS_i(P,V) = \xi(g_\theta(p, v_i))$$

其中 p 是点云要素，$V = v_1, ..., v_n$ 表示从 3D 形状提取的 n 个视图特征。函数 g_θ 推理了点云特征和每个视图特征之间的关系，并学习了一种有效的融合。在我们的网络中，g_θ 使用简单多层感知器，ξ 是归一化函数 (在作者的实验中使用 Sigmoid 函数)。对于每个视图，输出是从 0 到 1 的关系分数，这表示不同视图和点云之间的相关性的重要性。

关系得分有两个目的：1) 以残差的方式增强视图特征；2) 确定点云-多视图融合中将包括哪些视图特征。对于特征增强，与点云特征具有较强相关性的视图特征将被赋予更大的重要性。作者使用关系分数 $RS(P\ V)$ 通过残差连接来增强视图特征:

$$v_i' = v_i * (1 + RS_i(P, V))$$

其中 $v_i * RS_i(P\ V)$ 是通过关系得分对特征进行细化，然后与原始视图特征 vi 相加生成增强特征 v_i'，然后将增强后的视点特征、点云特征和关系得分送入点单视点融合模块和点多视点融合模块得到综合特征。点云-单视图融合和点云-多视点融合如图 3 所示。

点云-单视图融合 每个视点都有自己的局部特征，因此采用点云与不同视点的融合是合理的。考虑一个点视图集 $(PVSet)S_n = p\ v_1', ..., v_n'$，其中 n 表示该集合中的视图数，p 表示对应的点云信息。根据 PVSets 中的视点数目，分别进行了点云-单视图融合和点云-多视点融合。点云-单视图融合模块以包含点云和每个视点的 n 个 PVSets 为输入，聚合成对关系。当 PVSet 仅包括单视图 i 时，对集融合函数可定义为:

$$SF_i = h_\phi(p.v_i')$$

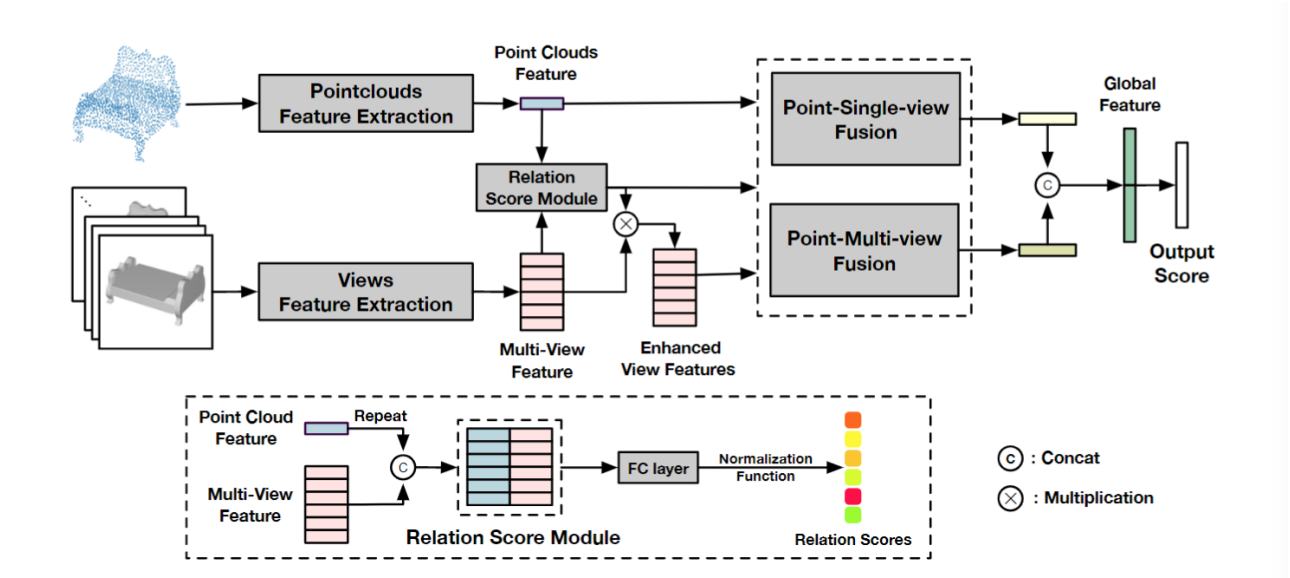

图 2 PVRNet 的框架结构

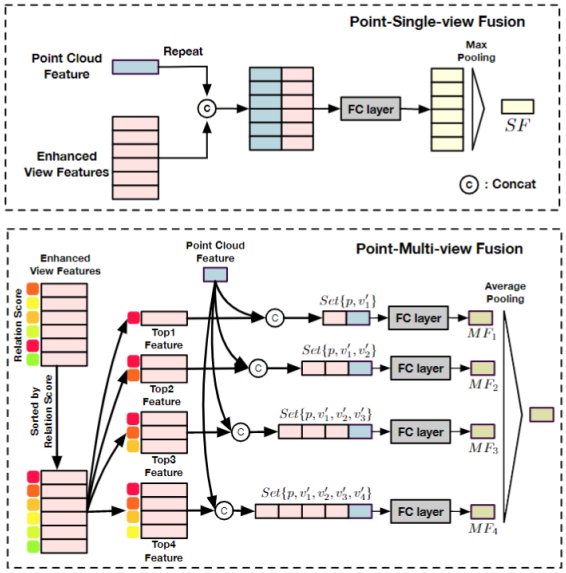

图 3 点云多视图融合和点云单视图融合模块。增强视图特征按关系分数排序。不同的颜色表示不同的关系得分显著性值。从红色到绿色，关系得分从 1 到 0 下降。

其中下标 i 表示融合是关于与第 i 个视图的点云-单视图集。作者使用 MLP 作为函数 h_{φ}。作者通过一个简单的最大池化操作聚合了覆盖所有视图的 n 个集合关系。该块的最终输出可以描述如下:

$$SF_{usion} = SF = MaxpoolingSF_1, ..., SF_n,$$

3.1.3 PointGrid

点云-多视图融合 进一步探讨点云与多视点的融合问题。在这种情况下，点云与多个视图组合形成 PVSet。此处的关系分数决定哪些视图特征包含在集合中。根据关系分数的值对视图特征进行排序。然后选取关联度较高的视图特征组合点云特征形成 PVSets。具有 k 个视图的集合可以表示为 $S_k = p, v'_1, ...v'_k$，其中包括具有最高关系分数的 top-k 个视图。融合函数可以在上述的点云-单视融合的基础上进行扩展:

$$MF_k = h_{\phi}(p, v'_1..., v'_k)$$

总共有 K 个 PVset，不同的点云-多视图组合叠加在一起:

$$MF_{usion} = \frac{1}{K} \sum Kk = 1MF_k$$

然后，通过拼接操作将两层融合特征聚合在一起，生成最终的点云视图关系特征:

$$F_{usion} = Concat(SF_{usion}, MF_{usion})$$

然后用 MLPs 和 Softmax 函数对最终特征进行分类，得到分类结果。

作者在 ModelNet40 上对分类和检索任务进行了综合实验，评估了该框架的有效性。消融结果和可视化结果表明，本文提出的基于点云视图关系的特征融合方案对该框架有重要贡献。

3.2 基于体素的方法

基于体素的点云分割的主要思想是将点云进行体素化，从而使点云变得有序和结构化，再对体素进行 3 维卷积，在点云分割领域中，从 2.5D

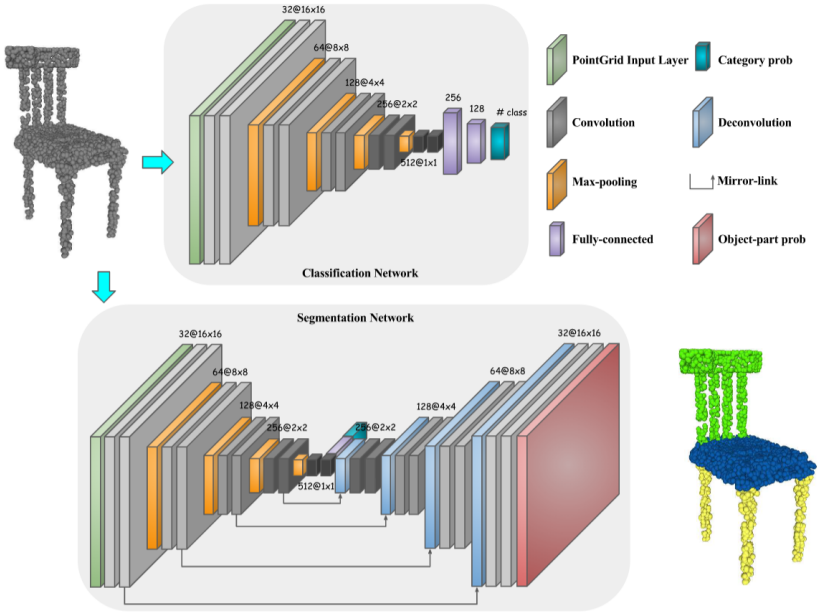

图 4 PointGrid 的架构示意图

图像中重构 3D 形状和用 3D 数据进行目标检测一直是一个难点，前人大部分依赖模型的部分标记和训练,3D ShapeNet[31] 采用卷积深度概率网络将 3D 几何形状映射为体素网格上的概率分布，从而更快速训练转化后的数据。基于体素的方法直接将点云体素作为 3 维卷积运算的基本单位，不需要对数据降维，维度信息保留好，特征还原度高，但发现该类方法已逐渐减少，综合分析有以下原因：
(1) 算法的时间和空间开销大，对于 3 维点的卷积运算需要更大的算力支持，高分辨率的体素网格需要耗费大量的存储空间，对于较大场景的适用性低
(2) 由于 3 维点云具有稀疏性，会产生大量冗余的体素网格，加剧运算和内存的开销。不过就方法本身而言，利用 3 维卷积处理来 3 维空间信息，可以很好的保留每个维度的信息特征，流程上也更加自然，随着计算性能和存储方法的不断升级，该类方法还是具有一定潜在的发展空间的。

体素网格因为它的规律性，在 3D 深度学习中被广泛应用。然而，使用相对较低阶的局部近似函数（例如分段常数函数）或者分段线性函数（距离场）来近似 3D 形状，意味着需要一个非常高分辨率的网格来表示更加精细的几何形状细节，这样会导致内存占用高，计算量大的问题。Truc Le 等人[14]提出了 PointGrid，一个 3D 卷积网络，在每个网络单元中包含恒定数量的点，从而允许网络学习更高阶的局部逼近函数，可以更好地表示局部几何形状细节。通过对流行的形状识别的基准进行测试，在现有的深度学习中，PointGrid 展示了良好的性能。

网络结构 新的深层架构使用为 3D 点云构建的 3D 网格。框架中唯一的预处理步骤是将点云数据归一化至 $[-1,1]^3$。与 VoxNet 不同的是，VoxNet 使用占用网格作为 3D 结构的主要表示，而本文作者希望将点的坐标堆叠为每个单元的特征。因此，具有 K 个点的单元格将具有对应 x、y 和 z 坐标的 3k 个特征。然而，每个单元具有不同的点数，构建这样的网格对于共享 3D 卷积核是不可行的。为了解决这个问题，作者使用了一种简单而有效的采样策略，称为点量化，在每个单元中保持固定数量的 K 个点。换种说法，如果一个单元格里有超过 K 个点，随机抽样其中的 K 个；如果少于 K 个点，则用零填充至 k 个点。这个操作模仿了 2 维卷积中的 padding 操作。

网络的主体由分类网络和分割网络组成，其框架结构如图 4 所示。PointGrid 是点和网格的混合表示，它可以更好地捕捉局部几何细节，同时展示易于学习的规则结构。在广泛使用的基准数据集上的实验表明，PointGrid 在分类和分割方面优于同时期的深度学习方法，并且比其他体积度量

方法占用的内存更小。

3.3 基于点处理的方法

由于点云的格式不规则，大多数研究人员通常会将这些数据转换为规则的三维体素网格或图像集合，然后再将其输入模型。而目前研究人员尝试直接从点云数据中提取特征信息，因而逐渐发展出一些直接处理点云的网络模型方法。其特点是直接输入原始数据，输入网络之前不对点云数据做任何变换。这也符合基于深度学习的端到端架构的思想。

3.3.1 PointNet

PointNet[6]是首个将点云数据直接输入进行训练的模型。完整网络架构如图5所示，其中分类网络和分割网络共享很大一部分结构。网络有三个关键模块：最大池层（作为聚合所有点信息的对称函数）、局部和全局信息组合结构，以及对齐输入点和点特征的两个联合对齐网络。我们将在下面的单独段落中讨论这些模块的设计与选择的原因。

无序输入的对称函数 为了使模型对输入转置保持不变，存在三种策略：1）将输入排序为规范顺序；2）将输入作为一个序列来训练 RNN，但通过各种排列来增加训练数据；3）使用简单的对称函数聚合每个点的信息。这里，对称函数将 n 个向量作为输入，并输出一个对输入顺序不变的新向量。例如，+ 和 * 运算符是对称的二元函数。

虽然排序听起来像是一个简单的解决方案，但在高维空间中，实际上不存在一般意义上稳定的排序。这很容易解释，如果存在这种排序策略，它将在高维空间和一维实线之间定义一个双射映射。不难看出，要求排序为稳定的排序相当于要求该图在维数减少时保持空间接近性，这是在一般情况下无法实现的任务。因此，简单的排序并不能完全解决排序问题，而且由于排序问题的持续存在，网络很难学习从输入到输出的一致映射。论文实验发现直接在排序点集上应用 MLP 的性能很差，尽管略优于直接处理未排序的输入。使用 RNN 的想法将点集视为一个序列信号，并希望通过使用随机排列的序列训练 RNN，RNN 将对输入顺序保持不变。然而虽然 RNN 对长度较小（几十个）的序列的输入排序具有相对较好的鲁棒性，但很难扩展到数千个输入元素，这仅仅是点集的常见大小。并且实验表明，基于 RNN 的模型的性能不如论文提出的方法。

论文的想法是通过对点集中的变换元素应用对称函数来近似定义在点集中的一般函数：

$$f(x_1, ..., x_n) \approx g(h(x_1), ..., h(x_n))$$

其中 $f : 2^{\mathbb{R}^N} \to \mathbb{R}, g : \underbrace{\mathbb{R}^K \times \cdots \times \mathbb{R}^K}_{n} \to \mathbb{R}$ 是对称方程。

论文的基本模块是实际上非常简单：用多层感知器网络来近似 h，用单变量函数和最大池函数的组合来近似 g。实验证明，这是行之有效的。通过 h 的集合，我们可以学习许多 f 来捕获集合的不同属性。

局部和全局信息聚合 上述部分的输出形成向量 $[f_1 \ldots f_K]$，这是输入集的全局特征，那么可以很容易地在形状全局特征上训练 SVM 或多层感知器分类器进行分类。然而，点分割需要结合局部和全局知识。论文通过了一种简单而高效的方式实现这一目标。

解决方案可以见图5（分段网络）。在计算全局点云特征向量后，通过将全局特征与每个点特征连接起来，将其反馈给每个点的特征。然后，我们基于组合点特征提取新的每点特征，而这一次每点特征同时学习局部和全局信息。

通过这种修改，网络能够预测依赖于局部几何和全局语义的每点数量。例如，我们可以准确预测每个点的法线，验证网络是否能够汇总来自点的局部邻域的信息。在实验环节中还表明了模型可以在形状部分分割和场景分割方面达到最先进的性能。

联合对齐网络 如果点云经历某些几何变换（如刚性变换），则点云的语义标记必须保持不变。因此，模型期望通过点集学习的表示对这些变换是不变的。自然的解决方案是在特征提取之前将所有输入集对齐到规范空间。而论文中不需要发明任何新的层，也不需要像图像那样引入别名，而是通过一个迷你网络（图5中的 T 网络）预测仿射变换矩阵，并将此变换直接应用于输入点的坐标。微型网络本身类似于大型网络，由与点无关的特征提取、最大池和完全连接层等基本模块组成。

该思想还可以进一步扩展到特征空间的对齐。可以在点特征上插入另一个对齐网络，并预测特征变换矩阵以对齐来自不同输入点云的特征。然

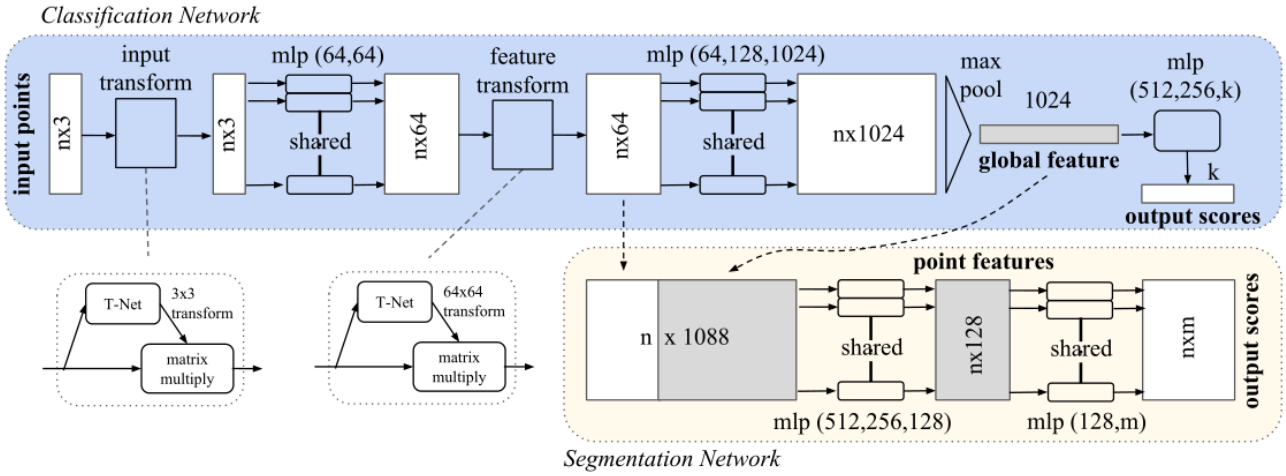

图 5 分类网络以 n 个点为输入，进行输入和特征变换，然后通过最大池化聚合点特征。输出为 k 类的分类分数。分割网络是分类网络的扩展。它连接全局和局部特征，并输出每个点的分数。“mlp” 括号中的数字是层大小。BatchNorm 用于具有 ReLU 的所有图层。Dropout 层用于分类网中的最后一个 mlp。

而，特征空间中的变换矩阵比空间变换矩阵的维数要高得多，这大大增加了优化的难度。因此，模型在 softmax 训练损失中添加了一个正则化项。将特征变换矩阵约束为接近正交矩阵定义为：

$$L_{reg} = \left\| I - AA^T \right\|_F^2$$

其中 A 是由微型网络预测的特征对齐矩阵。正交变换不会丢失输入中的信息，因此是需要的。实验发现，通过添加正则化项，优化变得更加稳定，模型获得了更好的性能。

同年，为了提高和改善分割效果，Qi 等人[15]在此基础上提出了 PointNet++，该网络由 PointNet 构成的特征提取块组成，并采用了 MSG、MRG 以及特征传播改进网络架构，输入沿着多分辨率层次以逐渐变大的比例捕获特征。虽然在一些数据集上的结果提高不是很多，但是也提高了架构对于稀疏点的鲁棒性。PointNet 网络操作虽然可以对点云数据直接进行处理且运算简单，利用对称函数解决了点云无序性的问题，但依旧存在着很多的缺陷：首先，其无法很好地捕捉由度量空间引起的局部结构信息；其次，没有考虑每个点与其邻近点的交互关系，从而导致无法高效刻画相关区域的语义结构；而且，统一的模板无法有效地解决密度不均一的数据。

3.3.2 ShapeContextNet

为了解决以上问题，研究者们从 PointNet 算法出发，进行了一系列的探索。Xie 等人[16]也将注意力机制引入点云分割中，并将 PointNet 作为骨干网络对点云数据进行多角度池化，然后采用共享权重的多层感知器获取自适应注意力权重，从而提升了网络性能。

论文首先简要介绍了经典的形状上下文描述符，该描述符被发明时是用于二维形状匹配和识别，一个主要贡献是设计了具有空间不均匀单元的形状上下文描述符。集合中每个点的邻域信息是通过计算每个单元内的相邻点的数量来获取的。因此，每个点的形状描述符是与单元数量相同维度的特征向量（直方图），每个特征维度描述每个单元内的点数量（归一化）。形状上下文描述符使用高维向量（直方图）对丰富的上下文形状信息进行编码，该向量特别适合于以分散点的形式匹配和识别对象。

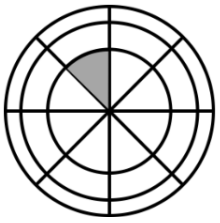

图 6 具有 24 个容器块（$n_r = 3$ $n_\theta = 8$）的 2D 形状上下文内核示例。

形状上下文使用对数极坐标系来设计容器块。图6显示了我们的方法中使用的基本 2D 形状上下文描述符（请注意，我们使中心单元格变大，这与中心单元格相对较小的原始形状上下文设计略有不同）。

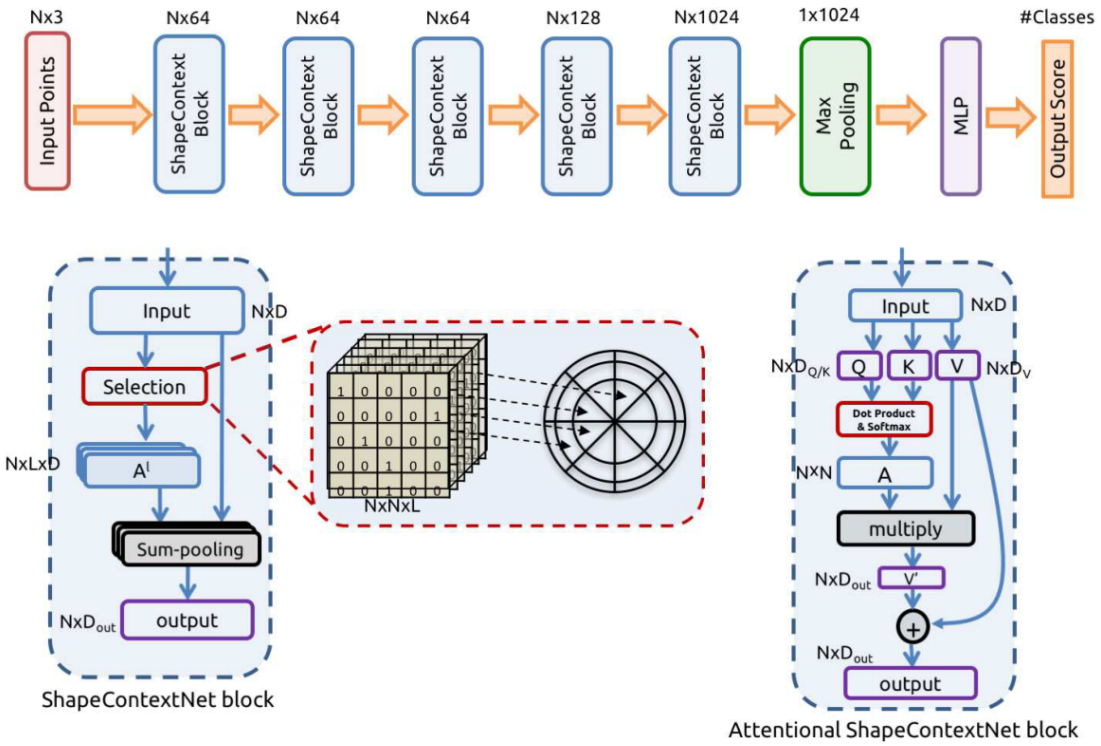

图 7 ShapeContextNet（SCN）和注意力 ShapeContextNet（A-SCN）架构。分类网络有 5 个 ShapeContext 块；每个块以 N 个点特征向量作为输入，并依次应用选择、聚合和变换操作。ShapeContext 块可以通过手工设计的形状上下文内核（SCN 块）或从数据中学习的自我注意机制（A-SCN 块）来实现。

也有人试图将形状上下文扩展到 3D，其中极角 ϕ 和方位角 θ 将空间划分为不同的象限。论文的采用了类似的设计。虽然形状上下文被认为是计算机视觉中最成功的描述符之一，但它与现代深度学习框架的整合还没有得到充分的探索。

之后来介绍形状上下文的通用公式，以构建深层 ShapeContextNet。设一个形状的给定点集（云）为 $P = \{p_1, p_2, ..., p_N\}$。每个 $pi \in R^3$ 是由其三维坐标表示的点。论文提出的 ShapeContextNet（SCN）是一种神经网络架构（如图7所示），其基本构建块是 SCN 块。每个 SCN 块由三个操作组成：选择、聚合和转换，下面将对此进行详细说明。

选择 选择对于 N 个点的点云 P，选择操作是生成亲和矩阵 $a \in \{0\ 1\}^{N \times N}$，其中 $A(i\ j) = 1$ 表示点 p_j 与参考点 p_i 有边，而 $A(i\ j) = 0$ 表示没有连接。以点 p_i 为中心的连接构件是全局形状排列的表示。在原始的形状上下文中，选择操作首先将空间划分为 L 个容器。在这种情况下，我们不需要一个单一的亲和矩阵，而是同时构建 L 个不相交的亲和矩阵，并且 $A^l(i\ j) = 1$ 表示在第 l 个容器两点相连。值得注意选择操作不一定依赖于任何预定义的空间划分，并且可以按照与注意机制相同的方式自动学习，其中 A 是 N×N 注意权重。注意力选择操作可以是硬任务，也可以是软任务。

聚合 在选择操作后，为了在参考点 pi 处形成形状排列的紧凑表示，需要聚合所选点的信息。将聚合函数表示为 m。在原始形状上下文中，对于 N 个点和 L 个容器块，以及一个参考点 p_i，我们有 L 个聚合函数 $m_i^l, l = 1, ..., L$，它们共同构成直方图表示。每个 m_i^l 是一个计数函数，用于计算 $bin(l)$ 中的点数，可以表示为和池化函数 $m_i^l = \sum_j \mathbb{1}[A(i\ j) = 1]$。

转换 在这里添加特征转换函数 f 以合并其他非线性并增加模型的容量。在原始形状上下文中，在构建局部描述符之后，可以为最终的分类任务添加一个判别分类器，例如支持向量机。这种变换可以通过径向基函数等核函数来实现。在深层神经网络的上下文中，MLP 或具有非线性激活函数的卷积层可用于特征转换。

形状上下文块 在介绍了上述三种操作之后，形状上下文描述符 SC 可以表示为：

$$SC_i = f(h_i) = f(h_i(1), ..., h_i(L)) = f([m_i^1, ..., m_i^L])$$

此公式中的每个组件都可以通过反向传播的神经

网络模块实现，因此，与卷积层类似，SC 是可用于构建形状上下文网络的合成块:

$$SCNet = SC_i(SC_i(SC_i(...)))$$

虽然概念简单，具有经典形状上下文描述符的良好特性，如平移不变性，但手工形状上下文内核并不直接，很难在不同的点云数据集（通常具有不同的大小和密度）上推广。这促使我们提出以下基于注意的模型。

受自然语言处理（seq2seq）任务研究的启发，传统的序列到序列模型通常采用递归神经网络（如 LSTM）、外部记忆或时序卷积来捕获上下文信息。积自注意是一种模型，它通过一种轻量级门控机制处理长路径长度上下文建模，其中注意权重矩阵是使用简单的点积生成的。值得注意的是，自注意对输入顺序也是不变的。与传统的基于注意的序列到序列模型不同，在自注意块中，查询向量 $Q \in R^{D_Q}$，键向量 $K \in R^{D_K}$（通常 $D_Q = D_K$）和值向量 $V \in R^{D_V}$ 从相同的输入中学习。在监督分类环境中，可以认为 Q、K 和 V 只是由三个独立的 MLP 层学习的三个特征向量。注意力权重由 Q 和 K 的点积计算，然后与 V 相乘，得到变换后的表示。

考虑在大小为 N 的整个点云 P 上计算自注意力。选择操作生成一个软亲和矩阵，即大小为 N×N 的自注意权重矩阵 a，聚合操作通过点积将值向量 V 与权重矩阵 a 转换:

$$Attention(Q,V,K) = Softmax(\frac{QK^T}{\sqrt{D_Q}}V)$$

模型利用多个子区间填充点云形状，构建区间之间的关联矩阵，之后聚合相关点的特征，完成特征提取，在保证性能的同时减少了空间开销；采用共享权重的多层感知器获取自适应注意力权重，有效从另一种角度解决了点云特征获取获取和无序性问题。

在神经架构搜索领域，搜索过程中的验证精度与评估过程中的测试精度之间存在很大差异，为了解决这一问题，顺序贪婪结构搜索 (Sequential Greedy Architecture Search,SGAS)[17] 利用 PointNet 中的基本单元感知机、最大值池化以及边缘卷积等模块来构造最优的网络结构，该方法在点云分类领域有着明显的性能提升，也为利用自动神经架构搜索来解决点云分割任务提供了一种新的思路。

3.4 基于优化 CNN 的方法

卷积神经网络在目前最主流的方法是用于对二维图像进行特征提取。然而，传统的卷积运算在三维点云数据上的适用性很弱，表征点间关联的能力差，平移和尺度不变性差。因此，一些研究者研究了卷积网络的优化方法，以解决这些问题。

3.4.1 Pointwise

3 维点云数据的无序性和无结构性限制了 CNN 在处理点云数据上的表现。Hua 等人[18] 利用逐点卷积 (pointwise convolution) 获取点的局部特征信息实现语义分割，避免了 PointNet 在训练过程中学习到的不必要的一个对称函数，提高了网络模型执行速度，同时保证了点云分割任务中的准确性。

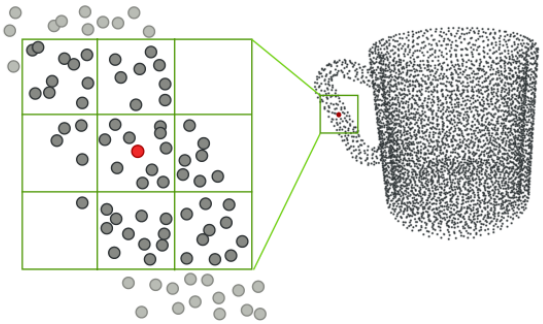

图 8 逐点卷积。对于每个点，动态查询最近邻居并将其分入核单元，然后再与核权重进行卷积。通过将逐点卷积算子叠加在一起，可以构建用于场景分割和点云目标识别的完全卷积神经网络。

卷积核以点云的每个点为中心。内核支持中的相邻点可以构成中心点。每个核都有一个大小或半径值，可以根据每个卷积层中不同数量的相邻点进行调整。图8显示了一个演示此想法的图表。形式上，逐点卷积可以写成

$$x_i^{\ell} = \sum_k w_k \frac{1}{|\Omega_i(k)|} \sum_{p_j \in \Omega_i(k)} x_j^{\ell-1}$$

其中 k 迭代内核支持中的所有子域；$\Omega_i(k)$ 是以点 i 为中心的核的第 k 子域；p_i 是点 i 的坐标；$|\cdot|$ 统计子域内的所有点；w_k 是第 k 子域的核权重，x_i 和 x_j 是点 i 和 j 的值，以及 $\ell 1$ 和 ℓ 输入和输出层的索引。

总结一下，Pointwise 针对不规则点云使用固

定大小的卷积核，对卷积核内的点都和权重相乘得出卷积核的平均值，论文提出的卷积操作简单有效，并证明排序对分类的作用，但是对于大规模点云场景理解中的应用还有待研究。

与上述思路不同，Xu 等人[19]提出了一种新的卷积结构 SpiderCNN。该卷积单元 SpiderConv 通过参数化一系列基于泰勒多项式的卷积滤波器，将卷积运算从规则的数据结构扩展到不规则 3 维点云数据上，捕获复杂的局部几何特征，能够提取语义深层特征。环形 CNN(Annularly-CNN)[20]提出了一种环形卷积，在 3 维点云上直接定义了计算卷积的新方法。这种新的卷积算子通过在计算中指定环形结构和方向，可以更好地捕捉每个点的局部邻域几何。在信号处理层面上，该算法能够适应几何变异性和可扩展性。然而上述方法在点云特征抽象时无法充分参考局部形状信息，因此对局部形状变化感知的鲁棒性较差，难以针对目标形状特征产生合适的卷积核。

3.4.2 RS-CNN

针对 CNN 网络欠缺表征点间联系的能力，Liu 等人[21]针对点云数据提出了关系形状 CNN(Relation Shape CNN,RS-CNN)，RS-CNN 的关键是从关系中学习，即点之间的几何拓扑约束。局部点集的卷积权重被迫从预定义的几何先验中学习该点集的采样点与其他点之间的高级关系表达式。通过这种方式，可以获得关于点的空间布局的具有显式推理的归纳局部表示，从而提高形状感知和鲁棒性。将这种卷积作为基本算子 RS-CNN，可以开发一种层次结构，以实现点云分析的上下文形状感知学习。通过这种方式，可以获得关于点的空间布局的具有显式推理的归纳局部表示，大大提高了网络的形状感知能力和鲁棒性。

建模 为了克服这个问题，我们对局部点子集 $P_{sub} \in \mathbb{R}^3$ 建模为球面邻域，采样点 x_i 为质心，周围点为邻域 $x_j \in \mathcal{N}(x_i)$。图最左边的部分解释了建模。论文的目标是学习这个邻域的 $f_{P_{sub}}$ 的归纳表示，它应该对底层的形状信息进行区分编码。为此，我们将一般卷积运算公式化为

$$\mathbf{f}_{P_{\text{sub}}} = \sigma\left(\mathcal{A}\left(\left\{\mathcal{T}\left(\mathbf{f}_{x_j}\right), \forall x_j\right\}\right)\right)^1, d_{ij} < r \forall x_j \in \mathcal{N}(x_i)$$

其中 x 是一个 3D 点，f 是一个特征向量。d_{ij} 是 x_i 和 x_j 之间的欧氏距离，r 是球面半径。这里，$f_{P_{sub}}$ 是通过首先用函数 $\mathcal{T}$ 变换 $\mathcal{N}(x_i)$ 中所有点的特征，然后用函数 $\mathcal{A}$ 聚集它们，然后用非线性激活器 σ 获得的。在此公式中，两个功能 $\mathcal{A}$ 和 $\mathcal{T}$ 是 $f_{P_{sub}}$ 的关键。也就是说，只有当 $\mathcal{A}$ 对称（例如求和）且 $\mathcal{T}$ 在 $\mathcal{N}(x_i)$ 中的每个点上共享时，才能实现点集的置换不变性。

通道提升映射 在上述公式中，$f_{P_{sub}}$ 的信道编号与输入特性 f_{x_j} 相同。这与经典的图像 CNN 不一致，CNN 增加了通道数，同时降低了图像分辨率以获得更抽象的表示。因此，作者在 $f_{P_{sub}}$ 添加了一个共享 MLP，用于进一步的信道提升映射。如图9的中间部分所示。

总的来说，RS-CNN 从几何关系中推理学习 3D 形状，使用点云的形状关系数据，利用多层感知机网络，学习出携带着点云形状关系信息的卷积核参数。它提出的卷积操作能学习点之间的几何拓扑约束，而在场景分割问题中的应用还有待研究。

而 Liu 等人提出的 DensePoint[22]通过泛化卷积运算符将规则网格 CNN 扩展到不规则点配置，该卷积运算符保留点的排列不变性，并实现有效的归纳学习局部特征。在结构上，它属于密集连接模式，反复聚合多级和深层次的多尺度语义。

3.4.3 3D-GCN

针对现有工作对点云平移和尺度不变性差的问题 Lin 等人[23]提出了一种 3 维图形卷积网络 3D-GCN，用于跨尺度从点云中提取局部 3 维特征，使用图最大池化机制定义可学习内核，使得网络具有很好的平移和尺度不变性。

Wu 等人[24]提出一个新的卷积操作 PointConv。其设计了一种新的计算权函数的方法，从而更好地扩展网络并提升网络的性能。利用多层感知器网络学习权函数，通过核密度估计学习密度函数，学习到的卷积核在 3 维点集上具有平移不变性和置换不变性。

3.5 基于图卷积的方法

点云数据与 2 维图像不同，属于非欧几何数据。如之前所说，传统的 CNN 并不适合直接处理点云学习任务，然而图结构可以很好的处理非欧数据。基于图卷积 (Graph Convolution Network, GCN) 的方法直接在图结构上进行卷积运算，依靠节点

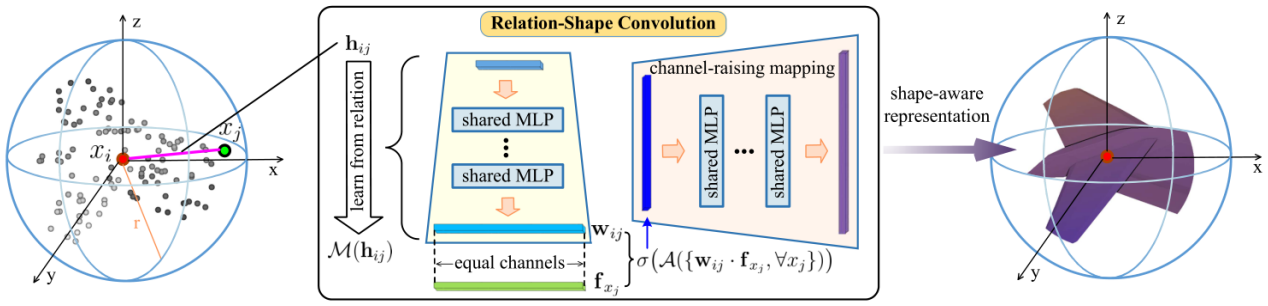

图 9 关系形状卷积（RS Conv）概述。关键是从关系中学习。具体而言，将 x_j 的卷积权重转换为 w_{ij}，$w_(ij)$ 学习预定义几何关系向量 h_{ij} 上的映射 M。这样，归纳卷积表示 $\sigma\left(\mathcal{A}\left(\left\{\mathcal{M}\left(\mathbf{h}_{ij}\right)\cdot\mathbf{f}_{x_j},\forall x_j\right\}\right)\right)$ 可以表达地推理点的空间布局，从而产生有区别的形状感知。如图所示，进一步进行通道提升映射，以获得更强大的形状感知表示。

间的信息传递获取图中的依赖关系，在近年来得到了广泛的应用。

Wang 等人[25]提出了图注意力卷积方法(Graph Attention Convolution Network, GACNet)，它是一种能在点云上端到端进行分割和分类的深度网络，其主要贡献在于将引入注意力机制引入图卷积，通过计算中心点与其邻接点的边缘权重使网络能精确分割的边缘部分。

图注意卷积 图10是 GAC 的图示。GAC 的表达式为：

$$h_i' = \sum_{j\in N(i)} a_{ij} * M_g(h_j) + b_i$$

其中 * 表示逐元素相乘，$M_g : R^F \to R^K$ 为一个特征变换函数，而

$$a_{ij} = \frac{\exp\left(\tilde{a}_{ij,k}\right)}{\sum_{l\in N(i)}\exp(\tilde{a}_{il,k})}, \tilde{a}_{ij} = \alpha\left(\Delta p_{ij}, \Delta h_{ij}\right) \in R^K$$

$$\alpha\left(\Delta p_{ij}, \Delta h_{ij}\right) = M_\alpha\left(\left[\Delta p_{ij} \| \Delta h_{ij}\right]\right)$$

$$\Delta h_{ij} = M_g\left(h_i\right) - M_g\left(h_j\right) \in R^K$$

其中 $\|$ 表示连接操作，$\Delta p_{ij} = p_i - p_j \in R^3$，即顶点 i，j 的坐标差。

图注意卷积网络 作者遵循常见的图像分割架构来组织点云语义分割网络，即图注意卷积网络（GACNet）。不同之处在于，论文的 GACNet 是在点云的图形金字塔上实现的，如图 3 所示。在图金字塔的每个尺度上，GAC 用于局部特征学习。然后使用图池化操作来降低每个特征通道中点云的分辨率。然后，将学习到的特征逐层插值回最佳比例。最后，考虑到多个图池和特征插值层导致的特征保真度损失，在最细尺度上应用额外的 GAC 层进行特征细化。

图池化旨在输出粗化图顶点上的聚合特征。将 H_l' 表示为图金字塔第 l 个尺度的输出特征集，(l+1) 个尺度的输入特征集 H_{l+1} 计算如下：

$$h_v = pooling\{h_l' : j \in \mathcal{N}_l(v)\}$$

其中 $h_v \in H_{l+1}$ 和 $\mathcal{N}_l(v)$ 表示顶点 v 在第 l 个尺度上的邻域。池化函数可以是 max 或 mean 函数，分别对应于最大和平均池化。

总结一下，GACNet 将图卷积注意力与点云任务结合，利用注意力学习边权重来代替固定权重，具体而言，将每个点与其周围点构建一个图结构，在计算中心点与其周围点的边缘权重时，引入注意力机制，差异化点与点之间的联系性。

为了处理更为复杂的大规模点云，有研究者引入了超图结构，现有图结构的边用于连接两个节点，然而在超图结构中，一条边可以有两个或多个节点。图结构中的信息传播可以看作是节点到节点的信息传播，超图神经网络[26] (HyperGraph Neural Networks, HGNN) 则可以看作节点到边再到节点的信息传播。将 HGNN 与图卷积网络等方法进行比较，HGNN 在处理多模态数据时具有优越性，特别是在处理复杂数据时，超图在数据建模方面更加灵活。在已有的关于图研究的工作中，特征之间隐藏的重要关系并没有直接表现在内在结构中。为了解决这个问题，Jiang 等人[27]提出了动态超图神经网络框架 (DynamicHGNN,DHGNN)，它由动态超图构造和超图卷积两个模块组成。超图在最初构造的时候可能需要进行很多调整，由此设计了动态超图构造模块在训练中动态调整超图，然后引入超图卷积对超图结构中的高阶数据关系进行编码。显然，动态超图可以增强模型的鲁棒性。Xu 等人[28]提出了 Grid-GCN 来实现快速其具

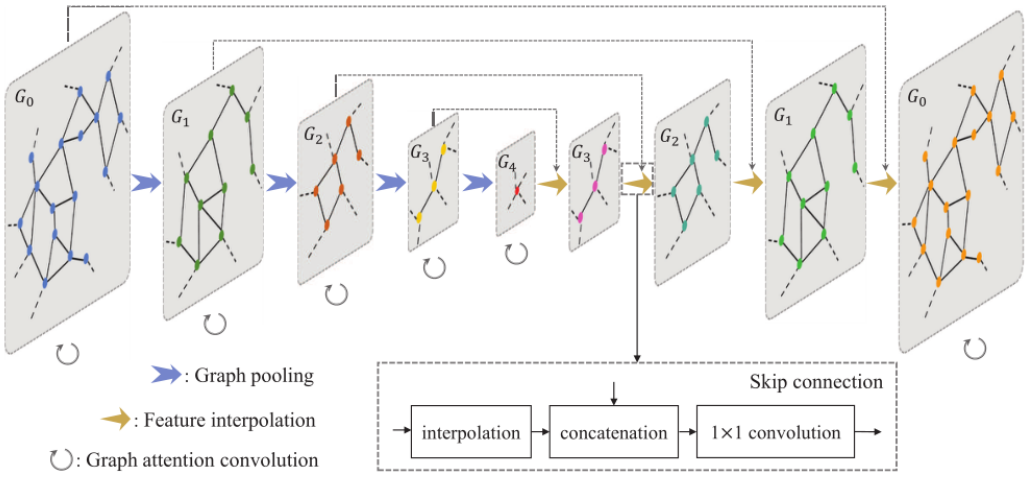

图 10 GACNet 架构。GACNet 构建在点云的图金字塔上。在图金字塔的每个尺度上，提出的 GAC 用于局部特征学习，然后在每个特征通道中使用图池来降低分辨率。然后，将学习到的特征逐层插值回最佳比例，以进行逐点标签分配。

有可拓展性的点云学习。Grid-GCN 通过利用高效的网格空间，不仅仅减低了时间复杂度，而且提升了其空间的覆盖率。Grid-GCN 也实现了当前点云分类和分割的数据集上的非常好的效果。

3.6 基于时序的方法

点云数据的上下文信息往往包含着不同尺度特征间的差异与联系。基于时序的点云分割方法便是将点云的上下文信息进行融合，从而实现更高精度的点云分割。与基于点处理的方法相比，基于时序的方法能够有效的利用点云的上下文信息，从而大大提高点云分割精度和效率。下面在这一部分中，将会介绍一些基于时序的点云分割方法。

3.6.1 Spacial-Context

在基于时序的点云分割方法研究中，Engelmann 等人[29]首次提出了一种分别在输入和输出部分应用空间上下文 (spacial-context) 的方法。基于 PointNet，Spacial-context 在输入部分使用多尺度模块和网格模块，在输出部分使用综合单元和循环综合单元来进一步提取上下文信息。实验结果表明，通过结合空间上下文信息，可以有效提高分割性能。在这里，我们将首先回顾 PointNet 模型，然后介绍扩展上下文的机制，最后介绍 Spacial-Context 模型的两个示例性体系结构。

Input-Level Context 在这个增添的简单模块中，Spacial-context 通过同时考虑一组块来增加网络的上下文信息，而不是像在 PointNet 中那样一次只考虑一个单独的块。上下文信息在组中的所有块之间共享。这些块组要么从同一位置以多个不同的比例（多比例块，见图11，左）选择，或从常规网格中的相邻单元格中选择（网格块，见图12，左）。对于每个输入块，Spacial-context 使用 PointNet 中的机制计算块特征；对于多尺度模块，模型为每个尺度单独训练一个块描述符，以获得与尺度相关的块特征；对于格网块，所有块特征都由共享的单尺度块描述符计算。最后，两种方法都输出与输入块对应的一组块特征。

Output-Level Context 在这一部分，模型进一步整合从上一阶段获得的块特征。这里有两种不同的整合方法：

合并单元（CU）使用一组点特征，通过 MLP 将它们转换为更高维的空间，并应用最大池化层来生成一个公共块特征，该特征再次与每个高维输入特征串联（参见图 8，蓝框）。此过程类似于 PointNet 的块功能机制。最为关键的是，CU 可以链接在一起形成一个序列 CU，形成一个更深层次的网络。我们可以如此理解：一开始，每个点只看到自己的特征。追加块要素后，每个点都会被额外告知其相邻点的要素。通过多次应用 CU，可以增强这种共享知识。

循环整合单元（RCU）是第二种上下文信息整合方法。RCU 将空间上接近区块的一系列区块特征作为输入，并返回一系列相应更新的区块要素。核心思想是创建同时考虑相邻块的块特征。更详细地说，RCU 是作为 RNN 实现的，特别是 GRU[30]，标准 LSTM[31]的更简单的变体。GRU 具有学习远程依赖关系的能力。该范围可以是随时间变化（如

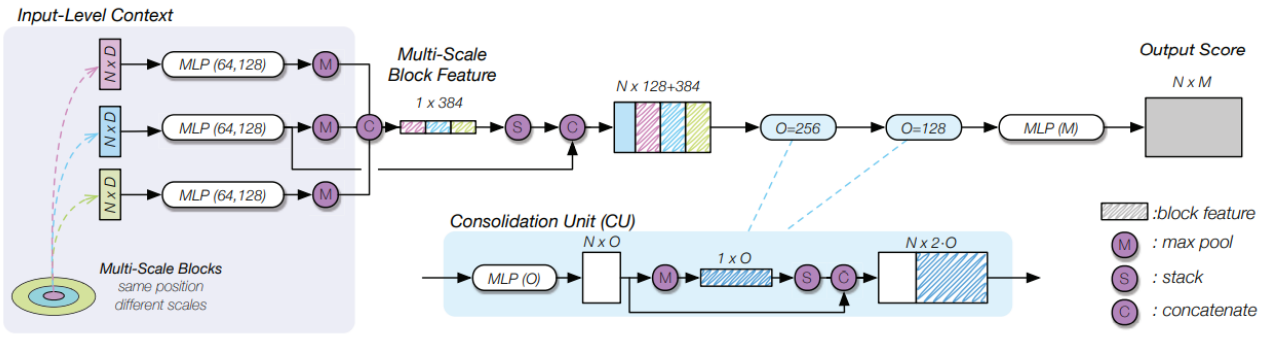

图 11 具有多尺度输入块和合并单元（ms-cu）的体系结构。该网络从多个尺度中获取三个块作为输入，每个块包含 N 个 D 维点。另外，对于每个比例，它学习一个类似于点网机制的块特征。串联的块特征被附加到输入特征，然后由一系列合并单位进行转换。网络输出每分分数。阴影字段表示块要素。

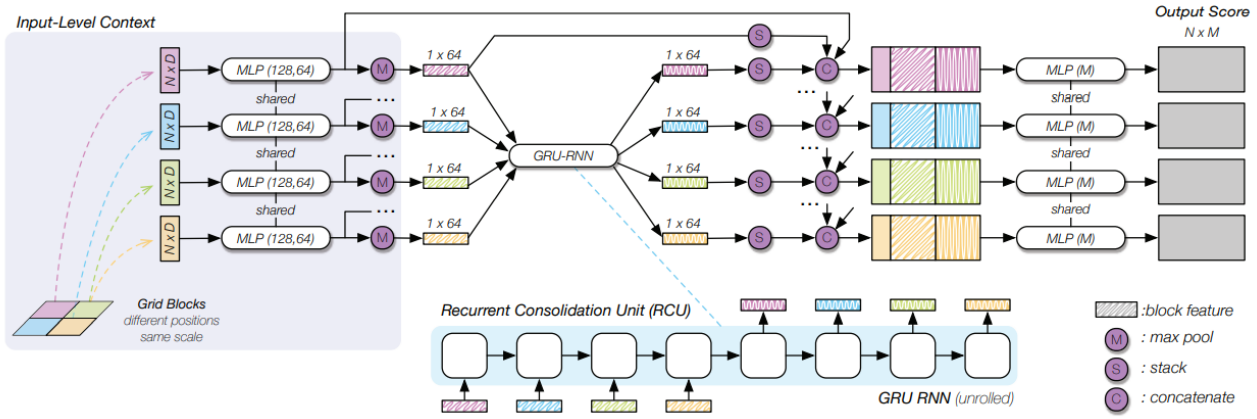

图 12 具有网格输入块和循环整合单元（gb-rcu）的架构。该网络从网格结构中获取四个块作为输入，每个块包含 N 个 D 维点。然后，它使用每个块的相同 MLP 权重来学习块特征。所有块要素都通过一个循环合并单元传递，该单元在所有块之间共享空间上下文信息并返回更新的块特征。更新后的区块要素与原始区块要素一起追加到输入要素中，并用于计算每点分数的输出。阴影字段表示块要素。为清楚起见，省略了一些跳过连接。

语音识别），也可以是空间范围，如这里的情况。展开的 GRU 的单元以不同步的多对多方式连接（参见图12，蓝色框）。这意味着更新的块特征仅在 GRU 看到块特征的整个输入序列时返回。直观地说，GRU 将有关场景的相关信息保留在内部存储器中，并根据新的观察结果进行更新。RCU 使用此存储机制来整合和共享所有输入块中的信息。例如，如果网络记得在房间内更深处看到了一张桌子，那么关于某个点是否属于椅子的决定就会改变。

3.6.2 3DCNN-DQN-RNN

为了实现大规模点云的高效语义解析，Liu 等人[32]将 3DCNN、深层 Q 网络 (Deep Q Network, DQN) 和残差递归神经网络 (Recurrent Neural Network,RNN) 进行融合。该方法将对象定位、分割和分类集成到一个框架中，在 3DCNN 和 DQN 控制下的视窗可以实现有效地定位和分割对象类的点，同时 3DCNN 和残差 RNN 进一步提取了视窗中点的鲁棒性和鉴别性特征，从而大大提高了大规模点云的解析精度。实验结果表明，该方法优于现有的点云分类方法。下面我们将首先介绍模型框架，随后分别介绍模型各个组成部分。

structure 专家预计，未来计算机视觉的进步将来自端到端训练的系统，并使用强化学习将 ConvNets 与 RNN 相结合[33]。这一思路同样可以应用到解析大规模点云中：首先，粗略地查看整个场景并找到目标的大致位置。接下来，专注于对象并将其与背景分开。基于这一思想，便诞生了 3DCNN-DQN-RNN 的基本框架。如下图13。这是一个深度强化学习框架，通过识别每个类对象来语义解析大规模 3D 点云。

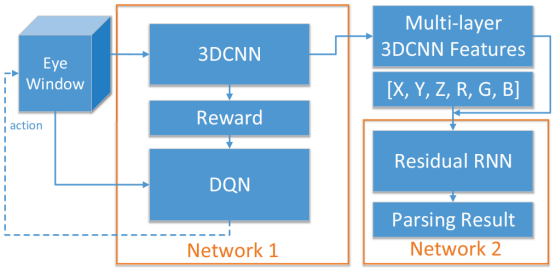

图 13 拟议方法的框架。XYZ 和 RGB 表示每个点的 3D 坐标和 RGB 颜色 P_i 在原始点云中。

Network 1: Recognition and Localization Through 3DCNN-DQN Network 1 的第一部分是 3D CNN。它被训练以确定不同的类对象。同时，每个类的点的特征表示由 3D CNN 获得。然后对这些功能进行编码，并将其应用于以下网络。Network 1 的第二部分是一个 DQN，其目标是检测和本地化对象。根据来自 3D CNN 的反馈，DQN 自动感知场景并调整其参数以定位对象。

为了高效、有效地解析点云，首先设置一定大小的窗口，让它在场景中移动，然后应用 3D CNN 识别窗口内的数据。DQN 获取窗口内包含目标类对象的概率（通过 3D CNN 输出的奖励向量计算）。接下来，它确定哪个区域值得查看，然后使视窗改变其大小并朝向该区域移动。模型重新应用 3D CNN，以获得新视觉窗口中积分的奖励向量。重复该过程，直到视觉窗口准确地包围类对象的点。最后，将 3D CNN 特征和视窗内的所有点作为 Network 2 的输入。图14显示了网络结构。

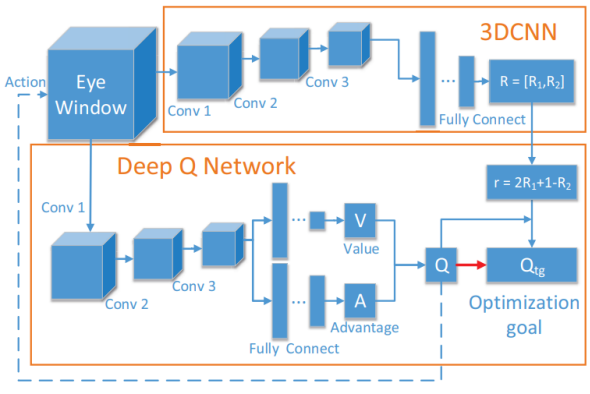

图 14 Network 1

Network 1: Recognition and Localization Through 3DCNN-DQN RNN 进一步学习眼窗口中点的特征。输入到 RNN 的点序列可以作为隐马尔可夫链。当根据其空间排列输入点时，RNN 可以识别多个尺度的特征之间的联系和差异。这些功能是融合或抽象的。然而，点云数据是无序的，它所携带的空间信息可能非常复杂，无法解密。为了完全模拟隐马尔可夫链，RNN 应该足够深，并且包含足够数量的参数来拟合相应的非线性转移函数。作者在模型中构建了一个多层残差 RNN 来满足要求。LSTM 单元用于防止梯度消失，并使网络具有长期记忆。残差块用于防止深度网络降级。

残差 RNN 结构: 每个点 P_k 在视觉窗口中对应于重建的向量 $Vk = [x_k, y_k, z_k, r_k, g_k, b_k, f_1, f_2, f_3]$。其中 x_k，y_k，z_k 是 P_k 的坐标；r_k，g_k，b_k 是 P_k 的颜色；f_1，f_2，f_3 是视觉窗口中每个点的 3D CNN 特征向量。在获得视窗内的所有重建向量后，模型按照原始空间安排将它们输入到残差 RNN 中以进行训练。所使用的残差 RNN 具有 7 个全连接层，3 个压差层，2 个残差块和一个 LSTM 单元，如图15所示。

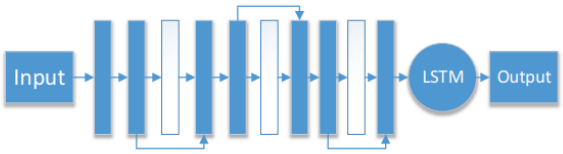

图 15 Network 2 的内部结构。蓝色矩形表示全连接的图层，白色表示下沉层。

LSTM 单元: 每个视觉窗口中的点仍然很大。因此，输入到残差 RNN 的点序列很长。Network 2 需要能够大规模地理解点之间的上下文信息。LSTM 单元使网络能够学习序列的长期依赖性，即点要素的连接和差异。

残差块: 网络太深存在降级问题，并且网络性能会随着层的加深而降低。在点云解析中，深度网络是必要的，但其太深的深度很容易导致高训练误差，这称为降级[34]。受深度残差网络[34]的启发，我们利用残差块的结构在单元之间建立一些重叠关节，以解决退化问题。

深度 RNN-A 多层分类器: 由于 Network 1 获得的特征向量来自不同类型和尺度的对象，因此特征应该在高维空间中有效地融合。如果将特征向量馈送到 SVM 或随机森林等分类器中，则特征表示的自调整可能会受到限制。也就是说，这些分类

器是相对浅层的模型，可能不适合点云分割的问题。而这里模型的多层神经网络可以学习点的位置、空间关系和颜色的判别特征表示，并很好地融合特征，以获得高质量的分类结果。

4 总结与展望

在这篇报告中，我们对基于深度学习点云分割方法进行了整理、分类和简要分析。在搜寻点云分割的相关研究成果，了解其研究进展的过程中，我们注意到，点云分割已经成为近两年三维视觉领域的一个新热点。在深度学习领域，3D 学习方法是一个较新的方向，虽然相对成熟的 2D 学习有所不足，但应用三维深度学习的点云分割方法，已经在各个方面取得了突破及进展。但是同样要注意到，虽然深度学习可以在点云分割任务上取得出色的成果，但由于深度学习本身的特性以及三维点云数据结构的复杂性，应用三维深度学习的点云分割将会带来极大的开销。而这也是之后的研究者们将致力解决的问题。

本文对各种应用三维深度学习的点云分割方法进行了简单的总结与介绍。在前面的分析中，我们已经可以了解到各种方法的特性与优劣之处：对于三维卷积方法，其可以展现出在不同情况下超越同期其他方法性能的能力，但过大的计算储存开销限制了它的深入发展；投影和视图的方法可以实现在一些简单的场景下得到极佳的性能，但由于其丢失空间特征的影响，此方法在处理复杂任务时便显得力不从心。

我们还可以看到，无序与有序的点方法近来出现融合的趋势。点云虽然无序，但点云所表达的物理含义却有着前后联系。因此，不能够忽视拓扑学信息所起到的作用。在深度学习技术的发展中，几何学、数学方法等也具有着相当的意义。最近的研究表明，通过数学的规范公理化语言描述，能增强深度学习的方法的理论性、确定性和可复现性。这对于点云分割的研究无疑是新的启示。

基于这些优势与不足，我们得以展望点云分割的未来发展方向与应用前景。

首先是训练数据集及应用场景。深度学习方法要求极大量的数据集进行训练，这在三维点云分割上更是如此。现有的数据集无法满足点云分割在深度学习领域的发展，因此，构建数据量丰富且全面的数据集是目前的当务之急. 同时，现有数据集对于不同场景的数据采集不够全面，这极大限制了点云分割方法的应用广泛性。在这种情况下，收集并标注相应的数据集对点云分割技术的发展十分重要。

其次是实时点云分割。在前面也有所提及，虽然应用深度学习方法的点云分割在精度等性能方面有了极大的进展，但随之而来的是模型的复杂度的提高以及运行速度的下降。近年来，物联网等技术的快速发展要求具有更高实时性的点云分割方法。因此，如何在保证精度的前提下将模型轻量化，以提高运行速度实现实时点云分割，是未来的研究方向之一。

随后是遥感点云分割。目前的点云分割方法在针对遥感点云的语义分割的问题上不够成熟，无法处理复杂的遥感图像。与此同时，现行的整体精度等评价指标在遥感领域并不适用，对于特定目标的准确度才是遥感领域中更为重要的问题。因此，如何开拓三维遥感点云分割这片荒地，将是未来点云分割的研究热点之一。

最后，弱监督或无监督点云分割。弱监督和无监督方法可以支持在少量的标注数据或无标注数据下进行训练，由此可以大大减少数据标注成本及时间。这种方法在计算机视觉领域已经有了很大的进展。在当前点云标注就成为点云分割研究的难题情况下，可以预见，随着对点云数据的分析和研究加深，弱监督或无监督点云分割将会成为未来研究的发展方向。

参考文献

[1] QI C R, LIU W, WU C, et al. Frustum pointnets for 3d object detection from rgb-d data[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 918-927.

[2] KLOKOV R, LEMPITSKY V. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models[J]. international conference on computer vision, 2017.

[3] CHEN X, CHEN Y, NAJJARAN H. 3d object classification with point convolution network[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 783-788.

[4] CHEN X, MA H, WAN J, et al. Multi-view 3d object detection network for autonomous driving[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2017: 1907-1915.

[5] ZHOU D, FANG J, SONG X, et al. Joint 3d instance segmentation and object detection for autonomous driving[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1839-1849.

[6] QI C R, SU H, MO K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[7] BHANU B, LEE S, HO C C, et al. Range data processing: Representation of surfaces by edges.[C]//1985.

[8] JIANG X, MEIER U, BUNKE H. Fast range image segmentation using high-level segmentation primitives[J]. workshop on applications of computer vision, 1996.

[9] FISCHLER M A, BOLLES R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of The ACM, 1981.

[10] SU H, MAJI S, KALOGERAKIS E, et al. Multi-view convolutional neural networks for 3d shape recognition[J]. international conference on computer vision, 2015.

[11] 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.

[12] HORN B K P. Extended gaussian images[J]. Proceedings of the IEEE, 1984, 72(12): 1671-1686.

[13] YOU H, FENG Y, ZHAO X, et al. Pvrnet: Point-view relation neural network for 3d shape recognition[J]. national conference on artificial intelligence, 2019.

[14] LE T, DUAN Y. Pointgrid: A deep network for 3d shape understanding[J]. computer vision and pattern recognition, 2018.

[15] QI C R, YI L, SU H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in neural information processing systems, 2017, 30.

[16] 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 (CVPR). 2018.

[17] LI G, QIAN G, DELGADILLO I C, et al. Sgas: Sequential greedy architecture search[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 1620-1630.

[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 (CVPR). 2018.

[19] XU Y, FAN T, XU M, et al. Spidercnn: Deep learning on point sets with parameterized convolutional filters[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[20] 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 (CVPR). 2019.

[21] LIU Y, FAN B, XIANG S, et al. Relation-shape convolutional neural network for point cloud analysis[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

[22] LIU Y, FAN B, MENG G, et al. Densepoint: Learning densely contextual representation for efficient point cloud processing[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2019.

[23] 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 (CVPR). 2020.

[24] 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 (CVPR). 2019.

[25] WANG L, HUANG Y, HOU Y, et al. Graph attention convolution for point cloud semantic segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

[26] FENG Y, YOU H, ZHANG Z, et al. Hypergraph neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 3558-3565.

[27] JIANG J, WEI Y, FENG Y, et al. Dynamic hypergraph neural networks.[C]//IJCAI. 2019: 2635-2641.

[28] XU Q, SUN X, WU C Y, et al. Grid-gcn for fast and scalable point cloud learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.

[29] ENGELMANN F, KONTOGIANNI T, HERMANS A. Exploring spatial context for 3d semantic segmentation of point clouds[C]//2017 IEEE International Conference on Computer Vision Workshops(ICCV). 2017.

[30] K C, B V M, Ç G, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[C]//EMNLP. 2014.

[31] S H, J S. Long short-term memory[C]//Neural computation. 1997.

[32] LIU F, LI S, ZHANG L. 3dcnn-dqn-rnn: A deep reinforcement learning framework for semantic parsing of large-scale 3d point clouds[C]//2017 IEEE International Conference on Computer Vision(ICCV). 2017.

[33] Y L, Y B, G H. Look wider to match image patches with convolutional neural networks[C]//Nature. 2015.

[34] K H, X Z, S R, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR). 2016.