3D点云综述

论文链接： Deep Learning for 3D Point Clouds: A Survey
作者及团队：国防科大 & 中山大学 & 牛津大学
会议及时间： Arxiv 2019
code： https://github.com/QingyongHu/SoTA-Point-Cloud

文章目录

Abstract

由于点云学习在计算机视觉，自动驾驶和机器人等许多领域的广泛应用，近来引起了越来越多的关注。深度学习作为AI中的主要技术，已成功用于解决各种2D视觉问题。但是，由于使用深度神经网络处理点云所面临的独特挑战，因此点云上的深度学习仍处于起步阶段。近年来，在点云上的深度学习甚至变得蓬勃发展，提出了许多方法来解决该领域的不同问题。为了激发未来的研究，本文对点云深度学习方法的最新进展进行了全面回顾。它涵盖了三个主要任务，包括3D形状分类，3D对象检测和跟踪以及3D点云分割。它还提供了一些可公开获得的数据集的比较结果，以及有见地的观察结果和对未来研究方向的启发.

索引词-深度学习，点云，3D数据，形状分类，对象检测，对象跟踪，场景流，实例分割，语义分割，场景理解。

1.Introduction

随着3D采集技术的飞速发展，3D传感器变得越来越可用和负担得起，包括各种类型的3D扫描仪，LiDAR和RGB-D相机（例如Kinect，RealSense和Apple深度相机）[1]。这些传感器获取的3D数据可以提供丰富的几何，形状和比例信息[2]，[3]。与2D图像互补，3D数据为更好地了解机器周围环境提供了机会。 3D数据在不同领域具有众多应用，包括自动驾驶，机器人技术，遥感，医学治疗和设计行业[4]。

3D数据通常可以用不同的格式表示，包括深度图像，点云，网格和体积网格。 作为一种常用格式，点云表示将原始几何信息保留在3D空间中，而不会进行任何离散化。因此，它是诸如自动驾驶和机器人技术之类的许多场景理解相关应用程序的首选表示法。最近，深度学习技术已经占据了许多研究领域，例如计算机视觉，语音识别，自然语言处理（NLP）和生物信息学。 因此，在3D点云上进行深度学习仍然面临数个重大挑战[5]，例如数据集规模小，维数高和3D点云的非结构化性质。 在此基础上，本文着重分析用于处理3D点云的深度学习方法。

点云上的深度学习一直吸引着越来越多的关注，尤其是在过去的五年中。还发布了一些公开可用的数据集，例如ModelNet [6]，ShapeNet [7]，ScanNet [8]，Semantic3D [9]和KITTI Vision Benchmark Suite [10]。这些数据集进一步推动了对3D点云的深度学习的研究，提出了越来越多的方法来解决与点云处理相关的各种问题，包括3D形状分

类，3D对象检测和跟踪以及3D点云分割。也很少有关于3D数据的深度学习调查，例如[11]，[12]，[13]，[14]。但是，我们的论文是第一个专门针对点云的深度学习方法的论文。此外，本文全面涵盖了分类，检测，跟踪和分段等不同应用。图1显示了3D点云的现有深度学习方法的分类。

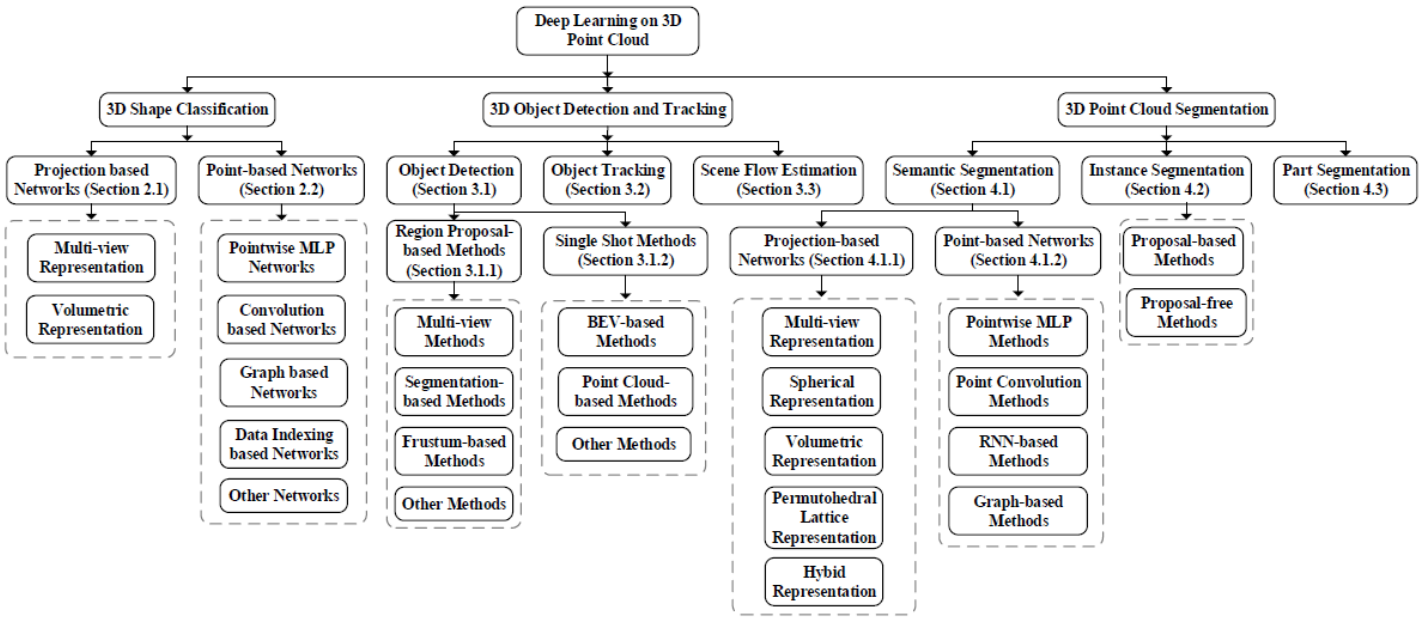


Fig. 1: A taxonomy of deep learning methods for 3D point clouds.

有文献相比，这项工作的主要贡献可以归纳如下：

1）据我们所知，这是第一份全面涵盖针对几个重要点云相关任务的深度学习方法的调查论文，包括3D形状分类，3D对象检测和跟踪以及3D点云分割。

2）与现有评论[11]，[12]相反，我们特别关注于针对3D点云的深度学习方法，而不是针对所有类型的3D数据。

3）本文涵盖了点云上深度学习的最新和最先进的进展。因此，它为读者提供了最新的方法。

4）提供了一些公开可用数据集上现有方法的全面比较（例如，表1、2、3、4），并给出了简短的摘要和有见地的讨论。

本文的结构如下。第2节回顾了3D形状分类的方法。第3节概述了3D对象检测和跟踪的现有方法。第4节概述了点云分割方法，包括语义分割，实例分割和零件分割。最后，第5节总结了论文。我们还在以下位置提供了定期更新的项目页面：https://github.com/QingyongHu/SoTA-Point-Cloud.

2.3D形状分类

这些方法通常首先学习每个点的嵌入，然后使用聚合方法从整个点云中提取全局形状嵌入。最后通过几个完全连接的层实现分类。基于对每个点进行特征学习的方式，现有的3D形状分类方法可以分为基于投影的网络和基于点的网络。图2说明了几种里程碑方法。

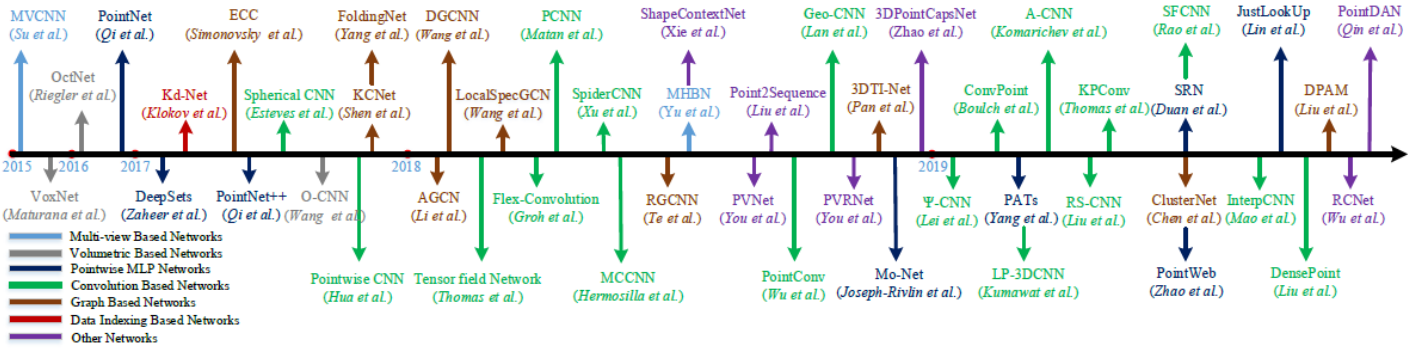


Fig. 2: Chronological overview of 3D shape classification networks.

基于投影的方法首先将非结构化的点云投影到中间的正则表示中，然后利用成熟的2D或3D卷积实现形状分类。相反，基于点的方法可直接在原始点云上运行，而无需任何体素化或投影。**基于点的方法不会造成明显的信息丢失，并且越来越受欢迎**。在本文中，我们主要关注基于点的网络，但为了完整起见，也很少包含基于投影的网络。

2.1 基于投影的网络

这些方法将3D点云投影到不同的表示形式（例如多视图，体积表示）中，用于特征学习和形状分类。

2.1.1多视图表示

这些方法首先将3D对象投影到多个视图中，并提取相应的按视图方向的特征，然后融合这些特征以进行准确的对象识别。如何将多个基于视图的功能聚合到一个可区分的全局表示中是一个关键挑战。 MVCNN [15]是一项开创性的工作，它只是将多视图特征最大池化为一个全局描述符。但是，最大池化只能保留特定视图中的最大元素，从而导致信息丢失。 MHBN [16]通过协调双线性池整合了局部卷积特征，以生成紧凑的全局描述符。杨等。 [17]首先利用关系网络来利用一组视图之间的相互关系（例如，区域-区域关系和视图-视图关系），然后将这些视图进行聚合以获得具有区别性的3D对象表示。另外，还提出了其他几种方法[18]，[19]，[20]，[21]，以提高识别精度。

2.1.2体积表示

早期方法通常在3D点云的体积表示基础上应用3D卷积神经网络（CNN）。 Daniel等文献[22]介绍了一种称为VoxNet的体积占用网络，以实现可靠的3D对象识别。 Wu等[6]提出了一种基于卷积深度信念的3D ShapeNet，以学习各种3D形状中点的分布。 3D形状通常由体素网格上二进制变量的概率分布表示。尽管已经实现了令人鼓舞的性能，但是这些方法无法很好地缩放到密集的3D数据，因为计算和内存占用量随分辨率呈三次方增长。为此，引入了层次结构和紧凑的图结构（例如八叉树）以减少这些方法的计算和存储成本。 OctNet [23]首先使用混合网格-八叉树结构对点云进行分层划分，该结构表示沿着规则网格具有多个浅八叉树的场景。八叉树的结构使用位字符串表示进行有效编码，并且每个特征向量体素通过简单的算术索引。 Wang等 [24]提出了一种基于Octree的CNN用于3D形状分类。在最细的叶子八分位数中采样的3D模型的平均法线向量被馈送到网络中，并将3D-CNN应用于3D形状表面所占据的八分位数。与基于密集输入网格的基准网络相比，OctNet对于高分辨率点云所需的内存和运行时间要少得多。 Le等[25]提出了一种称为PointGrid的混合网络，该网络集成了点和网格表示，以进行有效的点云处理。在每个嵌入的体积网格单元中采样恒定数量的点，这使网络可以使用3D卷积提取几何细节。

2.2 基于点的网络

根据用于每个点的特征学习的网络体系结构，该类别中的方法可分为点式MLP，基于卷积，基于图，基于数据索引的网络和其他典型网络。

2.2.1点对点MLP网络

这些方法使用几个多层感知器（MLP）独立地对每个点建模，然后使用对称函数聚合全局特征，如图3所示。这些网络可以实现无序3D点云的置换不变性。但是，没有完全考虑3D点之间的几何关系。

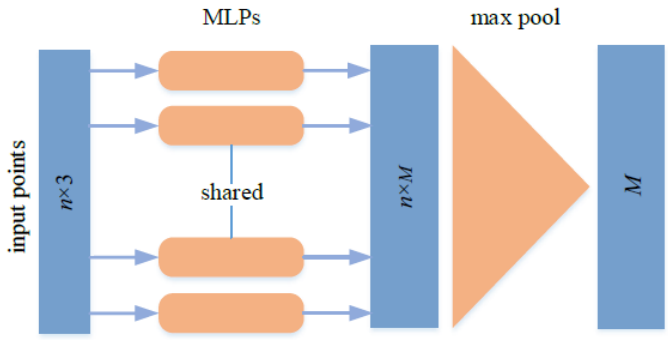


Fig. 3: The architecture of PointNet. n denotes the number of input points, M denotes the dimension of the learned features for each point. After max pooling, the dimension of the global feature of the whole point cloud is also M.

作为一项开创性的工作，PointNet [5]通过几个MLP层学习逐点特征，并通过最大池化层提取全局形状特征。使用几个MLP层获得分类分数。 Zaheer等。 [26]还从理论上证明了实现置换不变性的关键是对所有表示求和并应用非线性变换。他们还为包括形状分类在内的各种应用设计了一种基本架构DeepSets [26]。

由于对于PointNet [5]中的每个点都是独立学习特征的，因此无法捕获点之间的局部结构信息。因此，齐等。 [27]提出了一个层次网络PointNet ++来捕获每个点附近的精细几何结构。作为PointNet ++层次结构的核心，其集合抽象级别由三层组成：采样层，分组层和PointNet层。通过堆叠几个设置的抽象级别，PointNet ++可以从局部几何结构中学习特征，并逐层抽象局部特征。

由于其简单性和强大的表示能力，已经基于PointNet [5]开发了许多网络。 Achlioptas等[28]介绍了一种深度自动编码器网络来学习点云表示。它的编码器遵循PointNet的设计，并使用五个1-D卷积层，ReLU非线性激活，批归一化和最大池化操作独立学习点特征。在点注意变压器（PAT）[29]中，每个点都由其自身的绝对位置和相对于其邻居的相对位置表示。然后，使用组随机注意力（GSA）来捕获点之间的关系，并开发了排列不变，可区分且可训练的端到端Gumbel子集采样（GSS）层来学习分层特征。 Mo-Net [30]的体系结构与PointNet [5]类似，但是它需要有限的时间作为其网络的输入。PointWeb[31]也基于PointNet ++构建，并使用本地邻域的上下文来改进点使用自适应功能调整（AFA）的功能。段等。 [32]提出了一种结构关系网络（SRN）来学习使用MLP的不同局部结构之间的结构关系特征。 Lin等[33]通过为PointNet所学习的输入和函数空间构造查找表来加速推理过程。在中等机器上，与PointNet相比，ModelNet和ShapeNet数据集上的推理时间缩短了1.5毫秒和32倍。 SRINet [34]首先投影一个点云以获得旋转不变表示，然后利用基于PointNet的主干来提取全局特征，并利用基于图的聚合来提取局部特征。

2.2.2基于卷积的网络

与在2D网格结构（例如图像）上定义的内核相比，由于点云的不规则性，难以为3D点云设计卷积内核。 根据卷积核的类型，当前的3D卷积网络可以分为连续卷积网络和离散卷积网络，如图4所示。

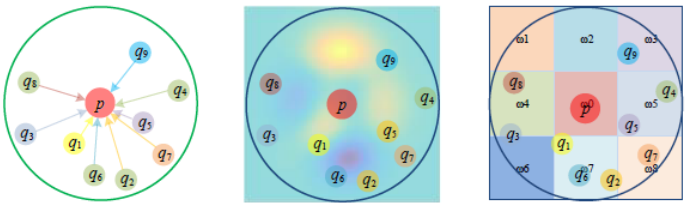


Fig. 4: An illustration of a continuous and discrete convolution for local neighbors of a point. (a) represents a local neighborhood; (b) and (c) represent 3D continuous and discrete convolution, respectively.

3D连续卷积网络。这些方法在连续空间上定义卷积核，其中相邻点的权重与相对于中心点的空间分布有关。

3D卷积可以解释为给定子集的加权和。 MLP是学习权重的一种简单方法。作为RS-CNN的核心层[35]，RS-Conv将某个点周围的局部点子集作为输入，然后通过学习映射使用MLP进行卷积。从低级关系（例如欧几里得距离和相对位置）到局部子集中点之间的高级关系。在[36]中，内核元素是在单位球体内随机选择的。然后使用基于MLP的连续函数在内核元素的位置和点云之间建立关系。在DensePoint [37]中，卷积定义为带有非线性激活器的单层感知器（SLP）。通过串联所有先前层的特征以充分利用上下文信息来学习特征。

一些方法还使用现有算法来执行卷积。在PointConv [38]中，卷积定义为相对于重要性采样的连续3D卷积的蒙特卡洛估计。卷积核由权重函数（通过MLP层学习）和密度函数（通过核化密度估计和MLP层学习）组成。为提高内存和计算效率，将3D卷积进一步减少为两个运算：矩阵乘法和2D卷积。使用相同的参数设置，其内存消耗可减少约64倍。在MCCNN [39]中，卷积被视为依赖样本密度函数（由MLP实现）的蒙特卡洛估计过程。然后使用Poisson磁盘采样来构建点云层次结构。该卷积运算符可用于在两种或多种采样方法之间执行卷积，并可以处理变化的采样密度。在SpiderCNN [40]中，提出了SpiderConv来将卷积定义为阶跃函数与在k个最近邻居上定义的泰勒展开式的乘积。阶跃函数通过对局部测地距离进行编码来捕获粗略的几何形状，泰勒展开通过在立方体的顶点处插值任意值来捕获固有的局部几何形状变化。此外，还基于径向基函数为3D点云提出了卷积网络PCNN [41]。托马斯等。 [42]使用一组可学习的核点为3D点云提出了刚性和可变形核点卷积（KPConv）运算符。
已经提出了几种方法来解决3D卷积网络面临的旋转等变问题。 [43]提出了3D球面卷积神经网络（Spherical CNN）来学习3D形状的旋转等变表示，它以多值球面函数为输入。通过在球形谐波域中用锚点对频谱进行参数化来获得局部卷积滤波器。提出了张量场网络[44]，将点卷积运算定义为可学习的径向函数和球谐函数的乘积，它们局部等价于点的3D旋转，平移和置换。[45]中的卷积定义为在球形互相关上使用通用快速傅里叶变换（FFT）算法实现。基于PCNN，SPHNet [46]通过在体积函数的卷积过程中合并球形谐波核来实现旋转不变性。 ConvPoint [47]将卷积核分为空间和特征部分。从单位球体中随机选择空间部分的位置，并通过简单的MLP学习加权函数。

为了加快计算速度，Flex-Convolution [48]将卷积核的权重定义为k个最近邻居上的标准标量积，可以使用CUDA对其进行加速。实验结果证明了它在具有较少参数和较低内存消耗的小型数据集上的竞争性能。

3D离散卷积网络。这些方法在常规网格上定义卷积核，其中相邻点的权重与相对于中心点的偏移量有关。

华等 [49]将非均匀的3D点云转换为均匀的网格，并在每个网格上定义了卷积核。与2D卷积（为每个像素分配权重）不同，建议的3D内核为落入同一网格的所有点分配相同的权重。对于给定的点，从上一层计算位于同一网格上的所有相邻点的平均特征。然后，对所有网格的平均特征进行加权和求和以产生当前层的输出。 [50]通过将3D球形邻近区域划分为多个体积仓并将每个仓与可学习的加权矩阵相关联，定义了球形卷积核。一个点的球形卷积核的输出由其相邻点的加权激活值平均值的非线性激活确定。在GeoConv [51]中，一个点及其相邻点之间的几何关系是基于六个基础显式建模的。沿基础每个方向的边缘特征根据相邻点的基础由可学习的矩阵独立加权。然后根据给定点及其相邻点形成的角度聚合这些与方向相关的特征。对于给定点，其当前层的特征定义为给定点的特征与其在上一层的相邻边缘特征的总和。 PointCNN [52]通过X-conv转换（通过MLP实现）实现了置换不变性。通过将点特征插值到相邻的离散卷积核量坐标，毛等人[53]提出了一个插值卷积算子InterpConv来测量输入点云和核重量坐标之间的几何关系。张等[54]提出了一个RIConv算子来实现旋转不变性，它以低层旋转不变几何特征作为输入，然后通过一种简单的装箱方法将卷积变成一维。

A-CNN [55]通过围绕查询点每个环上的核大小围绕邻居数组循环定义环形卷积。A-CNN学习局部子集中的相邻点之间的关系。 为了减少3D CNN的计算和存储成本，Kumawat等人（美国） [56]提出了一种基于3D短期傅立叶变换（STFT）的3D局部邻域中的相位提取整流局部相位体积（ReLPV）块，该参数可显着减少参数数量。 在SFCNN [57]中，将点云投影到具有对齐球坐标的规则二十面体网格上。 然后，通过卷积最大池-卷积结构，对从球形晶格的顶点及其相邻像素连接的特征进行卷积。SFCNN抵抗旋转和扰动。

2.2.3基于图的网络

基于图的网络将点云中的每个点视为图的顶点，并基于每个点的邻居为图生成有向边。 然后在空间或频谱域中进行特征学习[58]。 一个典型的基于图的网络如图5所示。

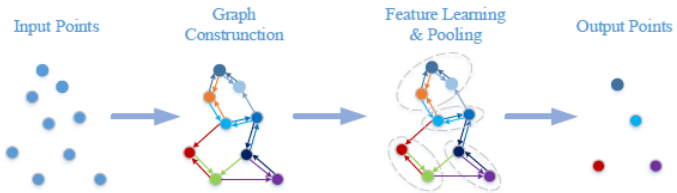


Fig. 5: An illustration of a graph-based network.

空间域中基于图的方法。 这些方法在空间域中定义操作（例如，卷积和池化）。 具体来说，卷积通常是通过对空间邻居的MLP来实现的，通过汇总来自每个点的邻居的信息，合并会生成新的粗化图。 通常为每个顶点的特征分配坐标，激光强度或颜色，而通常为每个边缘的特征分配两个连接点之间的几何属性。

作为开拓性的工作，Simonovsky等人 [58]将每个点视为图的顶点，并通过有向边将每个顶点连接到其所有邻居。然后，使用滤波器生成网络（例如，MLP）提出了边缘条件卷积（ECC）。采用最大池来聚集邻域信息，并基于VoxelGrid [59]算法实现图粗化。对于形状分类，首先对卷积和池进行交织。然后，遵循全局平均池和完全连接的层以产生分类分数。在DGCNN [60]中，在特征空间中构建图，并在网络的每一层之后进行动态更新。作为EdgeConv的核心层，MLP用作每个边缘的特征学习功能，通道方式的对称聚合也应用于与每个点的邻居相关联的边缘特征。此外，LDGCNN [61]删除了转换网络，并将DGCNN [60]中不同层的层次结构链接在一起，以改善其性能并减小模式大小。还提出了一种端到端无监督的深层自动编码器网络（即FoldingNet [62]），以使用向量化局部协方差矩阵和点坐标的级联作为其输入。

Hassani等人受Inception [63]和DGCNN [60]的启发。 [64]提出了一种无监督的多任务自动编码器来学习点和形状特征。编码器是基于多尺度图构造的。解码器是使用三个无监督任务构造的，包括聚类，自监督分类和重构，这些任务与多任务损失一起训练。刘等。 [65]建议
一个基于图卷积的动态点集聚模块（DPAM），将点集聚（采样，分组和合并）的过程简化为一个简单的步骤，该过程通过将集聚矩阵与点特征矩阵相乘来实现。与PointNet ++的分层策略相比，DPAM在语义空间中动态地利用了点之间的关系并聚集了点。

为了利用局部几何结构，提出了KCNet [66]来学习基于核相关性的特征。具体来说，一组表征局部结构的几何类型的可学习点被定义为核。然后，计算核与给定点邻域之间的亲和力。在G3D [67]中，卷积定义为邻接矩阵多项式的变体，池化定义为将Laplacian矩阵和顶点矩阵乘以一个粗化矩阵。 ClusterNet [68]利用严格旋转不变（RRI）模块提取每个点的旋转不变特征，并基于具有监督链接标准的无监督聚集层次聚类方法构造点云的层次结构[69]。首先通过EdgeConv块学习每个子集群中的功能，然后通过最大池聚合。

频谱域中基于图的方法。这些方法将卷积定义为频谱滤波，这是通过将图上的信号与图拉普拉斯矩阵的特征向量相乘来实现的[70]。

为了应对高计算量和非本地化的挑战，Defferrard等人 [71]提出了一个截断的切比雪夫多项式来近似频谱过滤。他们学习的特征图位于每个点的Khops邻居内。注意，特征向量是根据[70] [71]中的固定图拉普拉斯矩阵计算的。相反，RGCNN [72]通过将每个点与点云中的所有其他点连接来构造图，并更新每一层中的图拉普拉斯矩阵。为了使相邻顶点的特征更相似，在损失函数中添加了先验图信号平滑度。为了解决由数据的多种图形拓扑引起的挑战，AGCN [73]中的SGC-LL层利用可学习的距离度量来参数化

图形上两个顶点之间的相似度。从图获得的邻接矩阵使用高斯核和学习距离进行归一化。 [74]提出了一个超图神经网络（HGNN），并通过在超图上应用谱卷积来建立一个超边缘卷积层。

前述方法在全图上运行。为了利用当地的结构信息，王等。 [75]提出了一个端到端的频谱卷积网络LocalSpecGCN来处理一个本地图（它是由k个最近的邻居构造而成的）。此方法不需要对图拉普拉斯矩阵和图粗化层次进行任何离线计算。在PointGCN [76]中，基于来自点云的k个最近邻居构建图，并使用高斯核对每个边进行加权。卷积滤波器在图谱域中定义为Chebyshev多项式。全局池和多分辨率池用于捕获点云的全局和局部特征。 Pan等。 [77]提出了3DTI-Net，方法是在频谱域中对第k个最近的相邻图进行卷积。通过从相对的欧几里得距离和方向距离中学习，可以实现几何变换的不变性.

2.2.4基于数据索引的网络

这些网络是根据不同的数据索引结构（例如octree和kd-tree）构建的。在这些方法中，点特征是从叶节点到树的根节点进行分层学习的。 Lei等[50]提出了一种使用球面卷积核的八叉树引导的CNN（如2.2.2节所述）。网络的每一层都与八叉树的一层相对应，并且球形卷积核应用于每一层。当前层中神经元的值被确定为上一层中所有相关子节点的平均值。与OctNet [23]（基于octree）不同，Kd-Net [78]是使用多个Kd树构建的，每次迭代时都有不同的分割方向。按照自下而上的方法，使用MLP根据非子节点的子代表示来计算它的子代。根节点的特征（描述整个点云）最终被馈送到完全连接的层以预测分类分数。请注意，Kd-Net根据节点的拆分类型在每个级别共享参数。 3DContextNet [79]使用标准的平衡K-d树来实现特征学习和聚合。在每个级别上，首先通过MLP根据局部提示（模拟本地区域中点之间的相互依赖性）和全局上下文提示（模拟一个位置相对于所有其他位置的关系）来学习点特征。然后，使用MLP从非子节点的子节点计算其特征，并通过最大池化对其进行聚合。为了分类，重复以上过程直到获得根节点。

SO-Net网络的层次结构是通过执行点到节点k最近邻居搜索来构建的[80]。具体而言，修改后的置换不变自组织图（SOM）用于对点云的空间分布进行建模。通过一系列完全连接的层，从归一化的点到节点坐标中学习单个点特征。 SOM中每个节点的特征是使用通道方式最大池从与此节点关联的点特征中提取的。然后使用类似于PointNet [5]的方法从节点特征中学习最终特征。与PointNet ++ [27]相比，SOM的层次结构效率更高，并且可以充分利用点云的空间分布。

2.2.5其他网络

除上述方法外，还提出了许多其他方案。在3DmFV [82]中，将点云体素化为统一的3D网格，并根据在这些网格上定义的一组高斯混合模型的似然性来提取费舍尔向量。由于费舍尔向量的分量在所有点上求和，因此所得表示形式不变于点云的顺序，结构和大小。 RBFNet [86]通过聚集稀疏分布的径向基函数（RBF）内核中的特征来显式地建模点的空间分布。RBF特征提取层计算所有内核对每个点的响应，然后对内核位置和内核大小进行优化以在训练过程中捕获点的空间分布。与完全连接的层相比，RBF特征提取层可产生更多区分性特征，同时将参数数量减少几个数量级。赵等。 [85]提出了一种无监督的自动编码器3DPointCapsNet，用于3D点云的通用表示学习。在编码器阶段，首先将逐点MLP应用于点云以提取点无关特征，将其进一步馈送到多个独立的卷积层中，然后通过将多个最大池学习特征图进行级联来提取全局潜在表示。基于无监督的动态路由，学习了强大的代表性潜伏胶囊。 Xie等人从形状上下文描述符的构建中得到启发[89]。鲍勃科夫等人[81]提出了一种新颖的ShapeContextNet体系结构，该方法通过将亲和点选择和紧凑的特征聚合结合起来，并利用点积自关注实现了软对齐操作[90]。 [91]将基于手工制作的点对函数的4D旋转不变描述符输入到4D卷积神经网络中。 Prokudin等。 [92]首先从单位球中随机采样具有均匀分布的基点集，然后将点云编码为到基点集的最小距离，这将点云转换为固定长度相对较小的向量。然后可以使用现有的机器学习方法来处理编码的表示。 RCNet [88]利用标准的RNN和2D CNN来构建用于3D点云处理的置换不变网络。首先将点云划分为平行波束，并沿特定维度分类，然后将每个波束馈入共享的RNN。所学习的特征被进一步馈送到有效的2D CNN中以进行分层特征聚合。为了增强其描述能力，提出了RCNet-E沿不同分区和排序方向集成多个RCNet。 Point2Sequences [87]是另一个基于RNN的模型，可捕获点云局部区域中不同区域之间的相关性。它将从多个区域的局部区域中学习的特征视为序列，并将来自所有局部区域的这些序列馈送到基于RNN的编码器-解码器结构中，以聚合局部区域特征。秦等。 [93]提出了一种基于端到端无监督域自适应的网络PointDAN，用于3D点云表示。为了捕获点云的语义特性，提出了一种自我监督的方法来重构点云，该点云的各个部分已被随机重排[94]。

还提出了几种方法来从3D点云和2D图像中学习。 在PVNet [83]中，从多视图图像中提取的高级全局特征通过嵌入网络投影到点云的子空间中，并通过软关注掩模与点云特征融合。 最后，对融合特征和多视图特征采用残差连接以执行形状识别。 后来，进一步提出了PVRNet [84]，以利用3D点云及其多个视图之间的关系，这些关系是通过关系评分模块学习的。 基于关系得分，原始的2D全局视图功能得到了增强，可用于点单视图融合和点多视图融合。

TABLE 1: Comparative 3D shape classification results on the ModelNet10/40 benchmarks. Here, we only focus on point-based networks and the '#params' means the number of parameters of corresponding model. The 'OA' represents overall accuracy and the 'mAcc' represents mean accuracy in the table. The symbol '-' means the results are unavailable.

Methods		Input	#params (M)	ModelNet40 (OA)	ModelNet40 (mAcc)	ModelNet10 (OA)	ModelNet10 (mAcc)
Pointwise MLP Networks	PointNet [5]	Coordinates	3.48	89.2%	86.2%	-	-
	PointNet++ [27]	Coordinates	1.48	90.7%	-	-	-
	MO-Net [30]	Coordinates	3.1	89.3%	86.1%	-	-
	Deep Sets [26]	Coordinates	-	87.1%	-	-	-
	PAT [29]	Coordinates	-	91.7%	-	-	-
	PointWeb [31]	Coordinates	-	92.3%	89.4%	-	-
	SRN-PointNet++ [32]	Coordinates	-	91.5%	-	-	-
	JUSTLOOKUP [33]	Coordinates	-	89.5%	86.4%	92.9%	92.1%
Convolution-based Networks	Pointwise-CNN [49]	Coordinates	-	86.1%	81.4%	-	-
	PointConv [38]	Coordinates+Normals	-	92.5%	-	-	-
	MC Convolution [39]	Coordinates	-	90.9%	-	-	-
	SpiderCNN [40]	Coordinates+Normals	-	92.4%	-	-	-
	PointCNN [52]	Coordinates	0.45	92.2%	88.1%	-	-
	Flex-Convolution [48]	Coordinates	-	90.2%	-	-	-
	PCNN [41]	Coordinates	1.4	92.3%	-	94.9%	-
	Boulch [36]	Coordinates	-	91.6%	88.1%	-	-
	RS-CNN [35]	Coordinates	-	92.6%	-	-	-
	Spherical CNNs [43]	Coordinates	0.5	88.9%	-	-	-
	GeoCNN [51]	Coordinates	-	93.4%	91.1%	-	-
	Ψ-CNN [50]	Coordinates	-	92.0%	88.7%	94.6%	94.4%
	A-CNN [55]	Coordinates	-	92.6%	90.3%	95.5%	95.3%
	SFCNN [57]	Coordinates	-	91.4%	-	-	-
	SFCNN [57]	Coordinates+Normals	-	92.3%	-	-	-
	DensePoint [37]	Coordinates	0.53	93.2%	-	96.6%	-
	KPConv rigid [42]	Coordinates	-	92.9%	-	-	-
	KPConv deform [42]	Coordinates	-	92.7%	-	-	-
	InterpCNN [53]	Coordinates	12.8	93.0%	-	-	-
	ConvPoint [47]	Coordinates	-	91.8%	88.5%	-	-
Graph-based Networks	ECC [58]	Coordinates	-	87.4%	83.2%	90.8%	90.0%
	KCNet [66]	Coordinates	0.9	91.0%	-	94.4%	-
	DGCNN [60]	Coordinates	1.84	92.2%	90.2%	-	-
	LocalSpecGCN [75]	Coordinates+Normals	-	92.1%	-	-	-
	RGCNN [72]	Coordinates+Normals	2.24	90.5%	87.3%	-	-
	LDGCNN [61]	Coordinates	-	92.9%	90.3%	-	-
	3DTI-Net [77]	Coordinates	2.6	91.7%	-	-	-
	PointGCN [76]	Coordinates	-	89.5%	86.1%	91.9%	91.6%
	ClusterNet [68]	Coordinates	-	87.1%	-	-	-
	Hassani et al. [64]	Coordinates	-	89.1%	-	-	-
	DPAM [65]	Coordinates	-	91.9%	89.9%	94.6%	94.3%
Data Indexing-based Networks	KD-Net [78]	Coordinates	2.0	91.8%	88.5%	94.0%	93.5%
	SO-Net [80]	Coordinates	-	90.9%	87.3%	94.1%	93.9%
	SCN [81]	Coordinates	-	90.0%	87.6%	-	-
	A-SCN [81]	Coordinates	-	89.8%	87.4%	-	-
	3DContextNet [79]	Coordinates	-	90.2%	-	-	-
	3DContextNet [79]	Coordinates+Normals	-	91.1%	-	-	-
Other Networks	3DmFV-Net [82]	Coordinates	4.6	91.6%	-	95.2%	-
	PVNet [83]	Coordinates+Views	-	93.2%	-	-	-
	PVRNet [84]	Coordinates+Views	-	93.6%	-	-	-
	3DPointCapsNet [85]	Coordinates	-	89.3%	-	-	-
	DeepRBFNet [86]	Coordinates	3.2	90.2%	87.8%	-	-
	DeepRBFNet [86]	Coordinates+Normals	3.2	92.1%	88.8%	-	-
	Point2Sequences [87]	Coordinates	-	92.6%	90.4%	95.3%	95.1%
	RCNet [88]	Coordinates	-	91.6%	-	94.7%	-
	RCNet-E [88]	Coordinates	-	92.3%	-	95.6%	-

ModelNet10 / 40数据集是最常用的形状分类数据集。表1显示了通过不同的基于点的网络获得的结果。可以得出以下几点结论：

1）逐点*MLP*网络通常用作其他类型的网络的基本构建块，以学习逐点特征。

2）作为一种标准的深度学习架构，基于卷积的网络可以在不规则的*3D*点云上实现卓越的性能。对于不规则数据，应更加注意离散卷积网络和连续卷积网络。

3）由于其固有的强大能力来处理不规则数据，基于图形的网络近年来引起了越来越多的关注。然而，在频谱域中将基于图的网络扩展到各种图结构仍然是挑战。

4）大多数网络需要将点云下采样为固定的小尺寸。此采样过程将丢弃形状的详细信息。开发可以处理大规模点云的网络仍处于起步阶段*[95]*。

3.3D对象检测与跟踪

在本节中，我们将回顾3D对象检测，3D对象跟踪和3D场景流估计的现有方法。

3.1 3D对象检测

3D对象检测的任务是在给定场景中准确定位所有感兴趣的对象。类似于图像中的对象检测[96]，3D对象检测方法可以分为两类：基于区域提议的方法和单次拍摄方法。图6显示了几种里程碑方法。

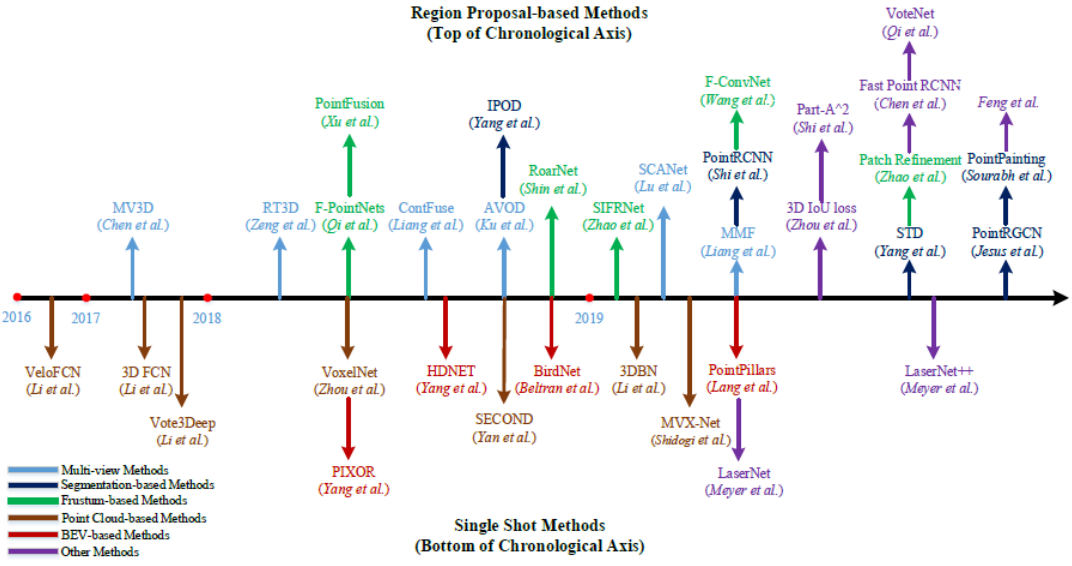


Fig. 6: Chronological overview of the most relevant deep learning-based 3D object detection methods.

3.1.1基于地区提案的方法

这些方法首先提议几个包含对象的可能区域（也称为提议），然后提取区域特征以确定每个提议的类别标签。根据它们的对象建议生成方法，这些方法可以进一步分为三类：基于多视图，基于分段和基于视锥的方法。

多视图方法。这些方法融合了来自不同视图地图的提议特征（例如，LiDAR前视图，鸟瞰图（BEV）和图像）以获得3D旋转框，如图7（a）所示。这些方法的计算成本通常很高。 [4]从BEV地图中生成了一组高度精确的3D候选框，并将其投影到多个视图的特征图（例如LiDAR前视图图像，RGB图像）。然后，他们将这些从不同视图获得的区域特征进行组合，以预测定向的3D边界框，如图7（a）所示。尽管此方法仅在300个提议的情况下在0.25的工会交叉点（IoU）上实现了99.1％的召回率，但其速度对于实际应用而言仍然太慢。随后，从两个方面开发了几种方法来改进多视图3D对象检测方法。

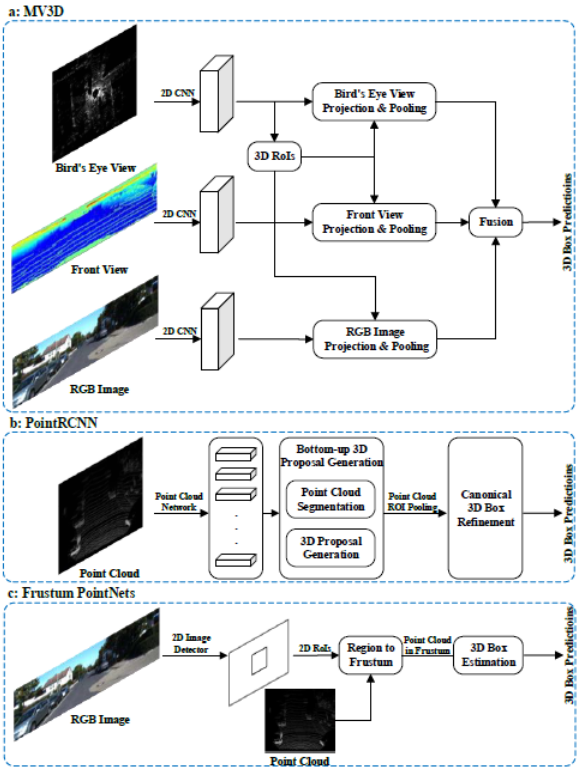


Fig. 7: Typical networks for three categories of 3D object detection methods. From top to bottom: (a) multi-view based, (b) segmentation-based and (c) frustum-based methods.

首先， 已经提出了几种方法来有效地融合不同模态的信息。为了生成对小物体具有较高召回率的3D建议，Ku等人[97]提出了一种基于多模式融合的区域提议网络。他们首先使用裁剪和调整大小操作从BEV和图像视图中提取了大小相等的特征，然后使用逐元素均值合并融合了这些特征。梁等。 [98]利用连续卷积来实现图像和3D LiDAR特征图在不同分辨率下的有效融合。具体而言，他们提取了BEV空间中每个点的最接近的对应图像特征，然后使用双线性插值法将图像特征投影到BEV平面中以获得密集的BEV特征图。实验结果表明，密集的BEV特征图比离散图像特征图和稀疏LiDAR特征图更适合3D对象检测。梁等。 [99]提出了一种用于端到端训练的多任务多传感器3D对象检测网络。具体而言，利用多种任务（例如，2D对象检测，地面估计和深度完成）来帮助网络学习更好的特征表示。进一步利用学习到的跨模态表示来产生高度准确的对象检测结果。实验结果表明，该方法在2D，3D和BEV检测任务上取得了显着改进，并且优于TOR4D基准[100]，[101]上的最新技术。

其次， 已经研究了不同的方法来提取输入数据的鲁棒表示。 Lu等。 [102]通过引入空间通道注意力（SCA）模块探索了多尺度上下文信息，该模块捕获了场景的全局和多尺度上下文并突出了有用的功能。他们还提出了扩展空间非采样（ESU）模块，通过组合多尺度低层特征来获取具有丰富空间信息的高层特征，从而生成可靠的3D对象建议。尽管可以实现更好的检测性能，但是上述多视图方法需要较长的运行时间，因为它们为每个建议执行功能池。随后，曾eng等。 [103]使用预RoI池卷积来提高[4]的效率。具体来说，他们将大多数卷积运算移到了RoI池模块的前面。因此，RoI卷积对于所有对象建议都执行一次。实验结果表明，该方法可以11.1 fps的速度运行，是MV3D的5倍[4]。

基于细分的方法。这些方法首先利用现有的语义分割技术去除大多数背景点，然后在前景点上生成大量高质量的建议以节省计算量，如图7（b）所示。与多视图方法相比[4]， [97]，[103]，这些方法实现了更高的对象召回率，并且更适用于对象被高度遮挡和拥挤的复杂场景。

杨等[104]使用2D分割网络来预测前景像素，并将其投影到点云中以去除大多数背景点。然后，他们在预测的前景点上生成建议，并设计了一个名为PointsIoU的新标准，以减少建议的冗余性和歧义性。继[104]之后，Shi等人[105]提出了PointRCNN框架。具体来说，他们直接分割3D点云以获得前景点，然后融合语义特征和局部空间特征以生成高质量3D框。继[105]的RPN阶段之后，耶稣等人 [106]提出了一项开拓性的工作，以利用图卷积网络（GCN）进行3D对象检测。具体来说，引入了两个模块以使用图卷积精炼对象建议。第一个模块R-GCN利用提案中包含的所有点来实现按提案的特征聚合。第二个模块C-GCN将所有提案中的每帧信息融合在一起，以通过利用上下文来回归准确的对象框。 Sourabh等[107]将点云投影到基于图像的分割网络的输出中，并将语义预测分数附加到这些点上。将绘制的点馈送到现有的检测器[105]，[108]，[109]中，以实现显着的性能改进。杨等。 [110]将每个点与球形锚点关联。然后，将每个点的语义分数用于删除多余的锚点。因此，与先前的方法[104]，[105]相比，该方法以较低的计算成本实现了更高的召回率。此外，提出了一个PointsPool层来学习提案中内部点的紧凑特征，并引入了一个并行的IoU分支以提高定位精度和检测性能。在KITTI数据集[10]的硬集（汽车类别）上优于其他方法[99]，[105]，[111]，并且以12.5 fps的速度运行。

基于视锥的方法。这些方法首先利用现有的2D对象检测器生成对象的2D候选区域，然后为每个2D候选区域提取3D视锥提案，如图7（c）所示。尽管这些方法可以有效地建议3D对象的可能位置，但分步流水线使其性能受到2D图像检测器的限制。

F-PointNets [112]是这个方向的开创性工作。它为每个2D区域生成一个视锥提案，并应用PointNet [5]（或PointNet ++ [27]）来学习每个3D视锥的点云特征以进行模态3D框估计。在后续工作中，Zhao等人[113]提出了一个Point-SENet模块来预测一组比例因子，这些比例因子还用于自适应地突出显示有用的特征并抑制信息量少的特征。他们还将PointSIFT [114]模块集成到网络中以捕获点云的方向信息，从而获得了强大的鲁棒性以进行形状缩放。与F-PointNets [112]相比，该方法在室内和室外数据集[10] [115]上均取得了显着改善。

徐等[116]利用2D图像区域及其对应的平截头体点来精确地回归3D框。为了融合点云的图像特征和全局特征，他们提出了用于框角位置直接回归的全局融合网络。他们还提出了一个密集的融合网络，用于预测每个角的逐点偏移。 Shin等。 [117]首先从2D图像中估计2D边界框和对象的3D姿态，然后提取多个在几何上可行的候选对象。这些3D

候选对象被输入到框回归网络中以预测准确的3D对象框。 Wang等。文献[111]沿着截头圆锥体轴为每个2D区域生成了一系列截头圆锥体，并应用PointNet [5]为每个截头圆锥体提取特征。对视锥级别的特征进行重新生成以生成2D特征图，然后将其输入到完全卷积的网络中以进行3D框估计。该方法在基于2D图像的方法中达到了最先进的性能，并在官方KITTI排行榜中排名第一。 Lehner等。 [118]首先在BEV图上获得了初步的检测结果，然后根据BEV预测提取了小点子集（也称为斑块）。应用局部优化网络来学习补丁的局部特征，以预测高度准确的3D边界框。

其他方法。得益于轴对齐IoU在图像目标检测中的成功，Zhou等人。 [119]将两个3D旋转边界框的IoU集成到几个最先进的检测器[105]，[109]，[120]中，以实现一致的性能改进。 Chen等。 [121]提出了一个两阶段的网络架构，以同时使用点云和体素表示。首先，将点云体素化并馈入3D骨干网络以产生初始检测结果。其次，进一步利用初始预测的内点特征进行框精炼。尽管此设计从概念上讲很简单，但在保持16.7 fps速度的同时，可达到与PointRCNN [105]相当的性能。

受基于Hough投票的2D对象检测器的启发，Qi等 [122]提出了VoteNet直接对点云中对象的虚拟中心点进行投票的方法，并通过汇总投票特征来生成一组高质量的3D对象建议。 VoteNet仅使用几何信息就大大超过了以前的方法，并在两个大型室内基准（即ScanNet [8]和SUN RGB-D [115]）上达到了最先进的性能。然而，对于部分遮挡的物体，虚拟中心点的预测是不稳定的。此外，冯等[123]添加了方向矢量的辅助分支，以提高虚拟中心点和3D候选框的预测精度。此外，在提案之间建立了3D对象-对象关系图，以强调用于精确对象检测的有用功能。 Shi等人的发现启发了3D对象的地面真相框提供对象内部零件的准确位置。 [124]提出了Part A2网络，它由一个部分感知阶段和一个部分聚集阶段组成。零件感知阶段使用具有稀疏卷积和稀疏反卷积的类UNet网络来学习点状特征，以预测和粗略生成对象内零件位置。零件汇总阶段采用RoI感知池，以汇总预测零件的位置，以进行盒评分和位置优化。

3.1.2 Single Shot Methods

这些方法使用单阶段网络直接预测类概率并回归对象的3D边界框。这些方法不需要区域提议的生成和后处理。因此，它们可以高速运行并且非常适合实时应用。根据输入数据的类型，单次拍摄方法可以分为两类：基于BEV的方法和基于点云的方法。

基于BEV的方法。这些方法主要以BEV表示为输入。杨等。 [100]离散化了具有等距间隔的场景的点云，并以类似的方式对反射率进行编码，从而得到规则的表示。然后应用完全卷积网络（FCN）来估计物体的位置和航向角。这种方法在以28.6 fps的速度运行时，胜过大多数单发方法（包括VeloFCN [125]，3D-FCN [126]和Vote3Deep [127]）。后来，杨等人。 [128]利用高清（HD）映射提供的几何和语义先验信息来提高[100]的鲁棒性和检测性能。具体来说，他们从HD地图中获取了地面点的坐标，然后用相对于地面的距离替换了BEV表示中的绝对距离，以弥补由道路坡度引起的平移差异。此外，他们沿通道维度将BEV表示与二进制路罩连接起来，以专注于移动物体。由于高清地图并非随处可见，因此他们还提出了在线地图预测模块，以从单个LiDAR点云中估计地图先验。该地图感知方法在TOR4D [100]，[101]和KITTI [10]数据集上明显优于其基线。但是，它对不同密度的点云的泛化性能很差。 [129]提出了一个标准化图，以考虑不同LiDAR传感器之间的差异。归一化贴图是具有与BEV贴图相同的分辨率的2D网格，它对每个单元中包含的最大点数进行编码。结果表明，该归一化图显着提高了基于BEV的检测器的归纳能力。

基于点云的方法。这些方法将点云转换为常规表示形式（例如2D地图），然后应用CNN预测对象的类别和3D框。

Li等[125]提出了使用FCN进行3D对象检测的第一种方法。他们将点云转换为2D点图，并使用2D FCN预测对象的边界框和置信度。后来，他们[126]将点云离散为一个具有长度，宽度，高度和通道尺寸的4D张量，并将基于2D FCN的检测技术扩展到3D域以进行3D对象检测。与[125]相比，基于3D FCN的方法[126]的准确性提高了> 20％，但是由于3D卷积和数据稀疏性，不可避免地要花费更多的计算资源。为了解决体素的稀疏性问题，Engelcke等人。 [127]利用以特征为中心的投票方案为每个非空体素生成一组投票，并通过累积投票获得卷积结果。它的计算复杂度方法与占用的体素数量成正比。 Li等。 [130]通过堆叠多个稀疏3D CNN构造了3D骨干网。此方法旨在通过充分利用体素的稀疏性来节省内存并加速计算。这个3D骨干网络提取了丰富的3D特征用于对象检测，而不会带来繁重的计算负担。

周等 [108]提出了一个基于体素的端到端可训练框架VoxelNet。他们将点云划分为等距的体素，并将每个体素中的要素编码为4D张量。然后连接区域提议网络以产生检测结果。尽管其性能强，但由于体素稀疏和3D卷积，该方法非常慢。 [120]使用稀疏卷积网络[134]来提高[108]的推理效率。他们还提出了正弦误差角损失，以解决0和方向之间的歧义。 Sindagi等[131]通过在早期融合图像和点云功能扩展了VoxelNet。具体来说，他们将[108]生成的非空体素投影到图像中，并使用预先训练的网络为每个投影体素提取图像特征。然后，将这些图像特征与体素特征相结合，以生成准确的3D框。与[108]，[120]相比，该方法可以有效利用多模式信息来减少误报和漏报。 Lang等人[109]提出了一种名为PointPillars的3D对象检测器。该方法利用PointNet [5]来学习以垂直列（支柱）组织的点云的特征，并将学习到的特征编码为伪图像。然后将2D对象检测管道应用于预测3D边界框。就平均精度（AP）而言，PointPillars优于大多数融合方法（包括MV3D [4]，RoarNet [117]和AVOD [97]）。而且，PointPillars在3D和BEV KITTI [10]基准上都可以以62 fps的速度运行，使其非常适合实际应用。

其他方法。 Meyer等。 [132]提出了一种称为LaserNet的高效3D对象检测器。该方法预测每个点在边界框上的概率分布，然后组合这些每点分布以生成最终的3D对象框。此外，将点云的密集范围视图（RV）表示用作输入，并提出了一种快速均值漂移算法来减少按点预测所产生的噪声。 LaserNet在0至50米的范围内实现了最先进的性能，其运行时间大大低于现有方法。 Meyer等。 [133]然后扩展LaserNet以利用RGB图像（例如50至70米）提供的密集纹理。具体来说，他们通过将3D点云投影到2D图像上来将LiDAR点与图像像素相关联，并利用这种关联将RGB信息融合到3D点中。他们还认为3D语义分割是学习更好的表示形式的辅助任务。这种方法在保持激光（LaserNet）的高效率的同时，在远距离（例如50至70米）目标检测和语义分割方面都取得了显着改善。

3.2 3D对象跟踪

给定对象在第一帧中的位置，对象跟踪的任务是估计其在后续帧中的状态[135]，[136]。由于3D对象跟踪可以使用点云中的丰富几何信息，因此有望克服基于2D图像的跟踪所面临的一些缺点，包括遮挡，照明和比例变化。

受到Siamese网络[137]成功用于基于图像的对象跟踪的启发，Giancola等人[138]提出了一种具有形状完成正则化的3D暹罗网络。具体来说，他们首先使用卡尔曼滤波器生成候选，然后使用形状正则化将模型和候选编码为紧凑的表示形式。余弦相似度然后用于搜索下一帧中被跟踪对象的位置。这种方法可以用作对象跟踪的替代方法，并且明显优于大多数2D对象跟踪方法，包括Staple-CA [139]和SiamFC [137]。为了有效地搜索目标物体，Zarzar等人[140]利用2D连体网络在BEV表示上生成大量的粗略候选对象。然后，他们通过利用3D连体网络中的余弦相似度来优化候选者。这种方法在精度（即18％）和成功率（即12％）方面均明显优于[138]。西蒙等[141]提出了一种语义点云的3D对象检测和跟踪架构。他们首先通过融合2D视觉语义信息生成体素化的语义点云，然后利用时间信息来提高多目标跟踪的准确性和鲁棒性。此外，他们引入了功能强大且简化的评估指标（即“标度-旋转-翻译得分（SRF）”），以加快训练和推理速度。他们提出的Complexer-YOLO提出了令人满意的跟踪性能，并且仍然可以实时运行。

3.3 3D场景流估计

与2D视觉中的光流估计类似，几种方法已经开始从一系列点云中学习有用的信息（例如3D场景流，时空信息）。

刘等[142]提出了FlowNet3D直接从一对连续的点云中学习场景流。FlowNet3D通过流嵌入层学习点级特征和运动特征。但是，FlowNet3D存在两个问题。首先，一些预测的运动矢量在方向上与地面真实情况大不相同。其次，很难将FlowNet应用于非静态场景，尤其是对于以可变形对象为主的场景。为了解决这个问题，王等人[143]引入了余弦距离损失以最小化预测和地面实况之间的角度。此外，他们还提出了点到平面的距离损失以提高刚性和动态场景的精度。实验结果表明，这两个损失项将FlowNet3D的准确性从57.85％提高到63.43％，并加快并稳定了训练过程。 Gu等[144]提出了一种分层多面体格流网（HPLFlowNet）来直接估计来自大规模点云的场景流。提出了几个双边卷积层以从原始点云恢复结构信息，同时降低了计算成本。

为了有效地处理顺序点云，Fan和Yang [145]提出了PointRNN，PointGRU和PointLSTM网络以及一个序列到序列模型来跟踪运动点。 PointRNN，PointGRU和PointLSTM能够捕获时空信息并为动态点云建模。同样，刘等[146]提出MeteorNet直接从动态点云中学习表示。该方法学习从时空相邻点聚合信息。进一步引入直接分组和链流分组来确定时间邻居。但是，上述方法的性能受到数据集规模的限制。米塔尔等[147]提出了两个自我监督的损失来训练他们的网络上的大型未标记的数据集。他们的主要思想是鲁棒的场景流估计方法应该在前向和后向预测中都有效。由于场景流注释的不可用，预测的变换点的最近邻居被视为伪地面实况。但是，真实的地面真实情况可能与最近的点不同。为避免此问题，他们计算了相反方向的场景流，并提出了循环一致性损失以将点转换为原始位置。实验结果表明，这种自我监督方法超越了基于监督学习的方法的最新性能。

3.4 小结

KITTI [10]基准是自动驾驶中最具影响力的数据集之一，已在学术界和工业界普遍使用。表2和表3分别显示了在KITTI 3D和BEV基准测试中，不同检测器所获得的结果。可以观察到以下几点：

- **基于区域提案的方法是这两个类别中研究最频繁的方法，并且在*KITTI*测试*3D*和*BEV*基准上都大大优于单发方法。**
- **现有的*3D*对象检测器有两个局限性：首先，现有方法的远程检测能力相对较差。其次，如何充分利用图像中的纹理信息仍然是 一个开放的问题。**
- **多任务学习是*3D*对象检测的未来方向。例如，*MMF [99]*学习了一种跨模态表示，通过合并多个来实现最新的检测性能 任务。**
- ***3D*对象跟踪和场景流估计是新兴的研究主题，自*2019*年以来逐渐引起越来越多的关注。**

TABLE 2: Comparative 3D object detection results on the KITTI test 3D detection benchmark. 3D bounding box IoU threshold is 0.7 for cars and 0.5 for pedestrians and cyclists. The modalities are LiDAR (L) and image (I). 'E', 'M' and 'H' represent easy, moderate and hard classes of objects, respectively. For simplicity, we omit the '%' after the value. The symbol '-' means the results are unavailable.

Method			Modality	Speed (fps)	Cars			Pedestrians			Cyclists		
					E	M	H	E	M	H	E	M	H
Region Proposal -based Methods	Multi-view Methods	MV3D [4]	L & I	2.8	74.97	63.63	54.00	-	-	-	-	-	-
		AVOD [97]	L & I	12.5	76.39	66.47	60.23	36.10	27.86	25.76	57.19	42.08	38.29
		ContFuse [98]	L & I	16.7	83.68	68.78	61.67	-	-	-	-	-	-
		MMF [99]	L & I	12.5	88.40	77.43	70.22	-	-	-	-	-	-
		SCANet [102]	L & I	11.1	79.22	67.13	60.65	-	-	-	-	-	-
		RT3D [103]	L & I	11.1	23.74	19.14	18.86	-	-	-	-	-	-
	Segmentation -based Methods	IPOD [104]	L & I	5.0	80.30	73.04	68.73	55.07	44.37	40.05	71.99	52.23	46.50
		PointRCNN [105]	L	10.0	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
		PointRGCN [106]	L	3.8	85.97	75.73	70.60	-	-	-	-	-	-
		PointPainting [107]	L & I	2.5	82.11	71.70	67.08	50.32	40.97	37.87	77.63	63.78	55.89
		STD [110]	L	12.5	87.95	79.71	75.09	53.29	42.47	38.35	78.69	61.59	55.30
	Frustum -based Methods	F-PointNets [112]	L & I	5.9	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01
		SIFRNet [113]	L & I	-	-	-	-	-	-	-	-	-	-
		PointFusion [116]	L & I	-	77.92	63.00	53.27	33.36	28.04	23.38	49.34	29.42	26.98
		RoarNet [117]	L & I	10.0	83.71	73.04	59.16	-	-	-	-	-	-
		F-ConvNet [111]	L & I	2.1	87.36	76.39	66.69	52.16	43.38	38.80	81.98	65.07	56.54
		Patch Refinement [118]	L	6.7	88.67	77.20	71.82	-	-	-	-	-	-
	Other Methods	3D IoU loss [119]	L	12.5	86.16	76.50	71.39	-	-	-	-	-	-
		Fast Point R-CNN [121]	L	16.7	84.80	74.59	67.27	-	-	-	-	-	-
		VoteNet [122]	L	-	-	-	-	-	-	-	-	-	-
		Feng et al. [123]	L	-	-	-	-	-	-	-	-	-	-
		Part-A^2 [124]	L	12.5	87.81	78.49	73.51	-	-	-	-	-	-
Single Shot Methods	BEV-based Methods	PIXOR [100]	L	28.6	-	-	-	-	-	-	-	-	-
		HDNET [128]	L	20.0	-	-	-	-	-	-	-	-	-
		BirdNet [129]	L	9.1	13.53	9.47	8.49	12.25	8.99	8.06	16.63	10.46	9.53
	Point Cloud -based Methods	VeloFCN [125]	L	1.0	-	-	-	-	-	-	-	-	-
		3D FCN [126]	L	<0.2	-	-	-	-	-	-	-	-	-
		Vote3Deep [127]	L	-	-	-	-	-	-	-	-	-	-
		3DBN [130]	L	7.7	83.77	73.53	66.23	-	-	-	-	-	-
		VoxelNet [108]	L	2.0	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
		SECOND [120]	L	26.3	83.34	72.55	65.82	48.96	38.78	34.91	71.33	52.08	45.83
		MVX-Net [131]	L & I	16.7	84.99	71.95	64.88	-	-	-	-	-	-
		PointPillars [109]	L	62.0	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
	Other Methods	LaserNet [132]	L	83.3	-	-	-	-	-	-	-	-	-
		LaserNet++ [133]	L & I	26.3	-	-	-	-	-	-	-	-	-

TABLE 3: Comparative 3D object detection results on the KITTI test BEV detection benchmark. 3D bounding box IoU threshold is 0.7 for cars and 0.5 for pedestrians and cyclists. The modalities are LiDAR (L) and image (I). 'E', 'M' and 'H' represent easy, moderate and hard classes of objects, respectively. For simplicity, we omit the '%' after the value. The symbol '-' means the results are unavailable.

Method			Modality	Speed (fps)	Cars			Pedestrians			Cyclists		
					E	M	H	E	M	H	E	M	H
Region Proposal -based Methods	Multi-view Methods	MV3D [4]	L & I	2.8	86.62	78.93	69.80	-	-	-	-	-	-
		AVOD [97]	L & I	12.5	89.75	84.95	78.32	42.58	33.57	30.14	64.11	48.15	42.37
		ContFuse [98]	L & I	16.7	94.07	85.35	75.88	-	-	-	-	-	-
		MMF [99]	L & I	12.5	93.67	88.21	81.99	-	-	-	-	-	-
		SCANet [102]	L & I	11.1	90.33	82.85	76.06	-	-	-	-	-	-
		RT3D [103]	L & I	11.1	56.44	44.00	42.34	-	-	-	-	-	-
	Segmentation -based Methods	IPOD [104]	L & I	5.0	89.64	84.62	79.96	60.88	49.79	45.43	78.19	59.40	51.38
		PointRCNN [105]	L	10.0	92.13	87.39	82.72	54.77	46.13	42.84	82.56	67.24	60.28
		PointRGCN [106]	L	3.8	91.63	87.49	80.73	-	-	-	-	-	-
		PointPainting [107]	L & I	2.5	92.45	88.11	83.36	58.70	49.93	46.29	83.91	71.54	62.97
		STD [110]	L	12.5	94.74	89.19	86.42	60.02	48.72	44.55	81.36	67.23	59.35
	Frustum -based Methods	F-PointNets [112]	L & I	5.9	91.17	84.67	74.77	57.13	49.57	45.48	77.26	61.37	53.78
		SIFRNet [113]	L & I	-	-	-	-	-	-	-	-	-	-
		PointFusion [116]	L & I	-	-	-	-	-	-	-	-	-	-
		RoarNet [117]	L & I	10.0	88.20	79.41	70.02	-	-	-	-	-	-
		F-ConvNet [111]	L & I	2.1	91.51	85.84	76.11	57.04	48.96	44.33	84.16	68.88	60.05
		Patch Refinement [118]	L	6.7	92.72	88.39	83.19	-	-	-	-	-	-
	Other Methods	3D IoU loss [119]	L	12.5	91.36	86.22	81.20	-	-	-	-	-	-
		Fast Point R-CNN [121]	L	16.7	90.76	85.61	79.99	-	-	-	-	-	-
		VoteNet [122]	L	-	-	-	-	-	-	-	-	-	-
		Feng et al. [123]	L	-	-	-	-	-	-	-	-	-	-
		Part-A^2 [124]	L	12.5	91.70	87.79	84.61	-	-	-	81.91	68.12	61.92
Single Shot Methods	BEV-based Methods	PIXOR [100]	L	28.6	83.97	80.01	74.31	-	-	-	-	-	-
		HDNET [128]	L	20.0	89.14	86.57	78.32	-	-	-	-	-	-
		BirdNet [129]	L	9.1	76.88	51.51	50.27	20.73	15.80	14.59	36.01	23.78	21.09
	Point Cloud -based Methods	VeloFCN [125]	L	1.0	0.02	0.14	0.21	-	-	-	-	-	-
		3D FCN [126]	L	<0.2	70.62	61.67	55.61	-	-	-	-	-	-
		Vote3Deep [127]	L	-	-	-	-	-	-	-	-	-	-
		3DBN [130]	L	7.7	89.66	83.94	76.50	-	-	-	-	-	-
		VoxelNet [108]	L	2.0	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55
		SECOND [120]	L	26.3	89.39	83.77	78.59	55.99	45.02	40.93	76.50	56.05	49.45
		MVX-Net [131]	L & I	16.7	92.13	86.05	78.68	-	-	-	-	-	-
		PointPillars [109]	L	62.0	90.07	86.56	82.81	57.60	48.64	45.78	79.90	62.73	55.58
	Other Methods	LaserNet [132]	L	83.3	79.19	74.52	68.45	-	-	-	-	-	-
		LaserNet++ [133]	L & I	26.3	-	-	-	-	-	-	-	-	-

4.3D点云分割

3D点云分割需要了解全局几何结构和每个点的细粒度细节。根据分割粒度，可以将3D点云分割方法分为三类：语义分割（场景级别），实例分割（对象级别）和部件分割（部件级别）。

4.1 3D语义分割

给定一个点云，语义分割的目标是根据它们的语义将点云分为几个子集。与3D形状分类的分类法相似（请参阅第2节），语义分割有两种范式，即基于投影的方法和基于点的方法。我们在图8中显示了几种代表性方法。

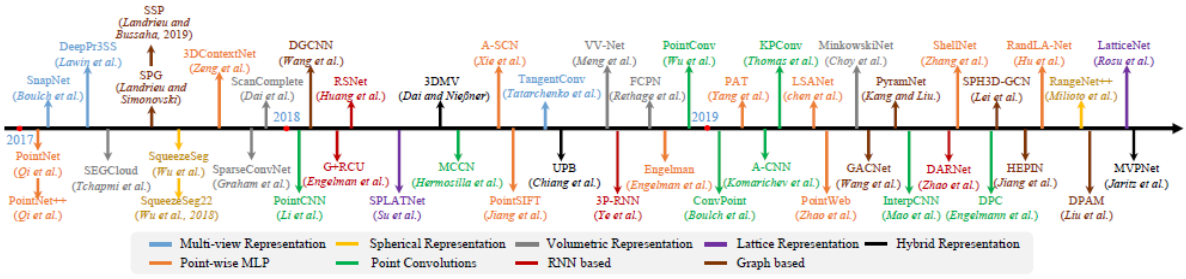
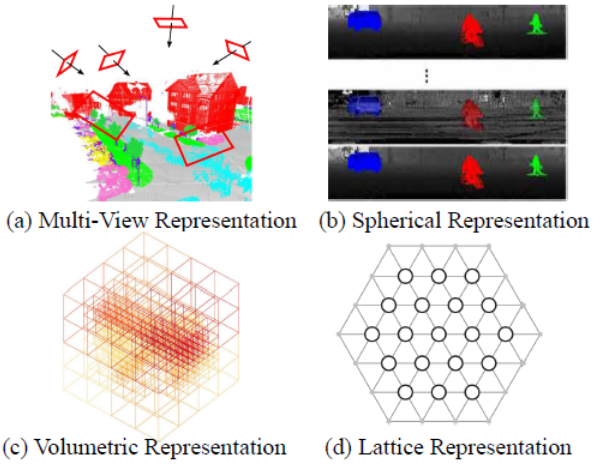


Fig. 8: Chronological overview of some of the most relevant deep learning-based point cloud semantic segmentation methods.

4.1.1基于投影的网络

中间正则表示可以组织或分类为多视图表示[148]，[149]，球形表示[150]，[151]，[152]，体积表示[153]，[154]，[155]，多面体如图9所示，可以使用点阵表示[156]，[157]和混合表示[158]，[159]。



(a) Multi-View Representation (b) Spherical Representation

(c) Volumetric Representation (d) Lattice Representation

Fig. 9: An illustration of the intermediate representation of projection-based methods.

多视图表示。 Felix等[148]首先从多个虚拟摄像机视图将3D点云投影到2D平面上。然后，将多流FCN用于预测合成图像上的逐像素评分。每个点的最终语义标签是通过将重新投影的分数融合到不同的视图上而获得的。同样，Boulch等。 [149]首先使用多个相机位置生成了点云的多个RGB和深度快照。然后，他们使用2D分割网络对这些快照执行了逐像素标记。从RGB和深度图像预测的分数将使用残差校正进一步融合[160]。 Tatarchenko等人基于点云是从局部欧几里得表面采样的假设。 [161]介绍了切线卷积的密集点云分割。该方法首先将围绕每个点的局部曲面几何投影到虚拟切线平面。切线卷积然后直接在曲面几何上进行。这种方法显示了很大的可伸缩性，并且能够处理具有数百万个点的大规模点云。总体而言，多视点分割方法的性能对视点选择和遮挡很敏感，此外，由于投影步骤不可避免地会导致信息丢失，因此这些方法还没有充分利用潜在的几何和结构信息。

球形表示。为了实现3D点云的快速准确分割，Wu等人[150]提出了一个基于SqueezeNet [162]和条件随机场（CRF）的端到端网络。为了进一步提高分割精度，引入了SqueezeSegV2 [151]，以利用无监督的域自适应流水线解决域移位问题。 Milioto等 [152]提出了RangeNet ++用于LiDAR点云的实时语义分割。

首先将2D范围图像的语义标签转移到3D点云，然后再使用有效的基于GPU的KNN基于后处理的步骤来减轻离散化错误和推理输出模糊的问题。与单视图投影相比，球形投影保留了更多信息，适合于LiDAR点云的标记。但是，这种中间表示不可避免地带来了一些问题，例如离散化误差和遮挡。

体积表示。黄等[163]首先将点云划分为一组占用体素。然后，他们将这些中间数据输入到全3D卷积神经网络中，以进行体素分割。最后，为体素内的所有点分配与体素相同的语义标签。该方法的性能受到由点云分区引起的体素的粒度和边界伪像的严重限制。此外，Tchapmi等 [164]提出了SEGCloud来实现细粒度和全局一致的语义分割。这种方法引入了确定性三线性插值法，将3D-FCNN [165]生成的粗体素预测映射回点云，然后使用完全连接CRF（FCCRF）来增强这些推断的点标签的空间一致性。孟等人[153]介绍了一种基于内核的内插变分自动编码器架构，以对每个体素内的局部几何结构进行编码。代替二进制占用表示，对每个体素采用RBF以获得连续表示和捕获每个体素中点的分布。 VAE还用于将每个体素内的点分布映射到紧凑的潜在空间。然后，对称组和等效CNN均用于实现鲁棒的特征学习。

良好的可伸缩性是体积表示的显着优点之一。具体来说，基于体积的网络可以自由地在具有不同空间大小的点云中进行训练和测试。在全卷积点网络（FCPN）[154]中，首先从点云中分层提取不同级别的几何关系，然后使用3D卷积和加权平均池来提取特征并合并远程依赖项。点云，在推理过程中具有良好的可伸缩性。安吉拉（Angela）等[166]提出了ScanComplete以实现3D扫描完成和每像素语义标注。该方法利用了全卷积神经网络的可扩展性，可以在训练和测试过程中适应不同的输入数据大小。从粗到精策略用于分层提高预测结果的分辨率。

体积表示自然是稀疏的，因为非零值的数量只占很小的百分比，因此在空间稀疏的数据上应用密集的卷积神经网络效率低下。为此，Graham等人 [155]提出了子流形稀疏卷积网络。该方法通过将卷积的输出限制为仅与占用的体素有关，从而大大减少了内存和计算成本。同时，其稀疏卷积还可以控制所提取特征的稀疏性。该子流形稀疏卷积适用于高维和空间稀疏数据的有效处理。此外，Choy等[167]提出了一种称为MinkowskiNet的4D时空卷积神经网络，用于3D视频感知。为了有效处理高维数据，提出了一种广义的稀疏卷积算法。三边平稳条件随机字段被进一步应用以增强一致性。

总体而言，体积表示自然保留了3D点云的邻域结构。它的常规数据格式还允许直接应用标准3D卷积。这些因素导致了该领域性能的稳步提高。然而，体素化步骤固有地引入了离散化伪像和信息丢失。通常，高分辨率会导致较高的内存和计算成本，而低分辨率会导致细节丢失。在实践中选择合适的网格分辨率并非易事。

四面体晶格表示。 Su等 [156]提出了基于双边卷积层（BCL）的稀疏格子网络（SPLATNet）。该方法首先将原始点云插值到四面体的稀疏晶格，然后将BCL应用于在稀疏填充的晶格的占据部分上进行卷积。然后将滤波后的输出内插回原始点云。另外，该方法允许灵活地联合处理多视图图像和点云。此外，Rosu等 [157]提出了LatticeNet来实现大点云的有效处理。还引入了一个称为DeformsSlice的依赖数据的插值模块，以将晶格特征反投影到点云。

混合表示。为了进一步利用所有可用信息，已经提出了几种方法来从3D扫描中学习多模式特征。 Angela和Matthias [158]提出了一个联合3D多视图网络，以结合RGB特征和几何特征。使用3D CNN流和几个2D流来提取特征，并提出了可微分的反投影层，以联合融合学习到的2D嵌入和3D几何特征。此外，Hung等。 [168]提出了一个基于点的统一框架，以从点云中学习2D纹理外观，3D结构和全局上下文特征。该方法直接应用基于点的网络，从稀疏采样的点集中提取局部几何特征和全局上下文，而无需任何体素化。 Jaritz等。 [159]提出了Multiview PointNet（MVPNet）来聚合2D多视图图像的外观特征和规范点云空间中的空间几何特征。

4.1.2基于点的网络

基于点的网络直接在不规则点云上工作。然而，点云是无序的和无组织的，因此直接应用标准的CNN是不可行的。为此，提出了开拓性的工作PointNet [5]来学习使用共享MLP的每点特征和使用对称池功能的全局特征。基于点网，最近已经提出了一系列基于点的网络。总体而言，这些方法可以粗略地分为按点MLP方法，点卷积方法，基于RNN的方法和基于图的方法。

逐点MLP方法。这些方法通常使用共享MLP作为其网络中的基本单位，以提高效率。然而，由共享的MLP提取的逐点特征无法捕获点云中的局部几何以及点之间的交互[5]。为了捕获每个点的更广泛的上下文并学习更丰富的局部结构，已引入了几个专用网络，包括基于相邻特征池，基于注意力的聚合以及局部全局特征串联的方法。

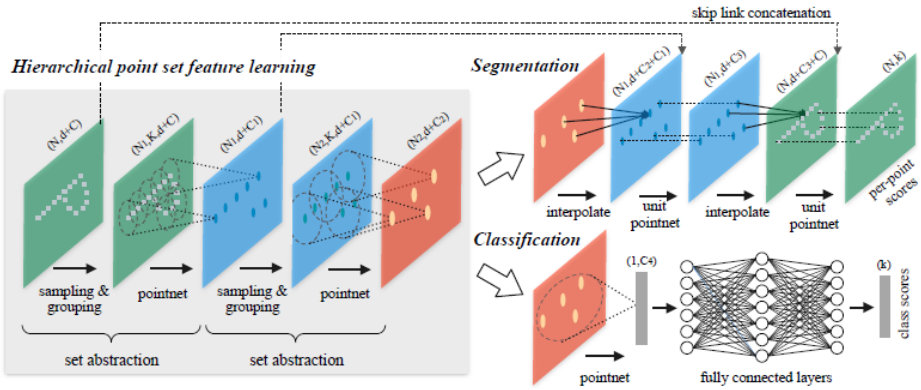


Fig. 10: An illustration of the PointNet++ [27] framework.

邻近特征池：为了捕获局部几何图案，这些方法通过汇总来自局部邻近点的信息来学习每个点的特征。特别是，PointNet ++ [27]对来自较大局部区域的点进行分层和渐进式学习，如图10所示。还提出了多尺度分组和多分辨率分组，以克服由不均匀和密度变化引起的问题。点云。后来，江等 [114]提出了一个PointSIFT模块来实现定向编码和尺度感知。该模块通过三阶段有序卷积运算对来自八个空间方向的信息进行堆叠和编码，提取并连接多尺度特征以实现对不同尺度的适应性。与PointNet ++中使用的分组技术（即球查询）不同，Francis等人。 [169]利用K-means聚类和KNN分别定义了世界空间和学习特征空间中的两个邻域。基于预期来自同一类的点在特征空间中更接近的假设，引入成对的距离损失和质心损失以进一步规范化特征学习。为了模拟不同点之间的相互作用，赵等人[31]提出了PointWeb，以通过密集构建本地完全链接的网络来探索本地区域中所有对点之间的关系。提出了一种自适应特征调整（AFA）模块来实现信息交换和特征细化。此聚合操作有助于网络学习区别性特征表示。张等[170]基于同心球壳的统计数据，提出了一个称为Shellconv的置换不变卷积。该方法首先查询一组多尺度的同心球，然后在不同的壳内使用最大池化操作汇总统计信息，使用MLP和一维卷积获得最终的卷积输出。 Hu等。 [95]提出了一种高效且轻量级的网络，称为RandLA-Net，用于大规模点云处理。该网络利用随机点采样在存储和计算方面实现了显着的效率。进一步提出了局部特征聚集模块以捕获和保留几何特征。

基于注意的聚合： 为了进一步提高分割的准确性，引入了一种注意机制[90]来进行点云分割。杨等。 [29]提出了一个小组改组注意力以建模点之间的关系的方法，并提出了一种排列不变，任务不可知且可区分的Gumbel子集采样（GSS）来代替广泛使用的最远点采样（FPS）方法。对异常值敏感，并可以选择代表点的子集。为了更好地捕获点云的空间分布，Chen等人。 [171]提出了一个局部空间感知（LSA）层来学习基于点云的空间布局和局部结构的空间感知权重。与CRF类似，Zhao等[172]提出了一种基于注意力的分数细化（ASR）模块，对网络产生的细分结果进行后处理。通过将相邻点的分数与学习的注意力权重合并在一起，可以细化初始分割结果。该模块可以轻松集成到现有的深度网络中，以提高最终的细分效果。

局部-全局串联：Zhao等[85]提出了一个排列不变的PS2-Net，以结合点云中的局部结构和全局上下文。 Edgeconv [60]和NetVLAD [173]反复堆叠以捕获局部信息和场景级全局特征。

点卷积方法。 这些方法倾向于为点云提出有效的卷积运算。 [49]提出了一种点式卷积算子，其中将相邻点合并到核单元中，然后与核权重进行卷积。 Wang等。 [174]提出了一个基于参数连续卷积层的称为PCCN的网络。该层的内核功能由MLP参数化，并跨越连续向量空间。休斯等。 [42]提出了一种基于核点卷积（KPConv）的核点全卷积网络（KP-FCNN）。具体地，KPConv的卷积权重由到核点的欧几里得距离确定，并且核点的数量不是固定的。核心点的位置被公式化为球空间中最佳覆盖率的优化问题。请注意，半径邻域用于保持一致的接收场，而网格二次采样用于每一层，以在变化的点云密度下实现高鲁棒性。在[175]中，弗朗西斯等人。提供了丰富的消融实验和可视化结果，以显示接受场对基于聚集的方法性能的影响。他们还提出了扩张点卷积（DPC）运算来聚集扩张后的邻近特征，而不是K个最近的邻居。该运算被证明在增加接收域方面非常有效，并且可以轻松地集成到现有的基于聚集的网络中。

基于RNN的方法。 为了从点云中捕获固有的上下文特征，递归神经网络（RNN）也已用于点云的语义分割。基于PointNet [5]，Francis等人。 [180]首先将点的块转换为多尺度块和网格块，以获得输入级别的上下文。然后，将PointNet提取的逐块特征顺序输入到合并单元（CU）或循环合并单元（RCU）中，以获得输出级别的上下文。实验结果表明，合并空间上下文对于提高分割效果非常重要。黄等。 [179]提出了一种轻量级的局部依赖建模模块，并利用切片池层将无序点特征集转换为特征向量的有序序列。 Ye等。 [181]首先提出了点向金字塔合并（3P）模块来捕获从粗到细的局部结构，然后利用双向分层RNN进一步获得远程空间依赖性，然后将RNN应用于实现末端然而，当将局部邻域特征与全局结构特征进行聚合时，这些方法会从点云中丢失丰富的几何特征和密度分布[189]。为了减轻刚性和静态合并操作引起的问题，Zhao等人。 [189]提出了一个动态聚合网络（DAR-Net）来考虑全局场景复杂性和局部几何特征。使用自适应的接收字段和节点权重来动态聚合中间特征。 Liu et al。[190]提出了3DCNN-DQN-RNN，用于大规模点云的高效语义解析。该网络首先使用3D CNN网络学习空间分布和颜色特征，DQN进一步用于对类对象进行定位。最终的级联特征向量被馈送到残差RNN中以获得最终的分割结果。

基于图的方法。 为了捕获3D点云的基本形状和几何结构，有几种方法可以求助于图形网络。 Loic等[182]将点云表示为一组相互连接的简单形状和超点，并使用属性有向图（即超点图）来捕获结构和上下文信息。然后，将大规模点云分割问题归结为三个子问题，即几何同构分割，超点嵌入和上下文分割。为了进一步改善分割步骤，Loic和Mohamed [183]提出了一种有监督的框架，将点云过度分割为纯超点。该问题被表述为由邻接图构成的深度度量学习问题。此外，还提出了一种图结构的对比损失，以帮助识别对象之间的边界。

为了更好地捕捉高维空间中的局部几何关系，Kang等人 [191]提出了一种基于图嵌入模块（GEM）和金字塔注意网络（PAN）的PyramNet。 GEM模块将点云公式化为有向无环图，并使用协方差矩阵替换欧几里得距离来构造相邻相似矩阵。 PAN模块中使用具有四个不同大小的卷积内核来提取具有不同语义强度的特征。在[184]中，提出了图注意力卷积（GAC）来从局部相邻集合中有选择地学习相关特征。通过基于它们的空间位置和特征差异，将注意力权重动态分配给不同的相邻点和特征通道，可以实现此操作。 GAC可以学习捕获区分特征以进行细分，并且具有与常用CRF模型相似的特征。

4.2 实例细分

与语义分割相比，实例分割更具挑战性，因为它需要更准确，更细粒度的点推理。特别是，它不仅需要区分具有不同语义含义的点，而且还需要分离具有相同语义含义的实例。总的来说，现有方法可以分为两类：基于提议的方法和不涉及提议的方法。图11中说明了几种里程碑方法。

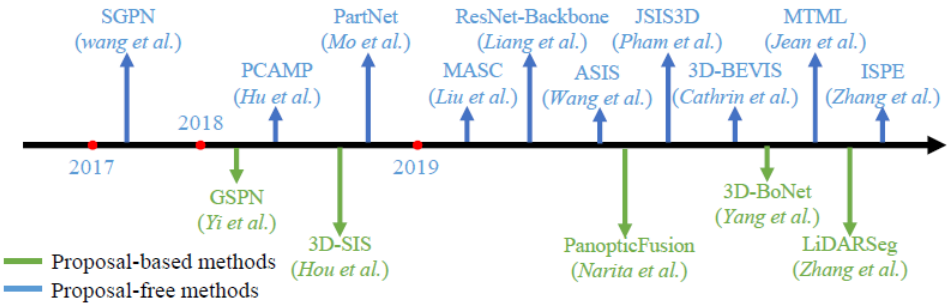


Fig. 11: Chronological overview of representative 3D point cloud instance segmentation methods.

4.2.1基于提案的方法

这些方法将实例分割问题转换为两个子任务：3D对象检测和实例掩码预测。侯等人。 [192]提出了一种3D全卷积语义实例分割（3D-SIS）网络，以在RGBD扫描上实现语义实例分割。该网络从颜色和几何特征中学习。与3D对象检测类似，3D区域提议网络（3D-RPN）和3D感兴趣区域（3D-RoI）层用于预测边界框位置，对象类别标签和实例蒙版。遵循综合分析策略，Yi等人。 [193]提出了一种可生成形状的提案网络（GSPN），以生成高对象的3D提案。这些建议由基于区域的PointNet（R-PointNet）进一步完善。最终标签是通过预测每个类标签的每点二进制掩码获得的。与从点云直接回归3D边界框不同，此方法通过加强几何理解来消除大量毫无意义的建议。通过将2D全景分割扩展到3D映射，Gaku等人。 [194]提出了一种单行立体3D映射系统，以共同实现大规模3D重建，语义标记和实例分割。他们首先利用2D语义和实例分割网络来获得按像素分类的全景标签，然后将这些标签集成到体积图上。进一步使用完全连接的CRF来实现准确的分割。该语义映射系统可以实现高质量的语义映射和区分对象识别。 [195]提出了一种称为3D-BoNet的单阶段，无锚定且端到端的可训练网络，以在点云上实现实例分割。该方法直接为所有潜在实例回归粗糙的3D边界框，然后利用点级二进制分类器获取实例标签。特别是，将边界框生成任务表述为最佳分配问题。还提出了多准则损失函数来规范生成的边界框。该方法不需要任何后处理，并且计算效率高。张等。 [196]提出了一个用于大型室外LiDAR点云分割的网络。该方法使用自注意块学习点云鸟瞰图上的特征表示。最终实例标签是根据预测的水平中心和高度限制获得的。总体上，基于建议的方法直观，简单，实例分割结果通常具有良好的客观性。但是，这些方法需要多阶段的训练和对冗余提议的修剪，因此它们通常很耗时且计算量大。

4.2.2 Proposal-free Methods

免提案方法[197]，[198]，[199]，[200]，[201]，[202]没有对象检测模块。相反，他们通常将实例分割视为语义分割之后的后续聚类步骤。特别是，大多数现有方法都是基于这样的假设，即属于同一实例的点应具有非常相似的特征。因此，这些方法主要集中于判别特征学习和点分组。

在一项开创性的工作中，Wang等人 [197]首先引入了一个相似性团体提案网络（SGPN）。该方法首先学习每个点的特征和语义图，然后引入一个相似度矩阵来表示每个配对特征之间的相似度。为了学习更多的鉴别特征，他们使用双铰损失来相互调整相似度矩阵和语义分割结果。最后，采用启发式和非最大抑制方法将相似点合并为实例。由于相似矩阵的构造需要大的存储器消耗，因此该方法的可扩展性受到限制。同样，刘等 [201]首先利用子流形稀疏卷积[155]来预测每个体素的语义分数和相邻体素之间的亲和力。然后他们引入了一种聚类算法，根据预测的亲和力和网格拓扑将点分组为实例。 [202]提出了一种学习判别式嵌入的结构感知损失。这种损失既考虑了特征的相似性，又考虑了点之间的几何关系。基于注意力的图CNN进一步用于通过汇总来自邻居的不同信息来自适应地精炼所学习的特征。

由于一个点的语义类别和实例标签通常相互依赖，因此提出了几种方法将这两个任务耦合为一个任务。 [198]通过引入端到端和可学习的关联分段实例和语义（ASIS）模块，整合了这两个任务。实验表明，通过此ASIS模块，语义特征和实例特征可以相互支持，从而提高性能。同样，Pham等。 [199]首先引入了多任务逐点网络（MT-PNet），为每个点分配标签，并通过引入判别性损失来对嵌入特征空间的规则进行规范[203]。然后，他们将预测的语义标签和嵌入融合到多值条件随机字段（MV-CRF）模型中，以进行联合优化。最后，均值场变分推理用于产生语义标签和实例标签。 Hu等。 [204]首先提出了一种动态区域增长（DRG）方法，将点云动态分离为一组不相交的补丁，然后使用无监督的K-means ++算法对所有这些补丁进行分组。然后在补丁之间的上下文信息的指导下执行多尺度补丁分段。最后，将这些标记的补丁合并到对象级别，以获得最终的语义和实例标签。

为了在完整的3D场景上实现实例分割，Cathrin等人 [200]提出了一种混合的2D-3D网络，可以从BEV表示和点云的局部几何特征共同学习全局一致的实例特征。然后将学习到的特征进行组合以实现语义和实例分割。注意，不是启发式GroupMerging算法[197]，而是更灵活的Meanshift [205]算法用于将这些点分组为实例。可替代地，还引入了多任务学习以进行实例分割。 Jean等。 [206]学习了每个实例的独特功能嵌入和指向对象中心的方向信息。提出了特征嵌入损失和方向损失来调整潜在特征空间中学习的特征嵌入。采用均值漂移聚类和非最大抑制将体素分组为实例。该方法可以达到ScanNet [8]基准的最新性能。此外，预测的方向信息对于确定实例的边界特别有用。张等。 [207]将概率嵌入引入到点云的实例分割中。该方法还结合了不确定性估计，并为聚类步骤提出了新的损失函数。

总之，无提议的方法不需要通常昂贵的区域提议组件。但是，由于这些方法没有显式检测对象边界，因此通过这些方法分组的实例段的客观性通常较低。

4.3 Part Segmentation

3D形状的 Part Segmentation 难度是双重的。首先，具有相同语义标签的形状零件具有较大的几何变化和模糊性。其次，该方法应对噪声和采样具有鲁棒性。

提出了VoxSegNet [208]，以在有限的解决方案上实现3D体素化数据的细粒度分割。提出了空间密集提取（SDE）模块（由堆叠的残差残差块组成），以从稀疏的体积数据中提取多尺度判别特征通过逐步应用注意力特征聚合（AFA）模块，可以对学习的特征进行进一步的加权和融合。 Evangelos等 [209]结合FCN和基于表面的CRF来实现端到端3D零件分割。他们首先从多个视图生成图像以实现最佳的表面覆盖率，然后将这些图像输入2D网络以生成置信度图。然后，这些置信度图由基于表面的CRF聚合，该CRF负责整个场景的一致标记。 [210]引入了一种同步频谱CNN（SyncSpecCNN）来对不规则和非同构形状图进行卷积。为了解决零件多尺度分析和形状间信息共享的问题，引入了卷积核和谱变换器网络的谱参数化方法。

Wang等[211]首先通过引入形状完全卷积网络（SFCN）并将三个低级几何特征作为其输入，在3D网格上执行形状分割。然后，他们利用基于投票的多标签图割来进一步细化细分结果。朱等。 [212]提出了一种用于3D形状共分割的弱监督CoSegNet。该网络将未分割的3D点云形状的集合作为输入，并通过迭代地最小化组一致性损失来生成形状零件标签。与CRF相似，提出了一个预训练的零件细化网络，以进一步细化和去除零件提案的噪声。 Chen等。 [213]提出了一种分支自动编码器网络（BAE-NET），用于无监督，单发和弱监督的3D形状共分割。该方法将形状共分割任务公式化为表示学习问题，旨在通过最大程度地减少形状重构损失来找到最简单的零件表示。基于编码器-解码器体系结构，该网络的每个分支都可以学习特定零件形状的紧凑表示。然后将从每个分支学习的特征和点坐标馈送到解码器以生成二进制值（指示该点是否属于此部分）。该方法具有良好的泛化能力，可以处理大型3D形状集合（多达5000多种形状）。但是，它对初始参数敏感，并且没有将形状语义合并到网络中，这阻碍了该方法在每次迭代中获得鲁棒和稳定的估计。

4.4小结

TABLE 4: Comparative semantic segmentation results on the S3DIS (including both Area5 and 6-fold cross validation) [176], Semantic3D (including both *semantic-8* and *reduced-8* subsets) [9] , ScanNet [8], and SemanticKITTI [177] datasets. Overall Accuracy (OA), Mean Intersection-over-Union (mIoU) are the main evaluation metric. For simplicity, we omit the '%' after the value. The symbol '-' means the results are unavailable.

Method			S3DIS				Semantic3D				ScanNet(v2)		Sem. KITTI (mIoU)
			Area5 (OA)	Area5 (mIoU)	6-fold (mIoU)	6-fold (mIoU)	sem. (OA)	sem. (mIoU)	red. (OA)	red. (mIoU)	OA	mIoU	
Projection -based Methods	Multi-view	DeePr3SS [148]	-	-	-	-	-	-	88.9	58.5	-	-	-
		SnapNet [149]	-	-	-	-	91.0	67.4	88.6	59.1	-	-	-
		TangentConv [161]	82.5	52.8	-	-	-	-	-	-	80.1	40.9	40.9
	Spherical	SqueezeSeg [150]	-	-	-	-	-	-	-	-	-	-	29.5
		SqueezeSegV2 [151]	-	-	-	-	-	-	-	-	-	-	39.7
		RangeNet++ [152]	-	-	-	-	-	-	-	-	-	-	52.2
	Volumetric	SegCloud [164]	-	48.9	-	-	-	-	88.1	61.3	-	-	-
		SparseConvNet [155]	-	-	-	-	-	-	-	-	-	72.5	-
		MinkowskiNet [167]	-	-	-	-	-	-	-	-	-	73.6	-
		VV-Net [153]	-	-	87.8	78.2	-	-	-	-	-	-	-
	Permutohedral lattice	SPLATNet [156]	-	-	-	-	-	-	-	-	-	39.3	18.4
		LatticeNet [157]	-	-	-	-	-	-	-	-	-	64.0	52.2
	Hybrid	3DMV [158]	-	-	-	-	-	-	-	-	-	48.4	-
		UPB [168]	-	-	-	-	-	-	-	-	-	63.4	-
		MVPNet [159]	-	-	-	-	-		-	-	-	64.1	-
Point -based Methods	Point-wise MLP	PointNet [5]	-	41.1	78.6	47.6	-	-	-	-	-	-	14.6
		PointNet++ [27]	-	-	81.0	54.5	85.7	63.1	-	-	84.5	33.9	20.1
		PointSIFT [114]	-	-	88.7	70.2	-	-	-	-	86.2	41.5	-
		Engelmann [178]	84.2	52.2	84.0	58.3	-	-	-	-	-	-	-
		3DContextNet [79]	-	-	84.9	55.6	-	-	-	-	-	-	-
		A-SCN [81]	-	-	81.6	52.7	-	-	-	-	-	-	-
		PointWeb [31]	87.0	60.3	87.3	66.7	-	-	-	-	85.9	-	-
		PAT [29]		60.1		64.3	-	-	-	-	-	-	-
		LSANet [171]	-	-	86.8	62.2	-	-	-	-	85.1	-	-
		ShellNet [170]	-	-	87.1	66.8	-	-	93.2	69.3	85.2	-	-
		RandLA-Net [95]	-	-	87.2	68.5	-	-	94.4	76.0	-	-	50.3
	Point convolution	PointCNN [52]	85.9	57.3	88.1	65.4	-	-	-	-	85.1	45.8	-
		PCCN [174]	-	58.3	-	-	-	-	-	-	-	-	-
		A-CNN [55]	-	-	87.3	-	-	-	-	-	85.4	-	-
		ConvPoint [47]	-	-	88.8	68.2	93.4	76.5	-	-	-	-	-
		KPConv [42]	-	67.1	-	70.6	-	-	92.9	74.6	-	68.4	-
		DPC [175]	86.8	61.3	-	-	-	-	-	-	-	59.2	-
		InterpCNN [53]	-	-	88.7	66.7	-	-	-	-	-	-	-
	RNN -based	RSNet [179]	-	51.9	-	56.5	-	-	-	-	84.9	39.4	-
		G+RCU [180]	-	45.1	81.1	49.7	-	-	-	-	-	-	-
		3P-RNN [181]	85.7	53.4	86.9	56.3	-	-	-	-	-	-	-
	Graph -based	DGCNN [60]	-	-	84.1	56.1	-	-	-	-	-	-	-
		SPG [182]	86.4	58.0	85.5	62.1	92.9	76.2	94.0	73.2	-	-	17.4
		SSP+SPG [183]	87.9	61.7	87.9	68.4	-	-	-	-	-	-	-
		GACNet [184]	87.8	62.9	-	-	-	-	91.9	70.8	-	-	-
		PAG [185]	86.8	59.3	88.1	65.9	-	-	-	-	-	-	
		HDGCN [186]	-	59.3	-	66.9	-	-	-	-	-	-	-
		HPEIN [187]	87.2	61.9	88.2	67.8	-	-	-	-	-	61.8	-
		SPH3D-GCN [188]	87.7	59.5	88.6	68.9	-	-	-	-	-	61.0	-
		DPAM [65]	86.1	60.0	87.6	64.5	-	-	-	-	-	-	-

表4显示了通过公开基准测试的现有方法所获得的结果，包括S3DIS [176]，Semantic3D [9]，ScanNet [102]和SemanticKITTI [177]。以下问题需要进一步调查：

- **基于点的网络是研究最频繁的方法。但是，点表示自然不具有显式的相邻信息，大多数现有的基于点的方法都必须诉诸昂贵的邻居搜索机制（例如*KNN [52]*或*Ball*查询*[27]*）。这会固有地限制这些方法的效率，因为邻居搜索机制需要很高的计算成本和不规则的内存访问*[214]*。**
- **从不平衡数据中学习仍然是点云分割中一个具有挑战性的问题。尽管有几种方法*[42]*，*[170]*，*[182]*取得了显着的总体表现，但它们在少数群体中的表现仍然有限。例如，*RandLA-Net [95]*在*Semantic3D*的*reduce-8*子集上实现了*76.0%*的总体*IoU*，但在*Hardscape*类上却达到了*41.1%*的非常低的*IOU*。**
- **现有的大多数方法*[5]*，*[27]*，*[52]*，*[170]*，*[171]*都适用于小点云（例如，具有*4096*个点的*1m*1m）。实际上，由深度传感器获取的点云通常是巨大且大规模的。因此，期望进一步研究大规模点云的有效分割问题。***
- **少数著作*[145]*，*[146]*，*[167]*已开始从动态点云中学习时空信息。期望时空信息可以帮助提高后续任务的性能，例如*3D*对象识别，分段和完成。**

5.结论

本文介绍了有关3D理解的最新方法的当代概况，包括3D形状分类，3D对象检测和跟踪以及3D场景和对象分割。已经对这些方法进行了全面的分类和性能比较。还涵盖了各种方法的优缺点，并列出了潜在的研究方向。