三维点云深度学习研究综述
论文：Deep Learning for 3D Point Clouds: A Survey

作者：Yulan Guo
时间：2019-12

引言

动机：

> Point cloud learning
> （点云学习）由于在视觉、自动驾驶、机器人等方面的广泛应用，近年来受到了广泛的关注。最近，随着点云的深度学习变得更加兴旺，人们提出了许多方法来解决这一领域的不同问题。为了促进未来的研究，本文对点云深度学习方法的最新进展进行了全面的综述。

挑战：

> 深度学习技术目前已经成为成功解决各种二维视觉问题的主流技术，点云的深度学习依然处于初级阶段。
> 深度神经网络处理点云所面临的独特挑战（例如数据集的小规模、高维和三维点云的非结构化性质）

意义

> 第一篇全面涵盖几个重要点云相关任务的深度学习方法的调查论文，包括三维形状分类、三维目标检测和跟踪以及三维点云分割。
> 与现有的综述不同，特别关注3D点云的深度学习方法，而不是所有类型的3D数据 介绍了点云深度学习的最新进展。因此，它为读者提供了最先进的方法
> 提供了在几个公开可用数据集上对现有方法进行的综合比较，并提供了简要总结和有洞察力的讨论

论述的三个主要的任务：

> 3D shape classification （三维形状分类） 3D object detection and tracking
> （三维对象检测和追踪） 3D point cloud segmentation （三维点云分割）

三维点云深度学习方法的分类：

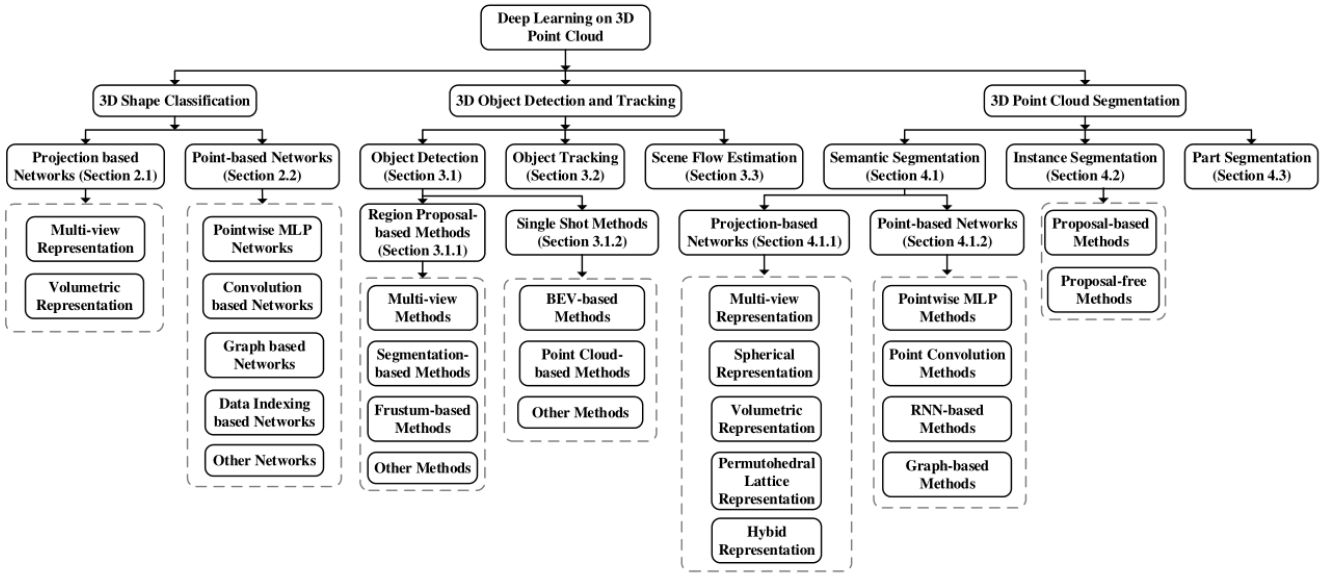


Fig. 1: A taxonomy of deep learning methods for 3D point clouds.

2、3D Shape Classification

介绍：这类的方法通常先学习 embedding of each point （每个点的嵌入），然后使用 aggregation method （聚合方法）从 whole point cloud （整个点云）中 extract a global shape embedding （提取全局形状嵌入），最终由几个全连接层来实现 classification （分类）。

基于对每个点进行 feature learning （特征学习）的方式，现有的 3D shape classification methods （三维形状分类方法）可分为 projection-based networks （基于投影的网络）和 point-based networks （基于点的网络）。在本文中，我们主要关注基于点的网络，但也包括一些基于投影的网络以保证完备性。

Projection-based methods： 首先将一个 unstructured （非结构化）的点云投影到一个规则中间的表示中，然后利用成熟的2D或3D卷积来实现形状分类。
point-based networks：基于点的方法直接作用于原始点云，而不需要任何体素化或投影。基于点的方法不会引入显式信息丢失，并且变得越来越流行。

按时间顺序概述3D shape classification 的一些里程碑的方法：

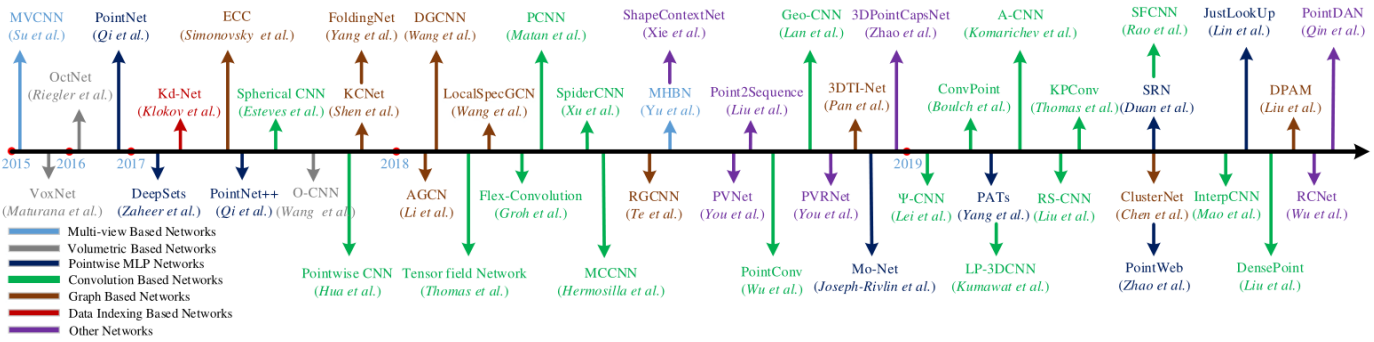


Fig. 2: Chronological overview of 3D shape classification networks.

2.1 Projection-based Networks 基于投影的网络

基于投影的网络将3D点云投影到不同的表示模式（例如，多视图、体积表示）中，以进行特征学习和形状分类。

2.1.1 Multi-view representation 多视角表示

这些方法首先将3D对象投影到多个 views (视图) 中并提取相应的 view-wise features（视域特征），然后融合这些特征以实现准确的对象识别。关键挑战是如何将多个 view-wise features 聚合到一个有识别力的全局表示 global representation 中。

现有的一些方法：

MVCNN： 开创性的工作，只是简单地 max-pools multi-view features(多视图特征) into a global descriptor（全局描述符），但是max-pooling 仅保留特定视图中的最大元素，从而会导致信息丢失。
MHBN： 通过协调双线性 pooling 来集成局部卷积特征（local convolutional features），以产生紧凑的全局描述符（global descriptor）。
首先利用关系网络（relation network）来发现一组视图上的相互关系(例如，区域-区域关系和视图-视图关系)，然后聚集这些视图以获得可辨别的 3D object representation。
……

2.1.2 Volumetric representation 体素表示

早期的方法通常使用建立在 3D point clouds （3D点云）的 volumetric representation（体表示：由称为体素的离散体组成）上的三维卷积神经网络(CNN)。
Wu et al. 提出了一种卷积深度 belief-based 的3D ShapeNets，用于从不同形状的三维形状中学习点的分布。虽然已经取得了令人鼓舞的性能，但是这些方法不能很好地扩展到密集的3D数据，因为计算和内存占用随着分辨率的提高而成倍增长。
为此，引入了一种层次紧凑的图结构(如八叉树 octree)来降低这些方法的计算和存储开销。eg：OctNet、Octree-based CNN... 与基于dense input grids的 baseline network 相比，OctNet对高分辨率点云所需的内存和运行时间要少得多 。
PointGrid的混合网络，该网络集成了点和网格表示，以实现高效的点云处理。

2.2 Point-based Networks 基于点的网络

根据用于每个点的特征学习的网络结构，这类方法可分为逐点MLP（pointwise MLP）、基于卷积（convolution-based）、基于图（graph-based）、基于数据索引的网络（data indexing-based networks）和其他典型网络。

2.2.1 Pointwise MLP Networks

这类方法使用多层感知器 MLP（Multi-Layer Perceptrons ）对各个点进行独立的建模，接着使用对称的函数来集成到全局特征。对于无序的3D点云数据，这类网络可以得到置换不变性。然而这样的方法并未考虑到3D点之间的几何关系，如下图3。

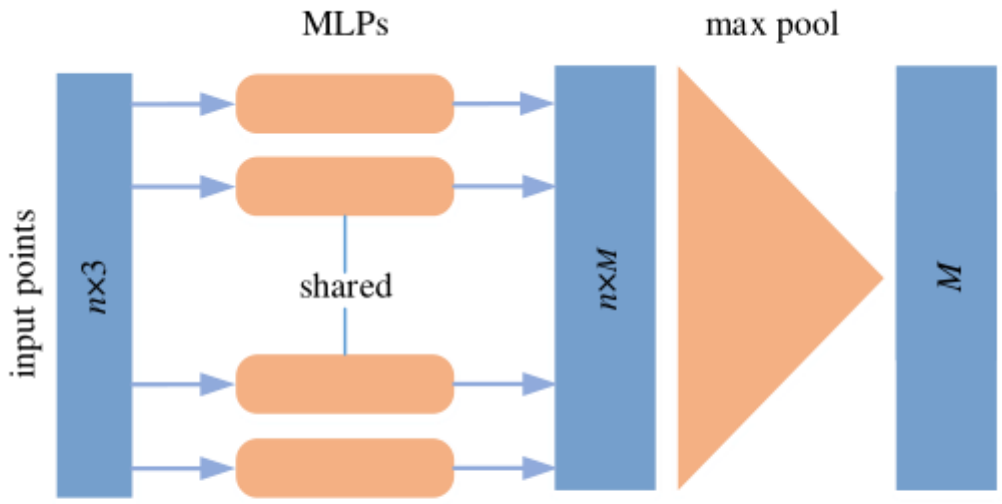


Fig. 3: The architecture of PointNet. n denotes the number of input points, M denotes the dimension of the learned features for each point. After max pooling, the dimension of the global feature of the whole point cloud is also M.

作为先驱工作，PointNet 使用MLP学习Pointwise特征，接着使用最大池化层来提取全局的形状特征。最后的分类结果也使用MLP来得到。[26]也论证了，得到置换不变性的关键在于将所有表示（representations）加起来并且使用非线性变化。[26]也设计了基础的网络DeepSets来进行多种应用的实现，包括形状分类。

由于特征是针对PointNet[5]中的每个点独立学习的，因此各个点之间的局部结构信息无法得到。[27]提出了一种分层次的网络PointNet++，从各个点之间的邻居来获取细粒度的几何特征。（PointNet++的核心，其abstraction level 由采样层（the sampling layer）、分组层（the grouping layer）和PointNet层三层组成。PointNet++通过堆叠多个abstraction level，可以从局部几何结构中学习特征，并逐层抽象局部特征。）

因为PointNet的简单和有效性，许多工作都基于PointNet开展。（这里介绍了一些网络）

2.2.2 Convolution-based Networks 基于卷积的网络

与2D卷积相比，由于点云的不规则性，3D点云的卷积核更难设置。根据卷积核的不同，目前的3D卷积网络可以被分为连续卷积网络（continuous convolution networks ）和离散卷积网络（discrete convolution networks），如下图所示。

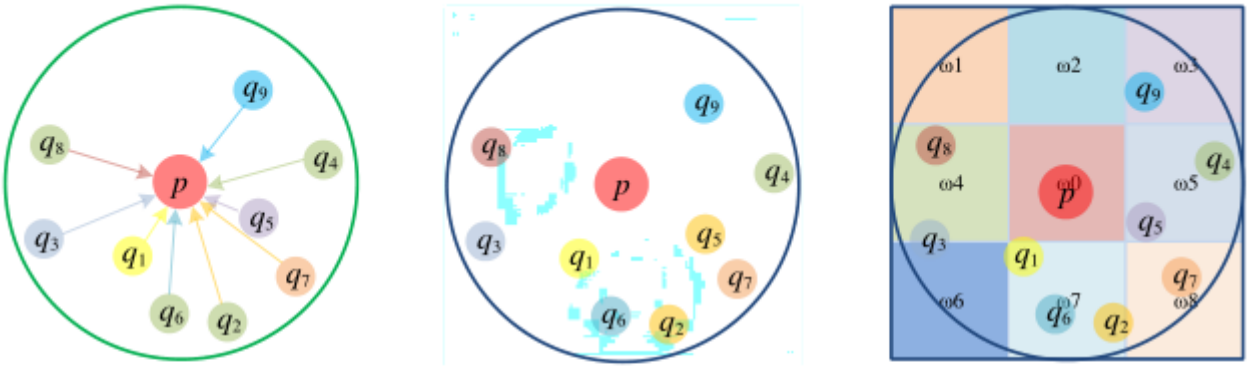


Fig. 4: An illustration of a continuous and discrete convolution for local neighbors of a point. (a) represents a local neighborhood; (b) and (c) represent 3D continuous and discrete convolution, respectively.

3D Continuous Convolution Networks. 3D连续卷积网络

这类方法在连续的空间中定义卷积核，其中邻居点的权重与它和中心点的空间分布有关。

3D卷积可以解释为给定子集上的加权和。MLP是学习权重的一种简单方法。作为RS-CNN[35]的核心层，RS-Conv将某个点周围的局部子集作为其输入，使用MLP的方法来进行卷积，学习低维关系到高维关系的映射。

一些方法还使用现有算法来执行卷积。在PointConv[38]中，卷积被定义为对重要性采样的连续3D卷积的蒙特卡洛估计。卷积核由加权函数（由MLP层学到）和密度函数（由核密度估计和MLP层学到）组成。为了提升内存和计算效率，3D卷积被简化成两部分：矩阵乘法和2D卷积，在相同的参数设置下，内存消耗可减小64倍。

3D Discrete Convolution Networks. 3D离散卷积网络

这类方法在标准的网格上定义卷积核，其中的邻居点的权重是其关于中心点的补偿（offset）。

[49]将非归一化的点云变换至归一化的网格，接着在各个网格上定义卷积核。与2D卷积不同（在各个像素上分配权重），所提的3D卷积核在网格内的所有点赋予相同的权重。

对于给定点，邻域内所有点（在相同网格上）的平均特征通过之前的层来计算得到。接着，所有网格的平均特征通过加权和产生当前层的输出。

2.2.3 Graph-based Networks 基于图的网络

基于图的网络将点云中的每个点视为图的一个顶点，并基于每个点的邻域来生成图的有向边。然后在空间域或谱域中执行特征学习[58]。典型的基于图的网络如图5所示。

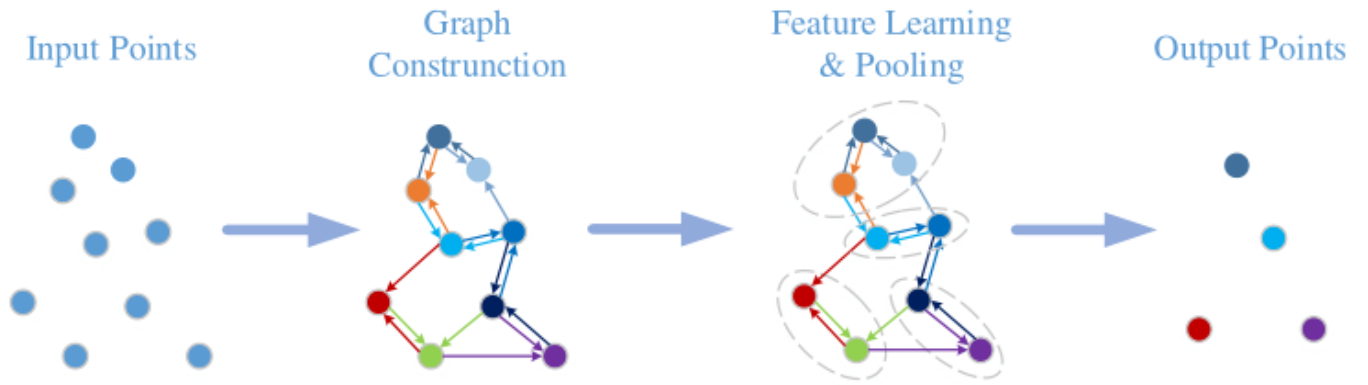


Fig. 5: An illustration of a graph-based network.

Graph-based Methods in Spatial Domain 空间域中的基于图的方法 .

这类方法在空间域中定义卷积和池化操作。卷积通过在空间邻域内的MLP实现，池化操作通过集成信息产生新的较粗的图。各个顶点的特征由坐标、激光强度、颜色来确定，各个边的特征由两个连接点的几何属性确定。

作为先驱工作，[58]将各个点视为图的顶点，利用有向边将顶点与其邻域内的点相连，接着使用Edge-Condition Convolution（使用生成filter的网络得到，MLP等）。最大池化用来集成邻域信息，图的粗化使用VoxelGrid[59]算法得到。首先通过卷积和池化的相互交错，再跟着为全局平均池化和全连接层来产生分类score。

Graph-based Methods in Spectral Domain 谱域中的基于图的方法.

这些方法将卷积定义为谱的滤波，将其实现为图上的信号与图的拉普拉斯矩阵的特征向量的乘法。

2.2.4 Data Indexing-based Networks 基于索引数据的网络

这些网络基于不同的数据索引结构(例如，八叉树和kd-树)来构建。在这些方法中，点特征是沿着树从叶节点到根节点分层学习得到的。

2.2.5 Other Networks

除了上述方法外，还提出了许多其他方案

表1：在ModelNet10/40基准上比较3D Shape Classification 结果，只关注基于点的网络（pointbased networks），“#params”指的是相应模型的参数个数。“OA”表示 overall accuracy，“MACC”表示表中的平均精度（mean accuracy）。符号‘-’表示结果不可用。

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%	-

3、3D Object Detection and tracking

3.1 3D Object Detection 物体检测

与普通2D中目标检测方法类似，3D中的目标检测也可以分为两类：基于候选区域的方法和直接映射方法。

3.1.1 Region Proposal-based Methods 基于候选区域

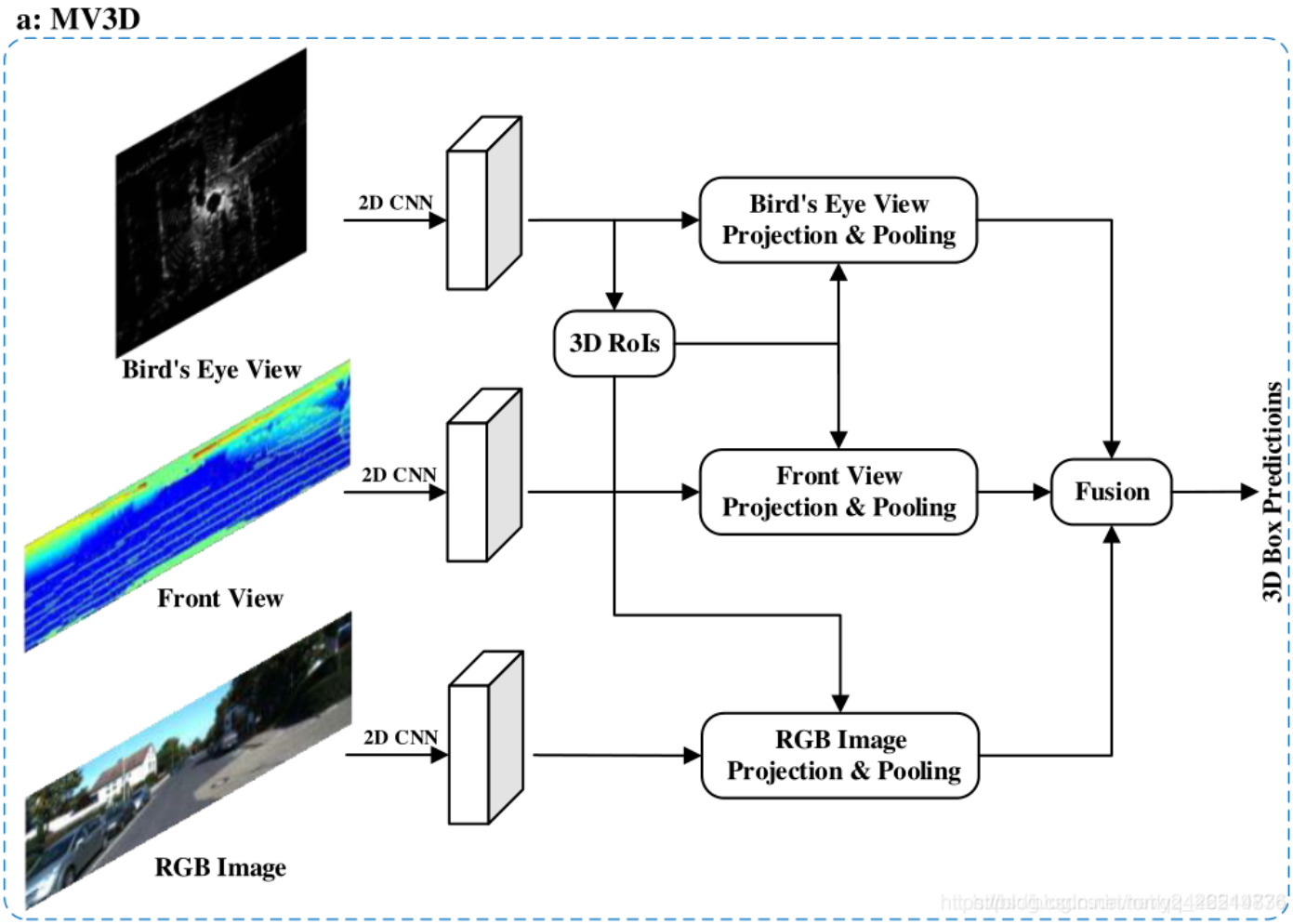
首先产生一些可能包含物体的区域（Proposals），接着对各个区域提取特征，来决定各个候选区域的物体类别。

根据不同的产生候选区域的方法，这些方法可进一步分为三类：基于多视角的方法（multi-view based）；基于分割的方法（segmentation-based）以及基于锥体的方法（frustum-based methods）。

Multi-view Methods 多视角的方法

这类方法从不同的视角图像（雷达前景图（LiDAR front view），鸟瞰图（bird’s eye view (BEV)），图像（image）等）中融合各个候选框的特征，来产生3D rotated boxes，如图7(A)所示。这些方法的计算成本通常很高。

在[4]中，Chen等人从鸟瞰图BEV中产生一组准确的3D候选框，并且将其投影到其它视角中（雷达前景图，RGB图像），接着将各个区域的特征组合到一起，来预测有方向的3D bounding boxes。尽管这种方法在0.25IOU， 300个候选框设置时达到了99.1%的recall，但是速度非常慢。



后续的基于多视角的3D物体检测方法主要从以下两个方面来提升。

（1）提出了很多方法来有效的融合不同模态之间的信息。

为了针对小物体产生有较高recall的候选框，[97]提出了一种多模态的基于融合的区域生成网络（ a multi-modal fusion-based region proposal network）。首先使用裁剪和大小调整操作从BEV视图和image视图中提取大小相等的特征，然后使用 mean pooling

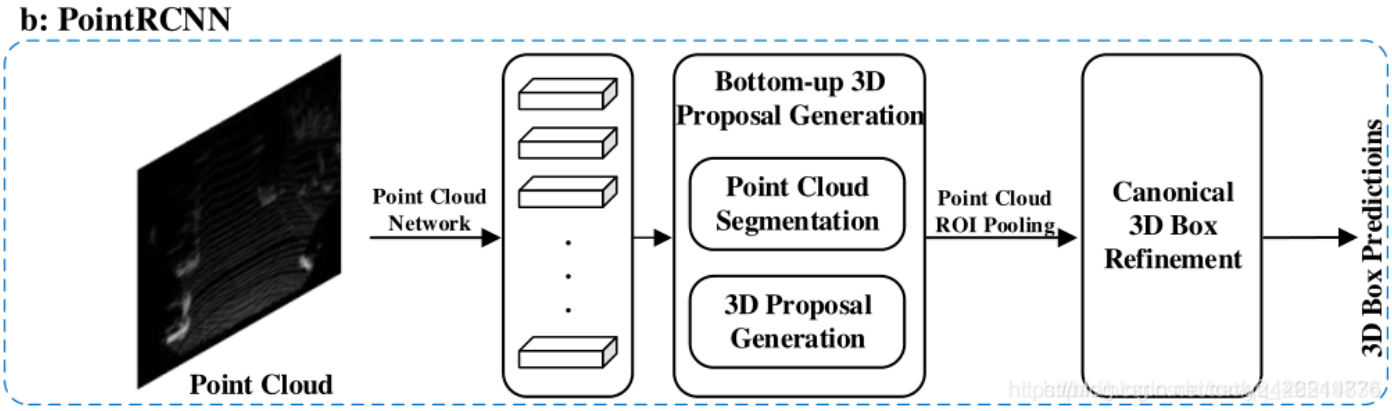
对这些特征进行融合。具体而言，他们对BEV（鸟瞰视角）空间中的每个点提取最近的对应点的图image 特征，接着通过将image特征投影至BEV空间的方法，使用双线性插值得到稠密的BEV的特征图。实验结果证明稠密的BEV特征图比起离散的image特征图和稀疏的LiDAR(雷达激光)特征图更加适合3D物体检测。
[99]提出了多任务，多感知器的3D物体检测网络来进行端到端的训练。具体而言，利用多种任务（2D物体检测，背景估计 ground estimation，深度补偿 depth completion），帮助网络学习到更好的特征表示。学习到的跨模态的表示，可进一步用来产生更准确的物体检测结果。实验证明这类方法在2D,3D,BEV detection 任务上有着非常好的提升，在TOR4D基准[100, 101]上超越了之前的SOTA。

（2）其它的一些方法致力于提取输入数据更鲁棒的表示 representations

[102]通过引入空间Channel注意力机制模块（Spatial Channel Attention (SCA) Module），探索了多尺度的环境信息，该模块可捕获全局的以及多尺度的场景环境，加强了有用的特征。同样的，他们还提出了一种 Extension Spatial Unsample (ESU) 模块，通过组合多尺度的低层特征来获得具有丰富空间信息的高层特征，从而生成更可靠的3D物体候选框 （proposals） 。尽管达到了更好的检测效果，但上述所提的多视角方法都需要较长的运行时间，因为他们对每个候选框都进行了特征的池化。因此，[103]使用了 预ROI池化卷积（pre-ROI pooling convolution）来提高[4]的效率。具体而言，他们将大部分的卷积操作移动到 RoI pooling 模块之前。因此，对于所有的物体候选框，ROI卷积只使用一次。实验结果显示这类方法可达到11.1fps, 速度达到了MV3D[4]的5倍。

Segmentation-based Methods 基于分割的方法

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候选框，以节省计算量，如图7(B)所示。



与刚刚的多视角Multi-view的方法[4],[97],[103]相比，这类方法达到了更好的物体recall，并且更适合一些目标高度遮挡和拥挤的复杂场景。

[104]中，Yang et al使用了2D的分割网络来预测前景（foreground pixels）的像素并将其投影至点云中，以此来剔除掉多数的背景点。接着在这些前景点中生成候选框，并且设计了一种新的标准称之为PointsIoU来减少候选框的冗余性和模糊性。

跟着[104]的脚步，[105]提出了PointRCNN的框架。具体而言，他们直接对3D点云进行分割，然后得到前景点，并且将语义特征和局部空间特征融合从而得到高质量的3D boxes。

[106] following [105]中的RPN，提出了一种利用图卷积网络来进行3D物体检测。具体而言，利用图卷积，引入了两个模块来改进refine物体的候选框。第一个模块R-GCN利用一个候选框中的所有点，得到每个候选框的特征集成。第二个模块C-GCN将所有候选框中的每一帧信息融合起来，利用环境来回归准确的物体boxes。

[107]将点云投影至基于图像 image-based 的分割网络的输出，并将语义预测值附加到这些点上。

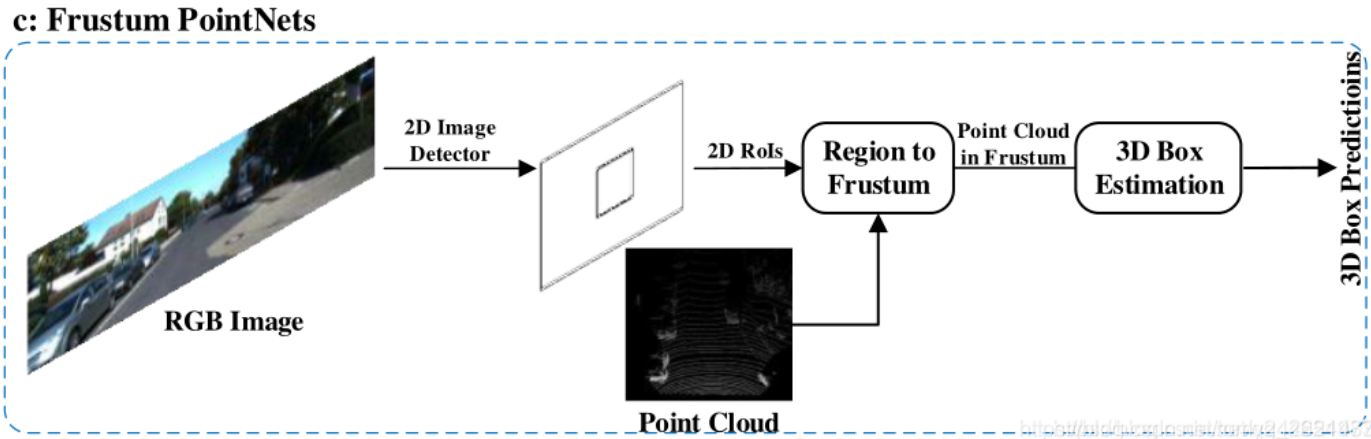
[109]得到了显著的性能提升，通过将涂色的点送入至一些检测器中[105, 108]。

[110]将每个点与spherical anchor相关联，每个点的语义值用来移除多余的anchors。这样的方法得到了更好的recall以及有着更小的计算消耗。与此同时，文中提出了PointsPool层，对候选框中的内部点学习相容的特征（compact features），并且引入了并行的IoU来提高位置的准确度的检测性能。

实验结果证实这样的方法在KITTI数据集[10]上较难的集合（car class）的性能比[99, 105, 111]的性能优越很多，并达到了12.5fps。

Frustum-based Methods 基于椎体的方法

这类方法首先利用现有的2D物体检测子，产生2D的候选矩形框，接着对每个2D的候选框提取3D的锥体候选框，如下图所示。尽管这类方法可以有效地给出3D物体的坐标，但step-by-step步进式的pipeline流水线使得它们的性能受到2D图像检测子的限制。



F-PointNets[112]为此类detection方向的先驱工作。它在每个2D区域上产生一个锥形的候选框（frustum proposal），并且应用PointNet[5]（或PointNet++[27]）来学习各个3D锥体的点云特征，从而进行3D box的估计。

在随后的工作中，[113]提出了Point-SENet模块，来预测一系列的缩放因子，从而被用来突出有用特征和抑制无用特征。同时他们也将PointSIFT[114]模块集成至网络中，来获取点云的方向信息，其可以得到对形状尺度的强鲁棒性。该方法在[10], [115]的数据集上，与F-PointNets[112]相比得到了显著的提高。

方法[116]利用了2D image 区域和对应的锥体点来回归3D boxes。为了融合image 特征和点云的全局特征，他们提出了全局的融合网络来直接回归box的角坐标。他们也提出了稠密的网络网络来预测各个点对于各个角的补偿（offsets）。

[117]第一次从2D图像中估计2D的bounding boxes和3D物体姿态，提取多个几何上可行的对象候选。这些3D候选对象被送入至box 回归网络来预测准确的3D物体boxes。

[111]对于各个2D区域，在锥体轴上产生一系列的锥体，并使用PointNet来对各个锥体提取特征。锥体层次的特征用来产生2D特征图，再被送入至FCN 全连接网络来估计3D box。该方法在基于2D图像的方法中达到了state-of-the-art的性能，并且在KITTI积分榜上排在很靠前的位置。

[118]首先在鸟瞰图BEV上得到初步的检测结果，接着基于鸟瞰图的预测结果，提取小部分点的子集，再应用局部的微调网络来学习局部特征，预测高精度的3D bounding boxes。

其他
……

3.1.2 Single Shot Methods 直接映射

这类方法使用单阶段的网络，直接预测类别概率和回归物体的3D bounding boxes。这类方法不需要产生区域候选框和后处理。结果是，这类方法有着很快的速度，很适合实时的应用。根据输入数据的形式，single shot方法可分为两类：基于鸟瞰图的方法和基于点云的方法。

BEV-based Methods 基于鸟瞰图的方法

这类方法将BEV表示作为输入。

[100]将场景的点云离散化，使用FCN来预测物体的位置和航向角。该方法超越了大多数 single shot 方法([125],[126],[127])并且达到了28.6fps。之后，[128]利用HP map（High-Definition 高清）提供的几何和语义先验信息，提高了[100]的鲁棒性和检测性能。

Point Cloud-based Methods. 基于点云的方法

这类方法将点云转换至一般的表示（例如2D map），接着使用CNN来预测对象的类别和3D boxes。

[125]提出了使用FCN进行 3D object detection 。他们将点云转换至2D point map，使用2D FCN来预测bounding boxes和物体的置信度。

之后，[126]将点云离散化至4D的张量，其维度分别为：长度，宽度，高度和channel，接着将2D FCN的方法延伸至3D来进行3D的物体检测（object detection）。与[125]相比，基于FCN的3D方法达到了大于20%准确率的收益，但是由于3D卷积核数据的稀疏性，消耗了更多的计算资源。

为了解决体素 voxels 稀疏性的问题，[127]利用了feature-centric voting scheme（特征为中心投票机制），为每个非空的体素生成一组的votes，最后通过将votes相加的方式得到卷积的结果。它的计算复杂度与被占据的体素数量成正比。

[130]通过堆叠多个稀疏3D CNN，构建了3D的backbone网络。这样的设计节约了内存并且加速了计算。这个3Dbackbone网络提取了丰富的物体检测的3D特征，并且并未引入计算量的负担。

[108]提出了基于体素的端到端的可训练框架VoxelNet。他们将点云分割成等间距的体素，将每个体素的特征编码成4D的张量。然后使用RPN（region proposal network）网络来产生检测结果（detection results）。尽管该方法效果很好，但由于体素的稀疏性和3D卷积操作，该方法运行速度很慢。之后，[120]使用了稀疏的卷积网络[134]来提高[108]的推断效率。

[131]通过将图像和点云特征在早期融合的方式，扩展了VoxelNet的工作。具体而言，他们将[108]产生的非空体素投影至图像，使用预训练的网络对各个投影的体素提取图像特征。这些图像特征与体素特征相级联，来预测准确的3D boxes。这类方法利用了多模态的信息，来减少false postivies and negatives。

[109]提出了3D物体检测子称为PointPillars。该方法利用了PointNet来学习点云的特征，将这些学到的特征编码伪图像（pesudo images）。然后使用2D的物体检测流水线（pipeline）来预测3D bounding boxes（边界框）。PointPillars在Average Precision（平均精度 AP）的指标上，超越了大多数的融合方法（MV3D[4], RoarNet[117], AVOD[97]）。并且，PointPillars在3D和BEV KITTI benchmarks上达到了62fps。

Other Methods

[132]提出了一种有效的3D目标检测子称之为LaserNet。该方法在各个点上预测bounding boxes的概率分布，然后结合各个点的分布来产生最后的3D object boxes。接着，点云的dense range view representation （密集视图(RV)表示）作为输入，使用fast mean-shift algorithm来降低逐点预测产生的噪声。LaserNet在0到50米的范围内实现了最先进的性能，其运行时间明显低于现有的方法。

[133]扩展LaserNet以利用RGB图像提供的密集纹理(例如，50到70米)。具体来说，通过将3D点云投影至2D图像使得LiDAR点和image点关联，并利用这种关联将RGB信息融合到3D点中。他们还将3D语义分割作为辅助任务以learn better representations。该方

法在保持LaserNet的高效率的同时，在长距离(例如50到70米)目标检测和语义分割方面都取得了显著的改进。

3.2 3D Object Tracking 3D物体跟踪
给定一个物体在第一帧时的位置，目标跟踪的任务是估计它在之后帧的状态。由于3D物体跟踪可以使用点云中丰富的几何信息，人们期待用它来克服在2D图像上追踪任务的困难，包括遮挡，光照以及尺度的变化。

Siamese network……

3.3 3D Scene Flow Estimation
类似于2D视觉中的光流估计，已经有几种方法开始从点云序列中学习有用的信息(如三维场景流、空间临时信息)。

[142]提出了FlowNet3D，在一系列连续点云中直接学习场景流（scene flows）。FlowNet3D通过flow embedding layer， 学习point-level的特征和运动特征（motion features）。然而FlowNet3D存在两个问题。第一，一些预测的运动向量（motion vectors）与真实值差别非常大；第二，很难将FlowNet应用至非静态的场景，尤其是有着可形变物体的场景。

为了解决该问题，[143]引入了余弦距离的损失函数来最小化预测值与真实值之间的夹角。同时，他们提出了point-to-plane的距离损失函数，来提高刚性的和动态的场景的准确率。实验结果显示这两种损失函数将FlowNet3D的准确率从57.85%提升至63.43%，并且加速和稳定了训练过程。

[144]提出了HPLFlowNet（Hierarchical Permutohedral Lattice FlowNet ），从大规模的点云中直接估计场景流。文中提出了一些bilateral convolutional layers来存储结构信息，同时降低计算消耗。

为了有效地处理序列点云，[145]提出了PointRNN, PointGRU和PointLSTM，以及一个sequence-to-sequence model 来追踪移动点（moving points）。PointRNN, PointGRU和PointLSTM能够捕捉空间-时间信息，并且建模动态的点云。

类似地，[146]提出了MeteorNet来直接从动态点云中学习表示。该方法试图从时间和空间上的邻近点学习总体特征。

[147]提出了两个自监督的损失函数，在大量无标签的数据集上训练网络。他们的主要思想是：一种鲁棒的场景流估计方法应该在向前预测和向后预测时均有效。由于场景流标注不可用，预测得到的转换后的点的最近点，被当做是假想的真实值。然而，真正的真实值可能与它不同。为了避免这个问题，他们在相反的方向计算场景流，并且提出了cycle consistency loss。实验结果显示这种自监督的方法超过了现有自监督学习方法中的SOTA（state-of-the-art）性能。

3.4 Summary

KITTI基准是自动驾驶领域中最有影响力的，并且在学术和工业领域有着广泛的应用。表2和表3展示了不同方法在KITTI test 3D and BEV benchmark上的结果。

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	-	-	-	-	-	-	-	-	-

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	-	-	-	-	-	-	-	-	-

可以观察到：

> Region proposal-based methods 是最常见的方法，在KITTI test 3D，
> BEV上的性能均超出了single shot methods。 现有的3D目标检测子（3D object detectors）有两个限制。第一，长范围的检测能力较弱。第二，如何充分利用图像中的纹理信息（texture information）仍然是个公开的问题。 多任务学习（ Multi-task learning）是在3D目标检测中未来的方向。例如，[99]通过合并多种任务，学习跨模态的表示来得到SOTA的检测效果。 3D物体跟踪（
> 3D object tracking）和场景流估计（scene flow estimation）是较新的研究方向，自2019年来受到越来越多的关注。

4、3D Point Cloud Segmentation

3D点云分割既需要了解全局的几何结构，又需要了解每个点的细粒度细节。根据分割的粒度，3D点云分割方法可分为以下三类：语义分割（场景级 scene level））、实例分割（物体级 object level）和 part segmentation（part level）。

4.1 3D Semantic Segmentation 3D 语义分割

给定一个点云，语义分割的目标是，根据语义信息，将各个点分成一定的子集。与3D

shape classification（第2节）的分类类似，语义分割可分为两种方法：基于投影的方法和基于点的方法。

4.1.1 Projection-based Networks 基于投影的网络

Intermediate regular representations（中间正则表示）可被分成以下几种：多视角(multi-view)表示[148], [149]、球状(spherical)表示[150], [151], [152]、体素(volumetric)表示[153], [154], [155]、超多面体晶格(permutohedral lattice)表示[156], [157]以及混合(hybrid)表示[158], [159]。具体可见下图。

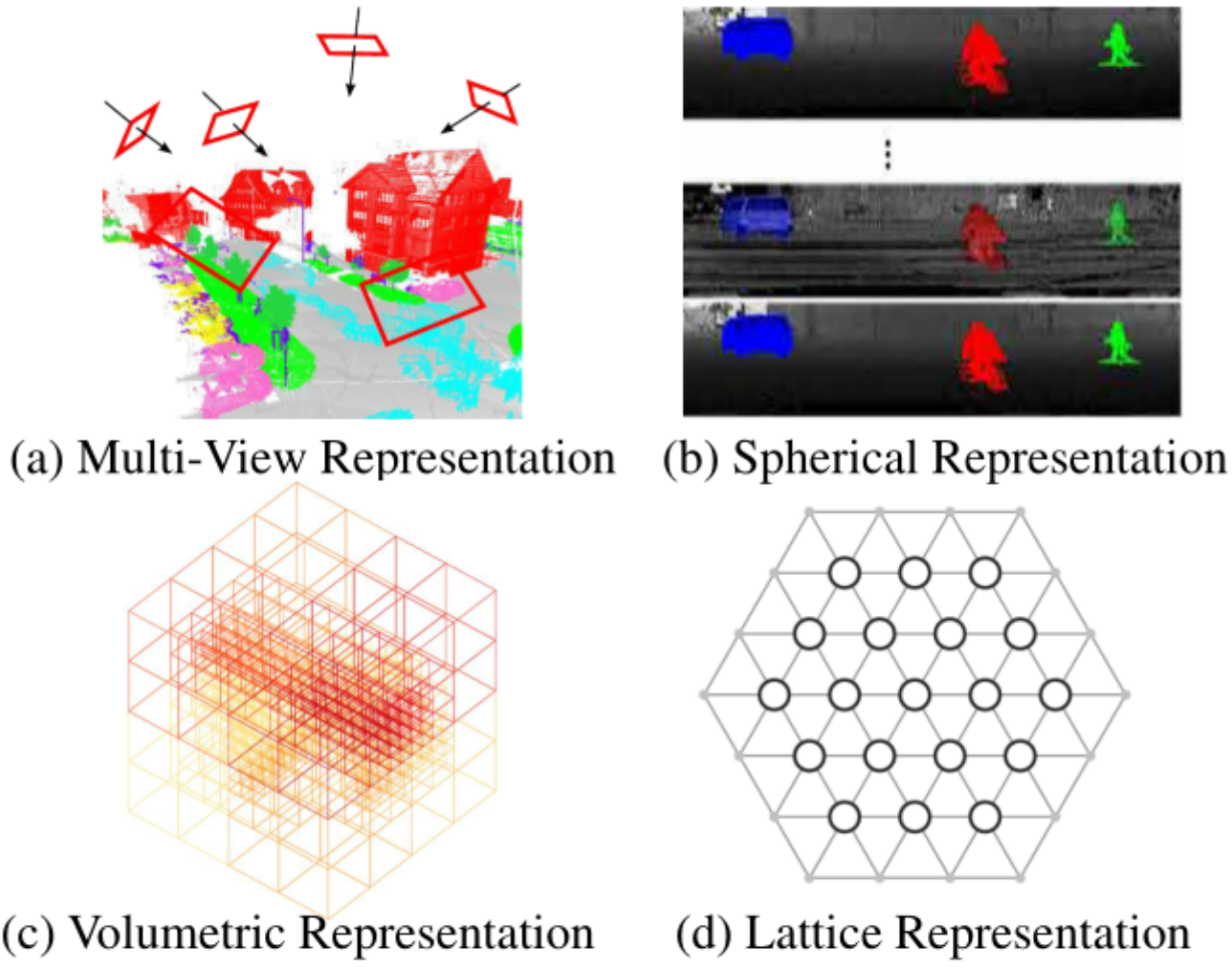


Fig. 9: An illustration of the intermediate representation of projection-based methods.

4.1.1.1 多视角表示 Multi-view Representation

[148]首先将3D点云从多个虚拟的相机视角投影至2D平面上，接着，使用 multi-stream FCN 对合成图像进行像素级分数预测。最终，通过融合不同视图上的重投影分数（re-projected scores ）来获得每个点的最终语义标签。

相似地，[149]首先利用多个相机位置，得到点云的一些RGB和深度图快照。接着使用2D segmentation networks ， 对这些快照进行像素级的标注label，使用残差校正(residual

correction)进一步融合从RGB和深度图像预测的分数。

基于点云是从局部欧式曲面上采样得到的假设， [161]引入了tangent convolutions进行稠密的点云分割。该方法首先将各个点周围的局部曲面投影至虚拟的切平面。Tangent convolutions在曲面上直接进行。该方法具有很强的可扩展性，能够处理几百万个点的大规模点云。

总的来说，多视角分割方法的性能对视角的选择(viewpoint selection)和遮挡(occlusions)非常敏感。同时，这类方法并未能完全利用潜在的几何和结构信息，因为投影操作不可避免地引入了信息损失。

4.1.1.2 球状表示 Spherical Representation

为了得到更快更准确的3D点云分割，[150]提出了基于SqueezeNet和条件随机场(Conditional Random Field (CRF))的端到端的网络。

为了进一步提升分割准确率，引入了SqueezeSegV2[151]，通过使用无监督的domain adaptationpipeline 解决domain shift 问题。

[152]提出了RangeNet++，针对LiDAR点云进行实时语义分割。首先将2D深度图像的语义标签转移至3D点云上，然后使用基于KNN的后处理步骤来减缓离散化误差和推理输出模糊的问题。

与单一的视角映射相比，球映射保持了更多的信息，并且更适合激光雷达（LiDAR）点云的标注。然而，这样的中间表示不可避免地引入了一些问题，比如离散化误差和遮挡问题。

4.1.1.3 体素表示 Volumetric Representation

[163]首先将点云分成一系列占有的体素（occupancy voxels）。接着将这些中间数据送入至fully-3D CNN中进行体素级别的segmentation。最后，为一格体素（a voxel）内的所有点分配与该体素相同的语义标签label。该方法的性能极其受限于体素粒度(granularity of the voxels)和点云分割引起的边界伪影(boundary artifacts)。

之后，[164]提出SEGCloud来得到更细粒度和全局一致(global consistent)的语义分割。该方法引入了确定性的三线性插值，将由3D-FCNN产生的粗糙的体素预测映射回点云中，接着使用Fully Connected CRF，确保推测出的点云有着空间上的一致性。

[153]引入了一种基于核的变分自编码器结构，对每个体素内部的局部几何结构进行编码。这里摒弃了binary occupancy representations， 使用RBF得到连续的表示，并捕获到每个体素中点的分布。再使用VAE将各个体素中的点分布映射至紧凑的隐空间，最后使用CNN得到鲁棒的特征表示。

良好的可扩展性是体素表示中的优点之一。具体而言，基于体素的网络（volumetric-based networks）可以在不同空间大小的点云中自由训练和测试。在Fully-Convolutional Point Network（FCPN）中，首先从点云中提取出来不同级别的几何相

关性，再使用3D卷积和加权的average pooling 来提取特征、合并依赖关系。该方法可处理大规模的点云，并且在推断时有着良好的尺度扩展性质(scalability)。

[166]提出了ScanComplete来实现3D补全，以及对各个体素进行语义标注。该方法利用了全卷积网络(fully-convolutional neural networks)的尺度扩展性(scalability)，在训练和测试阶段可以适应不同大小的输入数据。使用从粗到细的策略来提高预测结果的分辨率。

很自然地，体素表示是稀疏的，其中非零元素的数量仅仅占很小一部分。因此，在空间上稀疏的数据使用稠密的卷积网络是比较无效的。为此，[155]提出了子流形的稀疏卷积网络(submanifold sparse convolutional networks)。该方法通过限制卷积的输出只能与被占据的体素有关，从而显著降低了内存和计算成本。同时，该稀疏卷积还可以控制提取出的特征的稀疏性。该子流形稀疏卷积很适合处理高维度且空间较稀疏的数据。

更进一步，[167]提出了一种用于三维视频感知的4D时空卷积神经网络（4D spatio-temporal convolutional neural network） “Minkowski Net”。

综上所述，体素表示很自然地保留了3D点云的邻域结构。其规范的数据形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，体素化的过程内在地引入了离散化的伪影和信息损失。通常，高分辨率会导致较高的内存和计算消耗，低分辨率引入了信息的损失。在实际中如何选择合适的网格分辨(grid resolution)率是non-trivial(不平凡的)的。

4.1.1.4 超多面体晶格表示 Permutohedral Lattice Representation

[156]提出了基于双边卷积层（Bilateral convolution layers -BCLs）的稀疏晶格网络（Sparse Lattice Networks -SPLATNet）。该方法首先将原始点云插入至超多面体稀疏晶格(permutohedral sparse lattice)，再使用BCL对占据的部分进行卷积。得到的输出再重新插回原始点云。此外，该方法还允许灵活地联合处理多视图图像和点云。

更进一步，[157]提出了LatticeNet来实现有效的处理大规模点云。还引入了与数据相关的插值模块 DeformsSlice，将格点要素(lattice feature)反投影到点云中

4.1.1.5 混合表示 Hybrid Representation

为了进一步利用所有可用信息，许多方法试图学习多模态特征(multi-modal features)。

[158]提出了joint 3D-mult-view网络，来组合RGB 特征和几何特征。一个3D CNN stream 和一些2D CNN stream用来提取特征，另一个可微分的back-projection layer用来合并3D和2D特征。

更进一步，[168]提出了unified point-based network来学习2D纹理信息，3D结构和全局特征。该方法直接应用基于点的网络(point-based networks)来提取局部几何特征和环境信息。

[159]提出了Multiview PointNet（MVPNet）来集成2D多视角特征和空间几何特征。

4.1.2 Point-based Networks 基于点的网络

基于点的网络直接在点云上进行操作。然而，点云通常是无序且无结构的，使得直接应用标准的CNN不现实。为此，先驱的工作PointNet[5]用来对每个点进行特征学习，使用的是标准的MLP和全局特征。基于PointNet，一系列基于点的网络被提出。总体而言，这类方法可大致分为以下几类：基于各个点的MLP方法(pointwise MLP method)，基于点卷积的方法(point convolution methods)，基于RNN的方法(RNN-based methods)和基于图的方法(graph-based methods)。

4.1.2.1 Pointwise MLP Methods

这类方法通常利用共享的MLP作为网络中的基本单元。然而，由共享MLP提取出的各个点上的特征，并不能获取到点云中的局部几何关系(local geometry)，以及点与点之间的关系(mutual interactions)[5]。为了获取各个点周围更广泛的信息，以及学习到更丰富的局部结构(local structures)，有很多方法被提出，包括基于邻近点特征池化的方法(methods based on neighboring feature pooling)，基于注意力机制的集成(attention-based aggregation)以及局部-全局的特征级联(local-global feature concatenation)。

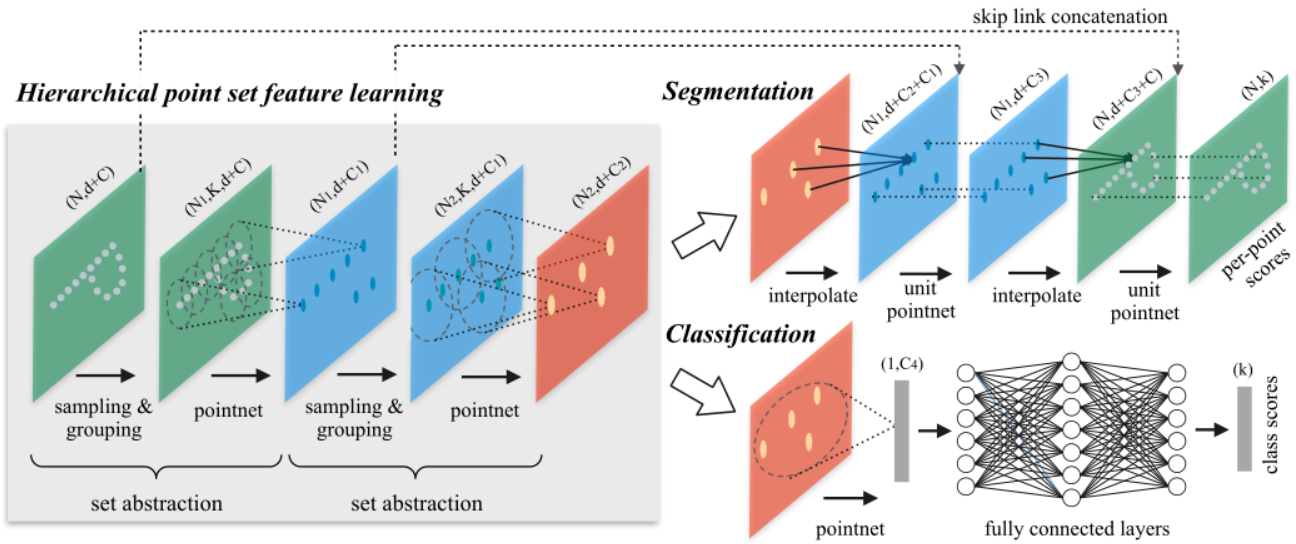


Fig. 10: An illustration of the PointNet++ [27] framework.

Neighboring feature pooling

为了获取局部的几何形式，这类方法通过将局部邻域点集成的方式，对各个点学习特征。具体而言，PointNet++[27]将点分层次，逐步地分成一些组，如下图所示。多尺度的

grouping和多分辨率的grouping来克服点云多样性造成的问题。

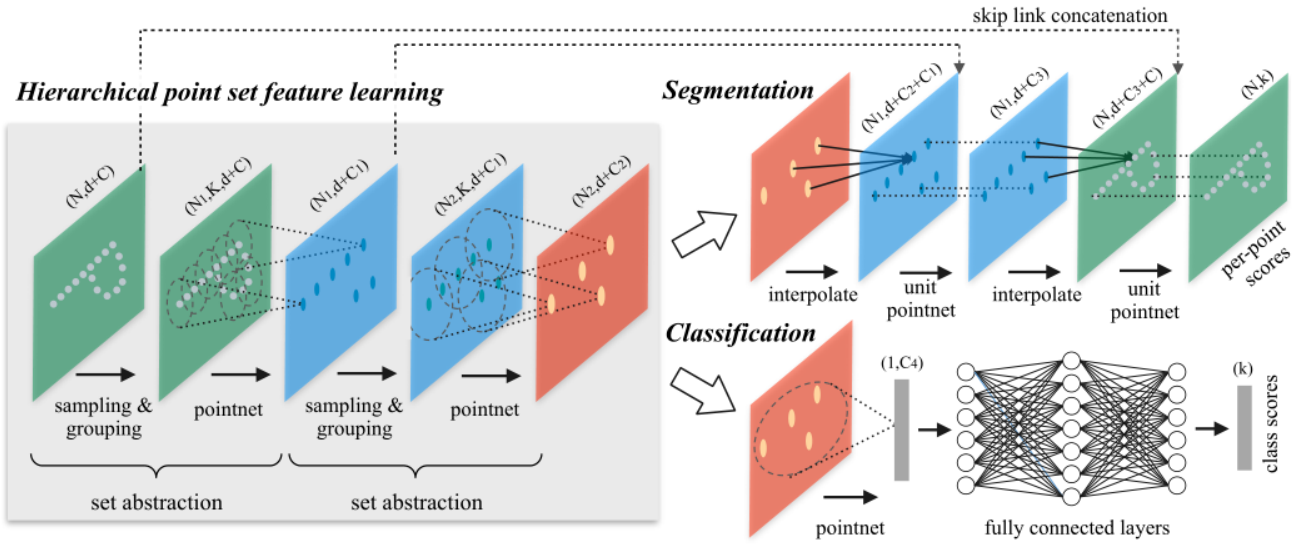


Fig. 10: An illustration of the PointNet++ [27] framework.

之后，[114]提出了PointSIFT模块来实现方向的编码和scale awareness。该模块通过使用3阶段的有向的卷积操作，将8个空间方向的信息堆叠并且编码，将多尺度的特征提取并级联来实现对不同尺度的适应性。

与PointNet++中使用GROUPING的方法不同，[169]利用K-Means聚类和KNN的方法在世界空间和特征空间定义两种邻域。基于这样的假设：来自于同一类的点在特征空间中应当接近，该论文提出了pairwise distance loss and a centroid loss来对特征学习进行正则。

为了建模点与点之间的相互关系，[31]提出了PointWeb来寻找局部区域内所有点对之间的关系。[170]提出了置换不变性的卷积称之为Shellconv。[95]提出了有效、轻量的网络称为RandLA-Net实现大规模的点云处理。该方法利用随机样本采样，在内存和计算方面提升很多。提出的局部特征集成用来获取和保持几何特征。

Attention-based aggregation

为了进一步提升分割的准确率，[90]针对点云分割，提出了基于注意力的机制。

[29]提出了组随机注意力机制(group shuffle attention)来建模点之间的关系，并且提出了具有置换不变性、task-agnostic以及可微分的Gumbel Subset Sampling(GSS)，来替代被广泛应用的Furthest Point Sampling(FPS)最远点抽样方法。该方法对离群点不敏感，并且可以选择具有代表性的点的子集。

为了更好地获取点云的空间分布，[171]提出了Local Spatial Aware(LSA)层来学习空间感知权重。

与CRF类似，[172]提出了Attention-based Score Refinement(ASR)模块对分割的结果进行后处理。初始分割结果通过pooling的方式进行修正。该模块很容易被集成至其他的深度网络中来提升分割效果。

Local-global concatenation

[85]提出了置换不变性的PS2-Net，将点云的局部结构(local structures)和全局信息(global context)合并。重复叠加Edgeconv[60]与NetVLAD[173]，以获取局部信息和场景级别的全局特征（scene-level global features）。

4.1.2.2 Point Convolution Methods 点卷积法

这类方法通常试图提出在点云上进行更有效的卷积操作。

[49]提出了一种逐点卷积算子，其中邻域点被合并至kernel cell，然后与核权重进行卷积。

[174]提出了称之为PCCN的网络，该网络基于参数化的连续卷积层。该层的核函数由MLP参数化，横跨连续向量空间。

[42]提出了Kernel Point Fully Convolutional Network(KP-FCNN)，基于Kernel Point Convolution(KPConv)。具体而言，KPConv的卷积权重由欧式空间的距离决定，卷积核的点数（number of kernel points）也并不固定。卷积核点（kernel points）的位置由一个最优化问题确定。

在[175]中，作者提供了丰富的消融实验（ablation experiments）和可视化结果展示了集成方法中，感受野的重要性。同时他们提出了Dilated Point Convolution(DPC)操作，来集成邻近点的特征，进而取代KNN(K nearest neighbours)的方法。该方法在提升感受野（the receptive field）上非常有效，并且可以容易地集成至 aggregation-based networks。

4.1.2.3 RNN-based Methods

为了从点云中获取固有的上下文特征(context features)，RNN也被用来进行点云的语义分割。

基于PointNet[5]， [180]首先将一大块点云转换成多尺度的块和网格块来获取输入级别的环境。接着，使用PointNet对各个块提取特征并送入Consolidation Units 或 Recurrent Consolidation Units来获取输出级别的环境信息。实验结果显示，这样处理空间环境信息的方法在提高分割性能时是很重要的。

[179]提出了一种轻量的模块，利用了slice pooling layer将无序的点云特征转换成有序的特征向量。

[181]提出了Pointwise Pyramid Pooling (3P)模块来获取从粗到细的局部特征，并利用双向的RNN来实现端到端学习。然而这类方法损失了丰富的几何特征和密度分布[189]。

[189]提出了Dynamic Aggregation Network(DAR-Net)来同时考虑全局场景复杂度和局部几何特征。

[190]提出了3DCNN-DQN-RNN。该网络首先使用3DCNN学习空间分布和颜色特征，使用DQN进一步定位类别物体。最后级联的特征向量送入RNN中获取最后的分割结果。

4.1.2.4 Graph-based Methods 基于图的方法

为了获取3D点云中潜在的形状和几何结构，一些方法使用了图神经网络（graph networks）。

[182]将点云看做是一些相连的简单形状和超点(Superpoint)的集合，并且使用属性有向图(attributed directed graph)（即超点图 superpoint graph ）获取结构和环境信息。接着，将大规模的点云分割问题分成三个子问题，即，geometrically homogeneous partition（几何均匀划分）, superpoint embedding（超点嵌入） and contextual segmentation（上下文分割）.

为了进一步提升，[183]提出了有监督的框架，来 oversegment a point cloud into pure superpoints（将点云过度分割为纯超点）。

为了更好地获取高维空间中的局部几何关系，[191]提出了基于Graph Embedding Module(GEM) 和 Pyramid Attention Network(PAN)的网络PyramNet。GEM模块将点云表述为有向无环图，并且在构建相似度矩阵时，利用协方差矩阵代替欧式距离。在PAN模块中，使用4个不同尺寸的卷积核来提取特征。

在[184]中，提出Graph Attention Convolution 用来从局部相邻集合中有选择性地学习相关特征。

4.2 Instance Segmentation 实例分割

与语义分割 semantic segmentation 相比，实例分割更具有挑战性因为它需要更准确和更小的细粒度，具体而言，他不仅需要将有着不同语义的点分辨出来，还需要将有着相同语义的实例 (instance)分出来。总体而言，目前的方法可分为两个方向：基于候选框的方法(proposal-based)以及不需要候选框的方法(proposal-free)。一些里程碑式的方法具体见下图。

(按时间顺序概述了典型的三维点云实例分割方法)

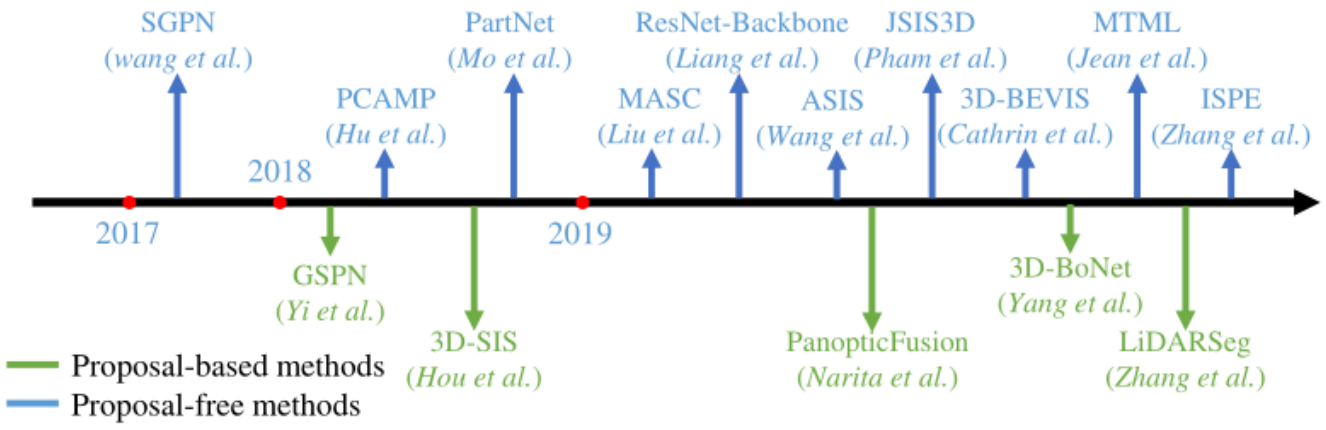


Fig. 11: Chronological overview of representative 3D point cloud instance segmentation methods.

4.2.1 Proposal-based Methods 基于候选框

这类方法将实例分割问题分成两个子任务：3D物体检测（3D object detection ）和实例mask的预测（instance mask prediction）。

[192]提出了3D fully-convolutional Semantic Instance Segmentation (3D-SIS) network，来实现在RGB-D数据上的语义实例分割。该网络从颜色和几何中学习特征。与3D object detection 类似，3D Region Proposal Network（3D-RPN）和 3D ROI layer用来预测bounding box的位置，物体类别和instance mask。

根据合成分析策略，[193]提出了Generative Shape Proposal Network(GSPN)来产生3D候选框。这些候选框再通过R-PointNet修正。最终的标签通过预测各个点的二进制mask来得到。与直接从点云数据回归三维边界框不同，该方法通过加强几何理解，去除了大量无用的候选框。

通过将2D全景分割(2D panoptic segmentation)扩展到3D映射，[194]为实现大规模三维重建（3D reconstruction）、语义标注（semantic labeling）和instance segmentation，提出了一种在线三维映射系统（oneline volumetirc 3D mapping system）。该方法首先利用2D语义和实例分割网络来获得像素级的全景标签（panoptic labels ），然后将这些标签整合到 volumtric map 上。进一步使用全连接的CRF来实现准确的分割，该语义映射系统能够实现高质量的语义映射（ semantic mapping）和具有判别性的目标检测（object recognition）。

[195]提出了单阶段的，不需要anchor的端到端可训练网络—3D-BoNet，来实现点云上的 instance segmentation。该方法对所有可能的instance 直接回归大致的3D bounding boxes，接着利用点级别的二分类器（binary classifier）来获取实例标签。特别地，该 bounding box generation task是被当做是最优分配问题。同时，使用了

multi-criteria 损失函数来正则化生成的bounding boxes。该方法不需要任何的后处理操作，并且有很高的计算效率。

[196]提出了针对大规模户外LiDAR点云进行instance segmentation的网络。该方法使用self-attention blocks，在点云的鸟瞰图上学习特征表示（feature representation），根据预测的水平中心和高度限制获得最终实例标签（instance labels）。

总的来说，基于候选框的方法较为直观，并且实例分割的结果通常较好。然而该方法需要多阶段的训练并且需要对多余候选框进行裁剪。因此通常都需要更多的时间和计算资源。

4.2.2 Proposal-free Methods 不需要候选框

不需要候选框的方法[197-202]并没有目标检测的模块（ object detection module）。作为替代的是，他们通常将instance segmentation 认为是semantic segmentation（语义分割）后的聚类步骤。具体而言，需要现有的方法都基于这样的假设：属于同一实例的点应当有着相似的特征。因此这类方法通常聚焦于判别式的特征学习（discriminative feature learning）和点云聚类（point grouping）。

……

总体而言，不需要候选框的方法不需要耗费资源的区域生成步骤。然而，因为该方法不检测物体的边界，导致该方法的准确率较低。

4.3 Part Segmentation

零件分割（part segmentation of 3D shapes）的主要困难来自于两方面。第一，有相同语义标签（ semantic label）的部件（shape parts）有着较大的几何变化和不确定性；第二，该方法需要对噪声和采样具有鲁棒性。

[208]提出了VoxSegNet，在3D体素数据上来实现细粒度的零件分割。

[209]将FCN与surface-based CRF组合，实现端到端的3D 零件分割。他们首先从不同的视角产生图像来实现optimal surface coverage，并将这些图片送入至2D网络产生置信图。接着，使用surface-based CRF 将置信图集成起来，用来对整个场景打标签。

[210]引入了Synchronized Spectral CNN(SyncSpecCNN)，在不规则非同构形状图上实现卷积。

[211]通过引入Shape Fully Convolutional Networks(SFCN),在3D网格上实现了形状分割，并且将三种低层次的几何特征作为输入。接着利用基于投票的多标签graph cut来修正分割结果。

[212]提出了弱监督的CoSegNet进行3D形状分割。该网络将一些未分割的3D点云形状作为输入，接着通过最小化group consistency loss，产生形状零件的标签。与CRF类似，预训练的part-refinement网络用来修正并且去噪。

[213]提出了Branched Auto-encoder network(BAE-NET)用来unsupervised，one-shot和weakly supervised 3D shape co-segmentation。

4.4 Summary

下表展示了已有方法在公开数据集上的结果，包括：S3DIS[176], Semantic3D[9], ScanNet[102]和SemanticKITTI[177].

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	-	-	-	-	-	-	-

接下来这些问题需要进一步的探索。

Point-based networks 是最常见的方法。然而，点的表示通常没有明确的邻域信息，现有的大多数基于点的方法不得不求助于昂贵的邻域搜索机制（KNN, ball query）。这自然地限制了这类方法的有效性，因为邻域查找方法需要很高的计算资源和内存。
在 point cloud segmentation 中，从不平衡的数据中学习仍然是具有挑战性的问题。尽管许多方法[42], [170], [182]达到了不错的结果，但性能在较小类别的数据上仍然较差。
大多数的方法[5], [27], [52], [170], [171]在较少点的点云上进行（4096）。实际上，从深度sensor上得到的点云是非常稠密的。因此需要寻求处理大规模点云的有效分割方法。
一些工作[145], [146], [167]开始在动态点云中学习空间-时间的信息，期望时空信息能够帮助提高后续任务（如3D对象识别[3D object recognition]、分割[segmentation]和补

全[completion]）的性能。

5、CONCLUSION

本文章提出了当前针对3D understanding的一些SOTA方法，包括3D shape classification ， 3D object detection & tracking以及3D scene and object segmentation。对这些方法进行了全面的分类和性能比较。文中还介绍了各种方法的优缺点，并指出了可能的研究方向。

参考：

https://arxiv.org/abs/1912.12033

https://github.com/QingyongHu/SoTA-Point-Cloud

http://yulanguo.me/