机器学习报告

多模态 Anchor-free 3D单阶段检测器

1 问题介绍与分析

本项目旨在提出一种无锚框（Anchor-free）的SSD算法。

精准，实时的3D检测在自动驾驶中发挥重要的作用，但是目前的算法主要存在以下瓶颈：

- 有些方法使用单目或立体相机进行3D物体检测。尽管**图像可以提供细粒度的语义上下文，但由于缺乏深度信息，此类方法的性能受到限制**
- 基于激光雷达的方法**可以获得更高的精度和精确的深度信息，但对分辨率低、纹理信息稀疏的小目标不友好**
- 在多模态融合领域，目前的瓶颈在于大多数3D探测器都在鸟瞰图上提取特征，而鸟瞰图**很难与摄像机的前视图对齐。**

并且， 大多数基于体素的检测器更喜欢基于锚的检测头，基于锚的方法生成密集的锚盒，以保证较高的recall。 然而，对于KITTI中的复杂场景，基于锚定的方法将根据每个网格中的类别和方向箱的数量生成几十个锚定，从而**导致参数的立方级增长，并减缓推理**。相比之下，现有的基于体素的无锚检测器直接回归热图前景区域的边界框，能够实时运行，但性能有限。 最近的研究证明，**合适的标签分配可以显著提高无锚检测头的性能，这对其在三维检测中的应用具有启发性**。

为了解决上述问题，我们提出了一种**无锚框3D目标检测SSD算法**。具体的工作是：

我们使用OpenPCDet 3D目标检测框架，修改和替换了其中的backbone和head部分，在此基础上做了以下工作：

- 在基于体素的方法中，我们创造性地设计了一个**由OTA-3D驱动的无锚检测头，在精度和速度之间实现了良好的平衡**。目前，在KITTI和nuScenes数据集上，**我们的模型已经达到了与最先进的单阶段方法相当的精度与推理速度。**
- 我们比较了不同的语义编码方式并对其性能进行了定量分析。

2 背景和论文调研

在进行我们的工作之前，我们对已有的相关论文进行了调研，调研结果总结如下：

2.1 3D 点云目标检测

我们通过调研，发现目前主要有两类目标检测算法，一是single-stage,直接输出预测分和边界框，二是two-stage,在第一阶段生成region proposals(候选区域)，然后在第二阶段通过以全分辨率重新使用点云重新确定RP，其中，

Single-stage 相关的工作包括：

- VoxelNet首先对点云进行体素化，并通过微小的PointNet提取体素特征。
- SECOND利用稀疏卷积和子流形卷积来加速3D卷积。
- Point GNN通过一种新型的图形神经网络提取特征，并实现与最先进的两级检测器相当的性能。
- SASSD使用辅助网络进行框中心回归和分割，旨在学习结构和定位信息

这启发我们利用点的监督来消除语义分割中的边界模糊效应。

Two-stage 相关的工作包括：

- Voxel-RCNN通过基于体素的高效CNN生成感兴趣区域（ROI）
- PV-RCNN利用3D体素CNN和基于点的集合抽象的优势来学习更多代表性特征。
- $Part-A^2$通过在第一阶段预测对象内部零件的位置来丰富RoI特征，减少边界框的模糊性。

2.2 多模态融合

调研发现目前主要有4类多模态融合的方法

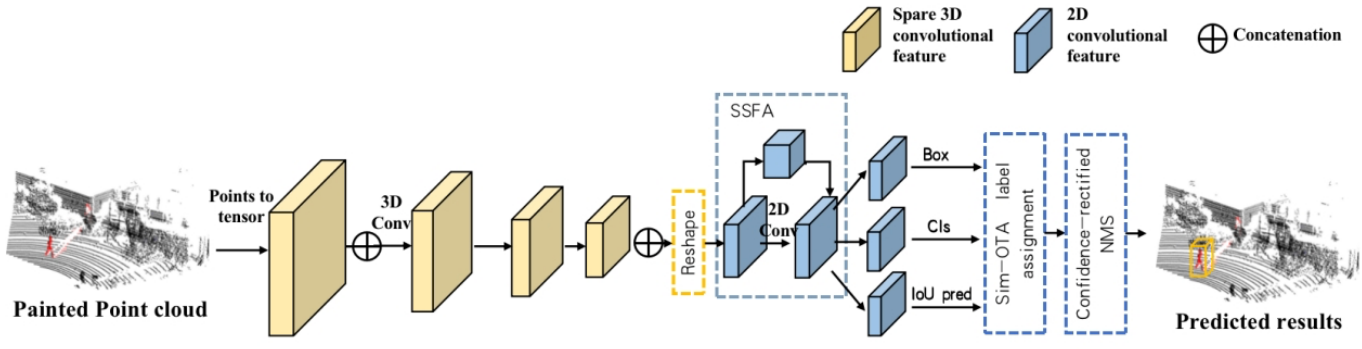
- Object-Centric Fusion (以对象为中心的融合)
- Continuous Feature Fusion (连续特征融合)
- Detection Seeding (检测种子)
- Sequential Fusion (顺序融合)

对于每个类别，其代表性的工作分别是：

- Object-Centric Fusion: MV3D以及AVOD, 其主要思路是从图像和点云投影视图中提取RoI特征，然后在BEV或前视图上执行深度特征融合。主要问题是投影过程会丢失空间信息，这大大降低了检测精度
- Continuous Feature Fusion: ContFuse，其主要思路是以不同的尺度融合了图像和激光雷达主干网络的特征。这些方法通常计算一种映射以将点云转换为图像平面。其主要问题是点云的BEV特征向量对应于二维图像中的多个像素，导致模糊特征对齐，从而限制了性能。
- Detection Seeding: Frustrum PointNet和ConvNet，其主要思路是利用2D检测来限制frustrum搜索空间，从而为3D提案设定种子。问题是这些检测器在很大程度上依赖于2D检测器的性能，且其召回率有一个上限
- Sequential Fusion: LRPD和PointPainting，它们都使用图像语义分割网络的输出来辅助目标检测。PointPaint将激光雷达点投影到语义分割结果中，并将绘制的点提供给3D对象检测器。然而，由于高级特征图的分辨率相对较低，在图像语义分割中会出现“边界模糊效应”。将它们重新投影到点云后，这种影响会变得更严重。

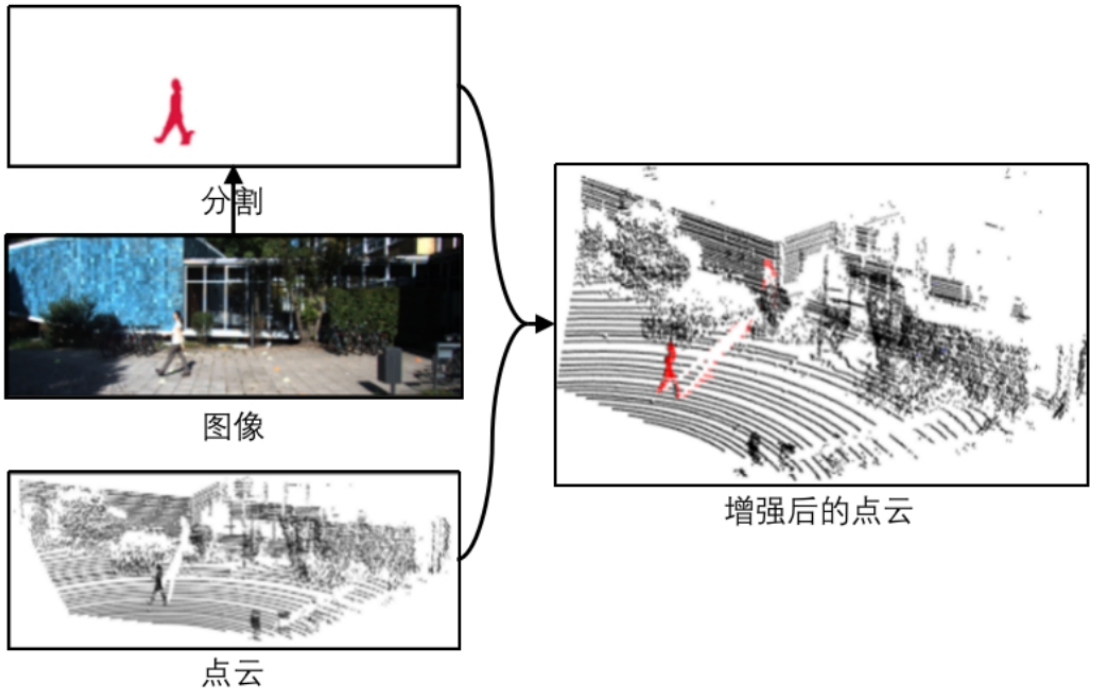
3 方法设计——网络结构

我们的框架包括数据处理， Backbone部分和Head 部分。



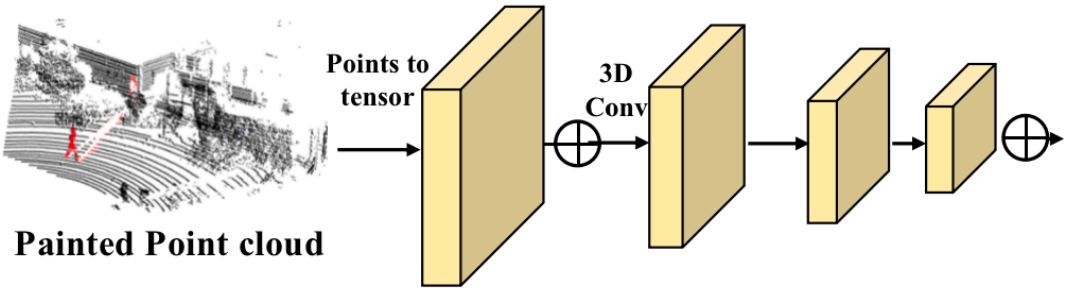
3.1 数据处理

在数据处理中，我们采用现成的图像语义分割器DeeplabV3+来输出每个像素语义类别的分数，然后将其附加到这些点上。在实际操作过程中，图像语义分割网络采用输入图像，并输出每个像素的语义类得分。 这些分数是图像的高级特征。 我们将LIDAR点转换为摄像机坐标系，将它们投影到图像中，然后将相应像素的分数附加到LIDAR点特征。



3.2 backbone 3D

我们的 backbone网络是一个典型的3D卷积网络。如下图所示，backbone网络由四个卷积块组成，每个卷积块都包含卷积核大小为3的子流形卷积。在高度这个维度进行特征的聚合产生BEV(鸟瞰图)特征图。



我们可以看到我们有两种不同的卷积类型，黄色的表示稀疏卷积，蓝色的表示我们常见的2D的卷积。在2D图像中，因为图像的信息都是比较密集的，每一个格子都会有对应的像素的信息；但是像点云这种结构，我们将它3D体素化之后会有很多格子都是空的，在这个时候我们如果使用3D卷积就会有很多资源的浪费，因为有很多空的空间，所以需要使用稀疏卷积。

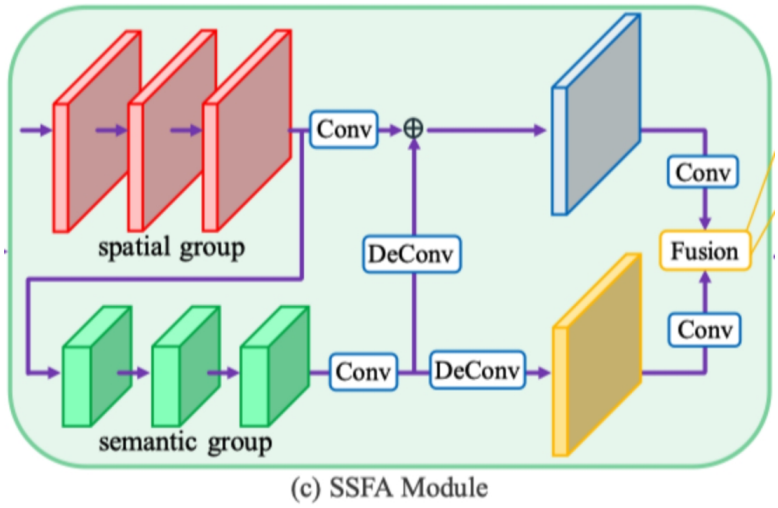
稀疏卷积是怎么计算的呢？就是我们只拿有点的格子进行 3D卷积，这样就节省了计算资源，但是空间中的空白信息同样重要，这个信息我们如何保存呢？我们取出有点的格子的时候需要记录下它的index，然后进行3D卷积计算，计算完成之后再通过它的 index 把把它还原到3D空间里。

> 对于稀疏卷积有两种：
>
> 第一种是空间稀疏卷积(Spatially Sparse Convolution)，在spconv中为SparseConv3d，例如经典论文 SECOND 中就用的是这个；

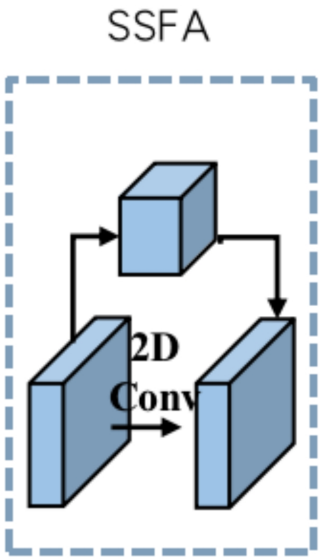
> 另一种是子流形卷积(Submanifold Sparse Convolution)， 在spconv中为SubMConv3d。只有当kernel的中心覆盖一个active site时(这里可以简单理解为卷积核覆盖的中心有值，这个概念在CVPR2018《Submanifold Sparse Convolutional Networks》被提出)，卷积输出才会被计算。可以看出子流形卷积保留了特征图的稀疏性。
>
> 我们使用的是子流形卷积。

3.3 backbone 2D——语义聚合模块（SSFA）

我们参阅了 AAAI 2021年的《CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud》，这篇论文提出了 SSFA 语义聚合模块。该模块可适应地融合**高级语义特征**和**低级空间特征**。相对于 SECOND 中的2D 特征提取模块，效果更好。



(c) SSFA Module



3.4 Head——OTA-3D 无锚检测头

我们首先阅读了CVPR2021年的这篇会议论文《Ota: Optimal transport assignment for object detection》，并将最佳运输分配（OTA）策略应用到我们的项目之中。 为了提高小物体的召回率，我们通过从固定区域内的网格中的预测来扩展小物体的正区域。

为了实现实时性能而不会降低精度，我们用OTA-3D构建了无锚检测头。 在此检测头中，我们将每个位置的预测从$N_{类} \times N_{方向}$组减少到1组。

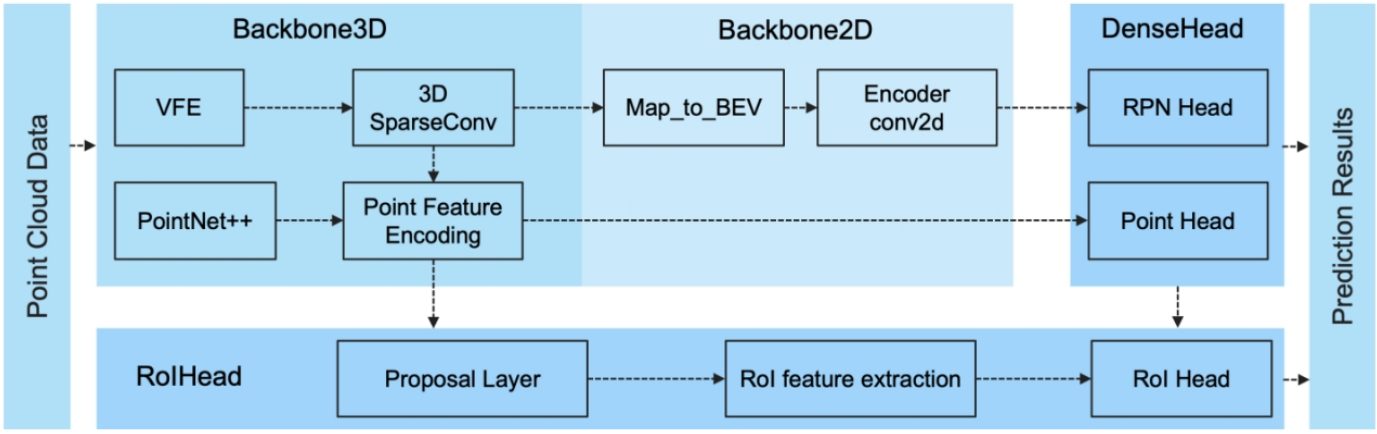
无锚点检测头有三个任务：分类，定位和与联合（IOU）预测，并具有置信函数，以纠正NMS（非最大抑制）后处理中使用的置信度。最后，我们使用全局最佳标签分配，以最少的全局成本找到最佳标签。

3.5 代码实现框架

我们的代码实现使用了OpenMMLab提供的模块化3D目标检测框架OpenPCDet，替换了Backbone 和 Head 部分。

> 该框架的作者是PV-RCNN的作者Shaoshuai Shi

我们替换了其中的 Backbone 部分和 Head部分，该框架的模块如下图所示：



我们最终提交的代码可以替换框架目录/OpenPCDet/pcdet/models下面的几个文件夹

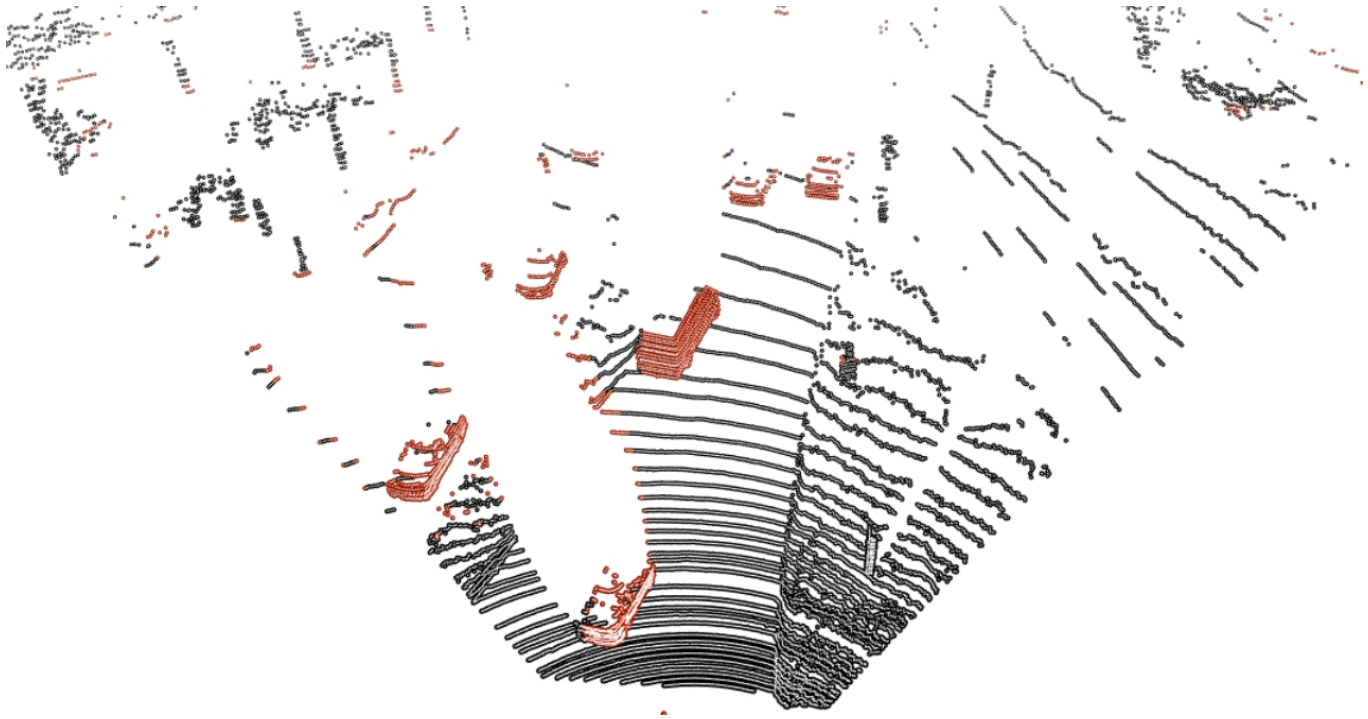
- backbones_2d
- backbones_3d
- dense_heads

4 实验结果

4.1 数据处理结果样例

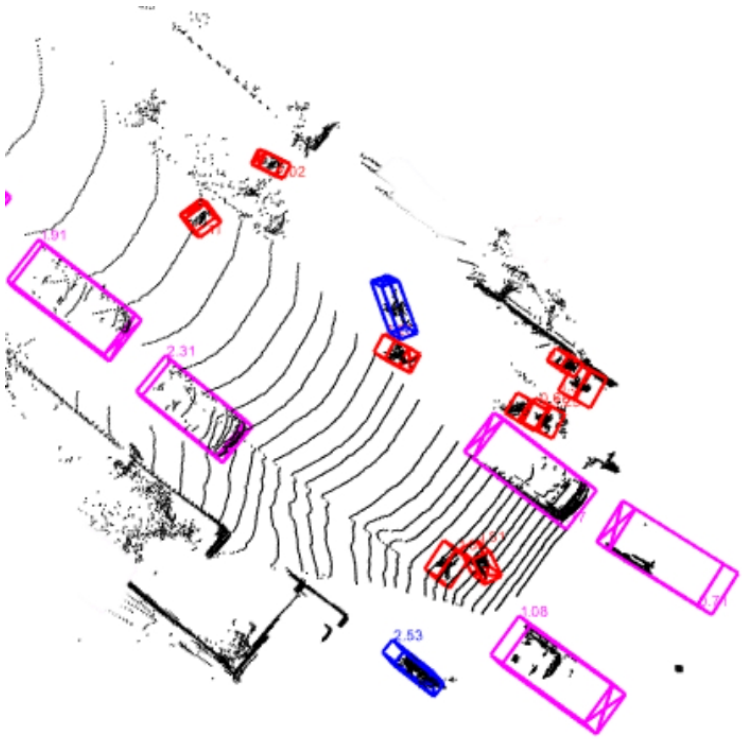
使用DeeplabV3+的语义分割结果：

增强后的点云：



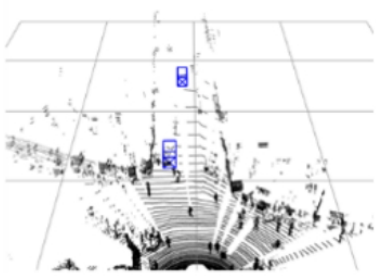
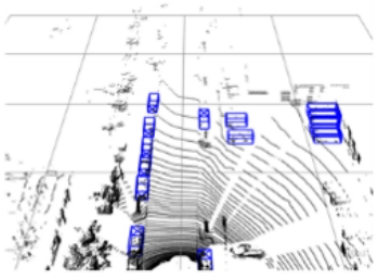
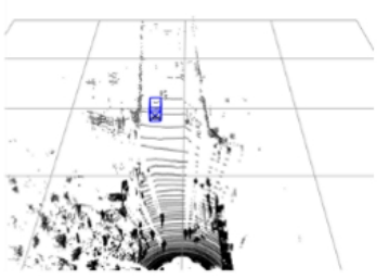
4.2 KITTI数据集样例结果展示

KITTI数据集的定性结果：粉红色，蓝色和红色框分别代表汽车，自行车和行人。

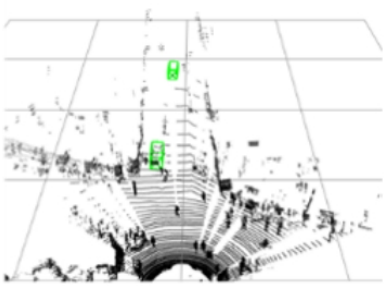
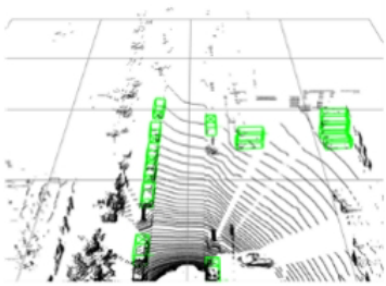
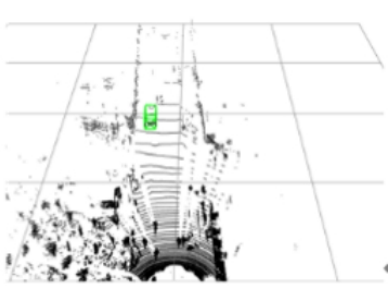


挑选的一个可视化结果：

1. 这是Groud Truth 真值

2. 这是我们的结果

4.3 KITTI测试集上的结果

如下图所示，我们的模型在“中等”和“困难”难度的“汽车”和”自行车“的检测水平大大提高大幅度优于其他相关模型。具体来说，我们的方法在mAP方面优于 PointPillars、SECOND分别为 6.03%、4.38% 。所以该模型更适合于检测比较有难度的模型，比如有更多的遮挡或者点云的分布更加稀疏点。

模型的推理速度与SECOND相比有很大的提升，主要是因为减少了大量的参数和并且采取了无锚框检测。然而，该模型在“行人”的测试集上的 mAP 低于PointPillars。原因之一是KITTI 的行人数量很多比汽车少很多，即使对数据采用数据增强，"行人"的多样性仍然缺乏；此外，BEV 上“行人”的体素分辨率比骑自行车的人或汽车差得多，所以要在 KITTI 数据集上通过体素化特征很好地学习"行人"检测器是比较困难的。

Method	Car-E	Car-M	Car-H	Ped-E	Ped-M	Ped-H	Cyc-E	Cyc-M	Cyc-H
SECOND	87.43	76.48	69.10	N/A	N/A	N/A	N/A	N/A	N/A
PointPillars	87.22	76.95	73.52	57.75	52.29	47.91	82.29	63.26	59.82
Ours	89.16	82.97	79.49	60.75	55.67	50.20	83.67	69.59	54.21

4.4 KITTI验证集上的结果

我们将我们的方法与其他单阶段方法进行了比较。可以看出，PointPillars 的行人检测的精度比较低，而我们的不同的语义编码方式成功解决了这个问题。此外，在与 SECOND 预测汽车类别的baseline相比，在数据集的三种难度下的AP的增长分别为（1.73%、6.49%、10.39%）。

Method	Car-E	Car-M	Car-H	Ped-E	Ped-M	Ped-H	Cyc-E	Cyc-M	Cyc-H
SECOND	87.43	76.48	69.10	N/A	N/A	N/A	N/A	N/A	N/A
PointPillars	87.22	76.95	73.52	57.75	52.29	47.91	82.29	63.26	59.82
Ours	89.16	82.97	79.49	60.75	55.67	50.20	83.67	69.59	54.21

5分析与讨论

5.1 OTA-3D无锚框检测头

当我们用了无锚框检测头和sim-OTA标签策略时，在中等难度的汽车检测方面的AP指标提高了2.08%。无锚框检测头和sim-OTA标签策略的结合不仅减少了参数量，而且在对比基于锚的方法如SA-SSD和CIA-SSD相比时有更好的性能。

此外，我们采用 3D D-IoU 损失和 IoU 预测分支来进一步提高每个类别的 AP。结果表明，这些策略对汽车之类的大型物体有很好的效果。

anchor-free	3D D-IoU	IoU pred	Car-M	Ped-M	Cyc-M
			76.48	51.14	66.74
√			78.56	54.55	67.65
√	√	√	81.85	53.70	68.13

5.2 不同语义编码方式的比较

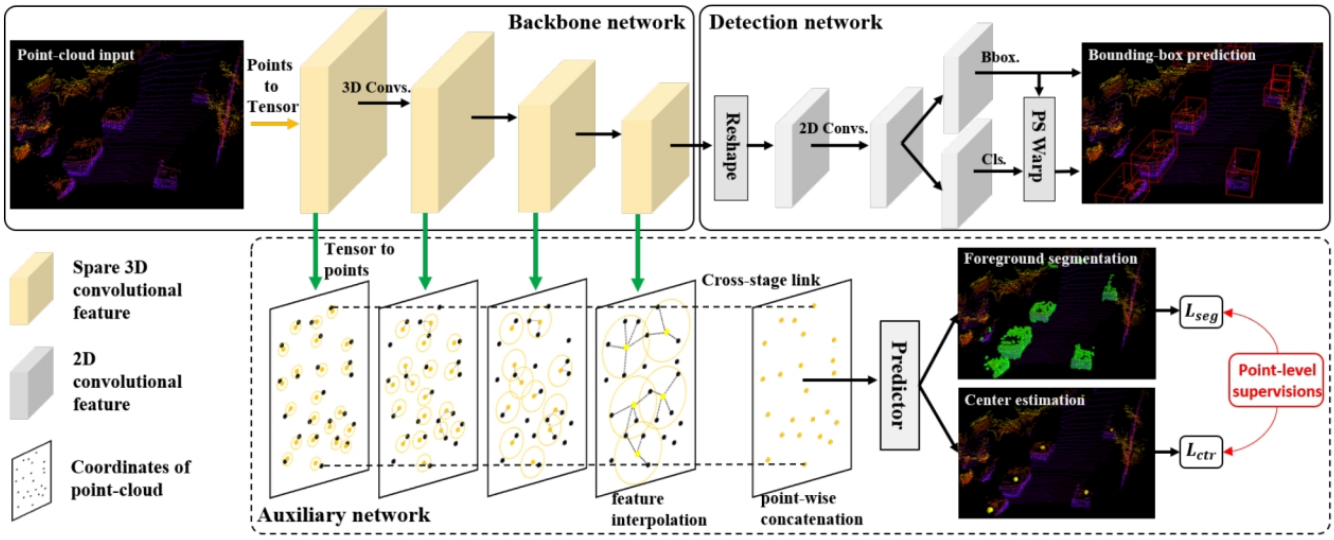
比较了三种不同的语义编码方式，分别为对象类别数值编码、分割分数以及one-hot 编码这四种不同的编码方式。其中，分割分数编码比用数值编码在汽车类别检测方面，AP值增幅最大，为0.9%。我们认为分割分数意味着分类置信度信息，指导模型区分类别本身。

类别编码	独热编码	分割分数	Car-M	Ped-M	Cyc-M
√			78.45	57.36	64.48
	√		77.92	55.59	65.12
		√	79.35	55.10	67.20

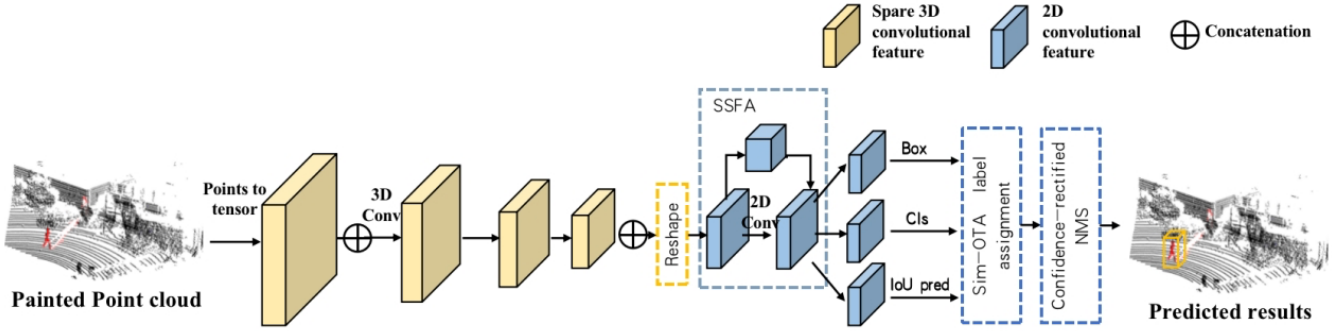
6 进一步研究

我们原本准备在这周移植2020 CVPR《SA-SSD：Structure Aware Single-Stage 3D Object Detection From Point Cloud》的辅助监督网络部分，实现我们原来的选题SPV-SSD，但是无奈期末周时间太紧，这一部分留给我们之后再去研究。

论文中的网络结构图：



我们可以看得出来，它的网络结构和我们很像，同样的4层3D稀疏卷积，但是主要在 backbone 3D部分增加了一个辅助网络。



7 小组分工与贡献度

组员	工作内容	贡献度
1951504 王凌	负责OpenPCDet环境的搭建，查阅以及应用相关论文，搭建基本的 Backbone 网络，主要负责Head部分的工作	$\frac{1}{3}$
1953714 杨茗溪	阅读相关论文，SSFA部分的的移植，OTA-3D 的修改与应用，结果可视化	$\frac{1}{3}$
1953902 高杨帆	阅读相关论文，Head部分的工作，Anchor-free的修改与应用，实验结果的分析	$\frac{1}{3}$