# Part-$A^2$ Net: 3D Part-Aware and Aggregation Neural Network for Object Detection from Point Cloud

Shaoshuai Shi[1] Zhe Wang[2] Xiaogang Wang[1] Hongsheng Li[1]
[1]The Chinese University of Hong Kong [2]SenseTime Research
{ssshi, xgwang, hsli}@ee.cuhk.edu.hk wangzhe@sensetime.com

## Abstract

*In this paper, we propose the part-aware and aggregation neural network (Part-$A^2$ net) for 3D object detection from point cloud. The whole framework consists of the part-aware stage and the part-aggregation stage. Firstly, the part-aware stage learns to simultaneously predict coarse 3D proposals and accurate intra-object part locations with the free-of-charge supervisions derived from 3D ground-truth boxes. The predicted intra-object part locations within the same proposals are grouped by our new-designed RoI-aware point cloud pooling module, which results in an effective representation to encode the features of 3D proposals. Then the part-aggregation stage learns to re-score the box and refine the box location based on the pooled part locations. We present extensive experiments on the KITTI 3D object detection dataset, which demonstrate that both the predicted intra-object part locations and the proposed RoI-aware point cloud pooling scheme benefit 3D object detection and our Part-$A^2$ net outperforms state-of-the-art methods by utilizing only point cloud data.*

## 1. Introduction

With the surging demand from autonomous driving and robotics, increasing attention has been paid to 3D object detection from point clouds obtained by the LiDAR sensors [2, 11, 15, 20, 32, 3, 12, 27, 33, 34, 19, 28]. Though significant achievements have been made in 2D object detection from images [5, 25, 18, 23, 24, 16, 17, 8, 1], directly extending these 2D object detection methods to 3D object detection from point clouds might lead to inferior performance. The key challenge is that the point cloud data captured by LiDAR sensors is sparse and irregular. How to extract discriminative features from the irregular points is still an open and challenging problem.

Existing 3D object detection methods have explored several ways to tackle these challenges. Several works [2, 11, 33, 34, 15] project the point cloud to obtain bird-view feature maps and apply 2D Convolutional Neural Networks (CNN) to these feature map for 3D object detection. Some others works [35, 29, 32] divide the 3D space into regular 3D voxels and apply 3D CNN or 3D sparse convolution [7, 6] to extract the features for 3D detection. There are also some works [20, 31] that utilize 2D detectors to detect 2D boxes from the image, and then PointNet [21, 22] is applied to the cropped point cloud to regress the parameters of 3D boxes. These works either suffer from the information loss during projection and quantization or depend heavily on the performance of 2D object detectors.

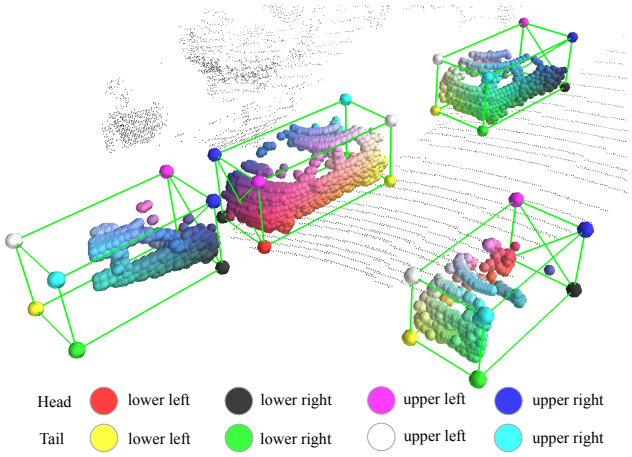


Figure 1. Intra-object part locations and segmentation masks can be robustly predicted by the proposed part-aware and aggregation network even when objects are partially occluded. Such part locations can assist accurate 3D object detection. Best view in colors.

Instead of detecting from the bird-view maps or 2D images, Shi *et al.* [27] proposed to directly generate 3D proposal from the point cloud by segmenting the foreground points, where the segmentation label is directly generated from the 3D box annotations. However, we observed that the 3D box annotations not only provide the segmentation masks, but also imply accurate intra-object part locations for all the points within the 3D boxes. This is totally different from the box annotations in 2D images, since some



[27]: PointRCNN, same main author.

parts of objects in the 2D images may be occluded. Using the ground-truth 2D boxes would generate inaccurate and noisy intra-object locations for each pixel within objects. Such 3D intra-object part locations are accurate, informative, and can be obtained for free, but were never explored in 3D object detection.

Motivated by this important observation, we propose the part-aware and aggregation neural network, *i.e.*, Part-$A^2$ net, a novel two-stage framework for 3D object detection from point cloud, which consists of the part-aware stage and the part-aggregation stage.

Specifically, in the first part-aware stage, as shown in Fig. 1, the network learns to estimate the intra-object part locations for all the foreground points, where the ground-truth part location annotations and segmentation masks are directly generated from the ground-truth 3D box annotations. The point features are learned by dividing the whole 3D space into small voxels and adopting a sparse convolution [7, 6] based backbone network. We also append a region proposal head to the backbone to generate coarse 3D proposals to group the parts for the following part-aggregation stage.

The motivation for the part-aggregation stage is that, given a group of points within a 3D proposal, the network should have the ability to evaluate the quality of this proposal and refine it by learning the spatial relationship of the predicted intra-object part locations of all these points. Hence, to group the points within the same 3D proposal, we propose a novel RoI-aware point cloud pooling module to eliminate the ambiguity when conducting region pooling on the point cloud. Unlike conventional pooling that pools over all the points or non-empty voxels, the RoI-aware pooling pools over all the voxels in the 3D RoI including both non-empty voxels and empty voxels. This is vital to obtain an effective representation for box scoring and location refinement, as the empty voxels also encode the box's geometry information. After the RoI-aware point cloud region pooling, the network aggregates the information of part locations with sparse convolution and further pooling. The experiments show that the aggregated part features could improve the quality of the proposals remarkably and achieve state-of-the-art performance on 3D detection benchmark.

Our main contributions could be summarized as follows:

- We propose a novel part-aware and aggregation neural network for 3D object detection from point cloud. With only the 3D box annotations as supervisions, our proposed method could predict the intra-object 3D part location accurately, which are then aggregated by our part-aggregation network to learn the spatial relationship between these parts for predicting accurate 3D object locations and confidences.

- We present the differentiable RoI-aware point cloud pooling module to eliminate the ambiguity in point cloud region pooling by encoding the position-specific features of 3D proposals. The experiments show that the pooled feature representation benefits the part-aggregation stage significantly.

- Our proposed part-aware and aggregation method outperforms all published methods with remarkable margins on the challenging 3D detection benchmark of KITTI [4] dataset as of July 9, 2019, which demonstrates the effectiveness of our method.

## 2. Related Work

**3D object detection from multiple sensors.** Several existing methods have worked on fusing the information from multiple sensors (*e.g.*, LiDAR and camera) to help 3D object detection. [2, 11] projected the point cloud to the bird view and extracted features from bird-view maps and images separately, which are then cropped and fused by projecting 3D proposals to the corresponding 2D feature maps for 3D object detection. [15] further explored the feature fusion strategy by proposing continuous fusion layer to fuse image feature to bird view features. Different from projecting point cloud to bird view map, [20, 31] utilized off-the-shelf 2D object detectors to detect 2D boxes first for cropping the point cloud and then applied PointNet [21, 22] to extract features from the cropped point clouds for 3D box estimation. Unlike these sensor fusion methods, our proposed part-aware and aggregation framework could achieve comparable or even better 3D detection results by using only point cloud as input.

**3D object detection from point clouds only.** Zhou *et al*. [35] for the first time proposed VoxelNet architecture to learn discriminative features from point cloud and detect 3D object with only point cloud. [32] improved VoxelNet by introducing sparse convolution [7, 6] for efficient voxel feature extraction. [34, 33, 12] projected the point cloud to bird-view maps and applied 2D CNN on these maps for 3D detection. Shi *et al*. [27] proposed the PointRCNN to directly generate 3D proposals from raw point cloud in a bottom-up manner by observing that the segmentation label could be obtained from 3D box annotations. Our proposed part-aware and aggregation framework further explores the abundant information provided by the 3D box annotations and learns to predict accurate intra-object part locations to improve the accuracy of 3D object detection.

**Point cloud feature learning for 3D object detection.** There are generally three ways of learning features from point cloud for 3D detection. [2, 11, 34, 33, 15] projected point cloud to bird-view map and utilized 2D CNN for feature extraction. [20, 31, 27] conducted PointNet [21, 22] to learn the point cloud features directly from raw point cloud.

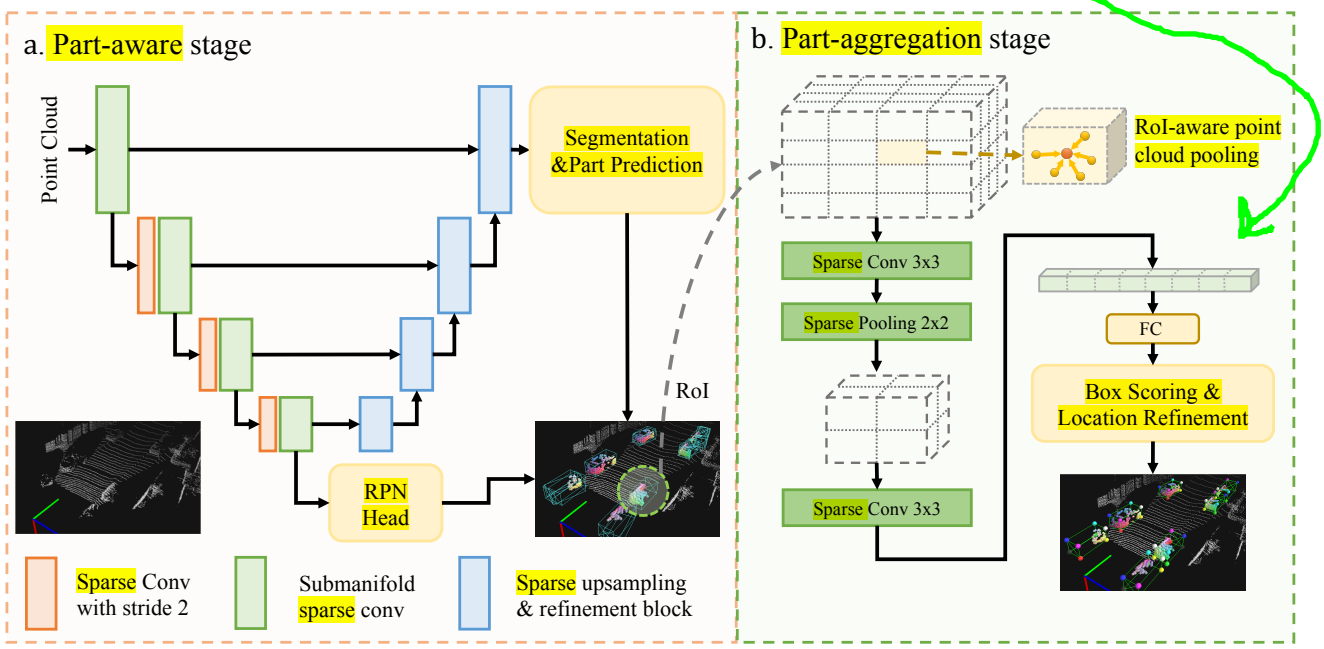


Figure 2. The overall framework of our part-aware and aggregation neural network for 3D object detection. It consists of two stages: (a) The first part-aware stage estimates intra-object part locations accurately and generates 3D proposals by feeding the raw point cloud to our newly designed backbone network. (b) The second part-aggregation stage conducts the proposed RoI-aware point cloud pooling operation to group the part information from each 3D proposal, then the part-aggregation network is utilized to score boxes and refine locations based on the part features and information.

[35] proposed VoxelNet and [32] applied sparse convolution to speed up the VoxelNet for feature learning. Inspired by VoxelNet, we designed a UNet-like [26] backbone network by using sparse convolution and deconvolution to extract discriminative point features for predicting intra-object part locations and 3D object detection.

## 3. Method

In this section, we present the proposed part-aware and aggregation neural network for 3D object detection from point cloud. The overall architecture of our method is shown in Fig. 2. Our key observation is that, the ground-truth boxes of 3D object detection automatically provide accurate intra-object part locations and segmentation mask for each 3D point since objects in the 3D space are naturally separated. This is very different from 2D object detection, where 2D object boxes might only contain a part of an object due to occlusion and thus cannot provide accurate intra-object location for each 2D pixel.

Motivated by this important observation, we propose a novel part-aware and aggregation 3D object detector, Part-$A^2$ net, for 3D object detection from point cloud. Specifically, we introduce the free-of-charge 3D part location labels and segmentation labels as extra supervisions to benefit the 3D proposal generation. The predicted 3D intra-part locations within each proposal are then aggregated in the second stage to score the box and refine the location. The experiments show that our designed part-aware network could estimate the 3D intra-object part locations accurately, which are effectively used by the part-aggregation network for the final confidence prediction and location refinement.

In Sec. 3.1, we first introduce the 3D intra-object part location estimation, which can benefit the 3D proposal generation. Then we present the RoI-aware point cloud pooling module in Sec. 3.2, to eliminate the ambiguity in point cloud pooling and aggregate 3D part information for 3D proposal refinement. In Sec. 3.3, we demonstrate the aggregation of predicted part information for box scoring and location refinement. Finally we describe the loss functions for training our proposed framework in Sec. 3.4.

### 3.1. Learning to estimate intra-object part locations

**Efficient point-wise feature learning via sparse convolution.** To segment the foreground points and predict their 3D intra-object part locations, we need learn the discriminative point-wise features from the point cloud. Inspired by [35, 32], we voxelize the 3D space as regular voxels and extract the features of non-empty voxels by sparse convolution [7, 6]. The center of each voxel is considered as a new point to form a new point cloud, which is approximately equivalent to the raw point cloud since the voxel size is much smaller (5cm×5cm×10cm) compared to the whole

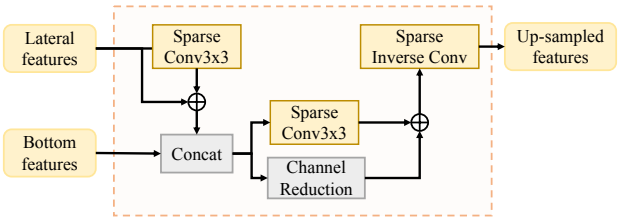


Figure 3. Sparse up-sampling and feature refinement block. This module is adopted in the decoder of our sparse convolution based UNet backbone. The lateral features and bottom features are first fused and transformed by sparse convolution. The fused feature is then up-sampled by the sparse inverse convolution.

3D space (∼70m×80m×4m). For each 3D scene in the KITTI dataset, there are generally about 16,000 non-empty voxels in the 3D space.

As shown in Fig. 2, we designed a UNet-like architecture [26] for learning point-wise feature representations with sparse convolution and sparse deconvolution on the obtained sparse voxels. The spatial resolution is downsampled 8 times by three sparse convolutions of stride 2, each of which is followed by several submanifold sparse convolutions. As illustrated in Fig. 3, we also designed a similar up-sampling block as that in [30] based on sparse operations for refining the fused features and saving computations.

**Semantic segmentation and intra-object part location prediction.** Intra-object part information is essential for the neural network to recognize and detect the objects. For instance, the side of the vehicle is also a plane perpendicular to the ground and the two round wheels are always close to the ground plane. By learning to estimate the foreground segmentation mask and the intra-object part location for each point, the neural network develops the ability of inferring the shape and pose of the objects, which is beneficial for 3D object detection.

Given the above sparse convolution based UNet backbone, two branches are appended for segmenting the foreground points and predicting their intra-object part locations. We adopt the focal loss [17] for point segmentation, where all points inside or outside ground-truth boxes are utilized as positive and negative samples for training.

The 3D ground-truth boxes automatically provide the 3D intra-object part location labels. We denote the part label of a foreground point $(p_x, p_y, p_z)$ as three continuous values $(O_x, O_y, O_z)$ to indicate its relative position in the corresponding object. Since a 3D box is represented by $(C_x, C_y, C_z, h, w, l, \theta)$ with $(C_x, C_y, Cz)$ as the center, $(h, w, l)$ as the size and $\theta$ as the orientation in bird-view, the ground-truth 3D part location labels could be formulated as follows:

Should be O_z = (p_z - C_z)/h + 0.5 I guess?

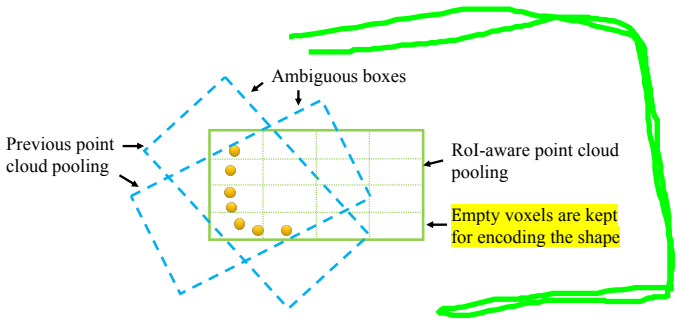


Figure 4. Illustration of RoI-aware point cloud feature pooling. Due to the ambiguity showed in the above BEV figure, We could not recover the original box shape by using previous point cloud pooling method. Our proposed RoI-aware point cloud pooling method could encode the box shape by keeping the empty voxels, which could be efficiently processed by following sparse convolution.

$$\begin{bmatrix} t_x & t_y \end{bmatrix} = \begin{bmatrix} p_x - C_x & p_y - C_y \end{bmatrix} \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix},$$
$$O_x = \frac{t_x}{w} + 0.5,\ O_y = \frac{t_y}{l} + 0.5,\ O_z = \frac{p_z - O_z}{h} + 0.5 \quad (1)$$

where we have $O_x, O_y, O_z \in [0, 1]$, and then the part location of the object center is $(0.5, 0.5, 0.5)$. Note that all coordinates are represented in the LiDAR coordinate system of KITTI, where the direction of $z$ is perpendicular to the ground, and $x$ and $y$ are parallel to the horizontal plane.

We apply the binary cross entropy loss as the part regression loss to learn the intra-object part locations along the 3 dimensions, which could be formulated as follows:

$$\mathcal{L}_{\text{part}}(P_u) = -\left(O_u \log(P_u) + (1 - O_u)\log(1 - P_u)\right) \quad \text{for } u \in \{x, y, z\}, \quad (2)$$

where $P_u$ is the predicted intra-object part locations after the sigmoid layer. Note that the part location prediction is only conducted for foreground points.

**3D proposal generation.** To aggregate the predicted intra-object part locations for 3D object detection, we need to generate 3D proposals to aggregate the part information from the estimated foreground points that belong to the same object. Hence, as shown in Fig. 2, we append the same RPN head with [32] to the feature map generated by the sparse convolution encoder. The feature map is 8 times downsampled and the features at different heights of the same bird-view position is aggregated to generate a 2D bird-view feature map for the 3D proposal generation.

### 3.2. RoI-aware point cloud feature pooling

Given the predicted intra-object part locations and the 3D proposals, we aim to conduct box scoring and proposal refinement by aggregating the part information of all the points within the same proposal. Shi *et al.* [27] proposed the point cloud region pooling operation to pool the point-wise features from the 3D proposals for refining the pro-

posal in a second stage. However, we argue that this operation loses the 3D proposal information since the points do not distribute regularly within the proposals and there exists ambiguity to recover the 3D boxes from the pooled points. As shown in Fig. 4, different proposals will result in the same pooled points, which introduces negative effects to the refinement network.

Therefore, we propose the RoI-aware point cloud pooling module to evenly divide the 3D proposal into regular voxels with a fixed spatial shape $(H \times W \times L)$, where $H, W, L$ are the height, width and length hyperparameters (*e.g.*, $14 \times 14 \times 14$ is adopted in our framework) of the pooling resolution in each dimension and independent of the 3D proposal size. The features of each voxel is calculated by aggregating (*e.g.*, max-pooling or average-pooling) the point features within this voxel, where the features of empty voxels will be set to zero and marked as empty. The proposed RoI-aware pooling module is differentiable, which enables the whole framework is end-to-end trainable.

The proposed RoI-aware point cloud pooling module normalizes different 3D proposals into the same local spatial coordinates, where each voxel encodes the features of a corresponding fixed grid in the 3D proposal. This position-specific feature pooling is more meaningful for the proposal encoding and results in an effective representation for the follow-up box scoring and location refinement in Sec. 3.3.

### 3.3. Part location aggregation for 3D box refinement

By considering the spatial distribution of the predicted intra-object part locations of all 3D points within a proposal, it is reasonable to evaluate the quality of this proposal by aggregating the predicted part locations. Actually we could formulate it as an optimization problem, and directly solve the parameters of 3D bounding box by fitting the predicted part locations of all points within the corresponding proposal. However, we found this optimization-based method is sensitive to outliers and the quality of predicted part locations.

To solve this problem, we propose a learning based method to robustly aggregate the part location information for box scoring and location refinement. For each 3D proposal, we apply the proposed RoI-aware point cloud pooling operation to the predicted point-wise part locations (average-pooling) and point-wise features (max-pooling) from the first stage separately, which results in two feature maps of sizes $(14 \times 14 \times 14 \times 4)$ and $(14 \times 14 \times 14 \times C)$, where the predicted part location map is 4-dimensional with 3 for the part locations $(x, y, z$ dimensions) and 1 for the foreground segmentation scores, and $C$ is the feature dimension for the point-wise features transformed by the first stage.

After the pooling operation, as shown in Fig. 2, the part-aggregation network is implemented in a hierarchical way to learn from the spatial distribution of the predicted intra-object part locations. Specifically, we first use sparse convolutional layer with kernel size $3 \times 3 \times 3$ to convert two pooled feature map to the same feature dimensions. After concatenating these two feature maps, we stack four sparse convolutional layers with kernel size $3 \times 3 \times 3$ to aggregate the part information gradually as the receptive filed increases. Here we also utilize a sparse max-pooling with kernel size $2 \times 2 \times 2$ and stride $2 \times 2 \times 2$ after the second convolutional layer to down-sample the feature map to $7 \times 7 \times 7$ for saving the computations and parameters. Then we vectorize it to a feature vector and append two branches for the final box scoring and location refinement.

Compared with the naive method to directly vectorize the pooled 3D feature map to a feature vector, our proposed part aggregation strategy could learn the spatial distribution of the predicted part locations effectively by aggregating features from local to global scales. It also saves many computations and parameters by using sparse convolution, since the pooled voxels are very sparse and the naive method could not ignore the empty voxels since each voxel encodes the features of a specific position in the 3D proposals.

### 3.4. Loss function

As mentioned in Sec. 3.3, we append two branches to the vectorized feature vector aggregated from predicted part information. For the box scoring branch, inspired by [9, 13], we use the 3D IoU between 3D proposal and its corresponding ground truth box as the soft label for the proposal quality evaluation, which is also learned by the binary cross entropy loss as Eq. (2).

For the 3D proposal generation and refinement, we adopt the similar regression targets as those in [35, 32] and use smooth-L1 loss to regress the normalized box parameters as

$$\begin{aligned} &\Delta x = \frac{x^g - x^a}{d^a}, \Delta y = \frac{y^g - y^a}{h^a}, \Delta z = \frac{z^g - z^a}{d^a}, \\ &\Delta l = \log(\frac{l^g}{l^a}), \Delta h = \log(\frac{h^g}{h^a}), \Delta w = \log(\frac{w^g}{w^a}), \\ &\Delta \theta = \theta^g - \theta^a, d^a = \sqrt{(l^a)^2 + (w^a)^2}, \end{aligned} \tag{3}$$

where $d^a = \sqrt{(l^a)^2 + (w^a)^2}$ normalizes the center offset in the bird view, $(x^a, y^a, z^a, h^a, w^a, l^a, \theta^a)$ are the parameters of 3D anchors/proposals and $(x^g, y^g, z^g, h^g, w^g, l^g, \theta^g)$ denote its corresponding ground truth box. Note that different from [27], here for proposal refinement, we directly regress the relative offsets or size ratios based on the parameters of 3D proposals, since our proposed RoI-aware point cloud pooling module already encodes the full geometric information of 3D proposals and transfers different 3D proposals to the same normalized spatial coordinate system.

Don't quite get this.

Hence, there are three losses for the part-aware stage with the equaling weights, including focal loss for foreground points segmentation, binary cross entropy loss for the regression of intra-object part locations and smooth-l1 loss for 3D proposal generation. For the part-aggregation stage, there are two losses also with the equal loss weight one, including binary cross entropy loss for IoU regression and smooth-l1 loss for location refinement.

# 4. Experiments

In this section, we evaluate our proposed part-aware and aggregation network on the challenging 3D detection benchmark of KITTI [4] dataset. In Sec. 4.1, we briefly introduce the implementation details of our method. In Sec. 4.2, we evaluate our method by comparing with state-of-the-art 3D detection methods on the KITTI benchmark. In Sec. 4.3, we present ablation studies and analysis to validate effectiveness of individual components in our method. Finally we visualize some qualitative results in Sec. 4.4.

## 4.1. Implementation details

**Network details.** As shown in Fig. 2, for the part-aware stage, the spatial feature maps have four scales with feature dimensions 16-32-64-64, and we use three sparse convolutions with kernel size 3 and stride 2 to downsample the spatial resolution by 8 times. We stack two submanifold convolutions in each level with kernel size 3 and stride 1. There are four sparse up-sampling blocks in the decoder with feature dimensions 64-64-32-16 from bottom to up, respectively. Note that the stride of the last up-sampling block is 1 and the stride of other three up-sampling blocks is 2.

For the part-aggregation stage, the pooling resolution of RoI-aware point cloud pooling module is $14 \times 14 \times 14$, which is downsampled to $7 \times 7 \times 7$ after processed by the sparse convolutions and max-pooling with feature dimensions 128. We vectorize the downsampled feature maps to a single feature vector for the final box scoring and location refinement.

**Training and inference details.** We train the entire network end-to-end with the ADAM optimizer and batch size 6 for 50 epochs. The cosine annealing learning rate strategy is used with an initial learning rate 0.001. We randomly select 128 proposals from each scene for the training of second stage, where 50% of the proposals that have 3D IoU with its corresponding ground truth box of at least 0.55.

We conduct common data augmentation during training, including random flipping, global scaling with scaling factor sampled from $[0.95, 1.05]$, global rotation around the vertical axis by an angle sampled from $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The whole training process of our proposed part-aware and part-aggregation networks takes about 17 hours on a single NVIDIA Tesla V100 card.

For inference, only 100 proposals are kept from part-aware stage with NMS threshold 0.7, which are then scored and refined by the following part-aggregation stage. We finally apply the rotated NMS with threshold 0.01 to remove redundant boxes and generate the final 3D detection results.

## 4.2. Experimental results on KITTI benchmark

There are 7481 training samples and 7518 test samples in the KITTI 3D detection benchmark. The training samples are divided into the *train* split (3712 samples) and *val* split (3769 samples) as the frequently used partition. We conduct all the experiments on the most important and challenging car category unless specified otherwise since it is the major focus of KITTI object detection benchmark. All models are trained on the *train* split, and evaluated on the *val* and *test* splits.

**Comparison with state-of-the-arts on 3D detection.** As shown in Table 1, we evaluate our method on the 3D detection benchmark and the bird's eye view detection benchmark of the KITTI test server. For the 3D object detection benchmark, by only using LiDAR point clouds, our proposed Part-$A^2$ net outperforms all previous peer-reviewed LiDAR only methods on all difficulty levels and outperforms all previous multi-sensor methods on the most important "moderate" difficulty level for both car and cyclist classes. For the bird's view detection of car and cyclist, our method outperforms previous state-of-the-art methods by large margins on almost all the difficulty levels. As of July 9th, 2019, our method ranks 1st among all methods on the most important car class of 3D object detection leader board of KITTI dataset [10].

We also report the performance of our method on KITTI *val* split including both 3D object detection (shown in Table 2) and 3D object localization (shown in Table 3). We could see that on the most important "moderate" difficulty level, our method outperforms previous state-of-the-art methods on both two tasks with large margins by using only the point cloud as inputs. Our methods achieves new state-of-the-art performance on all difficulty levels of the KITTI 3D object detection *val* split, which demonstrates the effectiveness of our proposed Part-$A^2$ net for 3D object detection. Note that due to the mismatched distribution of *test* data and *val* data of KITTI dataset, almost all the methods perform better on the *val* split and the performance would drop when tested on the *test* split of KITTI dataset.

**Evaluation of predicted intra-object part locations.** The intra-object part locations predicted by our part-aware stage are crucial for the part-aggregation stage to score the box and refine the box location. Here we evaluate the predicted intra-object part locations by the following metric:

$$\text{AbsError}_u = \frac{1}{||\mathcal{G}||} \sum_{i \in \mathcal{G}} |P_u^i - Q_u^i|, u \in \{x, y, z\}, \quad (4)$$

Why no results for pedestrians?

| Method | Modality | 3D Detection (Car) | | | BEV Detection (Car) | | | 3D Detection (Cyclist) | | | BEV Detection (Cyclist) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Mod. | Easy | Hard | Mod. | Easy | Hard | Mod. | Easy | Hard | Mod. | Easy | Hard |
| MV3D [2] | RGB + LiDAR | 62.35 | 71.09 | 55.12 | 76.90 | 86.02 | 68.49 | - | - | - | - | - | - |
| ContFuse [15] | RGB + LiDAR | 66.22 | 82.54 | 64.04 | 85.83 | 88.81 | 77.33 | - | - | - | - | - | - |
| AVOD-FPN [11] | RGB + LiDAR | 71.88 | 81.94 | 66.38 | 83.79 | 88.53 | 77.90 | 52.18 | 64.00 | 46.61 | 57.48 | 68.09 | 50.77 |
| F-PointNet [20] | RGB + LiDAR | 70.39 | 81.20 | 62.19 | 84.00 | 88.70 | 75.33 | 56.77 | 71.96 | 50.39 | 61.96 | 75.38 | 54.68 |
| UberATG-MMF [14] | RGB + LiDAR | 76.75 | 86.81 | 68.41 | 87.47 | 89.49 | 79.10 | - | - | - | - | - | - |
| VoxelNet [35] | LiDAR only | 65.11 | 77.47 | 57.73 | 79.26 | 89.35 | 77.39 | 48.36 | 61.22 | 44.37 | 54.76 | 66.70 | 50.55 |
| SECOND [32] | LiDAR only | 73.66 | 83.13 | 66.20 | 79.37 | 88.07 | 77.95 | 53.85 | 70.51 | 40.90 | 56.04 | 73.67 | 48.78 |
| PointPillars [12] | LiDAR only | 74.99 | 79.05 | 68.30 | **86.10** | 88.35 | 79.83 | 59.07 | 75.78 | 52.92 | 62.25 | 79.14 | 56.00 |
| PointRCNN [27] | LiDAR only | 75.76 | **85.94** | 68.32 | 85.68 | 89.47 | 79.10 | 59.60 | 73.93 | 53.59 | 66.77 | 81.52 | 60.78 |
| Part-$A^2$(Ours) | LiDAR only | **77.86** | **85.94** | **72.00** | 84.76 | **89.52** | **81.47** | **62.73** | **78.58** | **57.74** | **68.12** | **81.91** | **61.92** |

Table 1. Performance evaluation on KITTI 3D object detection test server (*test* split). The 3D object detection and bird's eye view detection are evaluated by average precision with rotated IoU threshold 0.7 for car and 0.5 for cyclist.

| Method | Modality | AP (IoU=0.7) | | |
|---|---|---|---|---|
| | | Moderate | Easy | Hard |
| MV3D [2] | RGB & LiDAR | 62.68 | 71.29 | 56.56 |
| ContFuse[15] | RGB & LiDAR | 73.25 | 86.32 | 67.81 |
| AVOD-FPN [11] | RGB & LiDAR | 74.44 | 84.41 | 68.65 |
| F-PointNet [20] | RGB & LiDAR | 70.92 | 83.76 | 63.65 |
| VoxelNet [35] | LiDAR | 65.46 | 81.98 | 62.85 |
| SECOND [32] | LiDAR | 76.48 | 87.43 | 69.10 |
| PointRCNN [27] | LiDAR | 78.63 | 88.88 | 77.38 |
| Part-$A^2$(Ours) | LiDAR | **79.47** | **89.47** | **78.54** |

Table 2. Performance comparison of 3D object detection on the car class of the KITTI *val* split set.

| Method | Modality | AP (IoU=0.7) | | |
|---|---|---|---|---|
| | | Moderate | Easy | Hard |
| MV3D [2] | RGB & LiDAR | 78.10 | 86.55 | 76.67 |
| F-PointNet [20] | RGB & LiDAR | 84.02 | 88.16 | 76.44 |
| ContFusion [15] | RGB& LiDAR | 87.34 | 95.44 | 82.43 |
| VoxelNet [35] | LiDAR | 84.81 | 89.60 | 78.57 |
| SECOND [32] | LiDAR | 87.07 | 89.96 | 79.66 |
| Part-$A^2$(Ours) | LiDAR | **88.61** | **90.42** | **87.31** |

Table 3. Performance comparison of bird-view object detection on the car class of the KITTI *val* split set.

| mAbsError$_x$ | mAbsError$_y$ | mAbsError$_z$ | mAbsError |
|---|---|---|---|
| 7.24% | 6.42% | 5.17% | 6.28% |

Table 4. Mean distance error of predicted intra-object part locations by part-aware stage on the KITTI *val* split. Here $x, y, z$ are along the direction of width, length and height of the object, respectively.

where $P_u^i$ is the predicted part location, $Q_u^i$ is the ground truth part location, and $\mathcal{G}$ is the set of foreground points for each sample. The final mAbsError$_u$ is the mean value of AbsError$_u$ for all the samples.

As shown in Table 4, the mean error of our predicted intra-object part locations is 6.28%, which shows that the part-aware network predicts the part locations accurately since the average error is only $\pm$6.28cm per meter. Based on this accurate intra-object part locations, our second part-aggregation stage could better score the boxes and refine the box locations by utilizing the predicted geometric information. Here we also report the detailed error statistics of predicted intra-object part locations on different difficulty levels of the KITTI *val* split in Fig. 5 for reference.

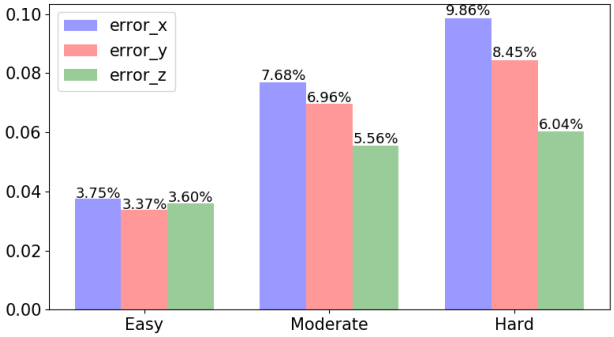


Figure 5. Statistics of predicted intra-object part location errors in the *val* split of KITTI dataset.

### 4.3. Ablation studies

**Does the part-aware and aggregation benefit for 3D object detection?** To validate the effectiveness of utilizing the part location information for 3D detection, we provide an ablation experiment which removes the part location supervisions as baseline. To design the baseline, we only removed the module for intra-object part location prediction from the part-aware stage and kept other modules unchanged. The predicted point-wise part locations $(14 \times 14 \times 14 \times 4)$ for the part-aggregation stage are replaced with the coordinates and segmentation scores of the points.

As shown in Table 5, compared with the model trained without intra-object part location supervisions ("Part-$A^2$" vs. "Part-$A^2$ w/o parts", our proposed part-aware and aggregation network achieves better recall and average precision on all difficulty levels of the *val* split. The remarkable improvements on recall and precision indicate that the network learned better discriminative features for scoring box and refining locations for 3D detection with detailed and accurate dense supervisions of the intra-object part locations.

We also conduct investigation by measuring the perfor-

The diffeence is not THAT big though.

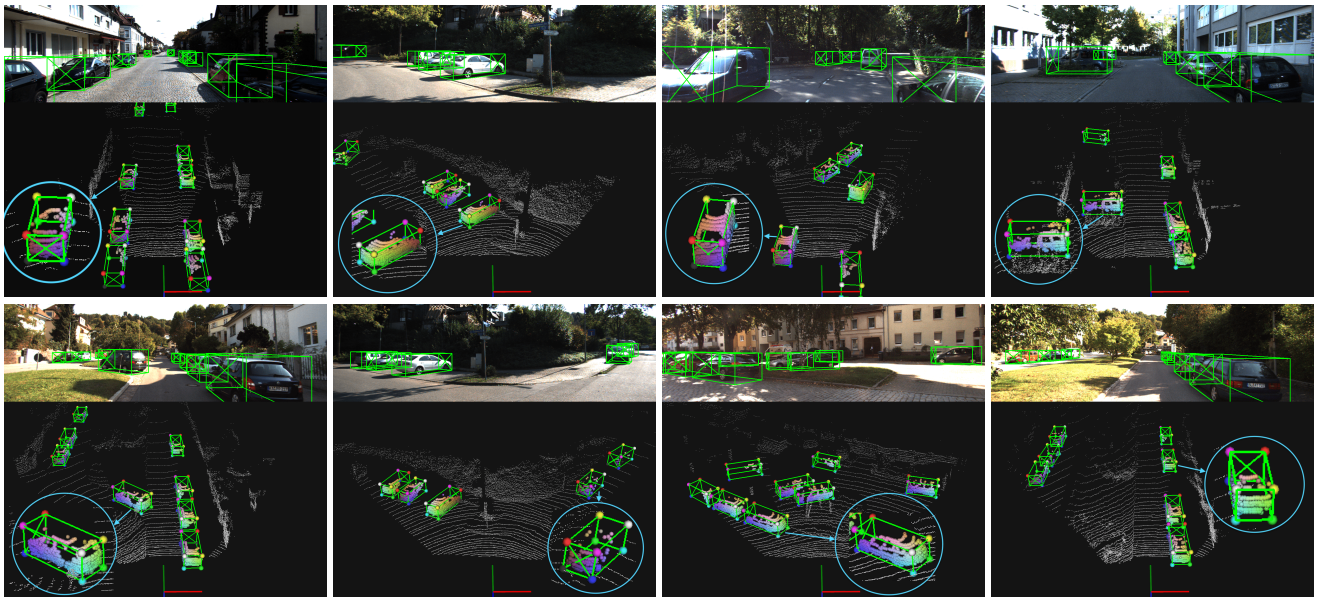

Figure 6. Qualitative results of Part-$A^2$ Net on the KITTI *test* split. The predicted 3D boxes are drawn with green 3D bounding boxes, and the estimated intra-object part locations are visualized with different colors according to the legend shown in Fig. 1. Best viewed in colors.

| Method | Recall (IoU=0.7) box#100 | AP (IoU=0.7) Easy | Moderate | Hard |
|---|---|---|---|---|
| Part-aware w/o parts | 80.90 | 88.48 | 77.97 | 75.84 |
| Part-aware | 80.99 | 88.90 | 78.54 | 76.44 |
| Part-$A^2$ w/o parts | 82.92 | 89.23 | 79.00 | 77.66 |
| Part-$A^2$ | **84.33** | **89.47** | **79.47** | **78.54** |

Table 5. Effects of removing the intra-object part location supervisions from our Part-$A^2$ net.

| Method | $AP_{Easy}$ | $AP_{Mod.}$ | $AP_{Hard}$ |
|---|---|---|---|
| RoI fixed-sized pool ($14 \times 14 \times 14$) & sparse conv | 88.78 | 78.61 | 78.05 |
| RoI-aware pool ($14 \times 14 \times 14$) & FCs | 89.45 | 79.32 | 78.35 |
| RoI-aware pool ($14 \times 14 \times 14$) & sparse conv | **89.47** | **79.47** | **78.54** |

Table 6. Effects of RoI-aware point cloud region pooling.

| Method | $AP_{Easy}$ | $AP_{Mod.}$ | $AP_{Hard}$ |
|---|---|---|---|
| RoI-aware pool $7 \times 7 \times 7$ & FCs | 89.17 | 79.11 | 78.03 |
| RoI-aware pool $7 \times 7 \times 7$ & sparse conv | 89.24 | 79.21 | 78.11 |
| Ours | **89.47** | **79.47** | **78.54** |

Table 7. Comparison of different part-aggregation network structures.

mance of directly utilizing the 3D proposals from our first part-aware stage as detection results. The detection results are obtained by directly applying NMS to the box proposals from our part-aware stage. We would like to investigate whether the intra-object part supervisions are also beneficial to the proposal generation. From the comparison ("Part-aware" vs. "Part-aware w/o parts"), we could see that, without the proposed intra-object part supervisions, the proposal recalls of our first proposal stage are comparable (80.90 vs. 80.99). However, the gap improves significantly (82.92 vs 84.33) after the 100 proposals are refined by the part-aggregation stage, which demonstrates that our part-aggregation stage could effectively aggregate the intra-object locations to achieve better 3D detection.

**Effects of RoI-aware point cloud region pooling.** The proposed RoI-aware point cloud pooling normalizes different 3D proposals to the same coordinate system to encode geometric information of proposals. It solves the ambiguous encoding by existing 3D point cloud pooling schemes as shown in Fig. 4. The 3D proposals are divided into regular voxels to better encode the 3D geometry. To validate the effects of the RoI-aware pooling module, we add the following comparison experiments. (a) We replace RoI-aware pooling by fixed-sized RoI pooling, *i.e.* pooling all 3D proposals with the same fixed-size 3D grid ($1.6m \times 3.9m \times 1.56m$) calculated from the mean object size of the training set. The center and orientation of the 3D grid are set as its corresponding 3D proposal's center and orientation, respectively. This is very similar to the pooling scheme used in PointRCNN [27], where not all geometric information is well preserved after pooling. (b) We replace sparse conv with several FC layers. As shown in Table 6, removing RoI-aware pooling greatly decreases detection accuracy, while replacing sparse conv with FC layers has similar performance, which proves the effectiveness of the proposed RoI-aware pooling for 3D object detection.

**Variants of part-aggregation network.** After applying the RoI-aware point cloud pooling module, there are several ways to implement the part-aggregation stage. The simplest strategy is to directly vectorize the pooled feature volumes to a feature vector followed by several fully-connected layers for box scoring and refinement. From Table 7, we could see that this naive way already achieved promising results, which are benefited from the effective representations of our

| RoI pooling size | $AP_{Easy}$ | $AP_{Mod.}$ | $AP_{Hard}$ |
|---|---|---|---|
| $6 \times 6 \times 6$ | 89.02 | 78.85 | 78.04 |
| $8 \times 8 \times 8$ | 89.09 | 78.97 | 78.15 |
| $10 \times 10 \times 10$ | 89.44 | 79.15 | 78.42 |
| $12 \times 12 \times 12$ | **89.61** | 79.35 | 78.50 |
| $14 \times 14 \times 14$ | 89.47 | **79.47** | 78.54 |
| $16 \times 16 \times 16$ | 89.52 | 79.45 | **78.56** |

Table 8. Effects of using different RoI-aware pooling sizes in our part-aggregation stage.

RoI-aware point cloud pooling scheme since each position of the feature vector encodes a specific intra-object position of the object of interest to help learn the shape of the box better. In the second row of Table 7, we further investigate using sparse convolution with kernel size 3 to aggregate the features from local to global scales gradually, which achieves better results. As we mentioned in Sec. 3.3, our final part-aggregation network adopts a higher pooling resolution $14 \times 14 \times 14$ to pool more details and then use sparse max-pooling to downsample the feature map for saving memory and computations, which achieves the highest performance as shown in Table 7.

**Effects of RoI pooling size.** The $14 \times 14$ pooling size was very commonly chosen for 2D object detection, and we follow the same setting to use $14 \times 14 \times 14$ as the RoI-aware pooling size. We also test different RoI pooling sizes as shown in Table 8. This pooling size shows robust performance for different 3D objects. Similar performances can be observed if the pooling sizes are greater than $12 \times 12 \times 12$.

### 4.4. Qualitative results

We present some representative results generated by our proposed Part-$A^2$ net on the *test* split of KITTI dataset in Fig. 6. From the figure we could see that our proposed part-aware network could estimate accurate intra-object part locations by using only point cloud as inputs, which are aggregated by our designed part-aggregation network to generate the accurate 3D bounding boxes.

## 5. Conclusion

We presented a novel 3D object detection method, part-aware and aggregation neural network (Part-$A^2$ net), for detecting 3D objects from point clouds. Our first part-aware stage learns to estimate the accurate intra-object part locations by using the free-of-charge location labels from the 3D box annotations. The predicted intra-object part locations of each object are pooled by the novel RoI-aware point cloud pooling scheme. The following part-aggregation stage can therefore consider the spatial relationship of the predicted intra-object part locations to score the boxes and refine their locations. The experiments show that our method achieves state-of-the-art performance on the challenging KITTI 3D detection benchmark, which demonstrates the effectiveness of our method.

## References


[1] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 1

[2] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017. 1, 2, 7

[3] X. Du, M. H. Ang, S. Karaman, and D. Rus. A general pipeline for 3d detection of vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3194–3200, May 2018. 1

[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 6

[5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1

[6] B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*, 2018. 1, 2, 3

[7] B. Graham and L. van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 1, 2, 3

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[9] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018. 5

[10] KITTI 3D object detection benchmark leader board. http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d, Accessed on 2019-07-09. 6

[11] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. *IROS*, 2018. 1, 2, 7

[12] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *CVPR*, 2019. 1, 2, 7

[13] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 5

[14] M. Liang*, B. Yang*, Y. Chen, R. Hu, and R. Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *CVPR*, 2019. 7

[15] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *ECCV*, 2018. 1, 2, 7

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 1

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 1, 4

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[19] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018. 1

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017. 1, 2, 7

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 1, 2

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 1, 2

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[24] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3, 4

[27] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1, 2, 4, 5, 7, 8

[28] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross. Complex-yolo: an euler-region-proposal for real-time 3d object detection on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 1

[29] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016. 1

[30] S. Sun, J. Pang, J. Shi, S. Yi, and W. Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. In *Advances in Neural Information Processing Systems*, pages 762–772, 2018. 4

[31] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 1, 2

[32] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 3, 4, 5, 7

[33] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *2nd Conference on Robot Learning (CoRL)*, 2018. 1, 2

[34] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 1, 2

[35] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 1, 2, 3, 5, 7