

# Focal Loss in 3D Object Detection

Peng Yun<sup>1</sup> Lei Tai<sup>2</sup> Yuan Wang<sup>2</sup> Chengju Liu<sup>3</sup> Ming Liu<sup>2</sup>

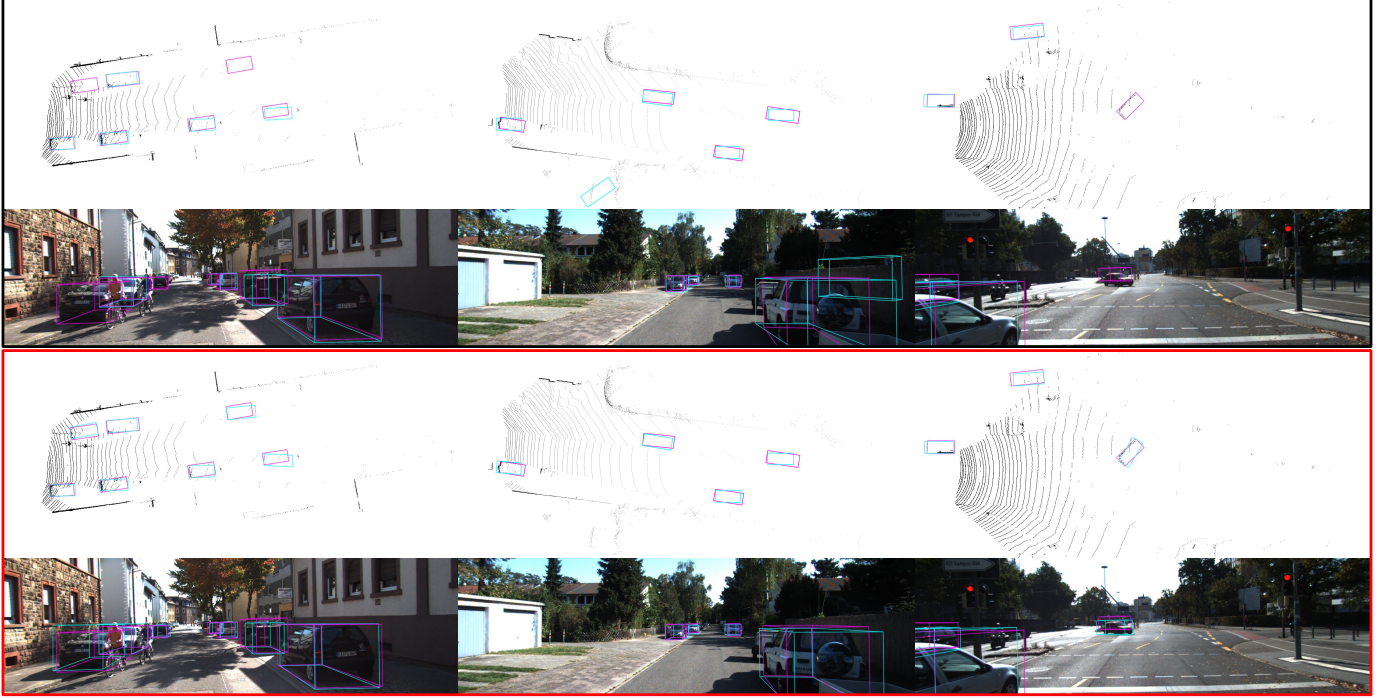


Fig. 1. Upper two rows show projected 3D object detection results from the detector trained with binary cross entropy. Lower two rows present related results from the detector trained with the focal loss. Purple and blue bounding boxes are the ground-truth and the estimated results respectively.

**Abstract**—3D object detection is still an open problem in autonomous driving scenes. When recognizing and localizing key objects from sparse 3D inputs, autonomous vehicles suffer from a larger continuous searching space and higher fore-background imbalance compared to image-based object detection. In this paper, we aim to solve this fore-background imbalance in 3D object detection. Inspired by the recent use of focal loss in image-based object detection, we extend this hard-mining improvement of binary cross entropy to point-cloud-based object detection and conduct experiments to show its performance based on two different 3D detectors: 3D-FCN and VoxelNet. The evaluation results show up to 11.2AP gains through the focal loss in a wide range of hyperparameters for 3D object detection.

**Index Terms**—Deep Learning in Robotics and Automation; Object Detection, Segmentation and Categorization; Recognition.

This work was supported by the National Natural Science Foundation of China (Grant No. U1713211), and was partially supported by Shenzhen Science Technology and Innovation Commission (SZSTI) JCYJ20160428154842603, the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, No. 16212815 and No. 21202816 awarded to Prof. Ming Liu. (*Corresponding author: Peng Yun.*)

<sup>1</sup>Peng Yun is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: pyun@ust.hk)

<sup>2</sup>Lei Tai, Yuan Wang and Ming Liu are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: {ltai, ywangeq, eelium}@ust.hk)

<sup>3</sup>Chengju Liu is with College of Electrical and Information Engineering, Tongji University, China (e-mail: liuchengju@tongji.edu.cn)

## I. INTRODUCTION

OBJECT detection in 3D is still challenging in robotics perception, the applied scenes of which widely include urban and suburban roads, highways, bridges and indoor settings. Robots recognize and localize key objects from data in the 3D form and predict their locations, sizes and orientations, which provides both semantic and spatial information for high-level decision making. The point cloud is one of the most commonly used 3D data forms, and can be gathered by range cameras, like LiDAR and RGB-D cameras. Since the coordinate information of point clouds is not influenced by appearance changes, point clouds are also robust in extreme weather and various seasons. In addition, it is naturally scale-invariant. The scale of an object is invariant anywhere in a point cloud, while it always changes in an image due to foreshortening effects. Moreover, the increasing perception distance and decreasing price of 3D LiDARs make them a promising direction for autonomous driving researchers [1].

Current image-based detectors benefit from translation invariance from convolution operations and can perform with human-comparable accuracy. However, the successful image-based architectures cannot be directly applied in 3D space. Point-cloud-based object detection consumes point clouds which are sparse point lists instead of dense arrays. If drawing

on the success of image-based detectors and conducting dense convolution operation to acquire translation invariance, pre-processing must be implemented to convert the sparse point clouds into dense arrays. Otherwise, special layers should be carefully designed to extract meaningful features from the sparse inputs. Additionally, the fore-background imbalance is much more serious than in 2D scenarios, since the new z-axis further enlarges the searching space and the extent of imbalance is different for each different z value.

Lin *et al.* [2] proposed focal loss to tackle the fore-background imbalance in image-based object detection, so that one-stage detectors could achieve state-of-the-art accuracy as two-stage detectors. As a hard-mining improvement of binary cross entropy, it helps the network focus on hard classified objects, in case they are overwhelmed by a large number of easily classified objects.

Similar to image-based detection methods, point-cloud-based detection methods can also be classified into two-stage [3], [4], [5] and one-stage detectors [6], [7]. In this paper, inspired by [2], we aim to solve the fore-background imbalance for 3D object detection through the focal loss. We claim the following contributions:

- We extend focal loss to 3D object detection to solve the huge fore-background imbalance in one-stage detectors, and conduct experiments on two different one-stage 3D object detectors, 3D-FCN [6] and VoxelNet [7]. The experiment results demonstrate up to 11.2AP gains from the focal loss in a wide range of hyperparameters.
- To further understand focal loss in 3D object detection, we analyze its effect towards foreground and background estimations, and validate that it plays a role similar to image-based detection. We also find that the special architecture of VoxelNet can naturally handle the hard negatives well.
- We plot the final posterior probability distributions of the two detectors and demonstrate that the focal loss with the increasing hyperparameter  $\gamma$  decreases the estimation posterior probabilities.

## II. RELATED WORK

### A. Two-Stage 3D Object Detection

When extending two-stage image detectors to the 3D space, researchers encounter the following problems: (1) the input is sparse and at low resolution; (2) the original image-based methods are not guaranteed to have enough information to generate region proposals. Ku *et al.* [4] proposed AVOD which fused RGB images and point clouds. It first proposes aligned 3D bounding boxes with a multimodal fusion region proposal network. Then, the proposed bounding boxes are classified and regressed with fully connected layers. Both the appearance and the 3D information are well-utilized to improve the accuracy and robustness of the proposed model in extreme scenes. Their hand-crafted features can be further improved to learn representations directly from raw LiDAR inputs to alleviate information loss.

Qi *et al.* [3] proposed F-PointNet and leveraged both 2D object detectors and 3D deep learning for object localization.

TABLE I  
IMAGE-BASED AND POINT-CLOUD-BASED OBJECT DETECTION

	Image-Based Object Detection	Point-Cloud-Based Object Detection	
Method	-	3D-FCN [6]	VoxelNet [7]
Dimension	2D	3D	3D
Input	Dense Grid	Dense Grid	Sparse Point List
Network	Dense Conv	Dense Conv	Heterogeneous
Pipeline	One/Two-Stage	One-Stage	One-Stage

They extracted the 3D bounding frustum of an object with a 2D object detector. Then 3D instance segmentation and 3D bounding box regression were applied with two variants of PointNet [8]. F-PointNet achieves state-of-the-art accuracy on the KITTI 3D object detection challenge [9], and also performs at real-time speed for 3D object detection. Their image detector needs to be carefully designed with a high recall rate, since the accuracy upper bound is determined by the first stage.

### B. One-Stage 3D Object Detection

Li [6] extended a 2D fully convolutional network to 3D. The voxelized point clouds are processed by an encoder-decoder network. The 3D fully convolutional network (3D-FCN) finally proposes a probability and a regression map for the whole detection region. It thoroughly consists of 3D dense convolutions with high computation and memory costs, so that the network depth is limited and hard to extract high-level features. Unlike 3D-FCN and AVOD, both of which adopt hand-crafted features to represent the point clouds, Zhou *et al.* [7] designed an end-to-end network to implement point-cloud-based 3D object detection with learning representations called VoxelNet. Compared to 3D-FCN [6], the computation cost is mitigated by the Voxel Feature Encoding Layers (VFELayers) and 2D convolution.

In this paper, we adopt 3D-FCN [6] and VoxelNet [7] as two different types of one-stage 3D detectors. As shown in Table I, 3D-FCN consumes dense grids and consists of only 3D dense convolution layers, where the 2D FCN architecture [10] is extended to 3D for dense feature extraction. In contrast, VoxelNet consumes sparse point lists and is a heterogeneous network, which firstly extracts sparse features with its novel VFELayers and then conducts 3D and 2D convolution sequentially.

### C. Imbalance between Foreground and Background

Image-based object detectors can be classified into two-stage and one-stage detectors. For two-stage detectors, like R-CNN [11], the first stage generates a sparse set of candidate object locations and the second stage classifies each candidate location as one of the foreground classes or as the background using a convolutional neural network. The two-stage detectors [12], [13] achieve state-of-the-art accuracy on the COCO benchmark. On the other hand, one-stage detectors, like YOLO [14] and SSD [15], aim to simplify the pipeline. They improve the training speed of deep models and also demonstrate promising results in terms of accuracy.

Lin *et al.* [2] explored both one-stage and two-stage detectors in image-based object detection, and claimed that the hurdle that obstructs the one-stage detectors from better accuracy is the extreme fore-background class imbalance encountered during training of dense detectors. They reshaped the standard cross entropy loss and proposed the focal loss such that the losses assigned to well-classified examples were down-weighted. This can be seen as a hard-mining improvement of binary cross entropy to help networks focus on hard classified objects in case they are overwhelmed by a large number of easily classified objects.

We extend focal loss to 3D object detection to tackle the fore-background imbalance problem. Different from image-based detection, point-cloud-based object detection is a more challenging perception problem in 3D space with sparse sensor data and suffers from more serious fore-background imbalance. To thoroughly evaluate the performance of the focal loss in this harder task, we conduct experiments based on two different types of one-stage 3D detectors: 3D-FCN and VoxelNet. We analyze the focal loss effect on these two 3D detectors following a similar method to that in [2], and further discuss the decreasing posterior probability effect of the focal loss.

### III. FOCAL LOSS

In this section, we first declare notations and revisit the focal loss [2], and then further analyze the fore-background imbalance in 3D object detection.

#### A. Preliminaries

We define  $y \in \{\pm 1\}$  as the ground-truth class, and  $p$  as the estimated probability for the class with label  $y = 1$ . For notational convenience, we define the posterior probability  $p_t$  as

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{if } y = -1, \end{cases} \quad (1)$$

where  $p$  is calculated with  $p = \text{sigmoid}(x)$ . The binary cross entropy (BCE) loss and its deviation can be formulated as

$$\mathcal{E}_{BCE}(p_t) = -\log(p_t) \quad (2)$$

$$\frac{d\mathcal{E}_{BCE}(p_t)}{dx} = y(p_t - 1). \quad (3)$$

As claimed in [2], when the network is trained with BCE loss, its gradient will be dominated by vast easy classified negative samples if a huge fore-background imbalance exists. Focal loss can be considered as a dynamically scaled cross entropy loss, which is defined as

$$\mathcal{E}_{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t) \quad (4)$$

$$\frac{d\mathcal{E}_{FL}(p_t)}{dx} = y(1 - p_t)^\gamma (\gamma p_t \log(p_t) + p_t - 1). \quad (5)$$

The contribution from the well classified samples ( $p_t \gg 0.5$ ) to the loss is down-weighted. The hyperparameter  $\gamma$  of the focal loss can be used to tune the weight of different samples. As  $\gamma$  increases, fewer easily classified samples contribute to the training loss. Obviously, when  $\gamma$  reaches 0, the focal loss

degrades to become same as the BCE loss. In the following sections, all the cases with  $\gamma = 0$  represent BCE loss cases.

Researchers have previously either introduced hyperparameters to balance the losses calculated from positive and negative anchors, or normalized positive and negative losses by the frequency of corresponding anchors. However, one essential problem that these two previous methods cannot handle is the gradient salience of hard negative samples. The gradients of hard negative anchors ( $p_t < 0.5$ ) are overwhelmed by a large number of easy negative anchors ( $p_t \gg 0.5$ ). Due to the dynamic scaling with the posterior probability  $p_t$ , a weighted focal loss can be used to handle both the fore-background imbalance and the gradient salience of hard negative samples with the following form,

$$\mathcal{E}_{FL}(p_t) = -\lambda (1 - p_t)^\gamma \log(p_t), \quad (6)$$

where  $\lambda$  is induced to weight different classes. In the following sections, we adopt hyperparameters  $\alpha$  and  $\beta$  to weight positive and negative focal loss respectively.

#### B. Fore-background Imbalance in 3D Object Detection

The methods for 3D object detection can be classified as one-stage [6], [7] and two-stage [3], [4], [5] detectors. The two-stage detectors first adopt an algorithm with a high recall rate to propose regions that possibly contain objects and adopt a convolution network to classify classes and regress bounding boxes. The one-stage detectors are end-to-end networks that learn representations and implement classification and regression in all anchors.

In one-stage methods, anchors are proposed at each location, and thus a huge fore-background imbalance exists. For instance, there are 50k bounding boxes proposed in each frame for 3D-FCN and 70k for VoxelNet, but less than 30 anchors among them contain positive objects (e.g. car, pedestrian, cyclist). Compared to image detectors, the extra estimation in z-axis further increases the fore-background imbalance. Additionally, positive samples always locate on the position with small z values in some specific scenes. For instance, cars and pedestrians are always on the road in autonomous driving scenes. In such situations, the distribution of fore-background imbalance is different along the z-axis: the extent of imbalance increases with higher z values.

The one-stage methods for 3D detectors are different from the 2D detectors because of their larger searching space, sparse input and different types of network architecture. Therefore, we select two different networks, 3D-FCN and VoxelNet, to conduct experiments to evaluate the performance of focal loss in 3D object detection. The features of these two 3D detectors are discussed in the following two sections, and the experimental details and results are shown in Section VI.

### IV. 3D-FCN FEATURES

In this section, we discuss the dense convolution network architecture of 3D-FCN and introduce our enhanced loss function for 3D-FCN. The details of 3D-FCN can be found in [6]. Please refer to APPENDIX for our implementation of 3D-FCN.

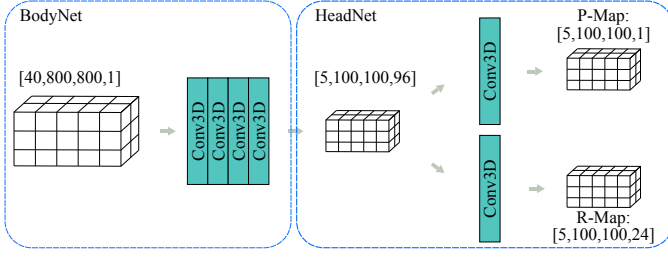


Fig. 2. The dense convolution network architecture of 3D-FCN [6]. The whole network consists of only 3D convolution layers. All intermediate tensors in the hidden space are dense 3D grids (which are represented by a tensor with dimensions as [height, width, length, feature]).

### A. Dense Convolution Network Architecture

3D-FCN [6] draws on experience from image-based recognition tasks, and extends the 2D convolution layer to 3D space to acquire translation invariance. The input point cloud is firstly voxelized into a 3D dense grid. In each voxel of the 3D dense grid, the values  $\{0, 1\}$  are used to present whether there is any point observed. The network architecture of 3D-FCN is shown in Figure 2. The voxelized point cloud is convolved by four Conv3D blocks sequentially. The output features are then processed by two Conv3D blocks individually to generate a probability map and a regression map (P-Map and R-Map). Different from image-based object detection, the probability map and regression map are all in 3D dense grids, so that the searching space is exponentially increased.

### B. Enhanced Loss Function

The original loss function for 3D-FCN [6] is shown in the left of Equation 7 to 11, where  $\epsilon_P$  and  $\epsilon_R$  represent the classification loss and regression loss, as well as  $\epsilon_{cls}$  and  $\epsilon_{reg}$  are the loss functions used for classification and regression respectively. In regression loss  $\epsilon_R$ ,  $\mathbf{u}_i$  and  $\mathbf{u}_i^*$  are the regression output and ground truth for positive anchors. In classification loss  $\epsilon_P$ ,  $p_i^{pos}$  and  $p_i^{neg}$  represent the posterior probability of positive and negative estimation.

$$\epsilon = \epsilon_P + \epsilon_R \rightarrow \epsilon = \epsilon_P + \epsilon_R \quad (7)$$

$$\epsilon_P = \eta(\epsilon_P^{pos} + \epsilon_P^{neg}) \rightarrow \epsilon_P = \eta(\epsilon_P^{pos} + \epsilon_P^{neg}) \quad (8)$$

$$\epsilon_R = \sum_i \epsilon_{reg}(\mathbf{u}_i, \mathbf{u}_i^*) \rightarrow \epsilon_R = \frac{1}{N_{pos}} \sum_i \epsilon_{reg}(\mathbf{u}_i, \mathbf{u}_i^*) \quad (9)$$

$$\epsilon_P^{pos} = \sum_i \epsilon_{cls}(p_i^{pos}, 1) \rightarrow \epsilon_P^{pos} = \alpha \frac{1}{N_{pos}} \sum_i \epsilon_{cls}(p_i^{pos}, 1) \quad (10)$$

$$\epsilon_P^{neg} = \sum_i \epsilon_{cls}(p_i^{neg}, 0) \rightarrow \epsilon_P^{neg} = \beta \frac{1}{N_{neg}} \sum_i \epsilon_{cls}(p_i^{neg}, 0) \quad (11)$$

In the original form, a large imbalance exists between  $\epsilon_P^{pos}$  and  $\epsilon_P^{neg}$ , which represent classification loss of positive and negative samples respectively. Therefore, we adopt the loss function used in VoxelNet [7], which normalizes sub-loss with corresponding frequency as well as balances  $\epsilon_P^{pos}$  and  $\epsilon_P^{neg}$  with two more hyperparameters  $\alpha$  and  $\beta$ . The adopted loss function is shown in the right of Equation 7 to 11.

In Section VI, we use the loss function in the right part of Equation 7 to 11 to demonstrate the focal loss improvement compared with BCE Loss, where  $\epsilon_{reg}$  denotes the square loss

and  $\epsilon_{cls}$  denotes the focal loss. We also show the enhanced loss function form improvement compared with the original loss function [6] in the APPENDIX, where  $\epsilon_{reg}$  denotes the square loss and  $\epsilon_{cls}$  denotes the BCE loss.

## V. VOXELNET FEATURES

In this section, we discuss the heterogeneous network architecture of VoxelNet, and its bird's-eye-view estimation. The details of VoxelNet can be found in [7]. Please refer to APPENDIX for our implementation of VoxelNet.

### A. Heterogeneous Network Architecture

The heterogeneous architecture overview of VoxelNet is shown in Figure 3. It consists of three main parts: FeatureNet, MiddleLayer and RPN.

*FeatureNet* extracts features directly from sparse point lists. It adopts Voxel Feature Encoding Layers (VFELayers) [7] to extract both point-wise and voxel-wise features directly from points, where fully connected layers are used to extract point-wise features and a symmetric function is used to aggregate local features from all points within a local voxel. Compared to sub-optimally deriving hand-crafted features from voxels, VFELayers can learn representations minimizing the loss function. The derived voxel-wise representations from VFELayers are sparse, which saves memory and time in the computation. In contrast, if a point cloud of KITTI dataset is partitioned into a  $[10, 400, 352]$  dense grid for vehicle detection, only around 5300 voxels (about 0.3%) are non-empty. However, the sparse representation is currently unfriendly to convolutional operations. In order to implement convolution, VoxelNet compromises on efficiency and converts the sparse representation to a dense representation at the end of FeatureNet. Each sparse voxel-wise representation is copied to its specific entry in the dense grid.

*MiddleLayer* consumes the 3D dense grid and converts it to a 2D bird's-eye-view form, so that further processing can be done in 2D space. The role of MiddleLayer is to learn features from all voxels in the same bird's-eye-view location. Therefore, the 3D convolutional kernel is of size  $[d, 1, 1]$ , if we denote the dense grid in the order of  $z, x, y$ . The 3D kernel of size  $[d, 1, 1]$  helps aggregate voxel-wise features within a progressively expanding receptive field along the  $z$ -axis and keeps the shape in the  $x, y$  dimension.

*RPN* predicts the probability and regression map from the 2D bird's-eye-view feature map. Since the increased invariance and large receptive fields of top-level nodes will yield smooth responses and cause inaccurate localization, it does not utilize max-pooling but adopts skip-layers [10] to combine high-level semantic features and low-level spatial features.

### B. Estimation in Bird's-Eye-View Form

The final probability and regression estimation maps are all in bird's-eye view form, which is similar to the final estimation of image-based detection methods. This saves both memory and time of the calculation compared to 3D maps, but only one object per location can be estimated in the bird's-eye view.



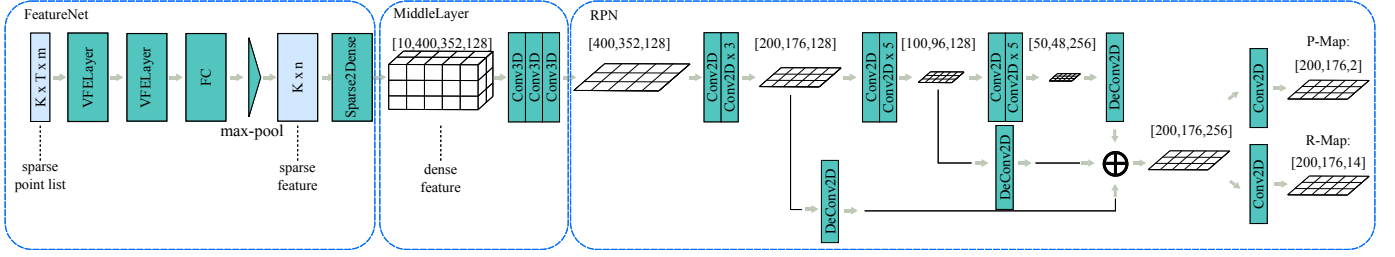


Fig. 3. VoxelNet heterogeneous architecture [7]. It consists of three main parts: FeatureNet (point-wise and voxel-wise feature transformation), MiddleLayer (3D dense convolution) and RPN (2D dense convolution). The probability and regression maps are in bird’s-eye-view form.

This is acceptable in autonomous driving scenes but will meet problems in indoor scenes, where objects can be stacked up (e.g., a mug on a stack of books).

MiddleLayer saves calculation for further processing by aggregating the 3D dense grid into a 2D bird’s-eye-view feature map. Otherwise, thoroughly 3D dense convolution in such a deep network (22 convolution layers) would bring exponentially more parameters and calculation. We note that MiddleLayer is still a bottleneck of the whole network as shown in Table VII because of its 3D dense convolution operations. The efficient sparse convolutional implementation is still an open problem and deserves effort to solve.

### C. Loss Function

We adopt the loss function form from the original VoxelNet [7], which is the same as the right half part from Equation 7 to 11. In Section VI, we use SmoothL1Norm [16] for  $\epsilon_{reg}$  as the original paper [7] and use the focal loss for  $\epsilon_{cls}$ .

## VI. EXPERIMENTS

In this section, we intend to answer two questions: 1) Can focal loss help improve accuracy in 3D object detection task? 2) Does focal loss have an equal effect in 3D object detection to its effect in image-based detection? To answer the former question, we conduct experiments to compare the performance of 3D-FCN and VoxelNet trained with BCE loss and focal loss on the challenging KITTI benchmark [9]. To answer the second question, we analyze the cumulative distribution curve of 3D-FCN and VoxelNet following a similar method to that in [13]. The code and weights for our experiments are available at <https://sites.google.com/view/fl3d>.

### A. BCE Loss vs. Focal Loss

The KITTI 3D object detection dataset [9] contains 3D annotations for cars, pedestrians and cyclists in urban driving scenarios. The sensor setup mainly consists of a wide-angle camera and a Velodyne LiDAR (HDL-64E), both of which are well-calibrated. The training dataset contains 7481 frames, including both raw sensor data and annotations. The KITTI 3D detection dataset contains some bad annotations which are empty bounding boxes containing few points. In order to avoid overfitting those bad annotations, we remove all bounding boxes containing few points (fewer than 10). Following [5], we split the dataset into training and validation sets, each containing around half of the entire set.

For simplicity, we conduct experiments only on the car class to show the focal loss improvement. We do such implement because both 3D-FCN and VoxelNet are trained class-specifically and extending them to other classes is only tuning techniques. Also, the focal loss in the form of Equation 6 is agnostic to the class of objects.

We set  $\alpha = 1$ ,  $\beta = 5$ ,  $\eta = 10$  in 3D-FCN and  $\alpha = 1$ ,  $\beta = 10$ ,  $\eta = 0.5$  in VoxelNet so that  $\epsilon_p^{pos}$  and  $\epsilon_p^{neg}$  as well as  $\epsilon_p$  and  $\epsilon_R$  will be of the same orders of magnitude. As claimed in [2], when training a network from scratch with the focal loss, it is unstable in the beginning. Therefore, we first train the network (both 3D-FCN and VoxelNet) for 30 epochs with the BCE loss and the learning rate  $lr$ , and then for another 30 epochs with the focal loss and a discounted learning rate  $0.1lr$ . The minimum overlap thresholds are 0.7, 0.5, 0.5 for 2D evaluation on image/ground plane and 3D evaluation. The network details of both 3D-FCN and VoxelNet are shown in Table VI and Table VII in APPENDIX. Non-maximum suppression with the threshold 0.8 is used at the end of 3D-FCN and VoxelNet for estimation refinement.

In order to control a single variable  $\gamma$ , we firstly make comparisons among last models, which are trained with the same amount of steps. Additionally, we also make comparisons among best models to make the conclusion more concrete. The best models are selected according to the mean value among easy, moderate and hard 3D detection APs (3D detection mAP).

We compare the results of the last models in Table II and Table III, where the rows with  $\gamma = 0$  and  $\gamma > 0$  represent the results from the BCE loss and the focal loss respectively. Bolded numbers are the results in which focal loss cases outperforms the BCE loss case. In general, VoxelNet outperforms 3D-FCN in accuracy, since the input of VoxelNet has the original point clouds, but 3D-FCN suffers from information loss when voxelizing the point clouds into binary representations. Additionally, VoxelNet benefits from its deeper network structure, which is able to extract more useful high-level features. In 3D-FCN, the focal loss helps improve accuracy in all metrics in a wide range of hyperparameters ( $0 < \gamma \leq 2.0$ ), providing gains from 0.3AP to 11.2AP. In VoxelNet, the cases with  $\gamma = 0.1, 0.5, 1$  show gains from the focal loss in all metrics, ranging from 0.6AP to 9.1AP. Both gains and losses happen when  $\gamma$  is 0.2 or 2. However, gains (up to 9.1AP) are generally much greater than losses (at most 2.7AP). The training processes include some randomness due to sample shuffling and the sophisticated gradient descent training scheme. We further evaluate all intermediate weights and select the best models

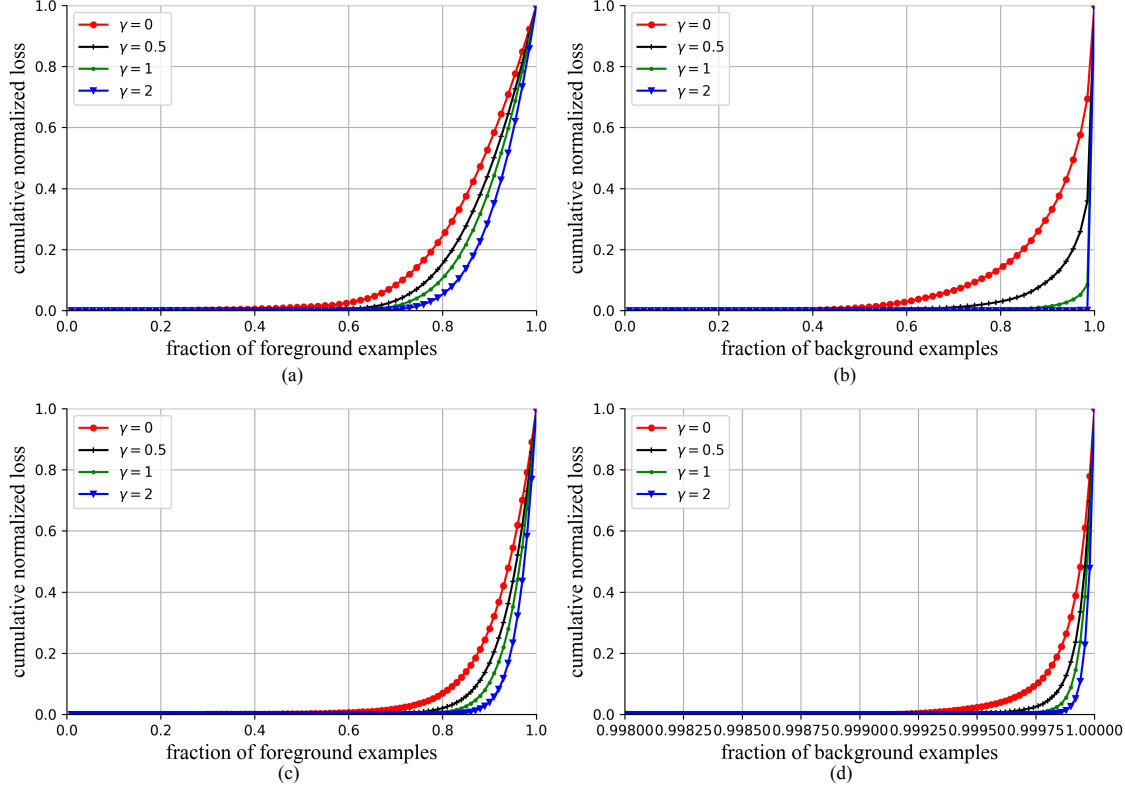


Fig. 4. Cumulative distributions of 3D-FCN and VoxelNet for different values of  $\gamma$ . In 3D-FCN (a, b), as  $\gamma$  increases, loss of both foreground and background samples concentrate on the harder partitions. The effect on the background is stronger. In VoxelNet (c, d), the effect of the focal loss increases as  $\gamma$  increases, but the effect on the foreground is stronger than on the background. Note that the VoxelNet background cumulative distribution (d) is in the range of  $[0.998, 1]$ .

TABLE II  
EVALUATION RESULTS ON KITTI VALIDATION DATASET  
FOR LAST MODELS OF 3D-FCN

$\gamma$	Bird's Eye View AP (%)			3D Detection AP (%)		
	Easy	Mod	Hard	Easy	Mod	Hard
0	32.11	31.67	27.78	24.22	21.96	18.63
0.1	<b>37.53</b>	<b>35.15</b>	<b>30.61</b>	<b>28.24</b>	<b>24.73</b>	<b>24.80</b>
0.2	<b>38.10</b>	<b>35.32</b>	<b>30.65</b>	<b>27.75</b>	<b>23.88</b>	<b>20.36</b>
0.5	<b>33.59</b>	<b>32.61</b>	<b>28.59</b>	<b>24.76</b>	<b>22.34</b>	<b>19.04</b>
1	<b>42.91</b>	<b>38.21</b>	<b>32.96</b>	<b>32.26</b>	<b>26.70</b>	<b>22.58</b>
2	<b>43.32</b>	<b>38.45</b>	<b>33.09</b>	<b>32.91</b>	<b>27.23</b>	<b>22.81</b>
5	25.18	24.38	20.62	18.77	16.47	17.27

TABLE III  
EVALUATION RESULTS ON KITTI VALIDATION DATASET  
FOR LAST MODELS OF VoxelNet

$\gamma$	Bird's Eye View AP (%)			3D Detection AP (%)		
	Easy	Mod	Hard	Easy	Mod	Hard
0	85.26	61.35	60.97	70.54	55.11	48.79
0.1	<b>85.93</b>	<b>69.65</b>	<b>68.83</b>	<b>72.67</b>	<b>56.31</b>	<b>56.11</b>
0.2	82.55	60.42	60.23	<b>72.66</b>	<b>56.67</b>	<b>50.41</b>
0.5	<b>86.80</b>	<b>69.40</b>	<b>61.79</b>	<b>75.86</b>	<b>58.28</b>	<b>57.92</b>
1	<b>87.28</b>	<b>70.46</b>	<b>61.93</b>	<b>74.16</b>	<b>57.01</b>	<b>56.20</b>
2	84.48	<b>68.76</b>	<b>61.04</b>	<b>70.82</b>	<b>55.25</b>	<b>54.67</b>
5	80.48	<b>62.56</b>	53.76	<b>75.04</b>	50.85	<b>50.53</b>

to make the comparison in Table IV. It shows that focal loss helps improve accuracy in all metrics with a proper  $\gamma$ . The performance losses of  $\gamma = 0.2$  in Table III might be caused by training randomness and model degradation with redundant training.

From Table II, Table III and Table IV, it shows that the focal loss in 3D object detection provides better or comparable results than BCE loss. Therefore, the focal loss works in 3D object detection and help improve accuracy in a wide range of  $\gamma$  (normally  $\gamma \leq 2$ ).

### B. Analysis of Focal Loss in 3D Detectors

We analyze the empirical cumulative distributions of the loss from the converged 3D-FCN and VoxelNet models as in [2]. We apply the two converged models trained with the focal loss

TABLE IV  
EVALUATION RESULT ON KITTI VALIDATION DATASET  
FOR BEST MODELS

Detector	$\gamma$	lr	Step	Bird's Eye View AP(%)			3D Detection AP(%)		
				Easy	Mod	Hard	Easy	Mod	Hard
3D-FCN	0	1e-2	126k	51.33	45.82	40.24	40.01	33.12	28.94
3D-FCN	2	1e-2	137k	<b>53.19</b>	<b>48.03</b>	<b>41.96</b>	<b>46.05</b>	<b>35.93</b>	<b>31.01</b>
VoxelNet	0	1e-4	134k	85.76	68.52	61.00	75.42	58.09	57.61
VoxelNet	0.2	1e-4	215k	<b>86.89</b>	<b>69.33</b>	<b>61.63</b>	<b>80.08</b>	<b>58.39</b>	57.60

Note that all cases in Table IV are the evaluation results of the best models selected among all intermediate weights. Thus the accuracy improvement is from the focal loss instead of longer training steps.

(row 2 and row 4 in Table IV) on the validation dataset and sample the predicted probability for  $10^7$  negative windows and

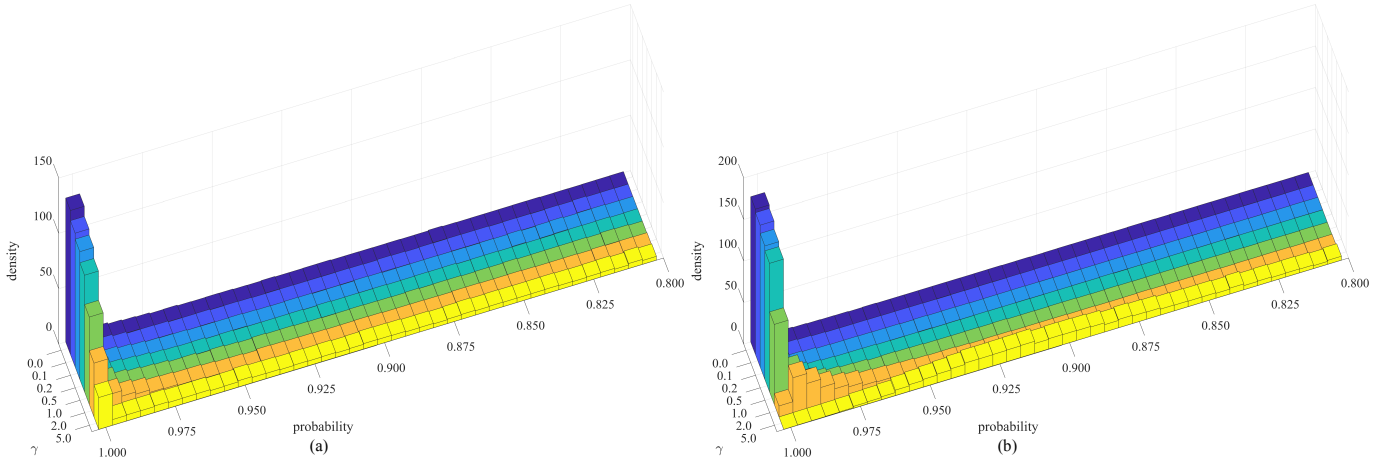


Fig. 5. Posterior probability histogram of 3D-FCN (a) and VoxelNet (b). As  $\gamma$  increases, the peak decreases and moves towards lower values in both 3D-FCN and VoxelNet.

$10^5$  positive windows. Then, we calculate the focal loss with these probability data. The calculated focal loss is normalized such that it sums to one and is sorted from low to high. We plot the cumulative distributions for 3D-FCN and VoxelNet for different  $\gamma$  in Figure 4.

In 3D-FCN, approximately 15% of the hardest positive samples account for roughly half of the positive loss. As  $\gamma$  increases, more of the loss gets concentrated in the top 15% of examples. However, compared to the effect of the focal loss on negative samples, its effect on the positive samples is minor. For  $\gamma = 0$ , the positive and negative CDFs are quite similar. As  $\gamma$  increases, more weight becomes concentrated on the hard negative examples. With  $\gamma = 2$  (the best result for 3D-FCN), the vast majority of the loss comes from a small fraction of samples. As claimed in [2], the focal loss can effectively discount the effect of easy negatives, so that the network focuses on learning the hard negative examples.

In VoxelNet, the condition is different. From c and d in Figure 4, we can see that the effect of the focal loss increases in both the positive and negative samples as  $\gamma$  increases. However, the cumulative distribution functions for the negative samples are quite similar among different values of  $\gamma$ , even though we adjust the x-axis to  $[0.998, 1]$ . This shows that VoxelNet trained with the BCE loss is already able to handle negative hard samples. Compared with the results on the negative samples, the effects of focal loss on the positive samples are stronger. Therefore, the accuracy gains of the focal loss in VoxelNet are mainly from the positive hard samples.

From the analysis of cumulative distributions, we believe that the focal loss in 3D object detection helps networks alleviate hard sample gradient salience in the training process.

### C. Focal Loss Decreases the Posterior Probabilities

When undertaking the experiments, we found networks trained with the focal loss should be set with a lower threshold for non-maximum suppression. This inspires us to explore the influence of the focal loss on the output posterior probabilities. We take the models in Table II and Table III, and evaluate them on the validation set. We record all the evaluation results and plot the probability histogram for positive bounding

boxes. The results are shown in Figure 5. As  $\gamma$  increases, the peak decreases and moves towards the lower values. This demonstrates that networks trained with the focal loss output positive estimation with lower posterior probabilities. A probable explanation is that objects with high posterior probabilities are easily classified, and the loss they contribute is down-weighted in the training process due to the focal loss. In other words, they will be relatively ignored in the training process if they are estimated with high posterior probabilities, so that their posterior probabilities cannot be further improved. However, they can also be accurately classified if we decrease the non-maximum suppression threshold in the final output step.

## VII. CONCLUSION

In this paper, we extended the focal loss of image detectors to 3D object detection to solve the fore-background imbalance. We conducted experiments on two different types of 3D object detectors to demonstrate the performance of the focal loss in point-cloud-based object detection. The experimental results show that the focal loss helps improve accuracy in 3D object detection, and it protects the network from fore-background imbalance and alleviates hard sample gradient salience both for positive and negative anchors in the training process. The posterior probability histograms show that the networks trained with the focal loss outputs positive estimation with lower posterior probabilities.

## REFERENCES

- [1] Z. Wang, Y. Liu, Q. Liao, H. Ye, M. Liu, and L. Wang, "Characterization of a rs-lidar for 3d perception," in *IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, July 2018.
- [2] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.
- [3] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 918–927.
- [4] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1–8.

- [5] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6526–6534.
- [6] B. Li, "3d fully convolutional network for vehicle detection in point cloud," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 1513–1518.
- [7] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 4490–4499.
- [8] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 77–85.
- [9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3354–3361.
- [10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.
- [11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.
- [12] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2980–2988.
- [13] T. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 936–944.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 779–788.
- [15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

## APPENDIX

### A. Improvement of Enhanced Loss Function for 3D-FCN

We demonstrate the improvement of adopting the loss function from VoxelNet [7] (normalization, new hyperparameters, BCE Loss) provides over its original loss function [6] for 3D-FCN. We set  $\alpha = 1$ ,  $\beta = 5$ ,  $\eta = 10$  in the enhanced 3D-FCN so that  $\epsilon_p^{pos}$  and  $\epsilon_p^{neg}$  as well as  $\epsilon_p$  and  $\epsilon_R$  can be of the same orders of magnitude. We set  $\eta = 0.1$  in the original 3D-FCN so that  $\epsilon_p$  and  $\epsilon_R$  can be of the same orders of magnitude.  $\gamma$  is set as 0 for using BCE loss. We train these two cases from scratch with 30 epochs. The threshold for non-maximum suppression is set as 0.994. The reason why  $\eta$  is  $100\times$  larger in the enhanced 3D-FCN is that we did normalization in the enhanced loss 3D-FCN and  $N_{neg}$  is much greater than  $N_{pos}$ . We compare the last models in Table V which shows the improvement of the enhanced loss function.

### B. Our 3D-FCN Implementation Details

The network details of 3D-FCN are shown in Table VI. Each Conv3D block in the BodyNet includes a 3D convolution layer, a ReLU layer and a batch normalization layer sequentially. In the HeadNet, each Conv3D block represents an individual 3D convolution layer. In the training phase, we create the ground

truth for P-Map by setting the object-voxel which contains an object center as 1. For the regression map, we create the ground truth by setting the object-voxels with 24-length residual vectors, each of which is the coordinates for the eight points of the bounding box with a fixed order. The result of the 3D-FCN baseline implemented by us is shown in the first row of Table IV.

### C. Our VoxelNet Implementation Details

The network details of VoxelNet are shown in Table VII. The FC block in VoxelNet consists of a fully connected layer, a batch normalization layer and a ReLU layer sequentially. Each Conv3D block in the MiddleLayer includes a 3D convolution layer, a ReLU layer and a batch normalization layer. The Conv2D block in the RPN consists of a 2D convolution layer, a ReLU layer and a batch normalization layer. The model of P-Map and R-Map is an individual 2D convolution layer. We adopt the original parameterization method and residual vector for regression of VoxelNet[7]. The result of our VoxelNet baseline is shown in the third row of Table IV.

TABLE V  
THE IMPROVEMENT OF THE ENHANCED LOSS FUNCTION FOR 3D-FCN

Detector	Bird's Eye View AP(%)			3D Detection AP(%)		
	Easy	Mod	Hard	Easy	Mod	Hard
Original	25.27	21.48	14.56	15.45	11.83	12.13
Enhanced	<b>28.90</b>	<b>27.33</b>	<b>27.47</b>	<b>18.23</b>	<b>16.54</b>	<b>14.48</b>

TABLE VI  
OUR IMPLEMENTATION DETAILS OF 3D-FCN

Block Name	Layer Name	Kernel Size	Strides	Filter	GFLOPs
Body	conv3d_1	[5,5,5]	[2,2,2]	32	25.8
	conv3d_2	[5,5,5]	[2,2,2]	64	204.9
	conv3d_3	[3,3,3]	[2,2,2]	96	16.6
	conv3d_4	[3,3,3]	[1,1,1]	96	24.9
Head-PMap	conv3d_obj	[3,3,3]	[1,1,1]	1	0.3
Head-RMap	conv3d_cor	[3,3,3]	[1,1,1]	24	6.2

TABLE VII  
OUR IMPLEMENTATION DETAILS OF VOXELNET

Block Name	Layer Name	Kernel Size / Output Unit	Strides	Filter	GFLOPs
FeatureNet	vfe	32	N/A	N/A	<0.1
	vfe	128	N/A	N/A	<0.1
	fc	128	N/A	N/A	<0.1
MiddleLayer	conv3d	[3,3,3]	[2,1,1]	64	311.5
	conv3d	[3,3,3]	[1,1,1]	64	93.5
	conv3d	[3,3,3]	[2,1,1]	64	62.3
	reshape	N/A	N/A	N/A	/
RPN	conv2d	[3,3]	[2,2]	128	41.6
	conv2d $\times$ 3	[3,3]	[1,1]	128	31.2
	deconv	[3,3]	[1,1]	256	20.8
	conv2d	[3,3]	[2,2]	128	10.4
	conv2d $\times$ 5	[3,3]	[1,1]	128	13.0
	deconv	[2,2]	[2,2]	256	5.2
	conv2d	[3,3]	[2,2]	256	5.2
	conv2d $\times$ 5	[3,3]	[1,1]	256	13.0
Prob-Map Reg-Map	deconv	[4,4]	[4,4]	256	2.6
Prob-Map	conv2d	[1,1]	[1,1]	2	0.1
Reg-Map	conv2d	[1,1]	[1,1]	14	0.8