RangeIoUDet: Range Image based Real-Time 3D Object Detector Optimized by Intersection over Union

Zhidong Liang Zehan Zhang Ming Zhang Xian Zhao Shiliang Pu*
Hikvision Research Institute
{liangzhidong, zhangzehan, zhangming15, zhaoxian, pushiliang.hri}@hikvision.com

Abstract

Real-time and high-performance 3D object detection is an attractive research direction in autonomous driving. Recent studies prefer point based or voxel based convolution for achieving high performance. However, these methods suffer from the unsatisfied efficiency or complex customized convolution, making them unsuitable for applications with real-time requirements. In this paper, we present an efficient and effective 3D object detection framework, named RangeIoUDet that uses the range image as input. Benefiting from the dense representation of the range image, RangeIoUDet is entirely constructed based on 2D convolution, making it possible to have a fast inference speed. This model learns pointwise features from the range image, which is then passed to a region proposal network for predicting 3D bounding boxes. We optimize the pointwise feature and the 3D box via the point-based IoU and box-based IoU supervision, respectively. The point-based IoU supervision is proposed to make the network better learn the implicit 3D information encoded in the range image. The 3D Hybrid GIoU loss is introduced to generate high-quality boxes while providing an accurate quality evaluation. Through the point-based IoU and the box-based IoU, RangeIoUDet outperforms all single-stage models on the KITTI dataset, while running at 45 FPS for inference. Experiments on the self-built dataset further prove its effectiveness on different LIDAR sensors and object categories.

1. Introduction

As the vital component of autonomous driving systems, 3D object detection from point clouds has attracted more and more attention. Many methods have been proposed for processing point clouds and achieved excellent performance. However, most of these methods are difficult to apply in practice, due to the complex framework, high memory complexity, and slow inference time (Fig. 1). The methods preferred by practical applications generally meet the following characteristics: simple framework for easy deployment, fast inference time, and 2D convolution based model without extra customized operations.

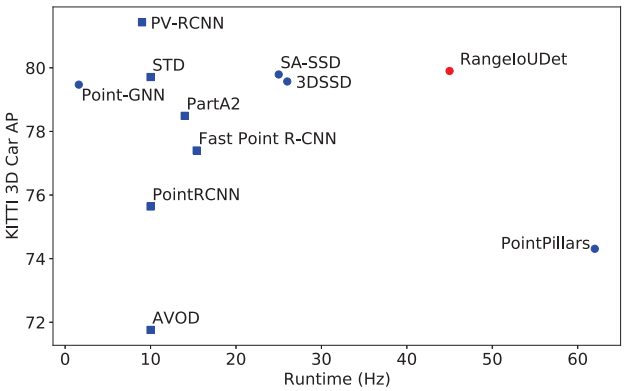

Figure 1. Speed (Hz) versus accuracy (AP) on the test set of KITTI 3D car detection. The single-stage methods are drawn as circles and the two-stage methods are drawn as squares. RangeIoUDet outperforms all methods except the top two-stage method PV-RCNN [24], and is much faster than PV-RCNN.

In terms of the input representations of point clouds, most existing methods can be divided into three types: voxel based, point based, and range image based methods. Voxel based [38, 32, 8, 26, 24] and point based methods [20, 31, 25, 27, 33] are currently the popular methods, but they are difficult to apply in practice due to the memory and time complexity issue. Range image based methods have been explored in early deep learning based 3D object detection [4]. As the raw data format of the LIDAR sensor, the range image is dense and compact, and retains almost all original information with minor loss. Operating on the range image enjoys the benefit of applying mature 2D convolution and does not suffer from the sparsity issue of point clouds, but this representation has been ignored for a long time for its unsatisfied performance [18]. Recently, several methods [18, 2, 29] rethink the advantages of this representation and propose effective frameworks based on the range

*Corresponding author. This work is supported by National Key R&D Program of China (Grant No.2020AAA010400X).

image. However, the inference speeds of these methods are still unsatisfactory due to the two-stage architecture [2] or multi-view fusion [29], making them still unable to meet the needs of practical applications. In this paper, we propose a high-performance and fast-speed single-stage 3D detection method based on the range image.

A simple idea utilizing the range image is to extract pointwise features from the range image [19] and then regress 3D bounding boxes from the bird's eye view, illustrated in the upper part of Fig. 2. Such a framework was proposed in our preliminary work [16] on the ArXiv. It only needs 2D convolution thanks to the dense representation of the range image. We introduce this framework in Sec. 3.1 of this paper. Although the framework is elegant, its performance is not satisfactory. The main drawback of the range image representation is the lack of the 3D local relationship, which means that points far away in the 3D space may be adjacent in the range image plane. It causes that although the pixels around the boundary of the object and background are far away in the 3D space, their features extracted from the range image may be similar due to the blurry issue of 2D feature extraction, which leads to the inaccurate pointwise features. The range image stores 3D spatial coordinates in its input channels, which means that it has the potential to learn more accurate features. To this end, we propose a point-based module to allow the network to learn the hidden spatial information encoded in the range image by explicit loss supervision, thereby indirectly enhancing the pointwise features. Specifically, the pointwise features are aggregated within the receptive field of 3D points, and then supervised by the point-based IoU [1]. It is worth noting that the point-based module is only used to supervise the learning of pointwise features during training, and is not needed for inference, so it will not bring extra computation cost or customized convolution.

Apart from enhancing the pointwise features, it is necessary to design power supervision losses to force the network to learn high-quality 3D boxes, especially for the single-stage model without the refinement stage. The performance of the 3D bounding box is mainly affected by the seven positioning parameters and the confidence score of the 3D box. Most current methods use smooth L1 loss to independently optimize the seven positioning parameters, and use the classification score to represent the confidence of the box. However, the positioning parameters are usually coupled with each other [35, 23], and the classification score cannot fully reflect the quality of the box. In order to address the above two challenges, we propose the 3D Hybrid GIoU loss based on the implementation of the differentiable 3D IoU. The "Hybrid" here consists of two meanings: the hybrid regression strategy and the combination of the regression and quality evaluation. On the one hand, we propose to use the smooth L1 loss to supervise the location of the box center and use the 3D GIoU loss to indirectly supervise the size of the 3D box. Such a combination avoids the local optimum of smooth L1 loss, and achieves better performance compared to the sole 3D GIoU. On the other hand, because most 3D objects have only partial point clouds, it is not easy to accurately regress the 3D bounding boxes, so a score that accurately measures the quality of the box is meaningful. 3D IoU is just a crucial indicator to measure the box quality, so we use the differentiable 3D IoU as the quality score to evaluate the quality of the 3D bounding box.

In summary, our contributions can be summarized into four-fold:

- We propose a single-stage 3D detection model RangeIoUDet based on the range image, which is simple, effective, fast, and only uses 2D convolution.
- We enhance pointwise features by supervising the point-based IoU, which makes the network better learn the implicit 3D information from the range image.
- We propose the 3D Hybrid GIoU (HyGIoU) loss for supervising the 3D bounding box with higher location accuracy and better quality evaluation.
- Our proposed single-stage model RangeIoUDet achieves state-of-the-art performance on the competitive KITTI 3D detection benchmark and the actual operation scenario dataset.

2. Related Work

2.1. 3D Object Detection

3D Object Detection based on 2D convolution. Early 3D object detection methods are mostly based on the 2D convolution. [4] directly projects the point clouds to the bird's eye view and predicts the 3D bounding box using 2D convolution. Many methods [10, 14, 13] improve this idea by multi-view aggregation and multi-sensor fusion. However, these 2D convolution based methods do not achieve the leading position in the benchmark. Also, the pipelines of these methods are complicated caused by the fusion strategy. [11] introduces a pillar-based encoding method and design a fast and effective 3D detection method. Recently, [18, 2] use the range image as the main representation to extract 3D features. [2] operates on the range image without conversion between views, and conducts experiments on the large-scale Waymo dataset [28]. The 2D convolution based methods have advantages in practical applications benefiting from the efficiency of 2D convolution.

3D Object Detection based on 3D convolution. The 3D convolution mainly includes two types: 3D voxel based convolution and 3D point based convolution. The pioneering work VoxelNet [38] divides the point cloud into 3D voxels, which are further processed by voxel feature encoder

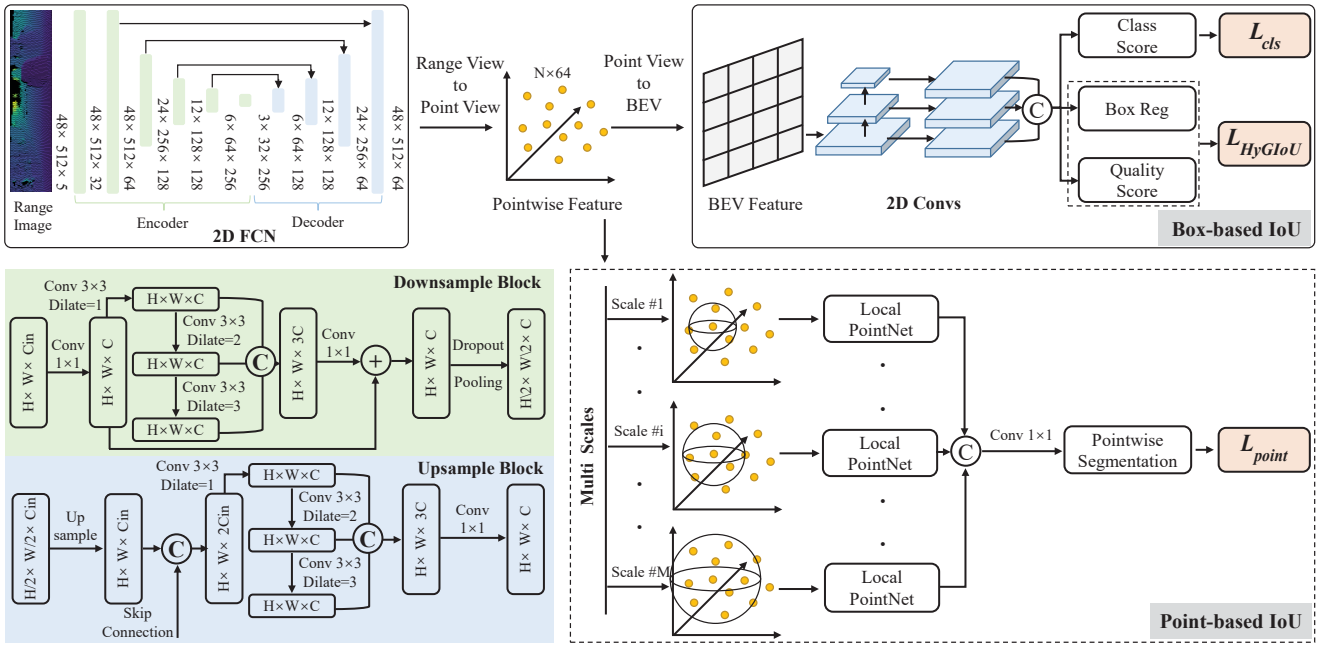

Figure 2. The architecture of RangeIoUDet. The range image is input to a 2D fully convolutional network (2D FCN) to extract high-dimensional features. The 2D FCN is an encoder-decoder structure that consists of a series of downsample blocks (green) and upsample blocks (blue). The pointwise feature is obtained according to the correspondence between the range image and point clouds. For improving the quality of the pointwise feature, a point-based IoU module is proposed to indirectly supervise the pointwise feature using Lovasz-Softmax loss. For improving the quality of the 3D bounding box, the 3D Hybrid GIoU loss is introduced to simultaneously provide accurate position and quality evaluation.

and 3D convolution. The resolution of the 3D voxel was limited by the memory complexity issue for a long time in the past. Afterward, the sparse convolution is proposed to reduce the memory and time complexity of 3D voxel based convolution, which greatly promotes the development of the voxel based methods [32, 8, 26, 24]. Until now, the 3D voxel based methods leads most of the 3D benchmark. The 3D point based convolution is a big innovation in the field of point cloud processing. The pioneering work in this field is PointNet [21]. Many following researches [22, 30, 12, 15] propose local operations based on PointNet. The point based methods [27, 33, 20, 31, 25] also have wide applications in 3D object detection. [20, 31] propose a two-stage framework which combines the 2D image detection and the 3D frustum detection. [25] proposes a bottom-up framework based on [22] for 3D bounding box generation.

Compared with the 2D convolution based methods, the 3D convolution based methods are not preferred by practical applications given the implementation and efficiency.

2.2. Intersection over Union

Intersection over Union (IoU) is widely used in many tasks. For the segmentation task, IoU is used for evaluating the accuracy of the pixel-wise or pointwise prediction, which reflects the quality of the learned feature. The cross entropy loss is a widely used loss function for the segmentation task. For alleviating the class-imbalance problem, focal loss [17] is proposed. By learning better on the hard example, the pixel-wise or pointwise IoU is improved. Lovasz-Softmax loss [1] directly optimizes the IoU measure for better performance. For the object detection task, IoU is used as a metric for the quality of the predicted bounding box. In 2D detection, [35] first introduces IoU Loss to replace the smooth L1 loss for the bounding box regression. [23] extends IoU loss to a generalized version. [37] extends 2D IoU loss to 3D object detection. [36, 6] are proposed for easy and clever implementation of 3D IoU, but suffer from the approximation error. Besides the approximation error, most of these methods directly replace the smooth L1 loss with 3D IoU related losses instead of exploring their optimal combination for 3D detection. Also, they only pay attention to the accuracy of the bounding box while ignoring the quality evaluation for the 3D box.

3. RangeIoUDet for 3D Object Detection

In this section, we describe the proposed method RangeIoUDet, a single-stage 3D object detector optimized by the point-based IoU and box-based IoU. In section 3.1, we describe the single-stage model based on the range image as our baseline. In section 3.2, we design a point-

based module to supervise the point-based IoU by Lovasz-Softmax loss, which indirectly enhances the pointwise feature passed to bird's eye view (BEV). In section 3.3, we introduce the 3D Hybrid GIoU loss to optimize the quality of the predicted 3D bounding box. Thanks to the optimization via IoU, the potential of the single-stage model is fully exploited, which achieves state-of-the-art performance in 3D object detection while maintaining a fast inference speed.

3.1. Baseline Model of RangeIoUDet

The range image representation has multiple advantages, especially on the network implementation and inference time. In this section, we describe the design of the range image based single-stage model, which uses the 2D convolution for feature extraction and box regression.

The input of the network is the range image, which is generated by spherical projection of point clouds [19]. The resolution of the range image is determined by the horizontal angular resolution and the number of vertical lasers. For the Velodyne 64E LIDAR, it produces 64 lasers and approximately 2000 points for each laser. Thus, the resolution of the range image is set to 64×2048, and the pixel in the range image and the point in the point cloud are approximately one-to-one. For each pixel, it encodes five channels including the 3D coordinates (x, y, z), range r and intensity e. Particularly, if we only pay attention to the front view of the vehicle whose horizontal field of view is 90 degrees and remove invalid vertical lasers, the size of the range image will be $48 \times 512 \times 5$. The compact representation of the range image dramatically reduces the computation cost.

The range image is input to a 2D fully convolutional neural network, shown in Fig. 2. The output of the 2D FCN is the pixel-wise high-dimensional features. The downsample block and the upsample block share a similar structure. The main difference is that the downsample block uses the average pooling to downsample the output of the block while the upsample block uses the bilinear upsampling to recover the resolution. They both apply a series of dilated convolutions [3] with different dilation rates to extract multi-scale features. Benefiting from the full-resolution output feature and the approximately one-to-one correspondence between the range image and the point clouds, the pointwise feature can be recovered from the pixel-wise feature with minor loss. The pointwise feature has multiple usages. One simple and effective usage is to project it to the x-y plane to generate the BEV feature and then apply the 2D convolution to predict the bounding box based on BEV.

The above single-stage model is extremely fast benefiting from the efficiency of 2D convolution and the compact representation of the range image. However, this model does not achieve a similar performance as state-of-the-art methods. We use it as the baseline model and optimize it by the proposed point-based IoU and box-based IoU.

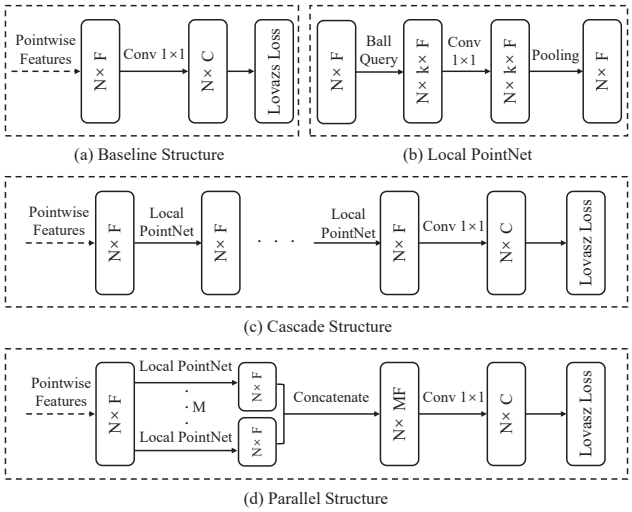

Figure 3. Network structure of the point-based IoU module. (a) Baseline structure with only 1×1 convolution. (b) Local PointNet with ball query, 1×1 convolution, and max pooling. (c) Cascade structure with several local PointNet layers. (d) Parallel structure with multi-scale Local PointNet layers applied in parallel.

3.2. Pointwise Feature Optimized by Lovasz-Softmax Loss

The 2D FCN outputs the pixel-wise feature of the range image, which is further transferred to the point cloud to obtain the pointwise feature. Due to the 2D receptive field in the range image plane, points far away in the 3D space may obtain similar features if they are adjacent in the range image. The pointwise feature is directly passed to the following module without any extra supervision. We argue that the implicit 3D position information encoded in the range image is not fully exploited. We propose to supervise the pointwise feature to make the 2D FCN learn better.

One simple idea is to directly apply a segmentation loss function to the pointwise feature, shown in Fig. 3(a). Directly supervising the pointwise feature of the point cloud is equivalent to supervise the pixel-wise feature of the range image, which does not further utilize the 3D position information of point clouds. In fact, this simple idea can not improve the detection accuracy. We analyze that the main problem is the lack of the 3D receptive field. However, utilizing the 3D receptive field needs to introduce the point based or voxel based convolution, which may slow down the inference speed and increase the difficulty of the deployment. Considering the above factors, we design the point-based IoU module shown at the bottom right of Fig. 2.

To make use of the 3D receptive field of point clouds, we search the local neighbors of each point using ball query and apply PointNet to extract local features (shown in Fig. 3(b)). We choose different radii for achieving multi-scale features. The multi-scale features are extracted in parallel and concatenated pointwisely. Finally, the features extracted in the

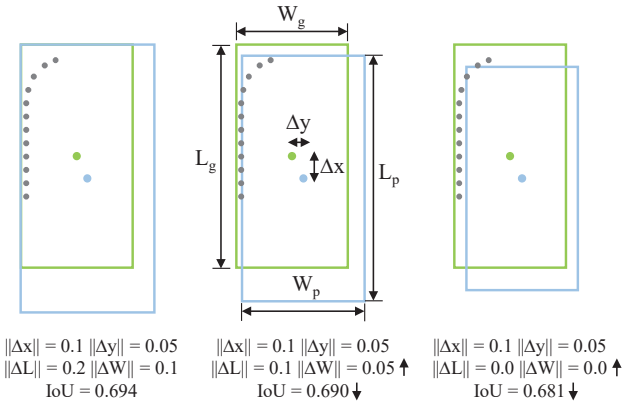

Figure 4. Illustration of the local optimum of smooth L1 loss. The green box is the ground truth. The blue box is the prediction. Under the supervision of Smooth L1 loss, the length L and width W learns better but the IoU becomes worse.

3D space is supervised by Lovasz-Softmax loss [1] to directly optimize the point-based IoU for better distinguishing the foreground and background. The local PointNet refines the pointwise feature which makes the final segmentation result better. Meanwhile, the better supervision promotes the 2D FCN to learn better through back-propagation. As a result, the pointwise feature passed to BEV becomes better even though the local PointNet is not directly applied to it.

When designing this module, we adopt the parallel structure (Fig. 3(d)) instead of the cascade strcuture (Fig. 3(c)) to extract the multi-scale feature. The parallel structure allows the gradient to be faster and more easily backpropagated to the 2D FCN, which leads to the point-based IoU supervision to have a more direct impact on the pointwise feature. The deeper structure may improve the quality of the pointwise segmentation but degrade the detection performance.

Discussion. Compared to the point representation, the range image representation has the drawback of lacking 3D local relationship. Introducing the proposed point-based IoU module in the training stage makes the network aware of the 3D receptive field. Although the point-based IoU module does not directly update the pointwise feature passed to BEV during the forward propagation, its function is indirectly reflected through gradient propagation. Moreover, this module can be ignored if there is no requirement for the segmentation result when applying the inference, which means that it does not slow down the inference speed.

3.3. 3D Bounding Box Optimized by 3D Hybrid GIoU Loss

The performance of 3D object detection is affected by two factors: (1) the positioning accuracy of the 3D bounding box, which is determined by seven parameters $(x, y, z, L, W, H, \theta)$; (2) the confidence score of the box, which has a great influence on the quality evaluation. The positioning accuracy and quality of the predicted box are both highly related to its IoU with the corresponding ground truth, which motivates us to explore the value of the box-based IoU in 3D detection. This section proposes a 3D Hybrid GIoU loss, including hybrid GIoU regression loss and 3D IoU quality loss to improve the above two aspects.

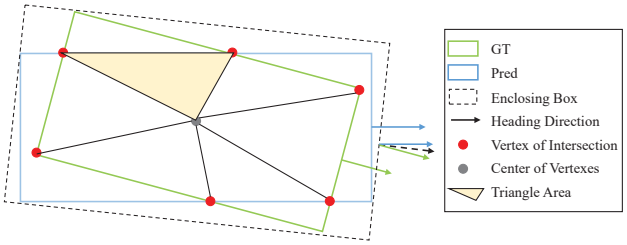

Figure 5. Illustration of calculating 3D GIoU loss. For simplicity, we visualize the example for the 2D rotated box. 3D IoU can be easily achieved by multiplying the height based on 2D rotated IoU.

Positioning Accuracy. The 3D bounding box has seven parameters. Optimizing these parameters independently using the smooth L1 loss may lead to the local optimum. The reason is that the seven parameters are mutually coupled. In Fig. 4, the x and y coordinates remain unchanged. When the length L and the width W become better, the IoU decreases. This motivates us to jointly optimize the seven parameters. Extending the IoU or GIoU loss in 2D detection to the 3D application [37] seems a natural choice. However, extending 2D IoU to 3D has two problems: (1) the calculation of analytic and differentiable 3D IoU is non-trivial, especially when considering the fast and parallel implementation; (2) using only 3D GIoU loss for regression still cannot effectively improve the accuracy of the box, because IoU loss is an indirect loss function for box regression.

For problem (1), we implement a parallelized and differentiable 3D IoU. 3D IoU can be easily achieved based on the 2D rotated IoU by multiplying the height. We briefly introduce the main calculation pipeline of the 2D rotated IoU (shown in Fig. 5): (a) Calculating the vertexes (shown in red in Fig. 5) of the intersection area by determining the valid intersections and corners; (b) Calculating the area of the intersection by sorting the vertexes clockwisely around their center and summing up the triangle areas; (c) Calculating the area of the predicted box and ground truth. Additionally, for 3D GIoU loss, we define the heading angle of the enclosing box [23] as the average of the GT and prediction angles. Such a definition is simple and easy to implement, and allows the enclosing box to better fit the geometry shape compared to the axis-aligned enclosing box. The above operations are implemented in a parallelized manner. The core of parallelization is to use the knowledge of plane geometry to allow most of the operations to be implemented in parallel based on the basic functions in Pytorch with automatic differentiation, avoiding time-consuming loops.

Based on the parallel implementation of calculating the

differentiable 3D IoU (IoU_{3D}) and the enclosing box of the predicted bounding box and the ground truth, we achieve the 3D GIoU loss which is nearly cost-free in runtime. The formula is as follows:

$$\mathcal{L}_{GIoU_{3D}} = 1 - \frac{A^i}{U} + \frac{A^c - U}{A^c} \tag{1}$$

where A^c is the area of the enclosing box [23]. A^i is the area of the intersection. The area of union U is equal to $A^p + A^g - A^i$, where A^p and A^g are the area of the predicted box and ground truth. IoU_{3D} is equal to $\frac{A^i}{U}$.

For problem (2), although 3D GIoU loss can serve as an independent loss for box regression, we find that the smooth L1 loss still has its advantages. In Fig. 4, the reason why a better length L does not lead to a better IoU is that it does not get a more accurate x coordinate. This rule also applied to (y, W) and (z, H). Therefore, predicting an accurate center position (x, y, z) is a prerequisite for obtaining an accurate box. Based on the above observation, we use the smooth L1 loss to directly supervise the position of the center, and use the 3D GIoU Loss to indirectly guide the network to learn the coupling relationship between the seven parameters $(x, y, z, L, W, H, \theta)$, so as to make full use of the advantages of the smooth L1 loss and the 3D GIoU Loss. A direction classifier L_{dir} [32] is used to judge whether the heading angle falls in $[0, \pi)$ or $[\pi, 2\pi)$. In summary, we propose the combination of the smooth L1 loss and 3D GIoU loss with an additional direction classifier to achieve a more global optimization. The formula is as follows:

$$\mathcal{L}_{HyGIoU_{reg}} = \sum_{b \in x,y,z} \mathcal{L}_{smoothL1(\Delta b)} + \mathcal{L}_{GIoU_{3D}} + \alpha \mathcal{L}_{dir} \tag{2}$$

Quality Evaluation. Besides accurate regression, the quality evaluation of the 3D box is also critical to the final result. As the object is usually partial, it does not always provide enough features for predicting an accurate box. Under such circumstances, assigning an accurate quality score for evaluating the quality of the predicted bounding box is very necessary. The quality score can guide the process of Non-Maximum Suppression (NMS) and the calculation of the AP metric. However, most current 3D detection algorithms only use the classification score to determine the quality of the box, which is obviously one-sided. Besides the classification quality, the positioning quality should also be considered, which motivates us to use the 3D IoU calculated in the 3D GIoU loss as the ground truth of the positioning quality score (QS). The formulation is as follows:

$$\mathcal{L}_{HyGIoU_{qs}} = \parallel IoU_{3D} - QS \parallel_2 \tag{3}$$

It should be noted that the ground truth (IoU_{3D}) and the predicted score (QS) are both differentiable, which means that they are jointly optimized. We introduce the positioning quality score to decide the priority of the predicted bounding box, which brings an obvious improvement. By using the differentiable IoU_{3D} as the ground truth, we surprisingly find that even though we do not use the positioning quality score for inference, the AP metric also increases. We analyze in detail in the experiment section.

In summary, the 3D Hybrid GIoU loss consists of the hybrid GIoU regression loss and IoU quality loss. These are both beneficial from the differentiable 3D box-based IoU. The box-based IoU does not only improve the accuracy of the 3D position but also learns a reasonable evaluation for the predicted box. The whole loss function is as follows:

$$\mathcal{L}_{HyGIoU} = \mathcal{L}_{HyGIoU_{reg}} + \mathcal{L}_{HyGIoU_{qs}} \tag{4}$$

3.4. Loss Functions

We utilize the multi-task loss function for jointly optimizing the box classification, box-based IoU, and point-based IoU. The total loss function is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \beta \mathcal{L}_{HyGIoU} + \gamma \mathcal{L}_{point} \tag{5}$$

where $\mathcal{L}_{cls}$ is focal loss with default hyper-parameters. $\mathcal{L}_{HyGIoU}$ is introduced in Sec. 3.3. $\mathcal{L}_{point}$ is Lovasz-Softmax loss for optimizing the IoU of foreground and background points (Sec. 3.2). β and γ is set to 2.0 and 1.0.

4. Experiments

In this section, we first describe the implementation details (Sec. 4.1) of RangeIoUDet. We compare with state-of-the-art methods on the challenging KITTI dataset (Sec. 4.2) and an actual operation scenario dataset (Sec. 4.3). In Sec. 4.4, we conduct extensive ablation studies to validate the effectiveness of different components.

4.1. Implementation Details

Network Details. The whole framework is shown in Fig. 2. The structure of the 2D FCN is illustrated in the figure. The size of the projected BEV feature map is 496×432 with resolution $[0.16m, 0.16m]$, ranging from $[0m, 69.12m]$ for the x axis and $[-39.68m, 39.68m]$ for the y axis. The BEV feature map is downsampled to 248×216, 124×108, 62×54 to extract multi-scale features. The three feature maps are all upsampled to 248×216 and then concatenated. For the local PointNet, we use two scales $[0.4m, 0.8m]$ for searching neighbors.

Training and Inference Details. The proposed RangeIoUDet is optimized with the ADAM optimizer. We train the network with the batch size 32, learning rate 0.01 for 80 epochs on 4 NVIDIA Tesla V100 GPUs, which takes about 2.3 hours for the KITTI dataset and 12 hours for the self-built dataset. A cosine warmup strategy is used in the first 32 epochs with a 0.1 ratio. For the rest epochs, we adopt the cosine annealing learning rate strategy for the learning rate decay. For data augmentation, we use the random flipping along the x axis, random global scaling following the uniform distribution $[0.95, 1.05]$, random global

Method	Reference	Modality	Car			Cyclist			mAP	FPS
			Easy	Moderate	Hard	Easy	Moderate	Hard	Moderate	
Two-stage:										
MV3D [4]	CVPR 2017	RGB+LIDAR	74.97	63.63	54.00	-	-	-	-	2.8
AVOD-FPN [10]	IROS 2017	RGB+LIDAR	83.07	71.76	65.73	63.76	50.55	44.93	61.16	10
F-PointNet [20]	CVPR 2018	RGB+LIDAR	82.19	69.79	60.59	72.27	56.12	49.01	62.96	5.9
F-ConvNet [31]	IROS 2019	RGB+LIDAR	87.36	76.39	66.69	**81.98**	**65.07**	56.54	70.73	-
UberATG-MMF [13]	CVPR 2019	RGB+LIDAR	88.40	77.43	70.22	-	-	-	-	-
EPNet [9]	ECCV 2020	RGB+LIDAR	89.91	79.28	74.59	-	-	-	-	-
PointRCNN [25]	CVPR 2019	LIDAR	86.96	75.64	70.70	74.96	58.82	52.53	67.23	10
Part-A2 [26]	TPAMI 2020	LIDAR	87.81	78.49	73.51	79.17	63.52	56.93	71.01	14
Fast Point R-CNN [5]	ICCV 2019	LIDAR	85.29	77.40	70.24	-	-	-	-	**15.4**
STD [34]	ICCV 2019	LIDAR	87.95	79.71	75.09	78.69	61.59	55.30	70.65	10
PV-RCNN [24]	CVPR 2020	LIDAR	**90.25**	**81.43**	**76.82**	78.60	63.71	**57.65**	**72.57**	10
Single-stage:										
SECOND [32]	Sensors 2018	LIDAR	83.34	72.55	65.82	71.33	52.08	45.83	62.32	20
PointPillars [11]	CVPR 2019	LIDAR	82.58	74.31	68.99	77.10	58.65	51.92	66.48	**62**
Point-GNN [27]	CVPR 2020	LIDAR	88.33	79.47	72.29	78.60	63.48	57.08	71.48	1.6
3D-SSD [33]	CVPR 2020	LIDAR	88.36	79.57	74.55	82.48	64.10	56.90	71.84	26
SA-SSD [8]	CVPR 2020	LIDAR	**88.75**	79.79	74.16	-	-	-	-	25
RangeIoUDet (ours)	-	LIDAR	88.60	**79.80**	**76.76**	**83.12**	**67.77**	**60.26**	**73.79**	45

Table 1. The average precision (AP) with 40 recall positions ($R40$) of 3D object detection on the KITTI test set.

rotation around the z axis following the uniform distribution $[-\frac{\pi}{4}, \frac{\pi}{4}]$ and the ground truth sampling [32].

For the post-processing stage, we keep the predicted boxes whose confidence scores are higher than 0.2. The threshold of NMS is set to 0.1 to remove the redundant boxes. The model runs at 45 FPS on a NVIDIA Tesla V100.

4.2. 3D Detection On the KITTI Dataset

KITTI dataset [7] is one of the most popular datasets for 3D object detection. It contains 7481 training samples and 7518 test samples. We follow the general split of 3712 training samples and 3769 validation samples. Table 1 shows the performance of RangeIoUDet on the test set. The AP of the moderate level is the most important metric which is chosen as the ranking basis on the benchmark. On the moderate level of Car, our method outperforms all previous single-stage and two-stage methods except PV-RCNN. On the hard level of Car, our method achieves almost the same accuracy as the top two-stage model PV-RCNN. We analyze that the compact and dense representation of the range image provides rich context information for hard examples, which alleviates the sparse issue of distant objects. While achieving high performance, our method is more than four times faster than PV-RCNN.

In order to further explore this advantage, we also train the model on the cyclist category. RangeIoUDet achieves the best performance. Besides the hard car examples, the dense representation of the range image also makes it easy to extract rich context information for small objects. Compared to state-of-the-art methods, the proposed RangeIoUDet has advantages both in accuracy and efficiency. More importantly, RangeIoUDet is a model based on 2D convolution, making it easy to implement and deploy.

In the field of the range image based detection, LaserNet [18] is a forward-looking work that points out the potential of the range image. It reports the result (74.52%) of BEV detection on the KITTI dataset. Our method outperforms it by a large margin (88.59%), which proves the effectiveness of the range image on the small-scale dataset.

4.3. 3D Detection On the Actual Operation Dataset

To show the generalization on different LIDAR sensors and the performance in actual scenarios, we use the self-built dataset to investigate the proposed RangeIoUDet. The LIDAR used in this dataset is Pandar40P, which generates 40 lasers, each with about 1800 points. Pandar40P provides the laser id on the driver, which makes the range image more accurate. In order to match the above network structure, we use the zero padding operation to extend the range image for the front view to the size of $48 \times 512 \times 5$.

The dataset is collected on the city road. It contains 18,000 training samples, 2,000 validation samples, and 3,000 testing samples. This section compares the proposed RangeIoUDet with PointPillars[11], SECOND[32], and PV-RCNN[25]. The pixel height of the bounding box on the 2D image is used to measure the difficulty. The categories Car reflects the detection performance for large objects, and Pedestrian and Cyclist reflect the detection performance when the number of point clouds is scarce.

Table 2 shows the comparison of the above methods. In the Car category, RangeIoUDet outperforms the other three methods. In the Pedestrian category, RangeIoUDet is weaker than PV-RCNN but outperforms the other two single-stage methods. We think that the two-stage refinement of PV-RCNN focuses on a local region, which can better learn the feature for small pedestrians. In the Cyclist category, the performance of RangeIoUDet is better than that of the other three methods. The above results prove that (1) even if the LIDAR is changed, the proposed method still has good detection performance, showing its good versatility,

Method	Car $AP_{0.7}$		Pedestrian $AP_{0.5}$		Cyclist $AP_{0.5}$	
	Height>40	Height>25	Height>40	Height>25	Height>40	Height>25
PointPillars[11]	96.57	95.37	50.31	40.03	73.26	66.58
SECOND[32]	98.54	95.46	51.50	42.52	76.67	70.80
PV-RCNN[25]	98.81	95.48	**56.21**	**45.34**	78.34	71.33
RangeIoUDet (ours)	**99.09**	**96.27**	53.38	44.83	**79.65**	**72.81**

Table 2. The average precision (AP) with 40 recall positions ($R40$) of 3D object detection on the test set of the actual self-built dataset. The height is used to measure the difficulty level.

Box-based IoU Loss	Confidence Score	Easy	Moderate	Hard
Baseline (SL)	Class	87.79	79.26	76.17
3D IoU	Class	88.49	80.57	77.31
3D GIoU	Class	89.61	80.97	78.01
HyGIoU_{reg}	Class	90.58	81.40	78.42
HyGIoU_{reg} + HyGIoU_{qs}	Class	90.59	81.72	78.71
HyGIoU_{reg} + HyGIoU_{qs}	Class * QS	**91.10**	**82.42**	**79.05**

Table 3. Comparision of different loss functions for the 3D bounding box. The AP with 40 recall positions (R40) is used.

Point-based IoU Module	Car IoU	Easy	Moderate	Hard
×	×	91.10	82.42	79.05
Fully Connected Layer	63.92	89.82	82.34	78.01
Local PointNet-SingleScale	70.58	91.58	82.99	79.36
Local PointNet-Cascade	**73.35**	91.48	82.77	79.75
Local PointNet-Parallel	72.72	**91.92**	**83.19**	**80.10**

Table 4. Comparision of different structures of the point-based IoU module. The point-based IoU for the Car category is provided.

(2) when an accurate scan id can be obtained, RangeIoUDet method will have better performance, even surpassing the current SOTA method PV-RCNN in some categories, (3) the proposed RangeIoUDet method also has good detection performance for the small and complex types.

4.4. Ablation Study

In this section, we conduct ablation studies to analyze the effectiveness of different components. We do the following experiments on the validation set of KITTI for the car class.

Effectiveness of the Box-based IoU Loss. In this paper, the 3D Hybrid GIoU loss is proposed to improve the location accuracy and quality evaluation of the 3D bounding box. We thoroughly investigate the effects of different components in this loss function. We use the smooth L1 loss as the baseline (Sec. 3.1). 3D IoU loss improves the quality of the 3D box by jointly optimization, and the 3D GIoU loss improves it by introducing the enclosing box with the average heading angle. Compared to the sole 3D GIoU loss, using the proposed hybrid regression ($HyGIoU_{reg}$) achieves a higher performance (4_{th} row of Table. 3), which verifies the effectiveness of our analysis and design. Surprisingly, adding the supervision for the quality score (QS) promotes the learning of the network (5_{th} row of Table. 3), even though we do not use the quality score for the confidence score. Finally, we set the confidence score to the product of the classification score and the quality score, which achieves the highest performance (6_{th} row of Table. 3).

Effectiveness of the Point-based IoU Module. We investigate the influence of different structures of the point-based IoU supervision based on the best performance in Table 3 (6_{th} row). Directly supervising the pointwise feature after a fully connected layer (Fig. 3(a)) degrades the performance (2_{nd} row of Table. 4) compared with the model without the point-based supervision (1_{st} row of Table. 4). Utilizing the 3D receptive field by local PointNet (Fig. 3(b)) can contribute to the 3D detection result. Using single-scale local PointNet already achieves a good improvement (3_{rd} row of Table. 4), while the multi-scale parallel structure (Fig. 3(d)) further improves the result (5_{th} row of Table. 4). The cascade structure does not lead to further improvement. The reason is that the function of the point-based module is to indirectly supervise the pointwise feature instead of directly updating it in the forward propagation. A deeper network will make the supervision have a weaker impact on the pointwise feature passed to BEV.

Accuracy of Semantic Segmentation. Although our network is designed for 3D object detection and the pointwise segmentation is ignored during the inference, we analyze the pointwise IoU for better understanding the result. The IoU of the car category is shown in Table 4. After introducing the local PointNet, the pointwise IoU improves significantly, from 63.92 to 70.58. Especially, the cascade structure performs better than the parallel structure for the pointwise IoU, but worse for the 3D detection result, which confirms the rationality of our network design.

5. Conclusion

We have presented RangeIoUDet, an efficient and accurate single-stage 3D object detector based on the range image. By optimizing the point-based IoU and the box-based IoU for the pointwise feature and the 3D bounding box respectively, the potential of the range image based single-stage model is well exploited. Benefiting from the compact representation of the range image and the efficiency of 2D convolution, our method runs at real-time frame rates. Experiments on the KITTI dataset and the actual operation dataset show the effectiveness and generalization on different LIDAR sensors and object categories. RangeIoUDet is simple to construct and can leverage vast design experience of mature image-based network structures, which makes it easy to be practically applied and continuously improved.

References

[1] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 2, 3, 5

[2] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection. *arXiv preprint arXiv:2005.09927*, 2020. 1, 2

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 4

[4] 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)*, pages 6526–6534, 2017. 1, 2, 7

[5] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 7

[6] Zhiming Chen, Kean Chen, Weiyao Lin, John See, Hui Yu, Yan Ke, and Cong Yang. Piou loss: Towards accurate oriented object detection in complex environments. *arXiv preprint arXiv:2007.09584*, 2020. 3

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 7

[8] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 1, 3, 7

[9] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 7

[10] 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)*, pages 1–8, 2018. 2, 7

[11] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, June 2019. 2, 7, 8

[12] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. 3

[13] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7345–7353, June 2019. 2, 7

[14] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 2

[15] Zhidong Liang, Ming Yang, Liuyuan Deng, Chunxiang Wang, and Bing Wang. Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8152–8158. IEEE, 2019. 3

[16] Zhidong Liang, Ming Zhang, Zehan Zhang, Xian Zhao, and Shiliang Pu. Rangercnn: Towards fast and accurate 3d object detection with range image representation. *arXiv preprint arXiv:2009.00206*, 2020. 2

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3

[18] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12677–12686, June 2019. 1, 2, 7

[19] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 2, 4

[20] 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*, pages 918–927, 2018. 1, 3, 7

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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. 3

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas 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. 3

[23] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 2, 3, 5, 6

[24] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1, 3, 7

[25] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object progposal generation and detection from point

cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1, 3, 7, 8

[26] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 3, 7

[27] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1711–1719, 2020. 1, 3, 7

[28] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2

[29] Yue Wang, Alireza Fathi, Abhijit Kundu, David Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. *arXiv preprint arXiv:2007.10323*, 2020. 1, 2

[30] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 3

[31] Z. Wang and K. Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749, 2019. 1, 3, 7

[32] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 3, 6, 7, 8

[33] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 1, 3, 7

[34] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019. 7

[35] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016. 2, 3

[36] Yu Zheng, Danyang Zhang, Sinan Xie, Jiwen Lu, and Jie Zhou. Rotation-robust intersection over union for 3d object detection. 3

[37] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pages 85–94, 2019. 3, 5

[38] 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*, pages 4490–4499, 2018. 1, 2