Rotation-robust Intersection over Union for 3D Object Detection

Yu Zheng[1,2,3], Danyang Zhang[1], Sinan Xie[1], Jiwen Lu[1,2,3] ⋆, and Jie Zhou[1,2,3,4]

[1] Department of Automation, Tsinghua University, China
[2] State Key Lab of Intelligent Technologies and Systems, China
[3] Beijing National Research Center for Information Science and Technology, China
[4] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
{zhengyu19,zhang-dy16,xsn18}@mails.tsinghua.edu.cn
{lujiwen,jzhou}@tsinghua.edu.cn

Abstract. In this paper, we propose a Rotation-robust Intersection over Union (*RIoU*) for 3D object detection, which aims to learn the overlap of rotated bounding boxes. In most existing 3D object detection methods, the norm-based loss is adopted to individually regress the parameters of bounding boxes, which may suffer from the loss-metric mismatch due to the scaling problem. Motivated by the IoU loss in the axis-aligned 2D object detection which is invariant to the scale, our method jointly optimizes the parameters via the *RIoU* loss. To tackle the uncertainty of convex caused by rotation, a projection operation is defined to estimate the intersection area. The calculation process of *RIoU* and its loss function is robust to the rotation condition and feasible for back-propagation, which only comprises basic numerical operations. By incorporating the *RIoU* loss with the conventional norm-based loss function, we enforce the network to directly optimize the *RIoU*. Experimental results on the KITTI, nuScenes and SUN RGB-D datasets validate the effectiveness of our proposed method. Moreover, we show that our method is suitable for the detection task of 2D rotated objects, such as text boxes and cluttered targets in the aerial images.

Keywords: 3D Object Detection, Loss Function, Rotation-robust

1 Introduction

Recent years have witnessed the advances in 2D object detection [11, 36, 25] along with the breakthrough of deep learning methods. However, detection of 3D objects, such as outdoor vehicles and pedestrians [10, 1], remains a challenging issue. The detection algorithms are designed to regress the translation, scale and yaw angle of the bounding boxes. Compared to the axis-aligned 2D targets, more attributes of 3D object are obtained attributed to the sufficient spatial information provided by the mounted lidar scanners [1]. Directly consuming lidar points as the detection input has drawn more attention recently [22, 51, 38].

⋆ Corresponding author

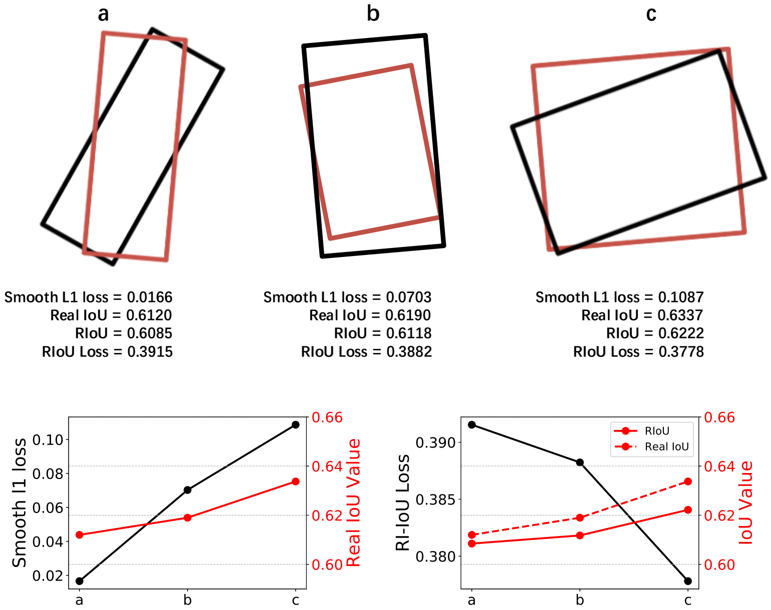


Fig. 1: Examples of inconsistency between the smooth-ℓ1 loss and real IoU. From left to right, the ℓ1 difference of bounding box parameters is becoming larger, while the overlap is becoming closer. The consistency is preserved between our proposed $RIoU$ loss and real IoU in this example. The result is calculated from numerical simulation.(*best viewed in color pdf file*)

State-of-the-art 3D object detection methods project the point cloud to a certain viewpoint [7, 49] for convolutional feature extraction in the 2D plane, or voxelize the point cloud and apply 3D convolution [29, 51]. Recently two-stage RCNN methods [38, 8] are proposed to better leverage the point-wise information. Compared to the monocular [4, 2, 43, 27, 20] or stereo [6, 24, 23, 35] methods, the attributes of 3D object are obtained from the lidar points with fewer stages, which makes it possible for real-time detection usage.

To regress the parameters of rotated bounding boxes, existing approaches of 3D object detection regress the translation, scale and yaw angle individually by using the smooth-ℓ1 loss [11, 36], which is based on the ℓ1-norm of parameter distance. While each parameter (i.e., height) might be normalized by the anchor parameters [51, 21], the size of the anchor is a pre-defined scalar. Therefore, the value of the ℓ1-norm is still sensitive to the scale of the bounding box [37]. As shown in Fig. 1, the conventional loss and evaluation metric are inconsistent. Addressing this loss-metric mismatch could provide insights into computer vision and machine learning tasks such as object detection [37] and metric learning [13].

In order to learn the bounding box parameters collaboratively as well as avoiding the scaling problem, directly optimizing Intersection over Union (IoU) is addressed in the axis-aligned cases [47, 18, 14, 37]. Such attempts significantly enhance the performance in the axis-aligned 2D object detection. However, due to the variance in shape, pose and environment condition, object targets are hardly axis-aligned, especially for 3D objects. Geometrically, the intersection area between a pair of rotated bounding boxes is non-trivial to calculate by using the numerical methods. In bird's eye view, as shown in the upper part of Fig. 1, the shape of the intersected convex is diverse due to the variance of location, size and angle. Currently, the accurate convex area between a pair of rotated bounding boxes is often calculated outside the training loop, or regarded as a constant [46] and not involved in the gradient descent of the back propagation. To tackle the aforementioned problem brought by rotation, some methods directly estimate the confident score of IoU by using deep networks [15] or calculate a simplified version of IoU [26] to select positive samples. But neither of them tries to directly learn on the overlap of the bounding boxes, and few similar attempts have been made in 3D object detection.

In this paper, we propose an IoU for 3D object detection called Rotation-robust Intersection over Union ($RIoU$) with its loss function format ($\mathcal{L}_{RIoU}$), and incorporate it into the conventional $\ell 1$ loss. Specifically, we define a pair of projected rectangles to calculate the intersection in the 2D plane. It is suitable for bounding box regression in arbitrary angles. Besides, it only comprises basic arithmetic operations and the $\min / \max$ function, which is feasible for back propagation during training. We also extend $RIoU$ to the volume and recent Generalized Intersection over Union [37] format. Experimental results on the KITTI [10] and nuScenes [1] datasets show that combined with our $\mathcal{L}_{RIoU}$, the performance of 3D object detection is improved by a large margin. Moreover, we test our method on the 2D rotated object detection to validate its applicability.

2 Related Work

Point-based 3D Object Detection: While 3D oriented objects can be detected from monocular [4, 2, 43, 27, 20] or stereo [6, 24, 23, 35] images, the spatial information is better preserved in point cloud data collected by the lidar scanners. It provides multiple projection viewpoints for feature aggregation [7]. Most state-of-the-art approaches consume raw lidar data as input. Early works directly apply 3D convolution to process the point cloud [9, 22]. Several methods group point cloud into stacked 3D voxels [29, 51] to generate more structured data, and [21] restricts the grouping operation within the ground plane to achieve real time detection. As for two-stage pipelines, some methods adopt detection results of 2D images to crop ROI regions in the 3D space [32, 44, 42], or fuse the image and point cloud feature to reduce the missing instances in the first stage [19]. Recently proposed RCNN methods [38, 8] adopt PointNet-based [33] module for better extracting and aggregating the point-wise feature. Similar to object detection in 2D images, all those methods adopt the $\ell 1$ regression loss [11, 36], which focuses on the difference of individual bounding box parameters.

Intersection over Union: Intersection over Union (IoU) is widely adopted as the evaluation metric in many visual tasks, such as object detection [11, 36, 25, 5], segmentation [33, 34, 3] and visual tracking [30]. Generally, it is calculated outside the training loop and not involved in the process of back-propagation. For example, it is adopted as a metric to discriminate between the positive and negative samples [18, 51]. Attempts towards directly learning IoU have been made in the scenario of axis-aligned 2D object detection, since IoU is invariant to the scale of the problem [37]. Instead of calculating IoU between the detected and ground-truth bounding boxes, [15] predicts the IoU as a metric in non-maximum suppression (NMS). IoU loss is adopted for axis-aligned face detection [47], visual tracking [18] and lumbar region localization [14]. [41] designs Intersection over Ground-truth (IoG) to penalize the wrongly matched detections. [37] proposes Generalized IoU (GIoU) to penalize poor detection instances. In the context of rotation-free bounding box regression, [50, 28] optimize axis-aligned IoU loss and angle loss seperately for text localization. [26] proposes a surrogate IoU, called Angle related IoU (ArIoU), to select prior boxes as positive samples for aerial image detection. More recently, [27] proposes *FQNet* to directly predict 3D IoU between samples and objects in monocular data, which is similar to [15]. The accurate 3D IoU loss is firstly proposed in [48], where the intersection can be calculated through traversing the vertices of the overlap area, which requires the sophisticated design of the forward and backward computation. It demonstrates superior performance over the $\ell1$-based loss function. Another alternative is to regard the accurate IoU as a constant coefficient [46], where the calculation of IoU is not involved in the back propagation of the training process.

3 Approach

In this section, we formulate the proposed approach in the bird's eye view of 3D space, which corresponds to the general 2D cases and can be easily extended to the 3D cuboid formulation.

3.1 Rotation-robust Intersection over Union

In the 3D space, a rotated bounding box B is defined by (x, y, z, l, h, w, r), where (x, y, z), (l, h, w) and r represent center coordinates, box size and rotation around z axis (yaw) respectively. As shown in (1), Intersection over Union (IoU) is calculated as an evaluation metric in object detection task generally.

$$IoU = \frac{B_1 \cap B_2}{B_1 + B_2 - B_1 \cap B_2}. \tag{1}$$

It is introduced as an optimized target in several conventional 2D cases [40, 37]. However, when extended to 3D or other cases, where the bounding box is rotatable, the calculation of the intersection area becomes non-trivial. The shape of the intersection polygon depends largely on the location, size and yaw angle

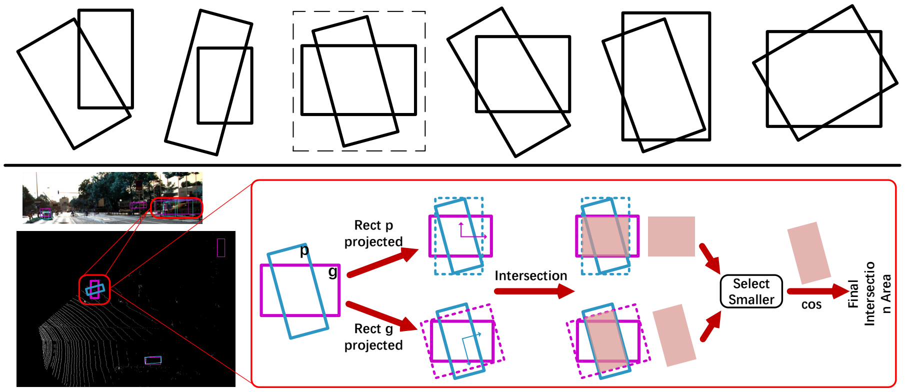


Fig. 2: **Upper**: Some of the overlapped situations between two rotated bounding boxes. **Bottom**: The calculation of Rotation-robust Intersection of rotated bounding box p and g corresponding to one of the upper situations, which is sampled from the KITTI [10] dataset. The projected rectangle $p^{'}$ (dotted) is defined in the canonical coordinate system built around g. And vice versa for $g^{'}$. The smaller of $p^{'} \cap g$ and $g^{'} \cap p$ is preserved and multiplied by a cosine coefficient to get the final intersection, which is the numerator of *RIoU(p,g)*. (*best viewed in color pdf file*)

of 2 bounding boxes. As shown in the upper part of Fig. 2, the polygon can be triangle, parallelogram, trapezoid or even pentagon, etc.

In bird's eye view or general 2D cases, a rotated bounding box is defined by (x, y, l, w, r). With the shape of 2×4, the coordinates of the bounding box corners C can be obtained from the parameters above. Given a pair of predicted bounding box $B_p(C_p)$ and ground-truth bounding box $B_g(C_g)$, a canonical coordinate system is firstly built around the center of B_g. In this system, B_g is axis-aligned and can be represented as $(0, 0, l_g, w_g, 0)$. Then we define a projected rectangle of B_p by satisfying the following properties: Firstly, B_p is inside the projected rectangle. Secondly, The projected rectangle is aligned to the axes of the new canonical coordinate system, i.e., the axes of B_g. Thirdly, The area of the projected rectangle is minimum. An example of defining such rectangles is shown in the bottom part of Fig. 2. As shown in (2), we first calculate the corner coordinates of B_p in the canonical coordinate system of B_g as $C_{p,a}$, including shifting the origin and rotating the axes. Here we denote the coordinate of i-th corner of predicted box as C_p^i. "$*$" denotes the matrix multiplication.

$$\begin{aligned} C_p^i &= C_p^i - [x_g, y_g]^T, i \in 1, 2, 3, 4 \\ C_{p,a} &= \begin{bmatrix} \cos r_g & \sin r_g \\ -\sin r_g & \cos r_g \end{bmatrix} * C_p. \end{aligned} \tag{2}$$

In the canonical coordinate system of B_g, the corners of the projected rectangle can be easily defined by extracting the min / max coordinate value of corners.

For example, $C^1_{p'}$, the top left coordinate of $B_{p'}$ is extracted as follows:

$$C^1_{p'} = [\min_x C_{p,a}, \max_y C_{p,a}]^T \tag{3}$$

$B_{p'}$ is axis-aligned in the canonical coordinate system of B_g. It is easy to calculate a pair of $(I1, Un1)$ from $B_{p'}$ and B_g, which corresponds to the area of intersection and the smallest enclosing box of $B_{p'}$ and B_g.

By swapping B_g and B_p and then repeating (2) and (3), we can get another pair of $(I2, Un2)$. The smaller intersection value is preserved and multiplied by a cosine coefficient to get the final intersection area. The final intersection and union area is calculated as follows:

$$\begin{aligned} I_{RIoU} &= \min(I1, I2) \cdot |\cos(2 \cdot (r_g - r_p))|, \\ U_{RIoU} &= \max(I_{RIoU}, l_g \cdot w_g + l_p \cdot w_p - I_{RIoU}), \\ RIoU &= \frac{I_{RIoU}}{U_{RIoU}}. \end{aligned} \tag{4}$$

As shown in bottom part of Fig. 2, the area of $p^{'} \cap g$ $(I1)$ or $g^{'} \cap p$ $(I2)$ is larger than $g \cap p$. To remedy this error, we preserve the minimum area of $p^{'} \cap g$ and $g^{'} \cap p$. Besides, we set the angle coefficient of cosine as 2. Therefore, the calculated area decreases more sharply as the angle difference becomes larger. Note that RIoU equals zero when the angle difference is 45 degree. However, the partial derivative of $\mathcal{L}_{RIoU}$ with regard to all parameters except the rotation angle is zero, which pushes the angle difference down to zero. And our $RIoU$ degrades to the conventional axis-aligned IoU when the boxes are parallel or orthogonal. We choose the cosine function in (4) based on its following properties: It decreases as the angle deviates from zero, which penalizes the fluctuation of angle difference. Moreover, it is periodic, which corresponds to the periodic orientation of objects. Please refer to the supplementary pages for more design details.

Apart from $RIoU$, we also implement its Generalized Intersection over Union ($GIoU$) format proposed in [37]. It is designed specifically to reveal and penalize the low intersection between 2 bounding boxes:

$$\begin{aligned} Un_{RIoU} &= \max(Un1, Un2), \\ RGIoU &= RIoU - \frac{Un_{RIoU} - U_{RIoU}}{Un_{RIoU}}. \end{aligned} \tag{5}$$

The complete process of calculation is summarized in Algorithm 1. The algorithm only comprises basic arithmetic operations and the min / max function, which is feasible for back propagation during training.

As the $RIoU$ above is implemented in bird's eye view, it can be easily extended to 3D IoU format by introducing another axis-aligned z dimension, as shown in (6). Here we denote the upper/lower z coordinate of ground-truth and predicted bounding box as $z_{g,u}/z_{g,l}$ and $z_{p,u}/z_{p,l}$. But in the experiment section we show that, the incorporation of $\mathcal{L}_{RIoU}$ implemented in the 2D plane enhances

Algorithm 1: *RIoU*, *RGIoU* and their loss function.

```
input  : Bounding box $B_g$, $B_p$ and their corners.
output: $RIoU$, $RGIoU$, $\mathcal{L}_{RIoU}$, $\mathcal{L}_{RGIoU}$.
 Function Project($B_1, B_2$):
    In the canonical coordinate system of $B_1$, set new the origin and corner
    coordinates for $B_2$ using (2);
    Locate $B_2^{'}$, the projected rectangle of $B_2$ using (3);
    Calculate $Intersection$ $B_1 \cap B_2^{'}$ as I;
    Calculate $Universal$ of $B_1$ and $B_2^{'}$ as Un;
    return $I, Un$
1 Calculate 2 pairs of intersection and universal:
    $I1, Un1 =$ Project $(B_g, B_p)$;
    $I2, Un2 =$ Project $(B_p, B_g)$;
2 Calculate $RIoU$ and $RGIoU$ using (4) and (5);
3 $\mathcal{L}_{RIoU} = 1 - RIoU$, $\mathcal{L}_{RGIoU} = 1 - RGIoU$.
```

the performance of both 2D and 3D target detection.

$$\begin{aligned} \delta z &= \min(z_{g,u}, z_{p,u}) - \max(z_{g,l}, z_{p,l}) \\ RIoU(v) &= \max(0, \delta z) \cdot RIoU \end{aligned} \tag{6}$$

Our $\mathcal{L}_{RIoU}$ and $\mathcal{L}_{RGIoU}$ is bounded in terms of stability. The final intersection area of *RIoU* is the minimum value of a subset of p and a subset of g. Hence I_{RIoU} is always smaller than the area of p or g, thus smaller than U_{RIoU} in (4). Therefore, the *RIoU* and $\mathcal{L}_{RIoU}$ are both bounded in $[0, 1]$. As for *RGIoU* and $\mathcal{L}_{RGIoU}$, since Un_{RIoU} is always larger than U_{RIoU} in (4), *RGIoU* is bounded in $[-1, 1]$. And $\mathcal{L}_{RGIoU}$ is bounded in $[0, 2]$.

3.2 Discussion

Comparison of RIoU loss and the smooth-ℓ1 loss. When rotation is introduced in the detection task, the smooth-ℓ1 loss [11, 36] is often adopted as the regression target, which focuses on the element-wise difference of the bounding box parameters. A typical set of element-wise difference in 3D cases is defined in (7) [51, 21], where *gt* and *a* denote the parameter of a ground-truth bounding box and its matched anchor box respectively.

$$\begin{aligned} &\Delta x = \frac{x^{gt} - x^a}{d^a}, \Delta y = \frac{y^{gt} - y^a}{d^a}, \Delta z = \frac{z^{gt} - z^a}{h^a}, \\ &\Delta w = \log \frac{w^{gt}}{w^a}, \Delta l = \log \frac{l^{gt}}{l^a}, \Delta h = \log \frac{h^{gt}}{h^a}, \\ &\Delta \theta = \sin\left(\theta^{gt} - \theta^a\right). \end{aligned} \tag{7}$$

Smooth-ℓ1 loss shares with ℓ1-norm difference the drawback demonstrated in [37]. As each parameter is optimized independently, smaller parameter difference can not guarantee bigger IoU (see Fig. 1). The normalization brought by the pre-defined anchor parameters d_a and h_a is not fully effective, because the size parameters of anchors are usually pre-defined scalars. Consequently, the $\ell 1$ difference is sensitive to the scale of the bounding box. Therefore, an approximate function that directly optimizes IoU may further enhance the performance.

Comparison of RIoU and other IoUs. Our RIoU shares the non-negativity, identity of indiscernibles and commutativity with the accurate IoU. The closest to our work is the angle-related IoU ($ArIoU$) proposed in [26]:

$$ArIoU_{180}(A, B) = \frac{area(\hat{A} \cap B)}{area(\hat{A} \cup B)} \cdot |cos(\theta_A - \theta_B)|. \tag{8}$$

$ArIoU$ is used for selecting positive anchors at the training period. $\hat{A}$ shares the parameters with A except that its rotation is the same with B. $RIoU$ and $ArIoU$ both take the angle difference into consideration. But $ArIoU$ is a non-commucative function, which means $ArIoU(A, B) \neq ArIoU(B, A)$. Besides, the area of $\hat{A} \cup B$ might be smaller than $A \cup B$. The cosine part further decays the intersection area, which makes the estimation even worse.

Recently [48] proposes to replace the $\ell 1$ loss with the accurate IoU loss function, which does not suffer from the approximation error. The forward computation of IoU and the backward propagation of the error are firstly implemented manually in this work. However, our proposed $RIoU$ can be easily implemented into the existing framework, and does not require the traversal of the vertices.

4 Experiment

To evaluate our $RIoU$ loss and its variant for rotated 3D object detection, we used the popular KITTI dataset [10] and the newly proposed challenging nuScenes [1] dataset. We plugged $\mathcal{L}_{RIoU}$ and $\mathcal{L}_{RGIoU}$ into the loss function of Frustum-PointNet v1 [32, 33], Frustum-PointNet v2 [32, 34], PointPillars [21] and VoteNet [31]. The weights for the $\ell 1$ loss and the proposed loss are the same. Here we denote the raw $\ell 1$ based regression function baseline as $\ell 1$, baseline incorporated with our proposed $\mathcal{L}_{RIoU}$ or $\mathcal{L}_{RGIoU}$ as $\ell 1 + iou$ or $\ell 1 + giou$.

4.1 Datasets and Settings

KITTI: The KITTI dataset [10] contains 7481 training and 7518 testing samples for 3D object detection benchmark. The evaluation is classified into Easy, Moderate or Hard according to the object size, occlusion and truncation. We followed [7] to split the training set into 3712 training samples and 3769 validation samples. As for input modality of the KITTI dataset, We took raw point cloud [21] and fusion [32] into consideration.

We reported the experiment results on the KITTI validation set. The evaluation took 3D average precision as the metric. The threshold for car, pedestrian and cyclist is 0.7, 0.5 and 0.5 respectively.

Table 1: Comparisons of different loss settings on car category of the KITTI validation set. The baseline model is Frustum-PointNet [32] (F-PointNet). The backbone network is PointNet [33](v1) and PointNet++ [34](v2) respectively. The results are reported on the task of **3D object localization** (Loc.) and **3D object detection** (Det.).

Method		Easy		Moderate		Hard	
		v1	v2	v1	v2	v1	v2
Loc.	ℓ1(baseline)	87.67	88.16	82.68	84.02	74.74	76.44
	ℓ1+ArIoU	**88.36**	88.17	82.81	84.12	74.23	76.39
	ℓ1+$RIoU$	88.01	88.56	82.83	**85.07**	75.45	**77.02**
	ℓ1+$RGIoU$	88.10	**88.67**	**82.93**	84.83	**75.65**	76.81
Det.	ℓ1(baseline)	83.47	83.76	69.52	70.92	62.86	63.65
	ℓ1+ARIoU	83.00	84.07	68.94	71.12	60.96	63.71
	ℓ1+$RIoU$	84.45	**84.83**	71.20	**72.13**	63.61	**64.35**
	ℓ1+$RGIoU$	**84.72**	84.10	**71.46**	71.71	**63.75**	64.00

NuScenes: The nuScenes dataset [1] contains 1k scenes, 1.4M camera images, 400k LIDAR sweeps, 1.4M RADAR sweeps and 40k key frames, which has 7x as many annotations as the KITTI dataset. Each key frame is annotated with 35 3D boxes on average, which is 2.6x as many as the KITTI dataset. The annotation of each instance comprises semantic category, parameters of 3D bounding box, velocity and attribute (parked, stopped, moving, etc.). Each scene is captured continuously for 20 seconds. The whole dataset is categorized into 23 semantic categories (car, pedestrian, truck, etc.) and 8 attributes.

For the nuScenes 3D object detection evaluation, we followed the evaluation protocol proposed along with the dataset [1]. While the average precision (AP) was calculated as final metric, 2D box center distance on the ground plane instead of IoU was used as threshold. We also evaluated the result following the KITTI protocol. The instance was regarded as easy, moderate or hard according to the number of points inside the bounding box.

SUN RGB-D: The challenging SUN RGB-D[39] dataset for scene understanding contains 10k RGB-D images, 5,285 for training and 5,050 for testing. It's densely annotated with 64k oriented 3D bounding boxes. The whloe dataset is categorized into 37 indoor object classes(bed,chair,desk,etc.). The standard evaluation protocol reports the performance on the 10 most common categories.

4.2 Results and Analysis

Frustum-PointNet: The training process took 200 epochs. We used the 2D object detections of the training and validation set provided by the authors[5]. As shown in Table 1, when the backbone network of F-PointNet is PointNet [33], $\mathcal{L}_{RIoU}$ outperforms these compared methods in both 2D localization and 3D

[5] github.com/charlesq34/frustum-pointnets/

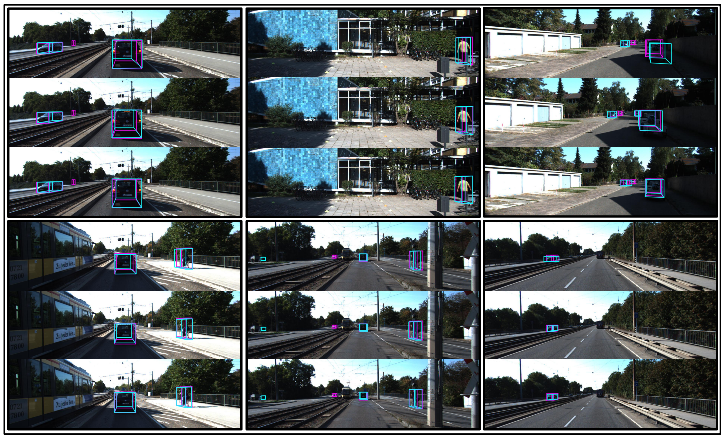


Fig. 3: Visualization of 6 predicted samples on the KITTI validation set (predictions in blue, ground-truths in pink). In each sample, we compare the baseline (**upper**) with the incorporation of $\mathcal{L}_{RIoU}$ (**middle**) or $\mathcal{L}_{RGIoU}$ (**lower**).

Table 2: Comparisons of different loss settings for **3D object detection** and **3D object localization** on car category of the KITTI validation set. The baseline model is PointPillars [21].

Method	3D detection			3D localization		
	Easy	Moderate	Hard	Easy	Moderate	Hard
$\ell 1$(baseline)	87.29	76.99	70.84	89.92	87.88	86.72
IoU [48]	87.88	**77.92**	**75.70**	90.21	**88.25**	**87.56**
$\ell 1$+$RIoU$	**88.02**	77.37	74.04	**90.45**	87.98	85.83

detection. The $\mathcal{L}_{RGIoU}$ further improves the detection performances. This is because the predicted boxes are generated from anchors in different sizes and angles but the same center, which could not guarantee the positive match between anchor and ground-truth. Then those negative matches are penalized more by $\mathcal{L}_{RGIoU}$ than $\mathcal{L}_{RIoU}$. For the experimental results of the *cyclist* and *pedestrian* categories, please refer to the supplementary pages.

Several visualization results are shown in Fig. 3. We projected the bounding boxes with confidence score larger than 0.3 into the raw images. In each of the 6 samples, the inaccuracy of localization exists in the **upper** baseline results. When incorporated with $\mathcal{L}_{RGIoU}$ (**middle**) or $\mathcal{L}_{RIoU}$ (**bottom**), the localization is notably improved. The results are from Frustum PointNet v2.

PointPillars: We used the SECOND [45] implementation[6] for the nuScenes and KITTI. We experimented car-only detection on both KITTI and nuScenes, using SECOND v1.5. The training process took 600k iterations for both datasets. And

[6] github.com/traveller59/second.pytorch

Table 3: Comparisons of different loss settings for **3D object detection** on car category of the nuScenes validation set. The Average Precision (AP) metric of 3D object detection is based on the matching of center distance(D) and IoU respectively. The baseline model is PointPillars [21].

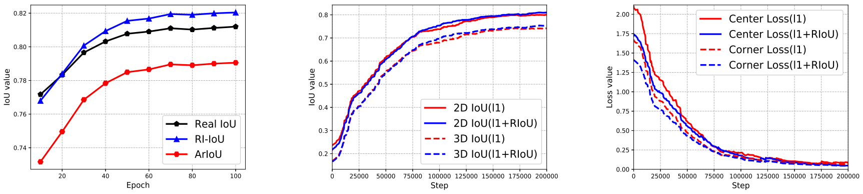
Method	3D detection(IoU)			3D detection(D)			
	Easy	Moderate	Hard	D=0.5	D=1.0	D=2.0	D=4.0
$\ell1$(baseline)	86.02	80.66	62.03	55.30	64.79	68.55	71.85
$\ell1$+$RIoU$	86.30	81.24	62.53	57.26	67.18	70.47	73.44
$\ell1$+$RGIoU$	86.42	81.03	62.65	57.9	67.75	71.07	73.91
$\ell1$+$RIoU$(v)	**86.82**	**82.97**	**65.15**	**61.87**	**72.89**	**75.76**	**77.73**
$\ell1$+$RGIoU$(v)	86.57	81.13	63.79	61.29	71.87	74.75	76.89

Table 4: Comparisons of different loss settings for 3D object detection on the SUN RGB-D dataset. The baseline model is VoteNet [31].

Method	bathtub	bed	bookshelf	chair	desk	dresser	nightstand	sofa	table	toilet	mAP	mAR
votenet	73.7	81.6	28.5	73.5	24.4	27.9	62.5	65.2	50.0	**90.1**	57.7	83.9
votenet+RIoU	73.5	**84.6**	31.6	**74.6**	24.8	**30.0**	**62.7**	**65.9**	**50.7**	89.3	**58.8**	**86.9**
votenent+RGIoU	**78.7**	84.1	**32.2**	73.5	**25.8**	29.4	60.4	65.5	49.5	88.6	**58.8**	85.4

we also evaluated a 9-class subset on the nuScenes using the proposed loss, which can be seen in the supplementary pages. The results are summarized in Table2 and Table3. All trials were evaluated on the whole validation set. While $\mathcal{L}_{RIoU}$ and $\mathcal{L}_{RGIoU}$ both improve the detection performance by an obvious margin, the enhancement by $\mathcal{L}_{RGIoU}$ is not superior to $\mathcal{L}_{RIoU}$. In Table 2, the combination of $\ell1$ loss and the RIoU loss achieves the competitive performance in the easy mode, compared to the IoU loss which computes the accurate IoU. Note that our proposed RIoU saves the sophisticated forward and backward computation.
VoteNet: We followed the official implementation[7] of VoteNet [31]. The training process took 180 epochs. The learning rate was decayed at epoch 40 and epoch 80 respectively. As shown in Table 4, both $\mathcal{L}_{RIoU}$ and $\mathcal{L}_{RGIoU}$ enhance the performance in terms of mAP and mAR, where $\mathcal{L}_{RIoU}$ achieves the best results. Besides, as for the AP of the individual categories, the baseline method only achieves the best result in the ”toilet” category. Especially in the ”bathtub” category, the $\mathcal{L}_{RGIoU}$ improves upon the baseline method by 5 AP.
Compared with ArIoU: We also implemented the loss format of angle-related IoU [26] (ArIoU) as $1 - ArIoU$, and incorporated it into regression loss function just like $\mathcal{L}_{RIoU}$. The experiment results on Frustum PointNet [32] is presented in Table1. When incorporated with $\mathcal{L}_{ArIoU}$, about half of the detection metrics of Frustum PointNet even drop compared with baseline. Compared with $RIoU$ loss function, ArIoU is not beneficial to the detection performance. Besides, during first 100 epochs before convergence, we collected the predicted bounding boxes and their corresponding ground-truths. We randomly picked 50k sample pairs with the interval of 10 epochs, and calculated the average real IoU, $ArIoU$ and

[7] github.com/facebookresearch/votenet



(a) Different IoUs.

(b) Average IoU curve

(c) $\ell 1$-based loss curve

Fig. 4: (a): The average real IoU, $ArIoU$ and our $RIoU$ of sampled bird's eye view prediction and ground-truth pairs. (b): The average IoU curve with regard to training steps. (c): The center loss and corner loss curve w.r.t. training steps.

our $RIoU$. The result in Fig. 4(a) shows that our $RIoU$ approximates better than $ArIoU$, especially during first 40 epochs.

3D assisted by 2D: While we implemented the $RIoU$ in 2D bird's eye view, it benefits 2D localization and 3D detection performance simultaneously. Specifically, our $RIoU$ improves 2D bird's eye view IoU (Fig. 4(b)), 3D IoU (Fig. 4(b)), center loss (Fig. 4(c)) and corner loss optimization (Fig. 4(c)) jointly. Note that the center loss and corner loss are both in $\ell 1$ format, which means our loss can in tern benefit the traditional $\ell 1$ based regression loss. When incorporated with our $\mathcal{L}_{RIoU}$, the overlap between the prediction and ground-truth is higher. The detection network also learns bounding box parameters more efficiently. The data was collected from the experiment on Frustum PointNet v2 [32]

Given the $RIoU$ in bird's eye view, we implemented the volume format of our $RIoU$ by introducing the z axis. The calculation process is the same with the axis-aligned 3D IoU. Following the setting, the training process took 40k iterations. We denote the volume format as $\ell 1$+$RIoU$(v) or $\ell 1$+$RGIoU$(v). As shown in the last 2 rows of Table 3, the volume format further enhanced the detection performance based on the localization improvement. When incorporated with the volume format of $\mathcal{L}_{RIoU}$, the network achieves the best detection performance.

4.3 Ablation Study

The goal of the ablation study is to verify the heuristic design of our $RIoU$. We did the ablation experiments on the 3D car detection task of the KITTI [10] dataset. We chose F-PointNet v1 [32] as the baseline method.

The minimum of intersection areas: While min, max and $mean$ function all hold the stability for the proposed method, we choose the min function in (4) to mitigate the estimation error of $Interseciton1$ and $Intersection2$. In the ablation experiment, we replaced the min function with max or $mean$. As shown in Table 5, except for the easy mode in 3D object localization, the min function gave the best detection result.

The cosine coefficient: The ablation study on the cosine coefficient in (4) focuses on 2 issues: the necessity of the cosine coefficient and the angle coeffi-

Table 5: Ablation experiment results on different function and coefficient value for 3D object detection and localization of car category. "w.o." denotes that the cosine function is removed in the calculation of I_{RIoU}.

Coef	Func	3D detection			3D localization		
		Easy	Moderate	Hard	Easy	Moderate	Hard
2	Max	83.53	70.79	63.08	87.90	82.67	75.01
2	Min	**84.45**	**71.20**	**63.61**	88.01	**82.83**	**75.45**
2	Avg	83.11	69.68	62.85	**88.09**	82.82	74.94
1	Min	83.86	70.73	63.22	87.96	82.53	74.67
2	Min	**84.45**	**71.20**	**63.61**	88.01	**82.83**	**75.45**
3	Min	83.79	70.10	63.11	87.83	82.31	74.93
4	Min	83.23	70.88	63.45	**88.33**	82.26	75.31
5	Min	82.90	70.39	63.08	87.67	82.27	74.22
w.o.	Min	82.41	69.02	61.13	87.87	82.67	75.01

cient value within the cosine function. As shown in the lower part of Table5, F-PointNet achieved the best detection performance when the cosine coefficient was preserved and the angle coefficient was 2. Note that the other 2 categories also achieved the best performance with Min function and coefficient 2, which is not shown in the table to save space. After we removed the cosine coefficient from the calculation of I_{RIoU} in (4), or modified the angle coefficient value, the detection performance dropped by an obvious margin. Note that F-PointNet gave the best 3D object localization performance under the easy mode when the angle coefficient value was 4. This might be due to similar degradation property when the 2 boxes are parallel or orthogonal. Besides, the larger angle coefficient value penalizes the angle difference more.

4.4 Applications in 2D Object Detection

As mentioned above, the proposed $RIoU$ loss function improves the 3D object detection performance by an obvious margin. To further verify its effectiveness in various rotated object detection tasks, we validated it on several 2D detection benchmarks, such as text localization and aerial image detection. The details of the latter are presented in the supplementary pages.

Datasets and settings: The ICDAR 2015 text localization benchmark [16] contains 1000 images for training and 500 images for testing. Each annotated text region is represented by 4 quadrangle corners. We chose EAST [50] as our baseline method, which optimizes the axis-aligned IoU and angle seperately. We used the implementation with RBOX and adopted ResNet-50 [12] as the feature extractor. Note that the network was trained and evaluated on the single ICDAR 2015 dataset when incorporated with the proposed loss. One of the baseline networks was also tuned on the the ICDAR 2013 dataset [17], which contains 229 training images with horizontal text annotations.

Results: The text localization results are reported in 3 metrics: recall, precision and F-score. As shown in Table 6, when incorporated with $\mathcal{L}_{RGIoU}$ and trained

Table 6: Text localization results, evaluated on the ICDAR 2015 test set. The 2013 and 2015 stand for the ICDAR 2013 and ICDAR 2015 dataset respectively.

Method	Trained on		Metric		
	2013	2015	Recall	Precision	F-Score
$\ell 1$(baseline)		✓	83.80	76.46	79.96
$\ell 1$(baseline)[1]	✓	✓	84.66	77.32	80.83
$\ell 1+RGIoU$		✓	82.86	77.52	80.10
$\ell 1+RIoU$		✓	**86.20**	**78.48**	**82.16**

[1] Reported on the ICDAR 2015 benchmark website.

only on the ICDAR 2015 dataset, the detection performance outperforms the baseline in terms of Precision and F-score. Especially when incorporated with $\mathcal{L}_{RIoU}$, all the 3 metrics even surpass the jointly-trained baseline by an obvious margin. The $RIoU$ helps to propose the text boxes more adequately and locate them more accurately, which demonstrates the effectiveness of our proposed loss function. The illustration is shown in the supplementary pages.

From the experiment results in text localization and detection in aerial images, we can see that the proposed loss function could significantly improve the detection performance of the rotated 2D targets. Apart from the 3D cuboids, it is also suitable for detection targets in the shape of narrow rectangle and large numbers of small, cluttered rotated targets in an image.

5 Conclusion

In this paper, we have proposed a loss function called Rotation-robust Intersection over Union ($RIoU$) for robust object detection. It is designed for bounding boxes in arbitrary rotation conditions. The implementation only comprises basic operations, and is feasible for back-propagation. It is suitable for both 2D target localization and 3D object detection. Incorporated into traditional $\ell 1$-based regression loss function, the proposed loss function achieves notable improvement over several state-of-the-art baselines. In the future, we will focus on the invariance of IoU with regard to the rotation. We will also explore its application in more abundant cases, such as object detection in indoor scenes.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 61822603, Grant U1813218, Grant U1713214, and Grant 61672306, in part by Beijing Academy of Artificial Intelligence (BAAI), in part by a grant from the Institute for Guo Qiang, Tsinghua University, in part by the Shenzhen Fundamental Research Fund (Subject Arrangement) under Grant JCYJ20170412170602564, and in part by Tsinghua University Initiative Scientific Research Program.

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
2. Chabot, F., Chaouch, M., Rabarisoa, J., Teuliere, C., Chateau, T.: Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In: CVPR. pp. 2040–2049 (2017)
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. TPAMI **40**(4), 834–848 (2017)
4. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: CVPR. pp. 2147–2156 (2016)
5. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. In: NeurIPS. pp. 424–432 (2015)
6. Chen, X., Kundu, K., Zhu, Y., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals using stereo imagery for accurate object class detection. TPAMI **40**(5), 1259–1272 (2017)
7. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: CVPR. pp. 1907–1915 (2017)
8. Chen, Y., Liu, S., Shen, X., Jia1;2, J.: Fast point r-cnn. In: ICCV. pp. 9775–9784 (2019)
9. Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H., Posner, I.: Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In: ICRA. pp. 1355–1361 (2017)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR. pp. 3354–3361 (2012)
11. Girshick, R.: Fast r-cnn. In: ICCV. pp. 1440–1448 (2015)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
13. Huang, C., Zhai, S., Talbott, W., Bautista, M.A., Sun, S.Y., Guestrin, C., Susskind, J.: Addressing the loss-metric mismatch with adaptive loss alignment. In: ICML (2019)
14. Janssens, R., Zeng, G., Zheng, G.: Fully automatic segmentation of lumbar vertebrae from ct images using cascaded 3d fully convolutional networks. In: ISBI. pp. 893–897 (2018)
15. Jiang, B., Luo, R., Mao, J., Xiao, T., Jiang, Y.: Acquisition of localization confidence for accurate object detection. In: ECCV. pp. 784–799 (2018)
16. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: ICDAR. pp. 1156–1160 (2015)
17. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: ICDAR. pp. 1484–1493 (2013)
18. Kosiorek, A., Bewley, A., Posner, I.: Hierarchical attentive recurrent tracking. In: NeurIPS. pp. 3053–3061 (2017)
19. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: IROS. pp. 1–8 (2018)

20. Ku, J., Pon, A.D., Waslander, S.L.: Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In: CVPR. pp. 11867–11876 (2019)
21. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019)
22. Li, B.: 3d fully convolutional network for vehicle detection in point cloud. In: IROS. pp. 1513–1518 (2017)
23. Li, P., Chen, X., Shen, S.: Stereo r-cnn based 3d object detection for autonomous driving. In: CVPR. pp. 7644–7652 (2019)
24. Li, P., Qin, T., et al.: Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In: ECCV. pp. 646–661 (2018)
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
26. Liu, L., Pan, Z., Lei, B.: Learning a rotation invariant detector with rotatable bounding box. arXiv preprint arXiv:1711.09405 (2017)
27. Liu, L., Lu, J., Xu, C., Tian, Q., Zhou, J.: Deep fitting degree scoring network for monocular 3d object detection. In: CVPR. pp. 1057–1066 (2019)
28. Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., Yan, J.: Fots: Fast oriented text spotting with a unified network. In: CVPR. pp. 5676–5685 (2018)
29. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IROS. pp. 922–928 (2015)
30. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: CVPR. pp. 4293–4302 (2016)
31. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV. pp. 9277–9286 (2019)
32. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: CVPR. pp. 918–927 (2018)
33. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. pp. 652–660 (2017)
34. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS. pp. 5099–5108 (2017)
35. Qin, Z., Wang, J., Lu, Y.: Triangulation learning network: From monocular to stereo 3d object detection. In: CVPR. pp. 11867–11876 (2019)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS. pp. 91–99 (2015)
37. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR. pp. 658–666 (2019)
38. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR. pp. 770–779 (2019)
39. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: CVPR. pp. 567–576 (2015)
40. Tychsen-Smith, L., Petersson, L.: Improving object localization with fitness nms and bounded iou loss. In: CVPR. pp. 6877–6885 (2018)
41. Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., Shen, C.: Repulsion loss: Detecting pedestrians in a crowd. In: CVPR. pp. 7774–7783 (2018)
42. Wang, Z., Jia, K.: Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. arXiv preprint arXiv:1903.01864 (2019)
43. Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. In: CVPR. pp. 2345–2353 (2018)

44. Xu, D., Anguelov, D., Jain, A.: Pointfusion: Deep sensor fusion for 3d bounding box estimation. In: CVPR. pp. 244–253 (2018)
45. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
46. Yang, X., Yang, J., Yan, J., Zhang, Y., Zhang, T., Guo, Z., Xian, S., Fu, K.: Scrdet: Towards more robust detection for small, cluttered and rotated objects. In: ICCV. pp. 8232–8241 (2019)
47. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: ACM MM. pp. 516–520 (2016)
48. Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., Yang, R.: Iou loss for 2d/3d object detection. In: 3DV. pp. 85–94 (2019)
49. Zhou, J., Lu, X., Tan, X., Shao, Z., Ding, S., Ma, L.: Fvnet: 3d front-view proposal generation for real-time object detection from point clouds. arXiv preprint arXiv:1903.10750 (2019)
50. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: an efficient and accurate scene text detector. In: CVPR. pp. 5551–5560 (2017)
51. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end ining for point cloud based 3d object detection. In: CVPR. pp. 4490–4499 (2018)