Deep Dive into Gradients: Better Optimization for 3D Object Detection with Gradient-Corrected IoU Supervision

Qi Ming[†], Lingjuan Miao, Zhe Ma, Lin Zhao,
Zhiqiang Zhou[*†] , Xuhui Huang, Yuanpei Chen, Yufei Guo[*]
School of Automation, Beijing Institute of Technology, China,
Intelligent Science & Technology Academy of CASIC,
chaser.ming@gmail.com,{miaolingjuan,zhzhzhou}@bit.edu.cn,zhaolins@foxmail.com,yfguo@pku.edu.cn

Abstract

Intersection-over-Union (IoU) is the most popular metric to evaluate regression performance in 3D object detection. Recently, there are also some methods applying IoU to the optimization of 3D bounding box regression. However, we demonstrate through experiments and mathematical proof that the 3D IoU loss suffers from abnormal gradient w.r.t. angular error and object scale, which further leads to slow convergence and suboptimal regression process, respectively. In this paper, we propose a Gradient-Corrected IoU (GCIoU) loss to achieve fast and accurate 3D bounding box regression. Specifically, a gradient correction strategy is designed to endow 3D IoU loss with a reasonable gradient. It ensures that the model converges quickly in the early stage of training, and helps to achieve fine-grained refinement of bounding boxes in the later stage. To solve suboptimal regression of 3D IoU loss for objects at different scales, we introduce a gradient rescaling strategy to adaptively optimize the step size. Finally, we integrate GCIoU Loss into multiple models to achieve stable performance gains and faster model convergence. Experiments on KITTI dataset demonstrate superiority of the proposed method. The code is available at https://github.com/ming71/GCIoU-loss.

1. Introduction

Autonomous driving has received extensive attention in recent years due to its broad application scenarios. 3D object detection is a very important and promising topic of autonomous driving. Currently, LiDAR is the most commonly used sensor to obtain representations of object in the real world for 3D object detection [4,21,33,35,40,44]. The sparse point cloud data from LiDAR scanners can provide depth information of objects, which is critical for accurate 3D object detection.

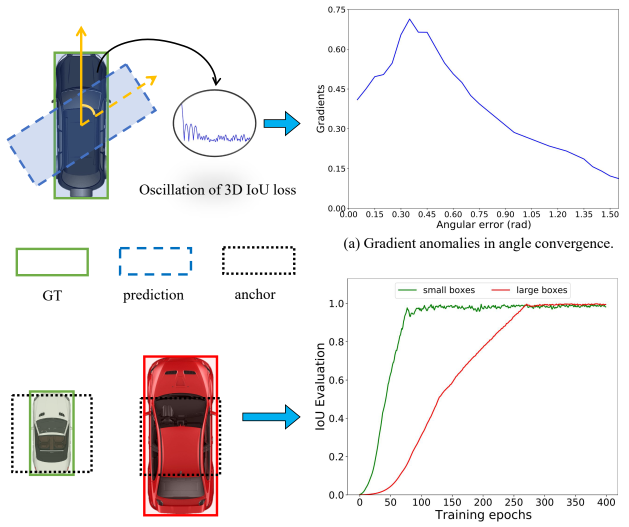

Figure 1. Visualization of some problems in current 3D IoU loss. (a) Gradient changes of 3D IoU loss w.r.t. angular error. Gradients are small under large angular errors. (b) Convergence of bounding boxes of different scales under supervision of 3D IoU loss. The optimization for multi-scale objects is different. IoU Loss converges slower for large objects.

Intersection-over-Union (IoU) is the most popular metric used to measure the detection accuracy and evaluate performance of the model. But in the training phase, current detectors usually use the L_{norm}-based loss function to optimize the bounding box regression [6, 24, 34, 43, 50]. The inconsistency between training loss and evaluation metric would lead to a misaligned optimization process. Specifically, a lower loss value does not guarantee better detection performance (higher IoU), which has also been discussed in some previous work [25,26,38,41,48,49]. For example, Yu

[*] Corresponding author.
[†] Contribute equally.

et al. [41] suggested out that IoU, as an evaluation metric in 2D object detection, can naturally be used in loss function to achieve consistency between training and testing. Zhou *et al.* [49] extended the case to 3D object detection and designed 3D IoU loss for autonomous driving.

IoU-based loss and its variants are studied and widely used in 2D object detection [9, 16–18, 25, 36, 48]. There are some work to introduce 3D IoU into 3D object detection and achieve performance improvement [12, 13, 15, 32, 49]. However, the IoU loss in 3D object detection is quite different from that in 2D object detection. Firstly, the 3D bounding boxes have higher degree of freedom than 2D horizontal bounding boxes, which makes the IoU calculation between two 3D bounding boxes very complicated. Secondly, the IoU-based losses generally converges slowly. The larger search space and higher degrees-of-freedom of 3D bounding box parameters further increases the difficulty of model convergence. There are also some previous work devoted to the improvements on 3D IoU loss [26, 47]. For instance, Rotation-robust Intersection over Union (RIoU) [47] uses a projection operation to estimate the intersection area to simplify the calculation of 3D IoU, which is both robust and feasible for back-propagation. Recently, Rotation-Decoupled IoU (RDIoU) [26] decouples the rotation variable to solve the negative coupling effect of rotation on the 3D IoU.

However, most of 3D IoU based losses directly apply 3D IoU or use its variants to supervise bounding box regression, and do not give an in-depth analyze to the loss gradient changes during training. As shown in Fig. 1(a), we found that the 3D IoU loss suffers from abnormal gradient changes during training. Firstly, when the angular error between predicted box and the ground-truth (GT) box is large, 3D IoU loss produces small gradients for optimization, which leads to a very slow convergence. Secondly, as the model converges, the angular error decreases, but there is a period where the gradient increases abnormally. In this case, the angle prediction is difficult to be refined, and it even leads to the loss oscillation (see Fig. 1(a)). Furthermore, IoU is generally considered to be scale-invariant, but we observed that the optimization of the 3D IoU loss performs differently for objects of different scales. As shown in Fig. 1(b), for large scale objects, more iterations are required to make the bounding box regressed well.

In this paper, we give detailed discussion of gradient changes of 3D IoU loss from both experimental results and mathematical analysis. Then, we propose a gradient-correction IoU (GCIoU) loss to solve the mentioned problems. GCIoU loss contains two parts, gradient correction strategy and gradient rescaling strategy. Gradient correction strategy corrects the gradient of 3D IoU loss w.r.t. angular error to ensure smooth angle convergence. The GCIoU loss is then obtained by integrating the expected gradients. Further, to optimize the convergence of objects of different scales, we use a gradient rescaling strategy to modify the gradient of IoU loss w.r.t. scale to adaptively update the variables. Experimental results on KITTI dataset demonstrate the superiority of our method.

The main contributions of this paper are summarized as follows:

- We found the abnormal gradient changes of 3D IoU loss w.r.t. angular error during the optimization process through experiments, and further gave analysis and proof for it.
- We observed different performances of the 3D IoU loss for the optimization of objects of different scales, and concluded that it is also caused by abnormal gradients via mathematical analysis.
- The gradient-correction IoU (GCIoU) loss is proposed to alleviate abnormal gradients and accelerate convergence of 3D IoU based loss. The parameter update strategy is then adjusted to adaptively optimize multi-scale objects with different steps.

2. Related Work

2.1. LiDAR-Based 3D Object Detection

Existing LiDAR-based 3D object detection mainly processes 3D point clouds through point-based [28, 30, 39], voxel-based [3, 7, 37, 45] or projection-based [4, 13] approaches. To generate 3D proposals directly from raw LiDAR point clouds in a bottom-up manner, PointRCNN [28] leverages the PointNet++ [23] network as an encoder that learns point cloud representations. STD [39] further designs spherical anchors to reasonably seed anchors for each LiDAR point. It then collects LiDAR points within spherical anchors for 3D object classification and regression. However, these PointNet-based 3D detectors rely on hand-designed parameters for frequent point sampling and grouping operations. Point-GNN [30] instead explores graph representations of LiDAR point clouds for compact encoding of position and spatial relationship information. Recently, sparse convolutions on 3D LiDAR point clouds [37] have show powerful feature extraction capabilities and reduced the computational cost by ignoring non-empty voxels. [7] [45] look into encoding the input 3D points with stacked sparse convolutional backbone networks. SA-SSD [7] utilizes an auxiliary network that combines point-level foreground segmentation with center estimation to guide the sparse voxel features to be aware of 3D structural information. CIA-SSD [45] presents an IoU-aware confidence refinement function to make the classification confidence and 3D bounding box regression more consistent. Among the recent two-stage 3D detection methods, Voxel R-CNN [3] aggregates fine-grained spatial information from encoded voxel features via a voxel RoI refinement module. Moreover, PV-RCNN [27] integrates both the PointNet-based point features and 3D sparse voxel convolutional features to capture rich contextual information for 3D proposals. Some other projection-based work [4, 13] projects LiDAR point clouds to range image via spherical projection [13]. In their method, point-wise feature learning can be exploited with efficient 2D convolutions on compact range image representations.

2.2. 3D Intersection-over-Union(IoU) Based Loss

Considering the misalignment between the refined bounding boxes from the second stage of the 3D detector and the estimated IoUs, Li *et al.* [12] leverages a 3D RoI alignment module to generate RoI-Aligned features that are incorporated into the corner embeddings for accurate position encoding. In RangeIoUDet [13], a point-based IoU and a 3D hybrid GIoU loss are adopted to improve point-wise feature learning and provide quality geometry information. It further leverages a 3D hybrid GIoU loss to improve localization accuracy and provide reasonable evaluation for predicted 3D Boxes. Zhou *et al.* [49] presents a unified IoU computation framework for rotated 2D and 3D object detectors. Also, [47] utilizes a projection operation to estimate the interaction region of IoUs and alleviate the convex uncertainty in the rotation computation. While the introduction of 3D IoU brings some improvements to 3D object detection, it is still sensitive to rotated bounding boxes, which may lead to instability in training stage. Most recently, RDIoU [26] introduces a rotation decoupling method to simplify the complicated interactions of box regression while improving the training stability of rotated 3D box estimation.

IoU loss in 3D object detection is still in its infancy and suffers from slow convergence. Current IoU-based methods have initially optimized and accelerated the 3D IoU computation. However, these methods do not examine in depth the specific performance of IoU loss on the 3D envelope regression process and some optimization issues from a gradient perspective.

3. Analysis of 3D IoU Loss

3.1. Preliminaries

In the 3D object detection task, given the prediction box $\boldsymbol{P} = \{x_p, y_p, z_p, w_p, h_p, l_p, \theta_p\}$ and GT box $\boldsymbol{G} = \{x_t, y_t, z_t, w_t, h_t, l_t, \theta_t\}$. (x, y) denotes center point of box. $\{w, h, l, \theta\}$ are the width, length, height, and orientation of box. The IoU between the two boxes is as follows:

$$\boldsymbol{IoU}(\boldsymbol{P}, \boldsymbol{G}) = \frac{\boldsymbol{I}(\boldsymbol{P}, \boldsymbol{G})}{\boldsymbol{U}(\boldsymbol{P}, \boldsymbol{G})} = \frac{\boldsymbol{P} \cap \boldsymbol{G}}{\|\boldsymbol{P}\| + \|\boldsymbol{G}\| - \boldsymbol{P} \cap \boldsymbol{G}} \quad (1)$$

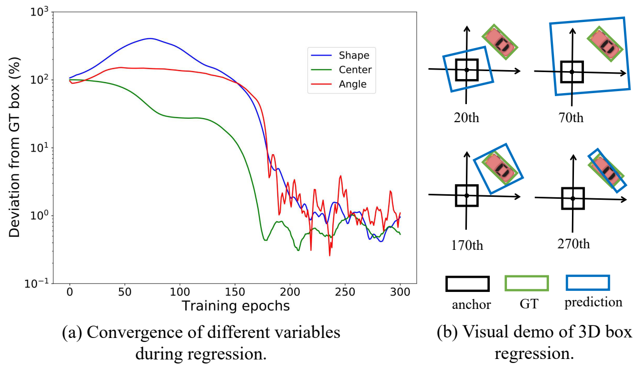

Figure 2. Illustration of the convergence process for different variables under 3D IoU supervision. (a) shows the convergence speed of different variables of 3D box. 'Center' refers to the center point deviation, 'Shape' is the volume offset between two boxes, and 'Angle' is orientation error. (b) visualizes the regression steps of the boxes under IoU supervision.

where $\boldsymbol{I}$ denotes the overlap area between $\boldsymbol{P}$ and $\boldsymbol{G}$, and $\boldsymbol{U}$ is the union of the two boxes. IoU intuitively reflects the spatial overlap of two geometries, and therefore it is the most commonly used metric to evaluate the localization accuracy.

Meanwhile, IoU can also be directly used as a loss function to supervise bounding box regression. The simplest IoU loss is as follows:

$$\boldsymbol{L}_{IoU}(\boldsymbol{P}, \boldsymbol{G}) = 1 - \boldsymbol{IoU}(\boldsymbol{P}, \boldsymbol{G}) \quad (2)$$

Also, there is another popular form of negative logarithm IoU loss, which is organized as follows:

$$\boldsymbol{L}_{lnIoU}(\boldsymbol{P}, \boldsymbol{G}) = -ln(\boldsymbol{IoU}(\boldsymbol{P}, \boldsymbol{G})) \quad (3)$$

In general 2D object detection task, horizontal bounding boxes are usually used to represent objects, and therefore IoU can be easily obtained. However, the calculation of intersection between two 3D bounding boxes is quite complicated. A variety of cases need to be considered separately, and it is difficult to give a specific calculation formula. In this paper, we suggest that only two cases which hinder the model convergence need to be considered. One is the case where the GT box is surrounded by the prediction box, and the other is the case where the two boxes are center-aligned. The reason is as follows.

We observed via experiments that the center points are relatively easy to be regressed. Shown in Fig. 2(a) is the convergence curve of model supervised by 3D IoU loss. We preset a series of anchors around the GT boxes, and then design a tiny model to regress the targets under the supervision of 3D IoU loss. A visualization example of regression process is given in Fig. 2(b). At the beginning of training, the

IoU loss forces the model to predict large-scale predictions to cover GT boxes and then the center point is accurately regressed. In this case, we need to guarantee a quick convergence of the IoU to speed up the regression process. After that, shape convergence is relatively slow compared with center point prediction, while angle regression even suffers from severe oscillations. Therefore, we only pay attention to optimization of shape and angle when center point is well located. To summarize, the calculation of 3D IoU can therefore be simplified into the following two cases: (a)GT box and predicted box are center aligned (b) GT is inside the predicted box.

In the first case, for a given prediction box $\boldsymbol{P} = \{x_p, y_p, z_p, w_p, h_p, l_p, \theta_p\}$ and its GT box $\boldsymbol{G} = \{x_t, y_t, z_t, w_t, h_t, l_t, \theta_t\}$, we can get the IoU based on the geometric relationship as follows:

$$\begin{cases} I = \dfrac{h_t h_p l}{sin\theta}, \\ U = w_t h_t l_t + w_p h_p l_p - I, \end{cases} \tag{4}$$

where $\theta = |\theta_p - \theta_t|$ denotes the angle deviation, and $l = min\{l_t, l_p\}$. Note that Eq. (4) only holds when the intersection is a parallelogram. Specifically, $\theta \in [\theta_0, \frac{\pi}{2}]$, θ_0 is the boundary conditions and can be calculated from the geometric relationship. As shown in Fig. 1(a), the gradient of 3D IoU loss w.r.t. angular error would converge well when the angular error is small. Therefore, we only consider the abnormal situations with $\theta \in [\theta_0, \frac{\pi}{2}]$.

For the another case where GT box is inside the predicted box, elements in IoU can be represented as follows:

$$\begin{cases} I = w_t h_t l_t, \\ U = w_p h_p l_p. \end{cases} \tag{5}$$

3.2. Analysis of Gradient of 3D IoU Loss

First, we get the partial derivatives of the IoU as follows:

$$\frac{\partial IoU}{\partial x} = \frac{\partial(\frac{I}{U})}{\partial x} = (\frac{\partial I}{\partial x} \cdot U - \frac{\partial U}{\partial x} \cdot I)/U^2, \tag{6}$$

where x is the variable in the 3D bounding box, such x_p, w_p. Next, we will analyze the gradients of IoU loss for the above two cases and demonstrate the experimental results in Fig. 1.

3.2.1 Gradient w.r.t. Angular Error

We first consider the center-aligned case, and calculate the partial derivatives of I and U w.r.t. the angular error θ as follows:

$$\begin{cases} \dfrac{\partial I}{\partial \theta} = -\dfrac{h_t h_p l \cdot cos\theta}{sin^2\theta} = -I \cdot \cot\theta, \\ \dfrac{\partial U}{\partial \theta} = -\dfrac{\partial I}{\partial \theta} = I \cdot \cot\theta. \end{cases} \tag{7}$$

For the common linear IoU loss in Eq. (2), combining Eq. (6) with Eq. (7), we obtain the gradient of the IoU loss w.r.t. angular error as follows:

$$\frac{\partial \boldsymbol{L}_{IoU}}{\partial \theta} = \text{IoU}(1 + \text{IoU}) \cdot \cot\theta. \tag{8}$$

As the model converges, the $IoU \uparrow$ and the angle error $\theta \downarrow$, therefore $\cot\theta \uparrow$. In the end, the gradient increases abnormally, which would lead to inaccurate angle predictions. Further, when θ is large, the gradient is very small, and therefore the regression loss converges very slowly in the early stage of training. The above analysis is consistent with the experimental results in Fig. 1(a), and accounts for the slow convergence of the IoU loss.

For the non-linear IoU loss in Eq. (3), we also obtain the gradient as follows:

$$\frac{\partial \boldsymbol{L}_{lnIoU}}{\partial \theta} = \frac{1}{\text{IoU}} \cdot \frac{\partial \boldsymbol{L}_{IoU}}{\partial \theta} = (1 + \text{IoU}) \cdot \cot\theta. \tag{9}$$

It can be seen that the same issues also exist in the non-linear IoU loss. But the gradient growth of non-linear IoU loss is slower than that of linear IoU loss. Therefore, in general non-linear IoU loss achieves relatively better performance than linear one.

Similarly, for the GT contained case, the partial derivatives of I and U w.r.t. the angular error θ can also obtained via Eq. (5): $\frac{\partial I}{\partial \theta} = \frac{\partial U}{\partial \theta} = 0$. Therefore, we get $\frac{\partial \boldsymbol{L}_{lnIoU}}{\partial \theta} = 0$. It means that the angle is not well optimized when the prediction box is expanding its area to find the GT box. Therefore, the IoU loss optimizes different variables of the predictions in different stages, which would lead to slow convergence speed.

In summary, current regression loss only ensures that the loss value decreases as error of parameter estimation decreases, ignoring that inappropriate changes in gradients also hinder the network convergence. Therefore, we suggest that the gradient of IoU loss w.r.t. angular error should also decrease as the angular error decreases.

3.2.2 Gradient w.r.t. Scale

Similar to the previous steps, we first consider the center-aligned case represented by Eq. (4). The scale of 3D bounding box contains three variables w, h, l. We denote that w, h, l are proportional to a scale factor s, and therefore $I, U \propto s^3$, and $IoU \propto s^0$. In this section, we give the results directly to avoid confusion caused by too many formulas.

Under the center-aligned case, we obtain the gradient of linear IoU loss w.r.t. the scale as follows:

$$\begin{cases} \dfrac{\partial \boldsymbol{L}_{IoU}}{\partial h_p} = -\text{IoU} \cdot \dfrac{V_t}{U h_p}, \\ \dfrac{\partial \boldsymbol{L}_{IoU}}{\partial w_p} = \text{IoU} \cdot \dfrac{h_p l_p}{U}. \\ \dfrac{\partial \boldsymbol{L}_{IoU}}{\partial l_p} = \begin{cases} \text{IoU} \cdot \dfrac{V_t}{U l_p}, & l = l_p, \\ -\text{IoU} \cdot \dfrac{w_p h_p}{U}, & l = l_t, \end{cases} \end{cases} \tag{10}$$

in which $V_t = w_t h_t l_t$ is the volume of GT box, and $V_t \propto s^3$. Obviously, we know that $|\frac{\partial \boldsymbol{L}_{IoU}}{\partial h_p}| \propto \frac{1}{s}$, $|\frac{\partial \boldsymbol{L}_{IoU}}{\partial w_p}| \propto \frac{1}{s}$, and $|\frac{\partial \boldsymbol{L}_{IoU}}{\partial l_p}| \propto \frac{1}{s}$. As a result, for large-scale objects, IoU loss produces small gradients to optimize the shape of the predictions, which would lead to slow convergence. For small-scale objects, the model applies large gradients to correct shape errors, which would lead to poor refinement of small bounding boxes and even loss oscillation. These results are consistent with the experimental observations in Fig. 1,

The same conclusion also holds for non-linear IoU loss. It is known that $\frac{\partial \boldsymbol{L}_{lnIoU}}{x} = \frac{1}{IoU} \cdot \frac{\partial \boldsymbol{L}_{IoU}}{x}$ in Eq. (9). Combining Eq. (10), we still get $|\frac{\partial \boldsymbol{L}_{lnIoU}}{\partial h_p}|$, $|\frac{\partial \boldsymbol{L}_{lnIoU}}{\partial w_p}|$, and $|\frac{\partial \boldsymbol{L}_{lnIoU}}{\partial l_p}| \propto \frac{1}{s}$. Hence, the non-linear IoU loss suffers from suboptimal regression of bounding boxes of different scales.

Under the GT box contained case, the gradients of linear IoU loss are as follows:

$$\begin{cases} \dfrac{\partial \boldsymbol{L}_{IoU}}{\partial h_p} = \text{IoU} \cdot \dfrac{w_p l_p}{U}, \\ \dfrac{\partial \boldsymbol{L}_{IoU}}{\partial w_p} = \text{IoU} \cdot \dfrac{h_p l_p}{U}. \\ \dfrac{\partial \boldsymbol{L}_{IoU}}{\partial l_p} = \text{IoU} \cdot \dfrac{w_p h_p}{U} \end{cases} \tag{11}$$

In this case, there is still the issue of different optimization of linear IoU loss for objects of different scales. Similarly, it is easy to prove that non-linear IoU loss also suffers from this problem.

4. Methodology

4.1. Gradient Correction for Angle Optimization

As discussed earlier, the negative logarithm IoU loss performs better than the linear IoU loss, so modifications are made based on the negative logarithm IoU loss. As shown in Eq. (9), the gradient anomaly of $\boldsymbol{L}_{lnIoU}$ mainly comes from two parts:

Prob. 1 Abnormal gradient growth as the angle converges.

Prob. 2 When the angular error is large, the small gradient is not conducive to the angle convergence.

To solve the above problems, we correct the $\boldsymbol{L}_{lnIoU}$ and obtain the gradient-corrected IoU (GCIoU) loss as follows:

$$\boldsymbol{L}_{GCIoU} = -ln(\text{IoU}) \cdot f(\theta) + g(\theta), \tag{12}$$

where $f(\theta)$ and $g(\theta)$ are the correction function and will be described below. Next, the first derivative of $\boldsymbol{L}_{GCIoU}$ w.r.t. θ can be obtained as follows:

$$\begin{aligned} \frac{\partial \boldsymbol{L}_{GCIoU}}{\partial \theta} = &(1+\text{IoU}) \cdot \cot\theta \cdot f(\theta) \\ &- ln(\text{IoU}) \cdot f(\theta)' + g'(\theta). \end{aligned} \tag{13}$$

As suggested before, we need to ensure that $\frac{\partial \boldsymbol{L}_{GCIoU}}{\partial \theta}$ is monotonically increasing w.r.t. θ, that is, $\frac{\partial^2 \boldsymbol{L}_{GCIoU}}{\partial \theta^2} > 0$. Hence, the second derivative is to be considered:

$$\begin{aligned} \frac{\partial^2 \boldsymbol{L}_{GCIoU}}{\partial \theta^2} = &\underbrace{2(1+\text{IoU})\cot\theta \cdot}_{B(\theta)} f(\theta)' \underbrace{-ln(\text{IoU})}_{C(\theta)} \cdot f(\theta)'' \\ &- \underbrace{(1+\text{IoU})(\text{IoU} \cdot \cot^2\theta + \csc^2\theta)}_{A(\theta)} \cdot f(\theta) + g''(\theta). \end{aligned} \tag{14}$$

For the original negative logarithm IoU loss in Eq. (3), $f(\theta) = 1$, $f(\theta)' = f(\theta)'' = 0$, and $g(\theta) = 0$. Therefore, $\frac{\partial^2 \boldsymbol{L}_{lnIoU}}{\partial \theta^2} < 0$, the model suffers from abnormal gradients.

Firstly, we set $g(\theta) = 0$ in Eq. (12) to determine $f(\theta)$. It's known that $\boldsymbol{L}_{GCIoU} > 0$, and thus $f(\theta) > 0$. Note that the targets of angle regression is not equivalent to that of the IoU optimization. Therefore, $f(0) \neq 0$ is required, otherwise the model cannot be further optimized when the angle is well regressed. Next, as angular error decreases, $\boldsymbol{L}_{GCIoU}$ should also decreases to ensure model convergence. Therefore, $\frac{\partial \boldsymbol{L}_{GCIoU}}{\partial \theta} > 0$. The straightforward candidates are functions that satisfy $f(\theta)' > 0$.

In Eq. (14), as $\theta \downarrow$ and IoU $\uparrow$, $A(\theta) \uparrow$, $B(\theta) \uparrow$, and $C(\theta) \downarrow$. Hence $C(\theta)$ is stable as model converges. For simplicity, we assume that $f(\theta)'' > 0$. If $\frac{\partial^2 \boldsymbol{L}_{GCIoU}}{\partial \theta^2} > 0$, we obtain a stronger consttrain from Eq. (14):

$$\frac{f(\theta)'}{f(\theta)} > \frac{\text{IoU} \cdot \cos^2\theta + 1}{2\sin 2\theta}. \tag{15}$$

Obviously, we cannot achieve the goal by simply choosing $f(\theta)$ that satisfies the above requirements. When $\theta \to 0$ or $\theta \to \frac{\pi}{2}$, there are always critical value that make Eq. (15) no longer hold. However, a suitable $f(\theta)$ can make this critical theta larger or smaller, thereby expanding the ideal optimization interval that meets the requirements. There is a suitable candidate that satisfies the above requirements: $f(\theta) = e^{P(\theta)}$. $P(\theta)$ can be θ, θ^2, etc. Then, GCIoU is as follows:

$$\boldsymbol{L}_{GCIoU} = -ln(\text{IoU}) \cdot e^{P(\theta)} + g(\theta). \tag{16}$$

For example, when $P(\theta) = \alpha\theta$,$\frac{f(\theta)'}{f(\theta)} = \alpha$. Therefore we can adjust the hyperparameter α to control gradients variation during regression. In our experiments, we finally select $P(\theta) = \theta^{\alpha}$ for better performance.

$f(\theta)$ is designed to alleviate the abnormal gradient increase in **Prob. 1**, while $g(\theta)$ is used to increase the gradient to speed up the convergence and solve the **Prob. 2**. Relatively, $g(\theta)$ helps to boost the gradients when the angle error is large, and thus accelerates the convergence speed. Therefore, $g(\theta)$ is an monotonically increasing function w.r.t θ to provide larger gradients at the beginning of training. Also, $g(\theta)$ should satisfy some requirements: (1) $g(\theta) > 0$ to avoid negative loss values, (2) $g(\theta)' > 0$ to ensure that angle convergence guarantees the loss decreases. There are many candidates for $g(\theta)$. For example, we take $g(\theta) = \tan\theta$ in Eq. (16) to get the final $\boldsymbol{L}_{GCIoU}$ as follows:

$$\boldsymbol{L}_{GCIoU} = -ln(\mathrm{IoU}) \cdot e^{\theta^{\alpha}} + \tan\theta, \tag{17}$$

The combination of $f(\theta)$ and $g(\theta)$ ensures that the gradient of 3D IoU loss changes adaptively during regression process thereby accelerating network convergence. Furthermore, formally the GCIoU loss simply adds two additional angular constraints to the initial negative log IoU loss. It would therefore not affect the intersection calculation, and helps steadily improve performance for all cases of 3D bounding box predictions.

4.2. Gradient Rescaling for Scale Optimization

IoU is scale-invariant and can evaluate the localization performance of objects at different scales. However, we found that the optimization process of the IoU loss is sensitive to scale. The GCIoU loss in Eq. (17) only optimizes the angle gradients and does not introduce shape information, so it cannot correct the suboptimal shape optimization of IoU loss. Hence $\frac{\partial \boldsymbol{L}_{GCIoU}}{\partial s} \propto \frac{1}{s}$.

Introducing shape information directly on the basis of Eq. (17) would change the gradient of the loss w.r.t. angular error. There is previous work that adjusts the variable update strategy during the regression process [22]. Similarly, we introduce gradient rescaling strategy to adjust gradients as follows:

$$s_{t+1} = s_t - \eta \cdot \frac{\partial \boldsymbol{L}_{GCIoU}}{\partial s} \cdot \mathrm{U}^{\frac{2}{3}}, \tag{18}$$

where U is the union between prediction and GT box. s_t denotes the shape of current predicted 3D bounding box (w,h or l), and s_{t+1} stands for the variables after updating. Since $\mathrm{U} \propto s^3$, $\mathrm{U}^{\frac{2}{3}} \propto s^2$, hence $\frac{\partial \boldsymbol{L}_{GCIoU}}{\partial s} \cdot \mathrm{U}^{\frac{2}{3}} \propto s$. The gradient rescaling strategy in Eq. (18) achieves adaptive scale optimization without affecting backpropagation. In this case, the parameter updating step is automatically adjusted according to size of object. Specifically, the model applies a larger step size to adjust large objects, while producing small optimization steps for small objects.

	GC		GR	Car (IoU=0.7)		
	$f(\theta)$	$g(\theta)$		Easy	Moderate	Hard
1				88.61	78.12	77.27
2	✓			89.21	78.56	78.22
3		✓		88.98	78.53	77.75
4	✓	✓		89.54	78.81	78.39
5			✓	89.16	78.54	78.01
6	✓	✓	✓	**89.85**	**80.03**	**78.66**

Table 1. Component-wise experiments of the proposed GCIoU loss on KITTI *val* set. 'GC' denotes the gradient correction strategy for angle regression. 'GR' is the gradient rescaling strategy for shape regression. We report the average precisions of 11 sampling recall points.

5. Experiments

5.1. Implementation Details

Dataset. Experiments are conducted to evaluate the proposed method on KITTI dataset. KITTI [5] is the most commonly used benchmark for 3D object detection, which contains 7481 LiDAR images for training and 7518 images for testing. Following the common dataset division strategy [2,11], training images are further divided into the training set and the validation set, containing 3712 images and 3769 images respectively.

Preprocessing. For a fair comparison with existing methods, all LiDAR point clouds are cropped according to the x-axis (0,70.4)m, y-axis (-40,40)m and z-axis (-3,1)m. The voxel size of raw LiDAR point cloud during voxelization is set to (0.05,0.05,0.1)m. For data augmentation during the training phase, random flipping along the X-axis is employed, together with global scaling of the input point cloud with a random scaling factor sampled from [0.95, 1.05]. Global rotation of the raw LiDAR point cloud around the Z-axis at a random angle is also adopted, where the random angle is sampled from $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

Training details. All experiments are conducted based on the OpenPCDet [31] toolbox. Following the work [26], we use the CT-Stack as the backbone model. Ablation experiments are trained on the *train* set from scratch, and then evaluated on the *val* set with 11 recall positions. We adopt the ADAM optimizer [10] with the learning rate set to 0.003. We train the model for 40 epochs with a batch size of 4 on NVIDIA 3090 GPU. Main results are collected from the *test* set, and then submitted to the server for performance evaluation and comparison with current state-of-the-art methods.

	$P(\theta)$	α	Car (IoU=0.7) Easy	Moderate	Hard
-ln(IoU)	0	—	88.61	78.12	77.27
GCIoU	$\alpha\theta$	1	88.86	78.19	77.22
		2	89.06	78.29	77.50
		3	89.19	78.43	77.62
	θ^α	1	89.38	78.62	77.52
		2	**89.54**	**78.81**	**78.39**
		3	89.15	78.33	77.41
		5	87.26	77.35	75.89

Table 2. Performance evaluation of different $P(\theta)$ in our method.

	$g(\theta)$	Car (IoU=0.7) Easy	Moderate	Hard
-ln(IoU)	—	88.61	78.12	77.27
GCIoU	θ	88.69	78.23	77.32
	$\tan\theta$	88.86	78.36	76.91
	$e^\theta - 1$	**88.98**	**78.53**	**77.75**

Table 3. Performance evaluation of different $g(\theta)$ in our method.

Methods	GCIoU	Car (IoU=0.7) Easy	Moderate	Hard
PointPillars [11]		88.95	77.35	76.11
	✓	**89.17**	**80.19**	**78.95**
SECOND [37]		88.91	78.85	77.62
	✓	**89.21**	**82.41**	**78.81**
SECOND-IoU [37]		**89.28**	79.21	78.36
	✓	89.26	**83.33**	**79.41**

Table 4. Performance improvements on different backbones.

5.2. Ablation Study

Evaluation of Gradient Correction Strategy. We perform ablation experiments on different modules of GCIoU loss as shown in Tab. 1. The baseline model trained with negative log IoU loss achieves the moderate AP of 78.12%. Gradient correction strategy consistently improves the performance, and achieves the moderate AP of 78.81%. Further, we conduct experiments to analyze the performance of functions for gradient correction in Tab. 2 and Tab. 3, respectively.

Experimental results in Tab. 2 show that performance gains are achieved for almost all of the different forms of $P(\theta)$ in GCIoU loss. Specifically, performance of $P_2(\theta) = \theta^\alpha$ is better than that of $P_1(\theta) = \alpha\theta$ as α=1 or α=2. Since $\frac{P_1'}{P_1} = \alpha$, while $\frac{P_2'}{P_2} = \alpha\theta^{\alpha-1}$. In Eq. 15, $\frac{P_2'}{P_2}$ dynamically adjusts thereby obtaining a larger interval of ideal gradient changes. Note that when α=5, moderate AP performance drops by 0.77 points. We suggest that a large α causes a sharp increase in the contribution of angular error in GCIoU loss. As a result, the model places too much emphasis on orientation regression and thus neglects centroid and shape prediction. The best performance is achieved when α=2, angle loss contribution and the gradient correction are well balanced in this case.

Further we conducted experiments to find the suitable $g(\theta)$. Results in Tab. 3 show that the best moderate AP is 78.53%, which is obtained when $g(\theta) = e^\theta - 1$. When $g_1(\theta) = \theta$, $g_1(\theta)' = 1$, which increases the gradient equally for all angular errors. Gradients are not well corrected, and thus the performance gain is negligible. For $g_2(\theta) = \tan\theta$, $g_2(\theta)' = \sec^2\theta$, it provides large gradients to accelerate model convergence at large angular errors. However, as $\theta \rightarrow \frac{\pi}{2}$, the gradients is too large and tends to make the regression process unstable. Finally, $g(\theta) = e^\theta - 1$ is a good trade-off and brings stable performance gains.

Evaluation of Gradient Rescaling Strategy. The results of the ablation experiments on the gradient rescaling strategy are shown in Tab. 1. Gradient rescaling strategy is a convenient plug-and-play module that directly adjusts the parameter update process. The application of gradient rescaling strategy improves the moderate AP by 0.42 points over the baseline IoU loss. Combined with the gradient correction strategy, we can obtain further performance improvements. Since the objects in the KITTI dataset do not have very obvious scale variation, so the performance gains are relatively small compared to the gradient correction strategy.

Evaluation of Different Backbones. We have also conducted experiments on different backbones to demonstrate the effectiveness of GCIoU loss. Experimental results are shown in Tab. 4. Experiments are conducted on three popular one-stage models PointPillars [11], SECOND [37], and SECOND-IoU [37]. SECOND-IoU is based on SECOND and extra IoU prediction is applied to achieve the consistent classification and regression predictions. Our method achieves significant performance gains on different backbone networks. For example, GCIoU loss improves moderate AP by 3.56 points on the KITTI *val* set on the basis of SECOND.

Comparison with Different IoU-based Losses. We further compare GCIoU loss with other IoU-based losses for 3D object detection. For a fair comparison, we reproduce the compared methods in OpenPCDet toolbox with the same backbone network. GIoU [25] and DIoU [48]are designed for 2D object detection, and we apply them to 3D object detection to improve performance. RIoU [47] adopts projections to approximate IoU between two 3D boxes, which suffers from inaccurate IoU calculation, and thus its hard AP even drops by 0.79 points. ODIoU [46] in-

Reg. Losses	Ref.	Car (IoU=0.7)		
		Easy	Moderate	Hard
1-IoU [49]	3DV2019	88.53	77.63	77.05
-ln(IoU) [49]	3DV2019	88.61	78.12	77.27
RIoU [47]	ECCV2020	88.42	77.81	76.26
GIoU [25]	CVPR2019	88.68	78.35	77.41
DIOU [48]	AAAI2020	89.16	78.78	77.95
ODIOU [46]	CVPR2021	89.32	78.92	78.01
RDIOU [26]	ECCV2022	89.71	79.61	78.40
GCIoU(ours)	—	**89.85**	**80.03**	**78.66**

Table 5. Performance comparison of different variants of 3D IoU losses on KITTI *val* set.

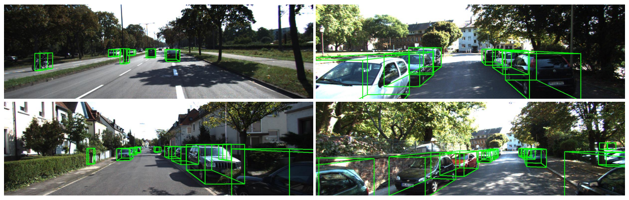

Figure 3. Visualization of some detections on KITTI *test* set.

troduces an additional orientation constraint based on DIoU loss, it obtains an improvement of 0.24 points on its basis. RDIoU [26] is the most recent approach, which decouples the angles to avoid the sub-optimal regression caused by rotation. RDIoU achieves a moderate AP of 79.61%. Our GCIoU loss solves the anomalous gradient variation, achieving fast and accurate model convergence, and therefore improves the performance of IoU-based losses. Finally, our method achieves the highest moderate AP of 80.03%.

5.3. Main Results on KITTI Dataset

We also compare the proposed method with some state-of-the-art models as listed in Tab. 6. We trained the model from scratch on *train* + *val* set, and then report results on *test* set on KITTI dataset. The model based on GCIoU loss achieves moderate AP of 80.43% and mAP of 81.47%, which outperforms the compared one-stage mehtods. Since our method is simple with minor modifications to the localization loss, and no complex data augmentation and pretraining strategies are applied, the final performance is still lower than some advanced detectors such as PV-RCNN [27] and PointFormer [20]. But GCIoU loss is general and flexible, and it can be applied to existing frameworks to achieve further performance gains. Some detections are shown in Fig. 3. The results illustrate that GCIoU loss helps to accurately locate objects in 3D scenes.

	Method	Reference	3D			
			Easy	Mod.	Hard	mAP
Two Stage	Part-A^2 [29]	TPAMI 2020	87.81	78.49	73.51	79.94
	STD [39]	ICCV 2019	87.95	79.71	75.09	80.92
	3D-CVF [40]	ECCV 2020	89.20	80.05	73.11	80.79
	PV-RCNN [27]	CVPR 2020	**90.25**	**81.43**	76.82	82.83
	PointFormer [20]	CVPR 2021	90.05	79.65	**78.89**	**82.86**
One Stage	SECOND	Sensors 2018	83.34	72.55	65.82	73.90
	PointPillars [11]	CVPR 2019	82.58	74.31	68.99	75.29
	TANet [14]	AAAI 2020	84.39	75.94	68.82	76.38
	HVPR [19]	CVPR 2021	86.38	77.92	73.04	79.11
	HotSpotNet [1]	ECCV 2020	87.60	78.31	73.34	79.75
	SA-SSD [7]	CVPR 2020	88.75	79.79	74.16	80.90
	CIA-SSD [45]	AAAI 2021	**89.59**	80.28	72.87	80.91
	SVGA-Net [8]	AAAI 2022	87.33	80.47	75.91	81.24
	IA-SSD [42]	CVPR 2022	88.87	80.32	75.10	81.43
	Ours	—	86.83	**80.83**	**76.77**	**81.47**

Table 6. Comparisons with some state-of-the-art methods of Car category on KITTI *test* set. We mark the best performance among one-stage detectors and two-stage detectors, respectively.

6. Conclusion and Limitation

In this paper, we discuss the issues of IoU based loss in 3D object detection. We demonstrate through experiments and mathematical proof that the gradients of 3D IoU loss w.r.t. angle error and the object scale change abnormally during training. These problems hinder the model convergence and degrade the detection performance. On this basis, a gradient-correction IoU loss is proposed to achieve fast and accurate angle convergence by optimizing the gradient. Then, a gradient rescaling strategy is adopted to adaptively adjust the optimization step size for objects of different sizes. Our method can be applied into existing models to achieve stable performance gains. Extensive experiments on the public KITTI dataset demonstrate its superiority.

However, there are certain limitations. Firstly, we only consider major cases of 3D IoU computation. A concise and unified way could be found in the future to calculate 3D IoU uniformly to avoid the gradient anomalies. Secondly, we set some assumptions when designing GCIoU loss, so the proposed loss function is not necessarily the optimal solution. This paper points out the potential drawbacks of IoU loss, and these issues could be future research directions.

Acknowledgement

The authors wish to thank Nicolas Vercheval and Aleksandra Pizurica for insightful discussions. Besides, we would also like to thank all members in GAIM group at Ghent University for their kind help and support.

References

[1] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *European conference on computer vision*, pages 68–84. Springer, 2020. 8

[2] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28, 2015. 6

[3] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1201–1209, 2021. 2

[4] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2918–2927, 2021. 1, 2, 3

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6

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

[7] 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. 2, 8

[8] Qingdong He, Zhengning Wang, Hao Zeng, Yi Zeng, and Yijun Liu. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 870–878, 2022. 8

[9] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–799, 2018. 2

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[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*, pages 12697–12705, 2019. 6, 7, 8

[12] Jiale Li, Hang Dai, Ling Shao, and Yong Ding. From voxel to point: Iou-guided 3d object detection for point cloud with voxel-to-point decoder. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4622–4631, 2021. 2, 3

[13] Zhidong Liang, Zehan Zhang, Ming Zhang, Xian Zhao, and Shiliang Pu. Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7140–7149, 2021. 2, 3

[14] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11677–11684, 2020. 8

[15] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4721–4730, 2021. 2

[16] Qi Ming, Lingjuan Miao, Zhiqiang Zhou, and Yunpeng Dong. Cfc-net: A critical feature capturing network for arbitrary-oriented object detection in remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 2

[17] Qi Ming, Lingjuan Miao, Zhiqiang Zhou, Junjie Song, Yunpeng Dong, and Xue Yang. Task interleaving and orientation estimation for high-precision oriented object detection in aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:241–255, 2023. 2

[18] Qi Ming, Zhiqiang Zhou, Lingjuan Miao, Hongwei Zhang, and Linhao Li. Dynamic anchor learning for arbitrary-oriented object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2355–2363, 2021. 2

[19] Jongyoun Noh, Sanghoon Lee, and Bumsub Ham. Hvpr: Hybrid voxel-point representation for single-stage 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14605–14614, 2021. 8

[20] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7463–7472, 2021. 8

[21] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393. IEEE, 2020. 1

[22] Hanyang Peng and Shiqi Yu. A systematic iou-related method: Beyond simplified regression for better localization. *IEEE Transactions on Image Processing*, 30:5032–5044, 2021. 6

[23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[25] 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/CVF conference*

on computer vision and pattern recognition, pages 658–666, 2019. 1, 2, 7, 8

[26] Hualian Sheng, Sijia Cai, Na Zhao, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Min-Jian Zhao, and Gim Hee Lee. Rethinking iou-based optimization for single-stage 3d object detection. *arXiv preprint arXiv:2207.09332*, 2022. 1, 2, 3, 6, 8

[27] 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. 3, 8

[28] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 2

[29] 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*, 43(8):2647–2664, 2020. 8

[30] 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. 2

[31] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 6

[32] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14615–14624, 2021. 2

[33] Li-Hua Wen and Kang-Hyun Jo. Three-attention mechanisms for one-stage 3-d object detection based on lidar and camera. *IEEE Transactions on Industrial Informatics*, 17(10):6655–6663, 2021. 1

[34] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[35] Hai Wu, Jinhao Deng, Chenglu Wen, Xin Li, Cheng Wang, and Jonathan Li. Casa: A cascade attention network for 3-d object detection from lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022. 1

[36] Shengkai Wu, Xiaoping Li, and Xinggang Wang. Iou-aware single-stage object detector for accurate localization. *Image and Vision Computing*, 97:103911, 2020. 2

[37] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 7

[38] Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. *Advances in Neural Information Processing Systems*, 34:18381–18394, 2021. 1

[39] 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/CVF international conference on computer vision*, pages 1951–1960, 2019. 2, 8

[40] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision*, pages 720–736. Springer, 2020. 1, 8

[41] 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. 1, 2

[42] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022. 8

[43] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3289–3298, 2021. 1

[44] Lin Zhao, Meiling Wang, and Yufeng Yue. Sem-aug: Improving camera-lidar feature fusion with semantic augmentation for 3d vehicle detection. *IEEE Robotics and Automation Letters*, 7(4):9358–9365, 2022. 1

[45] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 3555–3562, 2021. 2, 8

[46] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14494–14503, 2021. 7, 8

[47] Yu Zheng, Danyang Zhang, Sinan Xie, Jiwen Lu, and Jie Zhou. Rotation-robust intersection over union for 3d object detection. In *European Conference on Computer Vision*, pages 464–480. Springer, 2020. 2, 3, 7, 8

[48] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020. 1, 2, 7, 8

[49] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pages 85–94. IEEE, 2019. 1, 2, 3, 8

[50] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, pages 923–932. PMLR, 2020. 1