

# DCDet: Dynamic Cross-based 3D Object Detector

Shuai Liu, Boyang Li, Zhiyu Fang and Kai Huang\*

School of Computer Science and Engineering, Sun Yat-sen University

{liush376@mail2, liby83@mail, fangzhy9@mail2, huangk36@mail}.sysu.edu.cn

## Abstract

Recently, significant progress has been made in the research of 3D object detection. However, most prior studies have focused on the utilization of center-based or anchor-based label assignment schemes. Alternative label assignment strategies remain unexplored in 3D object detection. We find that the center-based label assignment often fails to generate sufficient positive samples for training, while the anchor-based label assignment tends to encounter an imbalanced issue when handling objects with different scales. To solve these issues, we introduce a *dynamic cross label assignment* (DCLA) scheme, which dynamically assigns positive samples for each object from a cross-shaped region, thus providing sufficient and balanced positive samples for training. Furthermore, to address the challenge of accurately regressing objects with varying scales, we put forth a *rotation-weighted Intersection over Union* (RWIoU) metric to replace the widely used  $L_1$  metric in regression loss. Extensive experiments demonstrate the generality and effectiveness of our DCLA and RWIoU-based regression loss. The Code is available at <https://github.com/Say2L/DCDet.git>.

## 1 Introduction

3D object detection plays a crucial role in enabling unmanned vehicles to perceive and understand their surroundings, which is fundamental for ensuring safe driving. Label assignment is a key process for training 3D object detectors. The dominant label assignment strategies in 3D object detection are anchor-based [Shi *et al.*, 2020a; Xu *et al.*, 2022; Zheng *et al.*, 2021] and center-based [Ge *et al.*, 2020; Hu *et al.*, 2022; Yin *et al.*, 2021; Wang *et al.*, 2021]. However, both of these label assignment schemes encounter issues that limit the performance of detectors.

*The anchor-based label assignment generally encounters an imbalanced problem when assigning positive samples to objects with different scales.* It employs the prior knowledge

of spatial scale for each category to predefine fixed-size anchors on the grid map. By comparing the intersection over union (IoU) between anchors and ground-truth boxes, positive anchors are determined to classify and regress objects. Consequently, the anchor-based label assignment tends to exhibit an uneven distribution of positive anchors across objects of different sizes. For example, car objects typically have a significantly higher number of positive anchors compared to pedestrian objects. This imbalance poses a challenge during training and leads to slow convergence for small objects. Moreover, the anchor-based label assignment scheme necessitates the recalculation of statistical data distribution for different datasets to obtain optimal anchor sizes. This requirement may reduce the robustness of a trained detector when applied to datasets with distinct data distributions.

*The center-based label assignment scheme often faces challenges in providing adequate positive samples for training.* This approach has recently been adopted by various 3D object detectors [Ge *et al.*, 2020; Hu *et al.*, 2022; Yin *et al.*, 2021; Wang *et al.*, 2021]. It focuses solely on object centers as positive samples (similar to positive anchors). As a result, the number of positive samples remains consistent across objects of different scales, solving the issue of imbalanced positive sample distribution encountered in anchor-based label assignment. However, the center-based label assignment overlooks many potential high-quality positive samples, as only one positive sample per object is responsible for regressing object attributes. This leads to an inefficient utilization of training data and sub-optimal network performance.

To simultaneously address the aforementioned challenges, this paper introduces a dynamic cross label assignment (DCLA), which aims to provide balanced and ample high-quality positive samples for objects of different scales. Specifically, DCLA dynamically assigns positive samples for each object within a cross-shaped region. The size of this region is determined by a distance parameter, which represents the Manhattan distance from the object’s center point. Given the varying scale and potential missing points in point clouds, a dynamic selection strategy is employed to adaptively choose positive samples from the cross-shaped region. As a result, each object is assigned sufficient positive samples, and objects of different scales receive a similar number of positive samples, effectively mitigating the issue of posi-

\*Corresponding author.

Appendix is presented at <https://arxiv.org/abs/2401.07240>.

tive sample imbalance.

Moreover, a rotation-weighted IoU (RWIoU) is introduced to accurately regress objects. In the 2D domain, the IoU-based loss [Rezatofghi *et al.*, 2019; Zheng *et al.*, 2020; Zhang *et al.*, 2022a] is confirmed to be better than the  $L_{norm}$  loss. However, in 3D object detection, the development of the IoU-based loss lags behind its 2D counterpart. This challenge arises due to the increased degrees of freedom in the 3D domain. The proposed RWIoU utilizes the idea of rotation weighting, thus elegantly integrating the rotation and direction attributes of objects into the IoU metric. The RWIoU loss can replace the  $L_{norm}$  and direction losses to help detectors achieve higher accuracy. Finally, a 3D object detection framework dubbed DCDet is proposed which combines the DCLA and RWIoU.

The contributions of this work are summarized as follows:

- We thoroughly investigate the current widely used label assignment strategies and analyze their pros and cons. Based on experimental observations, we introduce a new label assignment strategy called dynamic cross label assignment (DCLA).
- We propose a rotation-weighted IoU (RWIoU) to better measure the proximity of two rotation boxes compared to the  $L_1$  metric. RWIoU takes rotations and directions of 3D objects into consideration simultaneously.
- A 3D object detector dubbed DCDet is proposed which combines the DCLA and RWIoU. Extensive experiments on the Waymo Open [Sun *et al.*, 2020] and KITTI [Geiger *et al.*, 2012] datasets demonstrate the effectiveness and generality of our methods.

## 2 Related Work

### 2.1 3D Object Detection

VoxelNet [Zhou and Tuzel, 2018] encodes voxel features using PointNet [Qi *et al.*, 2017a], and then extracts features from 3D feature maps through 3D convolutions. SECOND [Yan *et al.*, 2018] efficiently encodes sparse voxel features by proposed 3D sparse convolution. PointPillars [Lang *et al.*, 2019] divides a point cloud into pillar voxels, avoiding the use of 3D convolution and achieving high inference speed. 3DSSD [Yang *et al.*, 2020] significantly improves inference speed by discarding upsampling layers and refinement networks commonly used in point-based methods. PointRCNN [Shi *et al.*, 2019] produces proposals from raw points using PointNet++ [Qi *et al.*, 2017b], and then refines bounding boxes in the second stage. PV-RCNN [Shi *et al.*, 2020a] uses features of internal points to refine proposals. Voxel RCNN [Deng *et al.*, 2021] replaces the features of raw points in the second-stage refinement with 3D voxel features in the 3D backbone.

### 2.2 Label Assignment

Label assignment, which is fundamental to 2D and 3D object detection, significantly influences the optimization of a network. Its development is more mature in 2D object detection, with RetinaNet [Lin *et al.*, 2017] assigning anchors

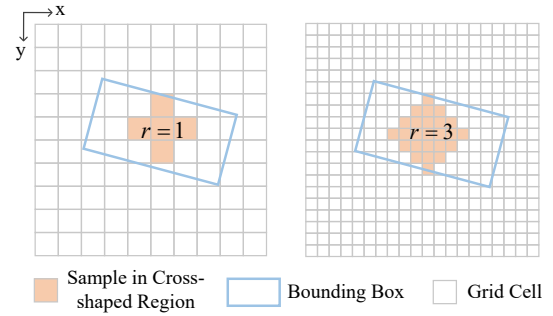


Figure 1: Cross-shaped region for different grid cell sizes.

on the output grid map, FCOS [Tian *et al.*, 2019] designating grid points within the range of ground truth boxes as positive samples, and CenterNet [Zhou *et al.*, 2019b] identifying center points of ground truth boxes as positive samples. ATSS [Zhang *et al.*, 2020] and AutoAssign [Zhu *et al.*, 2020] propose adaptive strategies for dynamic threshold selection and dynamic positive/negative confidence adjustment, respectively. YOLOX [Ge *et al.*, 2021] introduces the SimOTA scheme for dynamic positive sample selection. Conversely, 3D object detection label assignment is less developed, grappling with unique challenges such as maintaining a balance of positive samples across various object sizes. Current methods in 3D object detection typically use either anchor-based [Yan *et al.*, 2018; Lang *et al.*, 2019; Deng *et al.*, 2021] or center-based [Yin *et al.*, 2021; Ge *et al.*, 2020; Hu *et al.*, 2022] label assignment schemes. However, these schemes have drawbacks: the anchor-based label assignment often results in unbalanced assignments, and the center-based label assignment may overlook high-quality samples. To simultaneously overcome the above two drawbacks, we propose the dynamic cross label assignment (DCLA). Details about the DCLA are described in the methodology section.

### 2.3 IoU-based Loss

IoU-based losses [Rezatofghi *et al.*, 2019; Zheng *et al.*, 2020; Zhang *et al.*, 2022a] without rotation have been well studied in 2D object detection. These methods not only ensure consistency between the training objective and the evaluation metric but also normalize object attributes, leading to enhanced performance compared to the  $L_{norm}$  loss. Due to their success in 2D object detection, some 3D object detection methods [Zhou *et al.*, 2019a; Sheng *et al.*, 2022; Shi *et al.*, 2022] incorporate IoU-based losses. 3DIoU [Zhou *et al.*, 2019a] extends IoU calculation from 2D to 3D by considering rotation. However, the optimization direction of 3DIoU-based loss can be opposite to the correct direction. To address this, RDIoU [Sheng *et al.*, 2022] decouples rotation from 3DIoU. It considers rotation as an attribute similar to object location, but it doesn't consider object direction. A direction loss is needed for classifying object directions. ODIoU [Shi *et al.*, 2022] combines  $L_1$  metric and axis-aligned IoU to regress objects. Our proposed RWIoU incorporates both rotation and direction into the IoU metric, eliminating the need for  $L_{norm}$  and direction losses. Details of RWIoU will be explained in the next section.

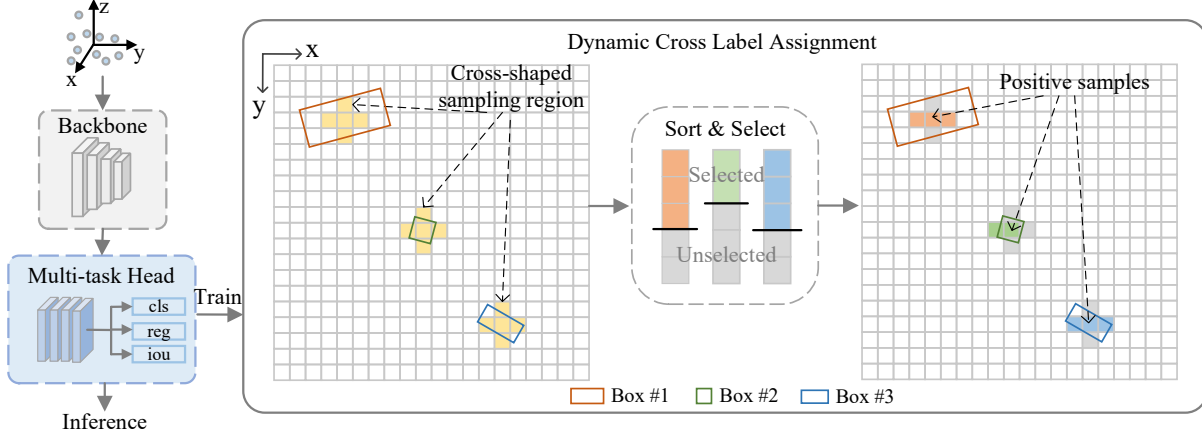


Figure 2: The overall framework of our DCDet. The dynamic cross label assignment scheme is only used in the training phase.

### 3 Methodology

This section will describe the dynamic cross label assignment (DCLA) and the rotation-weighted IoU (RWIoU) in detail. The overall framework is illustrated in Figure 2.

#### 3.1 Dynamic Cross Label Assignment

The label assignment schemes used in existing 3D object detection methods are generally based on prior information such as spatial ranges or object scales to manually select positive samples. For example, the anchor-based label assignment uses object scales to set the sizes of anchors and then uses anchors with IoU greater than a certain threshold as positive samples. The anchor-based label assignment generally produces unbalanced positive samples for different-scale objects, causing the model to prioritize large-scale objects. The center-based label assignment usually takes the center points of grounding truths as positive samples. This can result in a large number of good-quality samples being discarded, leaving inefficient utilization of training data.

The above label assignment schemes have a common property, they all use static prior information as the selection criteria. And the prior information is determined by human experience. Dynamic label assignment schemes [Zhang *et al.*, 2020; Zhu *et al.*, 2020; Ge *et al.*, 2021] have shown their advantages in 2D object detection. However, directly transferring these schemes to 3D object detection is not trivial. There are some challenges: 1) There is no space to dynamically select positive samples for small objects (e.g. pedestrians). Because small objects generally cover one or two grid points on the output map; 2) The coverage of objects with different scales varies greatly. This easily results in an imbalance of positive samples between different scale objects.

To dynamically select sufficient high-quality positive samples while maintaining the balance between different scale objects, we propose a dynamic cross label assignment (DCLA) scheme. Specifically, it limits the positive sampling range in a cross-shape region for each object. Typically, an object’s center region on a feature map contains enough features to identify it [Tian *et al.*, 2019], and objects in point clouds have regular shapes. Therefore, we only use the center

point and its surrounding points for positive sampling in the DCLA scheme. We refer to this sampling range as the cross region. It can be adjusted by a parameter  $r$  to adapt to outputs with different grid cell sizes as illustrated in Figure 1.  $r$  is the Manhattan distance away from the center point. When  $r = 1$ , the cross region covers the center and its top, down, left, and right neighbors. And when  $r = 0$ , the DCLA degenerates to the center-based label assignment.

The implementation steps of the DCLA are described in detail next. Given a ground truth  $\mathbf{b}^t$  and positions  $P$  in its cross region, calculate the selection cost as follows:

$$c_j = L_j^{cls} + \lambda_{reg} L_j^{reg}, j \in P, \quad (1)$$

where the  $L_j^{cls}$  and  $L_j^{reg}$  are the classification loss and regression loss between the ground truth  $\mathbf{b}^t$  and  $j$ -th prediction  $\mathbf{b}_j^o$  respectively, and  $\lambda_{reg}$  is the weight of regression loss. Then, sort the predictions in the cross region according to the selection costs. Next, sum the IoUs between the ground truth  $\mathbf{b}_i^t$  and predictions  $\mathbf{b}_j^o, j \in P$ :

$$k = \max\left(\left[\sum_{j \in P} \text{IoU}(\mathbf{b}_i^t, \mathbf{b}_j^o)\right], 1\right). \quad (2)$$

We utilize  $k$  as the number of positive samples for ground truth  $\mathbf{b}^t$ . Finally, select the top  $k$  predictions as positive samples. And the rest predictions are negative samples.

Specifically, given a point cloud input and the ground truth boxes  $\{\mathbf{b}_1^t, \mathbf{b}_2^t, \dots, \mathbf{b}_n^t\}$ , we assume that  $f(\mathbf{b}_i^t, \mathbf{b}_{ij}^o)$  represents the regression loss function, where  $\mathbf{b}_i^t$  and  $\mathbf{b}_{ij}^o$  denote the  $i$ -th ground truth box and its  $j$ -th predicted box, respectively. Therefore, the regression loss  $\ell$  for the point cloud is calculated as follows:

$$\ell = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{k_i} f(\mathbf{b}_i^t, \mathbf{b}_{ij}^o), \quad (3)$$

$$N = \sum_{i=1}^n \sum_{j=1}^{k_i} 1,$$

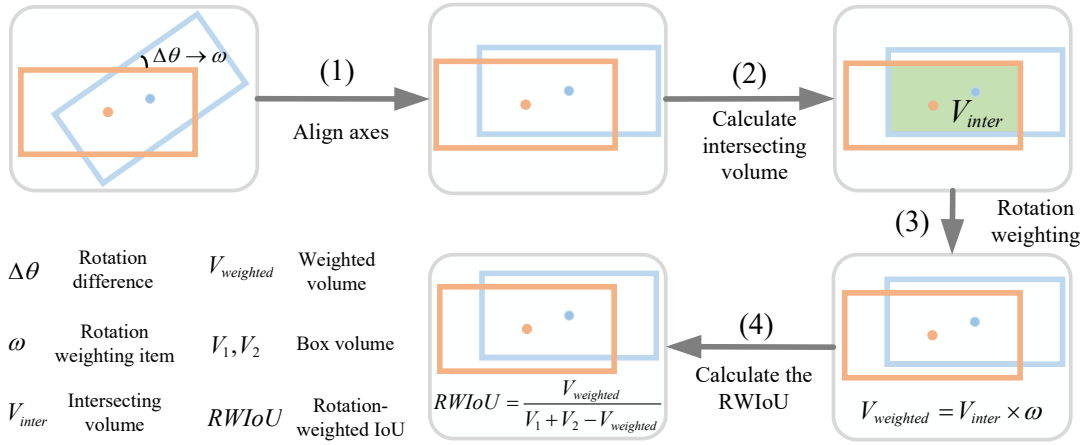


Figure 3: The calculation process of RWIoU.

where  $N$  represents the total number of positive samples in the input point cloud, and  $k_i$  denotes the number of positive samples assigned for the ground truth  $b_i^l$ . Notably,  $k_i$  is calculated independently for each ground truth, as in Eq. (2). It is related to the number of high-quality samples in the cross region and is not dependent on the ground truth scale. However, in the anchor-based label assignment,  $k_i$  varies significantly with the ground truth scale, resulting in a bias towards large-scale objects in the loss. For the center-based label assignment,  $k_i$  is always equal to 1, leading to inefficient utilization of training data.

We adopt the heatmap target for the classification task. The weights of positive samples are set to 1, and the weights of negative samples in cross regions are set to the values of IoU between predicted boxes and ground-truth boxes. As for the rest negative samples, the weights are all set to 0.

### 3.2 Rotation-Weighted IoU

In general, different object categories exhibit significant scale variations, and various attributes such as location, size, and rotation also possess scale differences. Many existing methods employ the  $L_{norm}$  loss as the regression loss. However, this loss function renders the model sensitive to differences in both object and attribute scales. Consequently, large objects and attributes dominate the total loss. The IoU metric can normalize object attributes, making it immune to scale differences. Moreover, the optimization objective of the IoU-based loss aligns with the evaluation metrics of detection models. Therefore, substituting the  $L_{norm}$  loss with the IoU-based loss often yields accuracy improvement [Zheng *et al.*, 2020; Rezatofghi *et al.*, 2019; Sheng *et al.*, 2022].

Utilizing IoU-based loss in 3D object detection poses several challenges. Firstly, calculating traditional IoU requires the computation of polyhedron volumes, which is a complex and computationally expensive task. Secondly, the traditional IoU-based loss, due to its tight coupling with rotation, can lead to misdirection in optimization, resulting in training instability [Sheng *et al.*, 2022]. Lastly, integrating the traditional IoU metric with object directions is not trivial. Therefore, the inclusion of  $L_1$  loss or direction loss becomes nec-

essary to aid models in classifying object directions.

To tackle the aforementioned challenges, we propose a rotation-weighted IoU (RWIoU). It thoroughly decouples the rotation from the IoU calculation, making the computation similar to the axis-aligned IoU computation. RWIoU can be implemented with just a few lines of code. By integrating sine and cosine values of rotations of objects into a rotation weighting item, our RWIoU can penalize rotation and direction errors simultaneously.

The RWIoU calculation process is shown in Figure 3. It first considers two rotation boxes  $B_1$  and  $B_2$  as axis-aligned boxes, and then calculates the intersecting volume of the two axis-aligned boxes as follows:

$$\begin{aligned}
 s_L &= \max(x_1 - l_1/2, x_2 - l_2/2), \\
 s_R &= \min(x_1 + l_1/2, x_2 + l_2/2), \\
 s_B &= \max(y_1 - w_1/2, y_2 - w_2/2), \\
 s_T &= \min(y_1 + w_1/2, y_2 + w_2/2), \\
 s_D &= \max(z_1 - h_1/2, z_2 - h_2/2), \\
 s_U &= \min(z_1 + h_1/2, z_2 + h_2/2), \\
 V_{inter} &= \max(s_R - s_L, 0) \times \max(s_T - s_B, 0) \\
 &\quad \times \max(s_U - s_D, 0),
 \end{aligned} \tag{4}$$

where  $(x_i, y_i, z_i)$ ,  $i \in \{1, 2\}$  denote the locations of box centers,  $(l_i, w_i, h_i)$ ,  $i \in \{1, 2\}$  represent the sizes of boxes, and  $V_{inter}$  denotes the intersecting volume of two axis-aligned boxes. Then, we update the  $V_{inter}$  according to the rotation difference of the two boxes as follows:

$$\begin{aligned}
 V_{weighted} &= \omega V_{inter}, \\
 \omega &= \omega_s \omega_c, \\
 \omega_s &= (1 - \alpha \frac{|\sin\theta_2 - \sin\theta_1|}{2}), \\
 \omega_c &= (1 - \alpha \frac{|\cos\theta_2 - \cos\theta_1|}{2}),
 \end{aligned} \tag{5}$$

where  $\theta_1$  and  $\theta_2$  represent rotations of two boxes,  $\omega_s$  and  $\omega_c$  denote the sine and cosine rotation error factor respectively that are all normalized to the range of  $[0, 1]$ ,  $\omega$  represents the

rotation weighting item,  $V_{weighted}$  is the rotation-weighted value of  $V_{inter}$ ,  $\alpha \in [0, 1]$  is a hyper-parameter which is used to control the contribution of rotation to the RWIoU. If  $\alpha = 0$ , the RWIoU degrades to axis-aligned IoU. After obtaining  $V_{weighted}$ , the value of RWIoU can be calculated as follows:

$$\begin{aligned} V_{union} &= V_1 + V_2 - V_{weighted}, \\ RWIoU &= \frac{V_{weighted}}{V_{union}}, \end{aligned} \quad (6)$$

where  $V_1$  and  $V_2$  represent the volumes of two boxes, respectively. The gradient analysis of the RWIoU is in Appendix.

### 3.3 Loss Function

Single-stage detectors typically encounter misalignment between classification confidence and localization accuracy. To solve the misalignment problem, we follow Zheng *et al.* to introduce an extra IoU prediction branch. The classification loss  $L_{cls}$  and IoU prediction loss  $L_{iou}$  are the same as those of CIA-SSD [Zheng *et al.*, 2021].

The regression loss  $L_{reg}$  is based on the RWIoU. It is calculated as follows:

$$L_{reg} = \frac{1}{N} \sum_{i=1}^N 1 - RWIoU_i + \left( \frac{D_i}{Diag_i} \right)^2, \quad (7)$$

where  $N$  is the total number of positive samples,  $RWIoU_i$  and  $D_i$  represent the RWIoU value and the  $L_2$  distance of centers, respectively. Additionally,  $Diag_i$  denotes the diagonal length of the minimal enclosing rectangle of the  $i$ -th predicted box and its ground truth. The term  $\frac{D_i}{Diag_i}$  is used to optimize the prediction of center locations. Since our RWIoU incorporates sine and cosine functions to represent the rotation angle of a bounding box, the need for a direction loss is eliminated. The overall loss function is calculated as follows:

$$L = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + \lambda_{iou} L_{iou}, \quad (8)$$

where  $\lambda_{cls}$ ,  $\lambda_{reg}$ , and  $\lambda_{dir}$  are the weight of classification, regression, and direction losses, respectively.

## 4 Experiments

In this section, we evaluate models on widely-used 3D object detection benchmark datasets including Waymo Open [Sun *et al.*, 2020] and KITTI [Geiger *et al.*, 2012].

### 4.1 Implementation Setup

#### Data Preprocessing

For the Waymo Open dataset, the detection range is  $[-74.88, 74.88]m$  for the  $X$  and  $Y$  axes and  $[-2, 4]m$  for the  $Z$  axis, the voxel size is set to  $(0.08, 0.08, 0.15)m$ . For the KITTI dataset, the detection range is  $[0, 70.4]m$  for the  $X$  axis,  $[-40, 40]m$  for the  $Y$  axis, and  $[-5, 3]m$  for the  $Z$  axis, the voxel size is set to  $(0.05, 0.05, 0.1)m$ .

### Training Details

The backbone of our DCDet is the same as that of CenterPoint [Yin *et al.*, 2021]. Following PillarNeXt [Li *et al.*, 2023], we use a feature upsampling in the detection head of DCDet, which increase the output resolution with only a little overhead. All models are trained from scratch in an end-to-end manner with the Adam optimizer and a 0.003 learning rate. And the parameter  $\alpha$  used in Eq. (4) is set to 0.5. The parameters  $\lambda_{cls}$  and  $\lambda_{iou}$  used in Eq. (7) are all set to 1. And the parameter  $\lambda_{reg}$  used in Eq. (1) and Eq. (7) is set to 3. For the Waymo Open and KITTI datasets, the parameter  $r$  used in DCLA is set to 1 and 3, respectively. On the Waymo Open and KITTI datasets, models are trained for 30 epochs with a batch size of 24 and 80 epochs with a batch size of 8, respectively. Hyper-parameters analysis is in Appendix.

### 4.2 Comparison with State-of-the-Art Methods

The baseline models presented in Table 1 primarily utilize either center-based or anchor-based label assignment. Moreover, they commonly employ  $L_{norm}$  regression loss. As depicted in Table 1, the center-based label assignment demonstrates a significant advantage over the anchor-based label assignment on the Waymo Open dataset. Nevertheless, our DCDet, featuring a lightweight single-stage network, surpasses the state-of-the-art center-based method DSVT, which employs a heavy backbone network. Notably, even our DCDet model trained on only 20% of the training samples outperforms both the center-based and anchor-based methods trained on the entire dataset. These results demonstrate the superior performance of our DCDet framework which employs DCLA and RWIoU-based regression loss.

We also evaluated our DCDet on the Waymo Open test set by submitting the results to the official server. The performance comparisons are presented in Table 2, revealing that our DCDet surpasses previous state-of-the-art methods significantly. Particularly, in the case of small-scale categories such as pedestrians and cyclists, our method demonstrates a substantial advantage due to the balanced and sufficient positive samples provided by DCLA.

### 4.3 Effect on Different Backbone Networks

To assess the generality of our DCLA and RWIoU, we conduct experiments by incorporating them into several widely used backbone networks, namely SECOND, PillarNet, and DSVT. All models are reproduced using the OpenPCDet [Team, 2020] codebase. We train these models using both 20% and 100% of the training data from the Waymo Open dataset and present the results in Table 3. As evident from the table, the integration of our DCLA and RWIoU yields significant improvements across all model groups. This underscores the generality and effectiveness of our proposed DCLA and RWIoU techniques. Notably, the DCLA and RWIoU-based regression loss belong to the learning strategies of models, resulting in cost-free improvements. Even when trained on only 20% of the training data, the models integrated with our DCLA and RWIoU techniques either surpass or catch up to the performance of models trained on the entire training data without these enhancements. This demonstrates that our learning strategies enhance the utilization of training data,

| Method  | Stages | LEVEL 2          |                  |                  | LEVEL 1          |                  |                  | LEVEL 2          |  |  |
|---|--------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|--|--|
|   |        | mAP/mAPH         | Vehicle          | Pedestrian       | Cyclist          | Vehicle          | Pedestrian       | Cyclist          |  |  |
| LiDAR R-CNN (a) [Li <i>et al.</i> , 2021]     | 2      | 65.8/61.3        | 76.0/75.5        | 71.2/58.7        | 68.6/66.9        | 68.3/67.9        | 63.1/51.7        | 66.1/64.4        |  |  |
| Part-A2-Net (a) [Shi <i>et al.</i> , 2020b]   | 2      | 66.9/63.8        | 77.1/76.5        | 75.2/66.9        | 68.6/67.4        | 68.5/68.0        | 66.2/58.6        | 66.1/64.9        |  |  |
| Voxel R-CNN† (a) [Deng <i>et al.</i> , 2021]  | 2      | 68.6/66.2        | 76.1/75.7        | 78.2/72.0        | 70.8/69.7        | 68.2/67.7        | 69.3/63.6        | 68.3/67.2        |  |  |
| PV-RCNN† (c) [Shi <i>et al.</i> , 2020a]      | 2      | 69.6/67.2        | 78.0/77.5        | 79.2/73.0        | 71.5/70.3        | 69.4/69.0        | 70.4/64.7        | 69.0/67.8        |  |  |
| PV-RCNN++† (c) [Shi <i>et al.</i> , 2023]     | 2      | 71.7/69.5        | 79.3/78.8        | 81.8/76.3        | 73.7/72.7        | 70.6/70.2        | 73.2/68.0        | 71.2/70.2        |  |  |
| FSD [Fan <i>et al.</i> , 2022b]               | 2      | 72.9/70.8        | 79.2/78.8        | 82.6/77.3        | 77.1/76.0        | 70.5/70.1        | 73.9/69.1        | 74.4/73.3        |  |  |
| SECOND* (a) [Yan <i>et al.</i> , 2018]        | 1      | 61.0/57.2        | 72.3/71.7        | 68.7/58.2        | 60.6/59.3        | 63.9/63.3        | 60.7/51.3        | 58.3/57.0        |  |  |
| PointPillars* (a) [Lang <i>et al.</i> , 2019] | 1      | 62.8/57.8        | 72.1/71.5        | 70.6/56.7        | 64.4/62.3        | 63.6/63.1        | 62.8/50.3        | 61.9/59.9        |  |  |
| IA-SSD (a) [Zhang <i>et al.</i> , 2022b]      | 1      | 66.8/63.3        | 70.5/69.7        | 69.4/58.5        | 67.7/65.3        | 61.6/61.0        | 60.3/50.7        | 65.0/62.7        |  |  |
| SST* (a) [Fan <i>et al.</i> , 2022a]          | 1      | 67.8/64.6        | 74.2/73.8        | 78.7/69.6        | 70.7/69.6        | 65.5/65.1        | 70.0/61.7        | 68.0/66.9        |  |  |
| CenterPoint‡ (c) [Yin <i>et al.</i> , 2021]   | 1      | 68.2/65.8        | 74.2/73.6        | 76.6/70.5        | 72.3/71.1        | 66.2/65.7        | 68.8/63.2        | 69.7/68.5        |  |  |
| VoxSet (c) [He <i>et al.</i> , 2022]          | 1      | 69.1/66.2        | 74.5/74.0        | 80.0/72.4        | 71.6/70.3        | 66.0/65.6        | 72.5/65.4        | 69.0/67.7        |  |  |
| PillarNet (c) [Shi <i>et al.</i> , 2022]      | 1      | 71.0/68.5        | 79.1/78.6        | 80.6/74.0        | 72.3/66.2        | 70.9/70.5        | 72.3/66.2        | 69.7/68.7        |  |  |
| AFDetV2 (c) [Hu <i>et al.</i> , 2022]         | 1      | 71.0/68.8        | 77.6/77.1        | 80.2/74.6        | 73.7/72.7        | 69.7/69.2        | 72.2/67.0        | 71.0/70.1        |  |  |
| CenterFormer (c) [Zhou <i>et al.</i> , 2022]  | 1      | 71.1/68.9        | 75.0/74.4        | 78.6/73.0        | 72.3/71.3        | 69.9/69.4        | 73.6/68.3        | 69.8/68.8        |  |  |
| SwinFormer (c) [Sun <i>et al.</i> , 2022]     | 1      | -/-              | 77.8/77.3        | 80.9/72.7        | -/-              | 69.2/68.8        | 72.5/64.9        | -/-              |  |  |
| PillarNeXt (c) [Li <i>et al.</i> , 2023]      | 1      | 71.9/69.7        | 78.4/77.9        | 82.5/77.1        | 73.2/72.2        | 70.3/69.8        | 74.9/69.8        | 70.6/69.6        |  |  |
| DSVT (Pillar) (c) [Wang <i>et al.</i> , 2023] | 1      | 73.2/71.0        | 79.3/78.8        | 82.8/77.0        | 76.4/75.4        | 70.9/70.5        | 75.2/69.8        | 73.6/72.7        |  |  |
| DCDet (20%) (ours)                            | 1      | 74.0/71.5        | 79.2/78.7        | 83.8/77.6        | 77.4/76.3        | 71.0/70.6        | 76.2/70.2        | 74.8/73.7        |  |  |
| DCDet (ours)                                  | 1      | <b>75.0/72.7</b> | <b>79.5/79.0</b> | <b>84.1/78.5</b> | <b>79.4/78.3</b> | <b>71.6/71.1</b> | <b>76.7/71.3</b> | <b>76.8/75.7</b> |  |  |

Table 1: Performance comparisons on the Waymo Open validation set. The results of AP/APH are reported. \*: reported by [Fan *et al.*, 2022b]. †: reported by [Shi *et al.*, 2023]. ‡: reported by [Wang *et al.*, 2023]. ‘a’ and ‘c’ denote the anchor-based and center-based label assignment, respectively. ‘20%’ denotes only 20% training samples are used.

| Method                                  | LEVEL 2          |                  |                  | LEVEL 1          |                  |                  | LEVEL 2          |  |  |
|---|------------------|------------------|------------------|------------------|------------------|------------------|------------------|--|--|
|   | mAP/mAPH         | Vehicle          | Pedestrian       | Cyclist          | Vehicle          | Pedestrian       | Cyclist          |  |  |
| CenterPoint [Yin <i>et al.</i> , 2021]  | -                | 80.2/79.7        | 78.3/72.1        | -                | 72.2/71.8        | 72.2/66.4        | -                |  |  |
| PV-RCNN [Shi <i>et al.</i> , 2020a]     | 71.2/68.8        | 80.6/80.2        | 78.2/72.0        | 71.8/70.4        | 72.8/72.4        | 71.8/66.1        | 69.1/67.8        |  |  |
| PillarNet-18 [Shi <i>et al.</i> , 2022] | 71.3/68.5        | 81.9/81.4        | 80.0/72.7        | 68.0/66.8        | 74.5/74.0        | 74.0/67.1        | 65.5/64.4        |  |  |
| AFDetV2 [Hu <i>et al.</i> , 2022]       | 72.2/70.0        | 80.5/80.0        | 79.8/74.4        | 72.4/71.2        | 73.0/72.6        | 73.7/68.6        | 69.8/68.7        |  |  |
| PV-RCNN++ [Shi <i>et al.</i> , 2023]    | 72.4/70.2        | 81.6/81.2        | 80.4/75.0        | 71.9/70.8        | 73.9/73.5        | 74.1/69.0        | 69.3/68.2        |  |  |
| DCDet (ours)                            | <b>75.7/73.3</b> | <b>82.2/81.7</b> | <b>83.4/77.8</b> | <b>77.3/76.1</b> | <b>74.8/74.4</b> | <b>77.5/72.1</b> | <b>74.7/73.5</b> |  |  |

Table 2: Performance comparisons on the Waymo Open test set by submitting to the official test evaluation server. The results are achieved by using single point cloud frames. No test-time augmentations are used.

which is particularly valuable considering the high cost associated with labeling 3D bounding boxes.

#### 4.4 Ablation Study

To further study the influence of each component of DCDet, we perform a comprehensive ablation analysis on the Waymo Open and KITTI datasets. For the Waymo Open dataset, following prior works [Shi *et al.*, 2020a; Wang *et al.*, 2023], models are trained on 20% training samples and evaluated on the whole validation samples. For the KITTI dataset, models are trained on the *train* set and evaluated on the *val* set.

##### Effect of RWIoU and DCLA

The baseline model adopts center-based label assignment and  $L_1$  regression loss. To evaluate the effectiveness of our proposed methods, we systematically integrate RWIoU-based regression loss and DCLA into the baseline model. The ablation results are presented in Table 4. We observe a notable performance improvement when incorporating RWIoU-based regression loss, as demonstrated by the results in the 1<sup>st</sup> and 2<sup>nd</sup> rows of Table 4. This suggests that the proposed loss function is better suited for the task of 3D object detection compared to the traditional  $L_1$  loss. Furthermore, models

trained with DCLA consistently achieve significantly better performance than the baseline, as illustrated in the 1<sup>st</sup> and 3<sup>rd</sup> rows of Table 4. This indicates that DCLA facilitates improved utilization of the available training data, thus enhancing the overall model performance. Notably, when both RWIoU-based regression loss and DCLA are used, the model achieves the highest performance among all evaluated models. These findings validate the effectiveness of our proposed methods and highlight the importance of carefully designing the loss function and label assignment for improving the performance of 3D object detectors.

##### Comparison with Other Regression Losses

Table 5 provides a comparison of different regression losses. All models utilize the DCLA scheme and the same backbone network. The results in the 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> rows of Table 5 reveal marginal differences between the  $L_1$ , RDIOU-based [Sheng *et al.*, 2022], and ODIOU-based [Shi *et al.*, 2022] regression losses. However, our RWIoU-based loss exhibits a significant performance improvement compared to the other regression losses, as demonstrated in the 4<sup>th</sup> row of Table 5. These results highlight the effectiveness of our RWIoU, which decouples the rotation from IoU calcula-

| Method        | Training Data | LEVEL 1   |           |             |             | LEVEL 2    |           |             |             |
|---------------|---------------|-----------|-----------|-------------|-------------|------------|-----------|-------------|-------------|
|               |               | mAP/mAPH  | Vehicle   | Pedestrian  | Cyclist     | mAP/mAPH   | Vehicle   | Pedestrian  | Cyclist     |
| SECOND        | 20%           | 64.8/60.4 | 70.9/70.3 | 65.8/54.8   | 57.8/56.2   | 58.7/54.7  | 62.6/62.0 | 57.8/48.0   | 55.7/54.2   |
| SECOND*       | 20%           | 73.4/70.0 | 74.0/73.3 | 77.0/69.1   | 69.2/67.7   | 67.1/64.0  | 65.7/65.2 | 68.7/61.3   | 66.9/65.4   |
| Improvement ↑ | N/A           | +8.6/+9.6 | +3.1/+3.0 | +11.2/+14.3 | +11.4/+11.5 | +8.4/+9.3  | +3.1/+3.2 | +10.9/+13.3 | +11.2/+11.2 |
| PillarNet     | 20%           | 71.6/68.0 | 72.9/72.3 | 73.0/64.1   | 68.9/67.6   | 65.6/62.3  | 64.9/64.4 | 65.3/57.2   | 66.5/65.2   |
| PillarNet*    | 20%           | 75.1/70.9 | 75.6/75.0 | 78.1/67.7   | 71.7/70.0   | 69.0/65.1  | 67.8/67.3 | 70.0/60.4   | 69.2/67.6   |
| Improvement ↑ | N/A           | +3.5/+2.9 | +2.7/+2.7 | +5.1/+3.6   | +2.8/+2.4   | +3.4/+2.8  | +2.9/+2.9 | +4.7/+3.2   | +2.7/+2.4   |
| DSVT          | 20%           | 78.3/75.3 | 78.1/77.6 | 82.3/74.8   | 74.6/73.5   | 72.2/69.3  | 69.8/69.3 | 74.7/67.7   | 72.0/71.0   |
| DSVT*         | 20%           | 79.8/76.5 | 79.2/78.7 | 83.6/75.3   | 76.5/75.4   | 73.7/70.6  | 71.1/70.7 | 76.2/68.3   | 73.9/72.8   |
| Improvement ↑ | N/A           | +1.5/+1.2 | +1.1/+1.1 | +1.3/+0.5   | +1.9/+1.9   | +1.5/+1.3  | +1.3/+1.4 | +1.5/+0.6   | +1.9/+1.8   |
| SECOND        | 100%          | 67.2/63.1 | 72.3/71.7 | 68.7/58.2   | 60.6/59.3   | 61.0/57.2  | 63.9/63.3 | 60.7/51.3   | 58.3/57.1   |
| SECOND*       | 100%          | 74.2/71.0 | 74.4/73.8 | 78.4/70.8   | 69.9/68.5   | 68.0/65.1  | 66.3/65.9 | 70.2/63.2   | 67.5/66.1   |
| Improvement ↑ | N/A           | +7.0/+7.9 | +2.1/+2.1 | +9.7/+12.6  | +9.3/+9.2   | +7.0/+11.9 | +2.4/+2.6 | +9.5/+12.9  | +9.2/+9.0   |
| PillarNet     | 100%          | 73.4/70.0 | 74.0/73.5 | 75.3/66.9   | 70.8/69.6   | 67.4/64.3  | 66.2/65.7 | 67.7/60.0   | 68.3/67.1   |
| PillarNet*    | 100%          | 75.7/71.9 | 75.8/75.3 | 79.1/69.7   | 72.2/70.7   | 69.7/66.1  | 68.2/67.6 | 71.1/62.4   | 69.8/68.4   |
| Improvement ↑ | N/A           | +2.3/+1.9 | +1.8/+1.8 | +3.8/+2.8   | +1.4/+1.1   | +2.3/+1.8  | +2.0/+1.9 | +3.4/+2.4   | +1.5/+1.3   |
| DSVT          | 100%          | 80.1/77.4 | 79.1/78.6 | 82.7/76.3   | 78.4/77.3   | 73.8/71.3  | 70.9/70.5 | 75.0/68.9   | 75.6/74.6   |
| DSVT*         | 100%          | 81.5/78.7 | 80.4/79.9 | 84.5/77.4   | 79.7/78.6   | 75.7/72.9  | 72.6/72.1 | 77.2/70.4   | 77.2/76.2   |
| Improvement ↑ | N/A           | +1.4/+1.3 | +1.3/+1.3 | +1.8/+1.1   | +1.3/+1.3   | +1.9/+1.6  | +1.7/+1.6 | +2.2/+1.5   | +1.6/+1.6   |

Table 3: Effect on different backbone networks. The results of AP/APH on the Waymo Open validation set are reported. ‘\*’ represents that our DCLA and RWIoU-based regression loss are applied.

| <i>RWIoU</i> | <i>DCLA</i> | Vehicle          | Pedestrian       | Cyclist          |
|--------------|-------------|------------------|------------------|------------------|
|              |             | 69.2/68.7        | 73.4/68.5        | 72.6/71.5        |
| ✓            |             | 69.9/69.3        | 74.3/68.5        | 74.1/73.1        |
|              | ✓           | 70.5/70.0        | 75.2/69.7        | 74.4/73.3        |
| ✓            | ✓           | <b>71.0/70.5</b> | <b>75.9/70.1</b> | <b>75.1/74.0</b> |

Table 4: Effect of different components of DCDet. *RWIoU* and *DCLA* denote RWIoU-based regression loss and dynamic cross label assignment, respectively. The LEVEL 2 AP/APH results on the Waymo Open validation set are reported.

tion by introducing rotation weighting. Notably, the RDIoU-based loss necessitates an additional direction classification loss, and the ODIoU-based loss requires an extra  $L_1$  loss. In contrast, our RWIoU-based loss is a pure IoU-based loss without any auxiliary losses. This simplification allows our approach to achieve superior performance without introducing additional complexity.

### Comparison with Other Label Assignment Schemes

Table 6 compares different label assignment schemes with all models using the RWIoU-based regression loss and the same backbone network. As depicted in the 1<sup>st</sup> and 3<sup>rd</sup> rows of Table 6, both anchor-based and box-based label assignment exhibit subpar performance when it comes to small objects like pedestrians and cyclists. This is mainly due to the unbalanced assignment of positive samples for objects with different scales. On the other hand, the center-based label assignment, as shown in the 2<sup>nd</sup> row of Table 6, achieves good results on the Waymo Open dataset but performs poorly on the KITTI dataset. We argue that this discrepancy arises from overlooking a large number of excellent samples, resulting in an insufficient number of positive samples for training on small-scale datasets like KITTI. Moreover, the poor performance of simOTA [Ge *et al.*, 2021] in 3D object detection, as demonstrated in the 4<sup>th</sup> row of Table 6, highlights the challenges of directly transferring methods from the 2D domain to the 3D domain. However, our DCLA outperforms these baseline label assignment schemes on both the Waymo Open and

| Regression Loss | Vehicle          | Pedestrian       | Cyclist          |
|-----------------|------------------|------------------|------------------|
| $L_1$           | 70.3/69.8        | 75.0/69.6        | 74.0/73.0        |
| RDIoU-based     | 70.2/69.7        | 74.8/69.3        | 74.3/73.2        |
| ODIoU-based     | 70.5/70.0        | 75.2/69.7        | 74.4/73.3        |
| RWIoU-based     | <b>71.0/70.5</b> | <b>75.9/70.1</b> | <b>75.1/74.0</b> |

Table 5: Comparison results of different regression losses. The LEVEL 2 AP/APH results on the Waymo Open validation set are reported.

| Label Assignment | Waymo            |                  |                  | KITTI        |
|------------------|------------------|------------------|------------------|--------------|
|                  | Vehicle          | Pedestrian       | Cyclist          | Mod. Car     |
| Anchor-based     | 67.8/67.3        | 63.4/55.5        | 67.7/66.5        | 85.37        |
| Center-based     | 69.9/69.3        | 74.3/68.5        | 74.1/73.1        | 75.49        |
| Box-based        | 67.8/67.4        | 66.2/61.4        | 69.9/69.0        | 85.32        |
| simOTA           | 68.7/68.3        | 67.8/63.1        | 72.2/71.2        | 85.45        |
| DCLA             | <b>71.0/70.5</b> | <b>75.9/70.1</b> | <b>75.1/74.0</b> | <b>85.82</b> |

Table 6: Comparison results of different label assignment schemes. The LEVEL 2 AP/APH results on the Waymo Open validation set and moderate AP<sub>R40</sub> results on the KITTI *val* are reported.

KITTI datasets, as illustrated in the last row of Table 6. This confirms that our DCLA can adapt to datasets of different scales by enabling balanced and adequate positive sampling.

## 5 Conclusion

In this paper, we propose a dynamic cross label assignment (DCLA), which dynamically assigns positive samples from a cross-shaped region for each object. The DCLA scheme mitigates the imbalanced issue in the anchor-based assignment and the loss of high-quality samples in the center-based assignment. Thanks to the balanced and adequate positive sampling, DCLA effectively adapts to different scale datasets. Moreover, a rotation-weighted IoU (RWIoU), which considers the rotation and direction in a weighting way, is introduced to measure the proximity of two rotation boxes. Extensive experiments conducted on various datasets demonstrate the generality and effectiveness of our methods.

## Acknowledgments

This work is supported by the Project of Guangxi Key R & D Program (No. GuikeAB24010324).

## References

- [Deng *et al.*, 2021] 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 *AAAI*, 2021.
- [Fan *et al.*, 2022a] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *CVPR*, 2022.
- [Fan *et al.*, 2022b] Lue Fan, Feng Wang, Naiyan Wang, and ZHAO-XIANG ZHANG. Fully sparse 3d object detection. In *NeurIPS*, 2022.
- [Ge *et al.*, 2020] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. Afdet: Anchor free one stage 3d object detection. *arXiv preprint arXiv:2006.12671*, 2020.
- [Ge *et al.*, 2021] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YoloX: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.
- [Geiger *et al.*, 2012] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.
- [He *et al.*, 2022] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *CVPR*, 2022.
- [Hu *et al.*, 2022] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *AAAI*, 2022.
- [Lang *et al.*, 2019] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.
- [Li *et al.*, 2021] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *CVPR*, 2021.
- [Li *et al.*, 2023] Jinyu Li, Chenxu Luo, and Xiaodong Yang. Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds. In *CVPR*, 2023.
- [Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.
- [Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.
- [Rezatofighi *et al.*, 2019] 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 *CVPR*, 2019.
- [Sheng *et al.*, 2022] 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. In *ECCV*, 2022.
- [Shi *et al.*, 2019] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019.
- [Shi *et al.*, 2020a] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020.
- [Shi *et al.*, 2020b] 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. *TPAMI*, 2020.
- [Shi *et al.*, 2022] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *ECCV*, 2022.
- [Shi *et al.*, 2023] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *IJCV*, 2023.
- [Sun *et al.*, 2020] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwala, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.
- [Sun *et al.*, 2022] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *ECCV*, 2022.
- [Team, 2020] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [Tian *et al.*, 2019] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.
- [Wang *et al.*, 2021] Qi Wang, Jian Chen, Jianqiang Deng, and Xinfang Zhang. 3d-centernet: 3d object detection network for point clouds with center estimation priority. *Pattern Recognition*, 2021.
- [Wang *et al.*, 2023] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023.
- [Xu *et al.*, 2022] Qiangeng Xu, Yiqi Zhong, and Ulrich Neumann. Behind the curtain: Learning occluded shapes for 3d object detection. In *AAAI*, 2022.



- [Yan *et al.*, 2018] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018.
- [Yang *et al.*, 2020] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 2020.
- [Yin *et al.*, 2021] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021.
- [Zhang *et al.*, 2020] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.
- [Zhang *et al.*, 2022a] Yi-Fan Zhang, Weiqiang Ren, Zhang Zhang, Zhen Jia, Liang Wang, and Tieniu Tan. Focal and efficient iou loss for accurate bounding box regression. *Neurocomputing*, 2022.
- [Zhang *et al.*, 2022b] 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 *CVPR*, 2022.
- [Zheng *et al.*, 2020] 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 *AAAI*, 2020.
- [Zheng *et al.*, 2021] 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 *AAAI*, 2021.
- [Zhou and Tuzel, 2018] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018.
- [Zhou *et al.*, 2019a] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *3DV*, 2019.
- [Zhou *et al.*, 2019b] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.
- [Zhou *et al.*, 2022] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *ECCV*, 2022.
- [Zhu *et al.*, 2020] Benjin Zhu, Jianfeng Wang, Zhengkai Jiang, Fuhang Zong, Songtao Liu, Zeming Li, and Jian Sun. Autoassign: Differentiable label assignment for dense object detection. *arXiv preprint arXiv:2007.03496*, 2020.