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
{lilush376@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-
tive sample imbalance.

Moreover, a rotation-weighted IoU (RWIoU) is introduced to accurately regress objects. In the 2D domain, the IoU-based loss [Rezatofighi et al., 2019; Zheng et al., 2020; Zhang et al., 2022a] is confirmed to be better than the $L_{\text{norm}}$ 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 $L_{\text{norm}}$ 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 R-CNN [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 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 [Rezatofighi 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 $L_{\text{norm}}$ 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 $L_{\text{norm}}$ and direction losses. Details of RWIoU will be explained in the next section.

![Figure 1: Cross-shaped region for different grid cell sizes.](image-url)
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, leading to 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 $b^i$ and positions $P$ in its cross region, calculate the selection cost as follows:

$$c_j = L^{cls}_j + \lambda_{reg} L^{reg}_j, j \in P,$$

where $L^{cls}_j$ and $L^{reg}_j$ are the classification loss and regression loss between the ground truth $b^i$ and $j$-th prediction $b^j$, 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 $b^i$ and predictions $b^j$, $j \in P$:

$$k = \max(\sum_{j \in P} \text{IoU}(b^i, b^j), 1).$$

We utilize $k$ as the number of positive samples for ground truth $b^i$. 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 $\{b^i_1, b^i_2, \cdots, b^i_n\}$, we assume that $f(b^i_j, b^i_\ell)$ represents the regression loss function, where $b^i_j$ and $b^i_\ell$ 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:

$$\ell = \frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{k_i} f(b^i_j, b^i_\ell),$$

$$N = \sum_{i=1}^{n} k_i.$$
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 \( b_i^j \). 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; Rezatofighi 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 necessary 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:

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

(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:

\[
\begin{align*}
    V_{weighted} &= \omega V_{inter}, \\
    \omega &= \omega_s \omega_c, \\
    \omega_s &= (1 - \alpha \frac{|\sin \theta_2 - \sin \theta_1|}{2}), \\
    \omega_c &= (1 - \alpha \frac{|\cos \theta_2 - \cos \theta_1|}{2}), \\
\end{align*}
\]

(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, $V_{\text{weighted}}$ is the rotation-weighted value of $V_{\text{inter}}$, $\alpha \in [0, 1]$ 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 $V_{\text{weighted}}$, the value of RWIoU can be calculated as follows:

$$V_{\text{union}} = V_1 + V_2 - V_{\text{weighted}},$$

$$\text{RWIoU} = \frac{V_{\text{weighted}}}{V_{\text{union}}},$$

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 $L_{\text{cls}}$ and IoU prediction loss $L_{\text{iou}}$ are the same as those of CIA-SSD [Zheng et al., 2021].

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

$$L_{\text{reg}} = \frac{1}{N} \sum_{i=1}^{N} 1 - \text{RWIoU}_i + \left(\frac{D_i}{\text{Diag}_i}\right)^2,$$

where $N$ is the total number of positive samples, $\text{RWIoU}_i$ and $D_i$ represent the RWIoU value and the $L_2$ distance of centers, respectively. Additionally, $\text{Diag}_i$ denotes the diagonal length of the minimal enclosing rectangle of the $i$-th predicted box and its ground truth. The term $\frac{D_i}{\text{Diag}_i}$ 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:

$$L = \lambda_{\text{cls}}L_{\text{cls}} + \lambda_{\text{reg}}L_{\text{reg}} + \lambda_{\text{iou}}L_{\text{iou}},$$

where $\lambda_{\text{cls}}$, $\lambda_{\text{reg}}$, and $\lambda_{\text{dir}}$ 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 $\lambda_{\text{cls}}$ and $\lambda_{\text{iou}}$ used in Eq. (7) are all set to 1. And the parameter $\lambda_{\text{reg}}$ 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 $L_{\text{norm}}$ 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.
compared to the traditional function is better suited for the task of 3D object detection regression loss, as demonstrated by the results in the 1st row of Table 4. We observe a notable performance improvement when incorporating RWIoU-based regression loss and DCLA into the baseline model. The ablation results are presented in Table 5. These results highlight 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.

Effect of RWIoU and DCLA

The baseline model adopts center-based label assignment and L1 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 1st and 2nd 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 L1 loss. Furthermore, models trained with DCLA consistently achieve significantly better performance than the baseline, as illustrated in the 1st and 3rd 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 1st, 2nd, and 3rd rows of Table 5 reveal marginal differences between the L1, 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 4th row of Table 5. These results highlight the effectiveness of our RWIoU, which decouples the rotation from IoU calculations.

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.
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 scales and moderate AP\textsuperscript{val} results on the Waymo Open validation set are reported.

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\textsuperscript{st} and 3\textsuperscript{rd} 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\textsuperscript{nd} 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\textsuperscript{th} 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 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Data</th>
<th>LEVEL 1</th>
<th>LEVEL 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP/mAPH</td>
<td>Vehicle</td>
<td>Pedestrian</td>
</tr>
<tr>
<td>SECOND</td>
<td>20% 64.8/60.4</td>
<td>70.9/70.3</td>
<td>63.8/53.4</td>
</tr>
<tr>
<td>SECOND*</td>
<td>20% 73.4/70.0</td>
<td>74.0/73.3</td>
<td>77.0/69.1</td>
</tr>
<tr>
<td></td>
<td>Improvement†  N/A +8.6/+9.6</td>
<td>+3.1/+3.0</td>
<td>+11.2/+14.3</td>
</tr>
<tr>
<td>PillarNet</td>
<td>20% 71.8/68.0</td>
<td>72.9/71.3</td>
<td>73.0/64.1</td>
</tr>
<tr>
<td>PillarNet*</td>
<td>20% 71.5/70.9</td>
<td>75.6/75.0</td>
<td>78.1/67.7</td>
</tr>
<tr>
<td></td>
<td>Improvement†  N/A +3.5/+2.9</td>
<td>+2.7/+2.7</td>
<td>+5.1/+4.6</td>
</tr>
<tr>
<td>DSVT</td>
<td>20% 78.3/75.3</td>
<td>78.1/77.6</td>
<td>82.3/74.8</td>
</tr>
<tr>
<td>DSVT*</td>
<td>20% 79.8/76.5</td>
<td>79.2/78.7</td>
<td>83.6/75.3</td>
</tr>
<tr>
<td></td>
<td>Improvement†  N/A +1.5/+1.2</td>
<td>+1.6/+1.1</td>
<td>+1.3/+0.5</td>
</tr>
<tr>
<td>SECOND</td>
<td>100% 67.2/63.1</td>
<td>72.3/71.7</td>
<td>66.7/58.2</td>
</tr>
<tr>
<td>SECOND*</td>
<td>100% 74.2/71.0</td>
<td>74.4/73.8</td>
<td>78.4/70.8</td>
</tr>
<tr>
<td></td>
<td>Improvement†  N/A +7.0/+7.9</td>
<td>+2.1/+2.1</td>
<td>+9.7/+12.6</td>
</tr>
<tr>
<td>PillarNet</td>
<td>100% 73.4/70.0</td>
<td>74.0/73.5</td>
<td>75.3/66.9</td>
</tr>
<tr>
<td>PillarNet*</td>
<td>100% 75.7/71.9</td>
<td>75.8/75.3</td>
<td>79.0/69.7</td>
</tr>
<tr>
<td></td>
<td>Improvement†  N/A +2.3/+1.9</td>
<td>+1.8/+1.8</td>
<td>+3.8/+2.8</td>
</tr>
<tr>
<td>DSVT</td>
<td>100% 80.1/77.4</td>
<td>79.1/78.6</td>
<td>82.7/76.3</td>
</tr>
<tr>
<td>DSVT*</td>
<td>100% 81.5/78.7</td>
<td>80.4/79.9</td>
<td>84.5/77.4</td>
</tr>
<tr>
<td></td>
<td>Improvement†  N/A +1.4/+1.3</td>
<td>+1.3/+1.3</td>
<td>+1.8/+1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RWIoU</th>
<th>DCLA</th>
<th>Vehicle</th>
<th>Pedestrian</th>
<th>Cyclist</th>
</tr>
</thead>
<tbody>
<tr>
<td>69.2/68.7</td>
<td>73.4/68.5</td>
<td>72.6/71.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ 69.9/69.3</td>
<td>74.3/68.5</td>
<td>74.1/73.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ 70.5/70.0</td>
<td>75.2/69.7</td>
<td>74.4/73.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓ ✓ 71.0/70.5</td>
<td>75.9/70.1</td>
<td>75.1/74.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Regression Loss</th>
<th>Vehicle</th>
<th>Pedestrian</th>
<th>Cyclist</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDIoU-based</td>
<td>70.2/69.7</td>
<td>74.8/69.3</td>
<td>74.3/73.2</td>
</tr>
<tr>
<td>ODIoU-based</td>
<td>70.5/70.0</td>
<td>75.2/69.7</td>
<td>74.4/73.3</td>
</tr>
<tr>
<td>RWIoU-based</td>
<td>71.0/70.5</td>
<td>75.9/70.1</td>
<td>75.1/74.0</td>
</tr>
</tbody>
</table>

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.

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.

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

Table 6: Comparison results of different regression losses. The LEVEL 2 AP/APH results on the Waymo Open validation set are reported.

Table 6: Comparison results of different label assignment schemes. The LEVEL 2 AP/APH results on the Waymo Open validation set are reported and moderate AP\textsuperscript{val} results on the KITTI validation set are reported.
Acknowledgments
This work is supported by the Project of Guangxi Key R & D Program (No. GuikeAB24010324).

References


[Hu et al., 2022] Yihan Hu, Zhuangzhuan Ding, Runzhou Ge, Wensin 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.


[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.


[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.


