AFDetV2: Rethinking the Necessity of the Second Stage for Object Detection from Point Clouds

Yihan Hu*, Zhuangzhuang Ding*, Runzhou Ge*
Wenxin Shao, Li Huang, Kun Li, Qiang Liu
Horizon Robotics
{yihan.hu96, dinghouzx, runzhouge, proliu}@gmail.com

Abstract

There have been two streams in the 3D detection from point clouds: single-stage methods and two-stage methods. While the former is more computationally efficient, the latter usually provides better detection accuracy. By carefully examining the two-stage approaches, we have found that if appropriately designed, the first stage can produce accurate box regression. In this scenario, the second stage mainly rescores the boxes such that the boxes with better localization get selected. From this observation, we have devised a single-stage anchor-free network that can fulfill these requirements. This network, named AFDetV2, extends the previous work by incorporating a self-calibrated convolution block in the backbone, a keypoint auxiliary supervision, and an IoU prediction branch in the multi-task head. We take a simple product of the predicted IoU score with the classification heatmap to form the final classification confidence. The enhanced backbone strengthens the box localization capability, and the rescoring approach effectively joins the object presence confidence and the box regression accuracy. As a result, the detection accuracy is drastically boosted in the single-stage. To evaluate our approach, we have conducted extensive experiments on the Waymo Open Dataset and the nuScenes Dataset. We have observed that our AFDetV2 achieves the state-of-the-art results on these two datasets, superior to all the prior arts, including both the single-stage and the two-stage 3D detectors. AFDetV2 won the 1st place in the Real-Time 3D Detection of the Waymo Open Dataset Challenge 2021. In addition, a variant of our model AFDetV2-Base was entitled the “Most Efficient Model” by the Challenge Sponsor, showing a superior computational efficiency. To demonstrate the generality of this single-stage method, we have also applied it to the first stage of the two-stage networks. Without exception, the results show that with the strengthened backbone and the rescoring approach, the second stage refinement is no longer needed.

1 Introduction

Object detection from point clouds has become a practical solution to robotics vision, especially in autonomous driving applications. Like the detection methods on 2D images, the 3D detection methods can also be divided into two groups: single-stage (Ge et al. 2020; Zhou and Tuzel 2018; Yan, Mao, and Li 2018; He et al. 2020; Zheng et al. 2021; Lang et al. 2019; Bewley et al. 2020; Fan et al. 2021; Chen et al. 2020a) and two-stage (Shi, Wang, and Li 2019; Yang et al. 2019; Qi et al. 2017; Shi et al. 2020c; Li, Wang, and Wang 2021; Shi et al. 2021; Deng et al. 2021a; Yin, Zhou, and Krahenbuhl 2021a; Sun et al. 2021), in terms of the model structure. The two-stage methods usually show better accuracy (Shi, Wang, and Li 2019; Shi et al. 2021; Deng et al. 2021a) in the classification confidence and the box regression, than the single-stage methods. In these methods, the first stage provides proposals of the bounding boxes, based on which the second stage pools feature and runs through a smaller network to classify it and refine the box regression. The features utilized by the second stage can be extracted from the feature maps (voxel-based) produced by the backbone (Deng et al. 2021a; Yin, Zhou, and Krahenbuhl 2021a), or from the raw point cloud (point-based) which need to go through another round of feature encoding (Yang et al. 2019; Shi, Wang, and Li 2019).

Why is the second stage needed? One argument is that the point features may recover the loss of positioning information, due to voxelization, striding operations, or lack of receptive field. Another argument is that the category classification and the box regression are usually realized in two separate branches, so the confidence map from the classification branch may not align well with the localization accuracy. These arguments are somewhat supported by the evidence that the second stage does boost detection accuracy.

But we wonder whether the above arguments mean the necessity of a second stage. For example, is the raw point necessary for a precise positioning? Some recent two-stage methods (Deng et al. 2021a; Yin, Zhou, and Krahenbuhl 2021a) suggest that voxel-based features could achieve the same positioning accuracy without using the point-based features (Shi, Wang, and Li 2019). What the second stage provides is to refine the classification score and enhance box regression with additional computational blocks. However, after a careful examination, we have found that the first stage is already capable of producing accurate box localization, when designed properly (Sec. 3.1). The actual contribution from the second stage lies in the enhancement of the classification scores. In another word, the second stage may not refine the positions of the boxes; instead, it rescores the clas-
sification confidence to better select the boxes.

This observation is consistent with the argument of classification-localization misalignment, in that the boxes with better IoU may be associated with a better score at the second stage. But, “can one resolve the misalignment problem within a single-stage approach?”. In fact, some previous works also asked the same question. For example, a recent 2D detection algorithm (Zhang et al. 2021) proposes to learn an IoU-aware classification score to join the object presence confidence with the localization accuracy. And, a 3D detection algorithm (Zheng et al. 2021) proposes an IoU regression branch, the output of which is combined with the classification score for the rescoring.

Our work is closely related to (Zheng et al. 2021) with a few major differences: first, our single-stage approach is an anchor-free network, extended from the 2020 award-winning work AFDet (Ge et al. 2020), while (Zheng et al. 2021) is anchor-based; second, we deploy a self-calibrated convolutional block with multi-scalability and spatial attention (Liu et al. 2020) in our backbone to enlarge the receptive field and improve the semantics, which is simple to plug into an existing network; third, to further enhance the box regression, we employ an auxiliary loss on the 4-corners and the center (i.e. keypoints) of the 3D boxes; and fourth, we conduct extensive experiments with both single-stage and two-stage approaches, to reveal that refining the classification confidence and box regression in a second stage is unnecessary. To align the classification score with the box regression, we have also utilized an IoU prediction branch, but with a modified rescoring formula to combine the classification heatmap and the IoU prediction map.

To demonstrate the effectiveness of our single-stage 3D detector, which is named AFDetV2, we have compared with other state-of-the-art models, including both single-stage and two-stage methods, on the Waymo Open Dataset (WOD) and the nuScenes Dataset. On these two datasets, all the experimental results show that AFDetV2 is superior to the prior arts. Our AFDetV2 with the test set accuracy of 73.12 APH/L2 and the latency of 60.06 ms won the 1st place in the Real-Time 3D Detection of the WOD Challenge 2021. In addition, a variant of our model AFDetV2-Base with the accuracy of 72.57 APH/L2 and latency of 55.86 ms was entitled the “Most Efficient Model” by the WOD Challenge Sponsor, showing a superior computational efficiency.

Also, we show that by plugging in the proposed components, i.e. self-calibrated block, keypoint auxiliary loss, and the IoU prediction branch, to the first stage of a two-stage approach, such as (Deng et al. 2021a), one can discard the second stage and still achieve similar or even better detection accuracy.

For the maximum paper length limit, refer to the complete arXiv version of this paper for more comparisons, results and details.

2 Related Work

2.1 Two-Stage/Singe-Stage LiDAR Detector

Inspired by 2D detection (Girshick 2015; Ren et al. 2016; Cai and Vasconcelos 2018), a two-stage LiDAR detector usually generates Region of Interests (RoIs) at the first stage, followed by a second stage to refine the first stage predictions. PointRCNN (Shi, Wang, and Li 2019) and STD (Yang et al. 2019) apply R-CNN style detector from 2D to 3D domain. After generating coarse 3D bounding box proposals using PointNet++ (Qi et al. 2017), point features within 3D proposals are directly pooled to second stage for refinement. However, different proposals might end up pooling the same group of points, which loses the ability to encode the geometric information of the proposals. To tackle this prob-
lem, Part-A (Shi et al. 2020c) designed an RoI-aware point cloud pooling operation, while LiDAR R-CNN (Li, Wang, and Wang 2021) devised a method with virtual point and boundary offset. Another point pooling method proposed by PV-RCNN (Shi et al. 2021) summarizes learned point and voxel-wise feature volumes at multiple neural layers into a small set of keypoint, then the keypoint features are aggregated according to RoI-grid. Pyramid R-CNN (Mao et al. 2021a) utilizes an RoI-grid pyramid to mitigate the sparsity problem. Instead of pooling from point features, Voxel R-CNN(Deng et al. 2021a) designs voxel-RoI pooling module to directly pool from voxel and BEV feature space according to the RoI-grid. To speed up, CenterPoint (Yin, Zhou, and Krahenbuhl 2021a) simplifies the pooling module by sampling five keypoints from BEV features using bilinear interpolation. Recently, RSN (Sun et al. 2021) utilizes a foreground segmentation as a first stage to sparsify the point clouds, which boosts the efficiency of the second stage sparse convolution.

For single-stage LiDAR detectors, VoxelNet (Zhou and Tuzel 2018) encodes the point cloud data as 3D voxels and uses 3D convolution to extract 3D features. However, 3D convolution is computationally expensive. Considering the sparsity nature of the point clouds, SECOND (Yan, Mao, and Li 2018) utilizes 3D sparse convolution to speed up the 3D convolution. To speed up the encoding process, PointPillars (Lang et al. 2019) encodes 3D point cloud data as BEV pillars, then conventional 2D CNN can be applied to the pseudo image. CIA-SSD (Zheng et al. 2021) utilizes an IoU prediction branch and a post-processing method to incorporate localization accuracy to confident scores. AF-Det (Ge et al. 2020, 2021; Ding et al. 2020; Wang et al. 2020b), which served as the base detector for several 1st place winner solutions of Waymo Open Dataset Challenge 2020, proposes an anchor-free and NMS free LiDAR detection framework for the first time. Recently, some works focus on the representation of the different views. Bewley et al.; Fan et al. exploit LiDAR detection in the range view. Chen et al.; Zhou et al. fuse the complementary information from the range view and the BEV.

2.2 Anchor-Free/Anchor-Based LiDAR Detector
The idea of anchor first appears in Faster R-CNN (Ren et al. 2016). Inspired by Ren, multiple works (Lang et al. 2019; Yan, Mao, and Li 2018; Zhou and Tuzel 2018; Kuang et al. 2020; Zhu et al. 2019; Qi et al. 2019; Zhou et al. 2020; Chen et al. 2017; Noh, Lee, and Ham 2021) utilize anchors to improve the performance of their LiDAR 3D detection networks. Different from 2D networks, anchors are extended into 3D space with a z-axis value. Yang et al. proposes a new spherical anchor to seed each point more efficiently and propose objects with higher recall. Shi et al. investigates both anchor-free and anchor-based strategies, in which anchor-based strategy achieves higher recall rates sacrificing memory and calculation costs. To improve the efficiency of networks, some works remove the process of anchor assignment. PointRCNN (Shi, Wang, and Li 2019) uses a foreground segmentation network to propose objects which can significantly reduce the number of proposals. PIXOR (Yang, Luo, and Urtasun 2018) assigns all pixels inside ground truth bounding boxes as positive samples in the BEV feature map. Similarly, some range-view-based methods assign pixels on range view maps that are inside 3D ground truth boxes as positive samples (Fan et al. 2021; Meyer et al. 2019; Bewley et al. 2020). Multiple anchor-free detection networks formulate the detection problem as a keypoint detection problem (Ge et al. 2020; Sun et al. 2021; Yin, Zhou, and Krahenbuhl 2021b). All features are acquired according to the “centerness” of objects so anchors can be deprecated. 3D-MAN (Yang et al. 2021) adopted an anchor-free single-stage detector as the first module of its multi-frame architecture to generate high-quality proposals.

Compared to the two-stage and anchor-based 3D detectors, the single-stage and anchor-free 3D detector has a much simpler structure and higher speed, which is more suitable for real-time deployment. Here we propose AF-DetV2, a fast and accurate anchor-free single-stage 3D detector, which surpasses all two-stage detectors in terms of accuracy and speed.

3 Methods
In this section, we first analyze the two-stage methods, focusing on the necessity of the second stage. Then we present our single-stage method.

3.1 Analysis on Two-stage Methods
There are two main causes for a second stage: (1) the RoI proposal from the first stage confines the spatial range so the PointNet based feature extraction can be afforded, which may restore the 3D context in better resolutions; (2) the second stage, as an extra computational block dedicated to an RoI, can refine the classification score and the box regression.
There are studies countering the former argument. In (Deng et al. 2021a), the authors compare the point-based methods, e.g. (Shi, Wang, and Li 2019; Yang et al. 2019; Shi et al. 2020b), to the voxel-based methods, e.g. (Shi et al. 2020c; Deng et al. 2021a), and show that with small voxel size (0.1m, 0.1m, 0.15m), the voxel-based methods can achieve similar accuracy as the point-based. Also, a more recent method (Yin, Zhou, and Krahenbuhl 2021a,b) demonstrates the state-of-the-art performances on multiple renowned datasets by using voxel-based features. These all suggest that the second stage may not necessarily provide a better spatial resolution than the first stage.

For the latter, there may be three factors of the possible improvements: the box regression, the classification score, and the alignment between these two. To differentiate the contributions of these factors, we conduct an experiment: we add the classification and the box regression branches to a bare single-stage network in turn and examined the corresponding improvements of the final average precision (AP). We find that adding the box regression alone in the second stage do not show any significance. But adding the score refinement in the second stage does improve the final AP significantly (Tab. 7). This suggests that the second stage may not necessarily enhance the box regression. Instead, it can enhance the classification confidence such that the box with better regression gets a higher score, and vice versa.

So the second stage can enhance the classification score, by properly pulling the context of the features and applying an extra computational block. We wonder if, by properly strengthening the feature extraction of the first stage, we may achieve a similar enhancement. With this thought, we conduct another experiment: we replace the convolutional block of the backbone in (Deng et al. 2021a) with a self-calibrated convolution block (Liu et al. 2020), to strengthen the semantics of the features. We observe that this simple replacement greatly improve the overall AP (Tab. 8). Although still slightly lower than the improvement of using the second stage, this result motivates a further experiment: adding an IoU alignment to the classification. So on top of the backbone enhancement, we add an IoU prediction branch to the first stage. And we combine the IoU prediction score with the classification score by a simple product (Eq. 1). This experiment wins the arguments against the second stage.

With the above observations, we believe a single-stage 3D detector can be as accurate as the current state-of-the-art two-stage 3D detectors. We present our single-stage framework in details in the next few sections.

## 3.2 Backbone

Our overall network structure follows the previous work AFDet (Ge et al. 2020), in which an anchor-free single-stage 3D detector is proposed. The whole network takes a point cloud input and voxelizes it. The voxel features are then sent into a 3D feature extractor (Ge et al. 2021), followed by the backbone, and finally into an anchor-free detection head. In this section, we present the details of the first three steps. We present anchor-free detection head in Sec. 3.3.

**Encoder** We first voxelize (Zhou and Tuzel 2018; Zhu et al. 2019) points into small voxels across 3D space. For this step, the position of voxels is determined by pre-defined grid size. In each voxel, the mean of all points is calculated and is used as the representative value. Thus all points with coordinate are quantized into fixed voxels.

After voxelization, inputs are sent into a 3D Feature Extractor. The 3D Feature Extractor is composed of 3D sparse convolutional layers (Yan, Mao, and Li 2018; Ge et al. 2021) and sub-manifold sparse convolutional layers (Graham and van der Maaten 2017). The extractor is designed to have fewer residual blocks with slightly more channels at the early stage. We set the stride of z-axis dimension to 8 to be more efficient. At the end of the 3D Feature Extractor, the resulting feature map is reshaped to form a BEV pseudo image.

**Self-calibrated convolutional backbone** After 3D feature extraction, the feature map is sent into a multi-stage backbone. The backbone has two stages and each stage has a convolutional layer and three blocks. For the second stage, the feature is downsampled and a transposed convolutional layer is used to restore the scale after the stage. To fully explore the potential of the single-stage framework, we apply the Self-Calibrated Convolutions (SC-Conv) (Liu et al. 2020) for each block. SC-Conv block efficiently enlarges the receptive field and adds channel-wise and spatial-wise attention, which increases the detection accuracy without sacrificing the computational costs. The structure of the SC-Conv block are shown in Fig. 2.

### 3.3 Anchor-Free Head

In addition to the five sub-heads introduced in AFDet (Ge et al. 2020), we devise an IoU-aware confidence score prediction, which is a key to removing the second stage. We also employ a keypoint auxiliary loss to add additional supervision, as shown in Fig. 1. The 5 common sub-heads in both AFDet and AFDetV2 are the heatmap prediction head, the location offset regression head, the z-axis location regression head, the 3D object size regression head, and the orientation regression head. Following (Yin, Zhou, and Krahenbuhl 2021a), the minimum allowed Gaussian radius (Law and Deng 2018) for the heatmap and keypoint head is set to 2. The regression target is set to \( \sin \) and \( \cos \) values of the object yaw angle for the orientation regression sub-head.

**IoU-aware confidence score prediction** The classification score is commonly used as the final prediction score in object detection tasks. However, the classification score lacks the localization information, which is not a good confidence estimate for object detection. Specifically, boxes with high localization accuracy but low classification scores may be deleted after Non-Maximum Suppression. Also, the misalignment harms the ranking-based metrics such as Average Precision. To alleviate the misalignment, most of the existing methods adopt an IoU-aware prediction branch in the second stage network (Jiang et al. 2018; Shi et al. 2020a; Yin, Zhou, and Krahenbuhl 2021a). However, an additional stage will increase the computational cost and the latency of the network. Also, special operators such as RoI Align (He
Table 1: The configurations of different variants of our model for Waymo Open Dataset. “L” and “LT” mean “all LiDARs” and “top-LiDAR only”, separately. “Ens” is short for ensemble. “Server Latency” is the latency measured by the official testing server in milliseconds. Values in training range, inference range and grid size columns are in meters with respect to \( x, y, z \)-axes, respectively. Our AFDetV2-Ens uses Test-Time Augmentation and model ensemble for better detection accuracy and is suitable for offboard usage (e.g. auto labeling). The other 3 variants are non-ensemble, real-time and suitable for onboard usage.

<table>
<thead>
<tr>
<th>Variants</th>
<th>Sensors</th>
<th>Frames</th>
<th>Usage</th>
<th>Scenario</th>
<th>Server</th>
<th>Training Range</th>
<th>Inference Range</th>
<th>Grid Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFDetV2-Lite</td>
<td>LT</td>
<td>1</td>
<td>X</td>
<td>Onboard</td>
<td>46.90</td>
<td>±75.2 ±75.2</td>
<td>- ±75.2 ±75.2</td>
<td>0.1 0.1 0.15</td>
</tr>
<tr>
<td>AFDetV2-Base</td>
<td>LT</td>
<td>2</td>
<td>X</td>
<td>Onboard</td>
<td>55.86</td>
<td>±75.2 ±75.2</td>
<td>- ±75.2 ±75.2</td>
<td>0.1 0.1 0.15</td>
</tr>
<tr>
<td>AFDetV2-Ens</td>
<td>L</td>
<td>2</td>
<td>✓</td>
<td>Offboard</td>
<td>60.06</td>
<td>±75.2 ±73.6</td>
<td>±80.0 ±76.16</td>
<td>0.1 0.08 0.15</td>
</tr>
</tbody>
</table>

Table 2: The LiDAR-only non-ensemble performance comparison on the nuScenes test set. The table is mainly sorted by nuScenes detection score (NDS) which is the official ranking metric.

<table>
<thead>
<tr>
<th>Methods</th>
<th>NDS</th>
<th>mAP</th>
<th>Car</th>
<th>Truck</th>
<th>Bus</th>
<th>Trailer</th>
<th>Cons. Veh.</th>
<th>Ped.</th>
<th>Motor.</th>
<th>Bicycle</th>
<th>Tr. Cone</th>
<th>Barrier</th>
</tr>
</thead>
<tbody>
<tr>
<td>WYSIYG 2020</td>
<td>41.9</td>
<td>35.0</td>
<td>79.1</td>
<td>30.4</td>
<td>46.6</td>
<td>40.1</td>
<td>7.1</td>
<td>65.0</td>
<td>18.2</td>
<td>0.1</td>
<td>28.8</td>
<td>34.7</td>
</tr>
<tr>
<td>PointPillars 2019</td>
<td>45.3</td>
<td>30.5</td>
<td>68.4</td>
<td>23.0</td>
<td>28.2</td>
<td>23.4</td>
<td>4.1</td>
<td>59.7</td>
<td>27.4</td>
<td>1.1</td>
<td>30.8</td>
<td>38.9</td>
</tr>
<tr>
<td>3DV2020</td>
<td>53.1</td>
<td>45.4</td>
<td>79.7</td>
<td>33.6</td>
<td>47.1</td>
<td>43.1</td>
<td>18.1</td>
<td>76.5</td>
<td>40.7</td>
<td>7.9</td>
<td>58.8</td>
<td>48.8</td>
</tr>
<tr>
<td>3DSSD 2020</td>
<td>56.4</td>
<td>42.6</td>
<td>81.2</td>
<td>47.2</td>
<td>61.4</td>
<td>30.5</td>
<td>12.6</td>
<td>70.2</td>
<td>36.0</td>
<td>8.6</td>
<td>31.1</td>
<td>47.9</td>
</tr>
<tr>
<td>Cylinder3D 2021</td>
<td>61.6</td>
<td>50.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CGBS 2019</td>
<td>63.3</td>
<td>52.8</td>
<td>81.1</td>
<td>48.5</td>
<td>54.9</td>
<td>42.9</td>
<td>10.5</td>
<td>80.1</td>
<td>51.5</td>
<td>22.3</td>
<td>70.9</td>
<td>65.7</td>
</tr>
<tr>
<td>Pointformer 2021</td>
<td>-</td>
<td>53.6</td>
<td>82.3</td>
<td>48.1</td>
<td>55.6</td>
<td>43.4</td>
<td>8.6</td>
<td>81.8</td>
<td>55.0</td>
<td>22.7</td>
<td>72.2</td>
<td>66.0</td>
</tr>
<tr>
<td>CVCSNet 2020a</td>
<td>64.2</td>
<td>55.8</td>
<td>82.6</td>
<td>49.5</td>
<td>59.4</td>
<td>51.1</td>
<td>16.2</td>
<td>83.0</td>
<td>61.8</td>
<td>38.8</td>
<td>69.7</td>
<td>69.7</td>
</tr>
<tr>
<td>CenterPoint 2021a</td>
<td>65.5</td>
<td>58.0</td>
<td>84.6</td>
<td>51.0</td>
<td>60.2</td>
<td>53.2</td>
<td>17.5</td>
<td>83.4</td>
<td>53.7</td>
<td>28.7</td>
<td>76.7</td>
<td>70.9</td>
</tr>
<tr>
<td>HotSpotNet 2020b</td>
<td>66.0</td>
<td>59.3</td>
<td>83.1</td>
<td>50.9</td>
<td>56.4</td>
<td>53.3</td>
<td>23.0</td>
<td>81.3</td>
<td>63.5</td>
<td>36.6</td>
<td>73.0</td>
<td>71.6</td>
</tr>
<tr>
<td>AFDetV2 (Ours)</td>
<td>68.5</td>
<td>62.4</td>
<td>86.3</td>
<td>54.2</td>
<td>62.5</td>
<td>58.9</td>
<td>26.7</td>
<td>85.8</td>
<td>63.8</td>
<td>34.3</td>
<td>80.1</td>
<td>71.0</td>
</tr>
</tbody>
</table>

3.4 Loss

Similar to AFDet (Ge et al. 2020), we apply different losses for different sub-heads. We use the Focal Loss (Lin et al. 2017) for the heatmap prediction and keypoint auxiliary heads, \( L_1 \) loss for the location offset, the \( z \)-axis location, the 3D object size, and orientation regression heads. We compute the target IoU by \((2 \cdot \text{iou} - 0.5) \in [-1, 1]\) for the IoU-aware head, where \( \text{iou} \) is the axis-aligned 3D IoU between the ground truth box and the predicted box. Smooth \( L_1 \) loss is used for this branch. For all sub-heads except the heatmap prediction head and keypoint auxiliary head, only \( N \) foreground objects that are in the detection range are used to compute the loss.

We use weighted sum of all losses as the final loss:

\[
L = L_{\text{heat}} + \lambda_{\text{off}} L_{\text{off}} + \lambda_{\text{ori}} L_{\text{ori}} + \lambda_{\text{size}} L_{\text{size}} + \lambda_{\text{kps}} L_{\text{kps}}
\]

where \( \lambda \) is the weight for each sub-head.

4 Experiments

4.1 Datasets

nuScenes nuScenes (Caesar et al. 2020) dataset contains 700 training sequences, 150 val sequences and 150 test sequences. The annotations include 10 classes. nuScenes de-
4.2 Experiment Settings

nuScenes We set max number of objects, max point per voxel and the max voxel number in each frame to 500, 10 and 160,000 respectively. As a convention, we accumulate 10 LiDAR sweeps to densify the point clouds (Caesar et al. 2020). The voxel size is set to $[0.075m, 0.075m, 0.2m]$ and the training and inference ranges are set to $[\pm 5m, \pm 5m, (\pm 5m, 3m)]$ for the $x, y, z$-axis. We do not use any model ensembling or Test-Time Augmentation on nuScenes Dataset.

Waymo Open Dataset All our models only use the top LiDAR during training. The max number of objects in each frame is set to 500. The max point per voxel and the max voxel number are set to 5 and 250,000 respectively during training. We do not set a limit for the max number of points at inference. AdamW (Loshchilov and Hutter 2019) optimizer and one-cycle policy (Gugger 2018) is used. The max learning rate, the division factor and momentum ranges are set to $3 \times 10^{-3}$, 10, and $[0.95, 0.85]$. The weight decay is fixed to 0.01. We set all $\lambda$ in Eq. 2 to 2.0. Besides, we replace max pooling with class-specific NMS for better AP. In our solution, we set the IoU threshold to 0.8, 0.55, 0.55, and $\alpha$ in the IoU rescoring branch to 0.68, 0.71, 0.65 for VEHICLE, PEDESTRIAN, and CYCLIST respectively. Models are trained with Nvidia V100 GPUs. We use the data augmentation strategy following (Ge et al. 2021).

Different model variants For Waymo Open Dataset, we report 4 different variants of our model: AFDetV2-Lite, AFDetV2-Base, AFDetV2 and AFDetV2-Ens. They have the same 3D Feature Extractor, backbone and detection head. We list their detailed configurations in Tab. 1. AFDetV2-Lite for test set evaluation is finetuned 18 epochs on trainval set after being trained on training set for 10 epochs. AFDetV2-Lite for val set evaluation is trained 18 epochs on training set. For AFDetV2-Base, we first train 10 epochs on training set and finetune 36 epochs on the whole trainval data. We further finetune AFDetV2-Base for another 36 epochs on the whole trainval set using smaller grid size indicated in Tab. 1 to get AFDetV2. AFDetV2-Ens is basically identical to AFDetV2 with two differences: we use all LiDAR points for AFDetV2-Ens instead of only using top-LiDAR points at inference; we use Test-Time Augmentation and model ensemble to further improve the detection accuracy for AFDetV2-Ens.

Table 3: The single-frame LiDAR-only non-ensemble L1 3D AP/APH performance comparison on the Waymo Open Dataset val set. “ALL” stands for the mean of all three classes. The table is mainly sorted by ALL APH. †: reported by PV-RCNN++ (2021). ‡: reported by LiDAR R-CNN (2021). Refer to our arXiv version for L2 3D AP/APH comparisons.
### Table 4: The L1 3D AP/APH performance comparison on the Waymo Open Dataset test set.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Sensors</th>
<th>Frames</th>
<th>Ens</th>
<th>Server Latency</th>
<th>ALL 3D AP/APH</th>
<th>VEH 3D AP/APH</th>
<th>PED 3D AP/APH</th>
<th>CYC 3D AP/APH</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Single-frame LiDAR-only Non-ensemble Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>^1PointPillars 2019</td>
<td>LT</td>
<td>1</td>
<td>✓</td>
<td></td>
<td>68.60/68.10</td>
<td>68.00/55.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Light-FMFNet 2021</td>
<td>L</td>
<td>1</td>
<td>✓</td>
<td></td>
<td>71.24/67.26</td>
<td>77.85/77.30</td>
<td>69.52/59.78</td>
<td>66.34/64.69</td>
</tr>
<tr>
<td>HIKVISION_LiDAR 2021</td>
<td>L</td>
<td>1</td>
<td>✓</td>
<td>54.13</td>
<td>75.19/72.58</td>
<td>78.63/78.14</td>
<td>76.00/69.90</td>
<td>70.94/69.70</td>
</tr>
<tr>
<td>CenterPoint 2021a</td>
<td>-</td>
<td>1</td>
<td>✓</td>
<td></td>
<td>80.20/79.70</td>
<td>78.30/72.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFDetV2-Lite (Ours)</td>
<td>LT</td>
<td>1</td>
<td>✓</td>
<td>46.90</td>
<td>77.56/75.20</td>
<td>80.49/80.03</td>
<td>79.76/74.35</td>
<td>72.43/71.23</td>
</tr>
<tr>
<td>(b) Multi-frame LiDAR-only Non-ensemble Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3D-MAN 2021</td>
<td>L</td>
<td>15</td>
<td>✓</td>
<td></td>
<td>78.71/78.28</td>
<td>69.97/65.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSN 2021</td>
<td>LT</td>
<td>3</td>
<td>✓</td>
<td></td>
<td>80.70/80.30</td>
<td>78.90/75.60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X_Autonomous3D 2021</td>
<td>L</td>
<td>2</td>
<td>✓</td>
<td>68.42</td>
<td>77.54/75.61</td>
<td>81.49/81.02</td>
<td>78.17/73.93</td>
<td>72.96/71.88</td>
</tr>
<tr>
<td>CenterPoint 2021a</td>
<td>L</td>
<td>2</td>
<td>✓</td>
<td>78.71/77.18</td>
<td>81.05/80.59</td>
<td>80.47/77.28</td>
<td>74.60/73.68</td>
<td></td>
</tr>
<tr>
<td>AFDetV2-Base (Ours)</td>
<td>LT</td>
<td>2</td>
<td>✓</td>
<td>55.86</td>
<td>79.24/77.67</td>
<td>81.27/80.82</td>
<td>81.08/77.87</td>
<td>75.35/74.33</td>
</tr>
<tr>
<td>Pyramid R-CNN 2021a</td>
<td>L</td>
<td>2</td>
<td>✓</td>
<td></td>
<td>81.77/81.32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CenterPoint++ 2021b</td>
<td>LT</td>
<td>3</td>
<td>✓</td>
<td>57.12</td>
<td>79.41/77.96</td>
<td>82.78/82.33</td>
<td>81.07/78.21</td>
<td>74.40/73.33</td>
</tr>
<tr>
<td>AFDetV2 (Ours)</td>
<td>LT</td>
<td>2</td>
<td>✓</td>
<td>60.06</td>
<td>79.77/78.21</td>
<td>81.65/81.22</td>
<td>72.68/75.05</td>
<td>76.41/75.37</td>
</tr>
<tr>
<td>(c) Ensemble Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PV-RCNN Ens 2020b</td>
<td>L</td>
<td>2</td>
<td>✓</td>
<td></td>
<td>78.82/76.90</td>
<td>81.06/80.57</td>
<td>80.31/76.28</td>
<td>75.10/73.84</td>
</tr>
<tr>
<td>RSN Ens 2021</td>
<td>LT</td>
<td>3</td>
<td>✓</td>
<td></td>
<td>81.38/80.97</td>
<td>82.41/77.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HorizonLiDAR3D 2020</td>
<td>CL</td>
<td>5</td>
<td>✓</td>
<td>83.28/81.85</td>
<td>85.09/84.68</td>
<td>85.03/82.10</td>
<td>79.73/78.78</td>
<td></td>
</tr>
<tr>
<td>AFDetV2-Ens (Ours)</td>
<td>L</td>
<td>2</td>
<td>✓</td>
<td>84.07/82.63</td>
<td>85.80/85.41</td>
<td>85.22/82.16</td>
<td>81.20/80.30</td>
<td></td>
</tr>
</tbody>
</table>

Following Ge et al., we use Stochastic Weights Averaging (SWA) (Izmailov et al. 2018; Zhang et al. 2021) in AFDetV2-Lite, AFDetV2 and AFDetV2-Ens before submitting to the official evaluation server for test set evaluation. We never use SWA in any evaluation against val set or in ablation studies. We don’t use SWA in AFDetV2-Base either.

**Model ensembling setting** We use Test-Time Augmentation (TTA) and ensemble to improve the performance following (Ding et al. 2020). We only use yaw rotation, global scaling and translation along z-axis. To be specific, we use $[0^\circ, ±22.5^\circ, ±45^\circ, ±15^\circ, ±157.5^\circ, ±180^\circ]$ for yaw rotation, $[0.95, 1, 1.05]$ for global scaling, and [-0.2m, 0m, 0.2m] for translation along z-axis. We merge point clouds from all LiDARs for model ensembling. The model is named as AFDetV2-Ens in Tab. 4.

**Latency optimization** Latency optimization is allowed by the server evaluation. Our fastest model is evaluated on the Waymo real-time 3D detection evaluation server and achieves 46.9 ms latency. We make the following efforts to accelerate our model’s inference speed. First, the conversion from range images to point clouds in Cartesian coordinate and voxelization is executed on GPU. Second, except for the last layers in each sub-heads, our model is cast to half-precision. Next, we merge Batch Normalization (Ioffe and Szegedy 2015) parameters into 3D Sparse Convolution and SubManifold 3D Sparse Convolution layers in 3D Feature Extractor. Finally, the keypoint auxiliary branch is disabled. Note that all other methods which were measured server latency in Tab. 4 were optimized similarly to achieve fast inference speed (e.g. half-precision inference).

### 4.3 Comparison with State-of-the-art Methods

**Evaluation on nuScenes test set** We also compare our AFDetV2 with previous LiDAR-only non-ensemble methods on the nuScenes test set. As shown in Tab. 2, our method outperforms all prior arts. Specifically, AFDetV2 surpasses HotSpotNet (Chen et al. 2020b) by 2.5 NDS or 3.1 mAP. To the best of our knowledge, AFDetV2 surpasses all the published LiDAR-only non-ensemble methods on the nuScenes Detection leaderboard.

**Evaluation on Waymo Open Dataset val set** We compare our AFDetV2-Lite with all published single-frame LiDAR-only non-ensemble methods on WOD val set in Tab. 3. Tab. 3 is mainly sorted by ALL APH. We can see that our AFDetV2-Lite significantly outperforms the previous state-of-the-art single-frame LiDAR-only detectors. To be specific, our AFDetV2-Lite achieves 74.83 APH for the mean of all three classes, surpassing prior art (Shi et al. 2021) by 3.83%.

**Evaluation on Waymo Open Dataset test set** We also compare our AFDetV2 variants with all published methods.
on WOD test set in Tab. 4. Tab. 4 is split into three sub-tables. The upper table is for (a) single-frame LiDAR-only non-ensemble methods; the middle table is for (b) multi-frame LiDAR-only non-ensemble methods; and the bottom table is for (c) ensemble methods. Table (a) and table (b) aim at onboard real-time scenario; table (c) is for offboard use cases.

For table (a), our AFDetV2-Lite outperforms all the previous single-frame LiDAR-only non-ensemble models with respect to both speed and accuracy. Our AFDetV2-Lite is 15.4% faster than HIKVISION_LiDAR (Xu et al. 2021) and also does better detection than it by 2.62 ALL APH/L2.

Table (b) is for multi-frame LiDAR-only non-ensemble models. As we can see our AFDetV2 surpasses all the other methods, ranking the 1st on the official Real-Time 3D Detection leaderboard.

After the evaluation of real-time variants of AFDetV2 for onboard scenarios, we further explore our model for offboard use cases. Under the offboard scenario, we could leverage ample computational resources. Detection accuracy plays a much more important role than detection speed. Thus, we leverage the Test-Time Augmentation and model ensemble for our AFDetV2-Ens to achieve more accurate detection. We list all published ensemble models in table (c) of Tab. 4. Our AFDetV2-Ens outperforms all published prior arts. It also ranks the 1st on the official Non-Real-Time 3D Detection leaderboard.

To validate the generality of AFDetV2, we conduct an experiment on a different encoder. In Tab. 6, we replace the 3D sparse convolution feature extractor with PointPillars encoder (PP) for AFDetV2. AFDetV2-PP w/o IoU already surpasses the SOTA using PP, showing a strong baseline of classification and box regression. With IoU-aware rescoring (IoU), AFDetV2-PP shows even more significant improvement. The results are 3D APH/L2 calculated by the official Waymo evaluation metrics on the entire val set.

Table 5: Ablation study of the effect of the AFDetV2 improvements. “KP” represents keypoint auxiliary loss. “SC” stands for self-calibrated convolutional backbone. All results are in 3D APH/L2 metric on full val set.

<table>
<thead>
<tr>
<th>Model</th>
<th>ALL</th>
<th>VEH</th>
<th>PED</th>
<th>CYC</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFDetV2-Lite</td>
<td>64.65</td>
<td>64.96</td>
<td>61.82</td>
<td>67.16</td>
</tr>
<tr>
<td>AFDetV2-Lite</td>
<td>67.43</td>
<td>64.02</td>
<td>62.26</td>
<td>67.29</td>
</tr>
<tr>
<td>AFDetV2-Ens</td>
<td>67.60</td>
<td>64.02</td>
<td>62.26</td>
<td>67.29</td>
</tr>
<tr>
<td>AFDetV2-Ens+IoU</td>
<td>67.60</td>
<td>64.02</td>
<td>62.26</td>
<td>67.29</td>
</tr>
</tbody>
</table>

Table 6: Ablation study of PointPillars-based AFDetV2 model with and without IoU-aware rescoring. We replaced the feature extractor with a PointPillars encoder (PP) for AFDetV2. AFDetV2-PP w/o IoU already surpasses the SOTA using PP, showing a strong baseline of classification and box regression. With IoU-aware rescoring (IoU), AFDetV2-PP shows even more significant improvement. The results are 3D APH/L2 calculated by the official Waymo evaluation metrics on the entire val set.

Table 7: Ablation study of the effect of the AFDetV2 improvements. “KP” represents keypoint auxiliary loss. “SC” stands for self-calibrated convolutional backbone. All results are in 3D APH/L2 metric on full val set.

<table>
<thead>
<tr>
<th>KP</th>
<th>SC</th>
<th>IoU</th>
<th>ALL</th>
<th>VEH</th>
<th>PED</th>
<th>CYC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>64.65</td>
<td>64.96</td>
<td>61.82</td>
<td>67.16</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td></td>
<td>67.43</td>
<td>64.02</td>
<td>62.26</td>
<td>67.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓</td>
<td>67.60</td>
<td>64.02</td>
<td>62.26</td>
<td>67.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>✓✓</td>
<td>67.60</td>
<td>64.02</td>
<td>62.26</td>
<td>67.29</td>
</tr>
</tbody>
</table>

4.5 Ablation Studies

Effect of AFDetV2 improvements To validate our method, we conduct experiments on the val set of WOD. Three improvements are proposed in our solution which are keypoint auxiliary loss, SC-Conv backbone, and IoU rescoring. We first validate the improvement brought by each of them. From Tab. 5, we can see that keypoint auxiliary loss brought a boost of 0.12 ALL APH/L2, utilizing SC-Conv backbone led to an increment of 3.12 ALL APH/L2. When all modules are deployed, the accuracy of the model was increased by 4.12 ALL APH/L2. We can see that the IoU rescoring brings significant improvement to the model.

To validate the generality of AFDetV2, we conduct an experiment on a different encoder. In Tab. 6, we replace the 3D sparse convolution feature extractor with PointPillars encoder (Lang et al. 2019). Our AFDetV2 surpasses state-of-the-art PointPillars-based model (Yin, Zhou, and Krahenbuhl 2021a) by a margin of 3.17 ALL APH/L2. Besides, the IoU-aware rescoring shows higher improvement (+1.42 ALL APH/L2) than the second stage refinement brought to the CenterPoint (+1.1 ALL APH/L2) with the same encoder.

4.4 Waymo Real-Time 3D Detection Challenge

The Waymo Open Dataset Real-time 3D Detection Challenge requires an algorithm to detect the 3D objects of interest as a set of 3D bounding boxes within 70 ms per frame. The model with the highest APH/L2 performance while satisfying this real-time requirement wins this challenge. In addition, the model with the lowest latency and APH/L2 > 70

and score refinement, respectively. All the results are 3D APH/L2 on the entire val set. All models take 2 frames as input. The first row is our AFDetV2 without the IoU branch. Comparing the first row and the second row, one can see that the box refinement in the second stage made a trivial change, in contrast to the score refinement. Comparing the last two rows, one can see that, for a well-designed single-stage network like AFDetV2, the box and score refinements in the second stage are no longer needed.

In Tab. 8, we migrate our improvements (SC-Conv and IoU-aware head) to a typical state-of-the-art two-stage network, Voxel R-CNN (2021a). Voxel R-CNN uses SECOND (2018) as the first stage. Then, a voxel-RoI pooling module is applied to both of the voxel and BEV features. The acquired RoI features are fed into a second-stage network. Similar to most R-CNN style networks, the second stage has two sibling branches: one for box regression and the other for confidence prediction. The box regression branch predicts the residue from 3D region proposals to the ground truth boxes, and the confidence branch predicts the IoU-related confidence score. As shown in Tab. 8, we divide the table into three major rows. In the top row, score and box refinement together would bring 1.61 ALL APH/L2 improvement. We validate that the single-stage detector with IoU-aware rescoring can beat its corresponding two-stage detector. Furthermore, by comparing the last two rows of Tab. 7, the single-stage detector with IoU-aware rescoring can even beat its corresponding two-stage detector with IoU-aware rescoring. In this scenario, the second stage network is unnecessary for a well-designed single-stage network.

Table 7: The analysis of each component of the second stage and the comparison with our single-stage method. “2S-b” and “2S-s” mean the second stage for box refinement and score refinement, respectively. All the results are 3D APH/L2 on the entire val set. All models take 2 frames as input. The first row is our AFDetV2 without the IoU branch. Comparing the first row and the second row, one can see that the box refinement in the second stage made a trivial change, in contrast to the score refinement. Comparing the last two rows, one can see that, for a well-designed single-stage network like AFDetV2, the box and score refinements in the second stage are no longer needed.

<table>
<thead>
<tr>
<th>SC</th>
<th>IoU</th>
<th>ALL</th>
<th>VEH</th>
<th>PED</th>
<th>CYC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>68.81</td>
<td>69.05</td>
<td>67.39</td>
<td>70.00</td>
</tr>
<tr>
<td>✔</td>
<td></td>
<td>68.79</td>
<td>68.78</td>
<td>67.55</td>
<td>70.04</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>70.42</td>
<td>69.63</td>
<td>70.62</td>
<td>71.01</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>71.01</td>
<td>70.10</td>
<td>71.71</td>
<td>71.20</td>
</tr>
<tr>
<td>✔</td>
<td>✔</td>
<td>71.10</td>
<td>70.73</td>
<td>71.24</td>
<td>71.33</td>
</tr>
</tbody>
</table>

5 Conclusion

We have proposed a real-time single-stage anchor-free 3D object detection model named AFDetV2. We have conducted extensive experiments to show that the second stage is unnecessary if we strengthen and fully utilize the first stage. Our model achieves the state-of-the-art performance on Waymo Open Dataset and nuScenes Dataset with high inference speed.

References


Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In CVPR.


Chai, Y.; Sun, P.; Ngiam, J.; Wang, W.; Caine, B.; Vasudevan, V.; Zhang, X.; and Anguelov, D. 2021. To the Point: Efficient 3D Object Detection in the Range Image With Graph Convolution Kernels. In CVPR.


3D object detection from LIDAR point clouds.


Deng, J.; Zhou, W.; Zhang, Y.; and Li, H. 2021b. From Multi-View to Hollow-3D: Hallucinated Hollow-3D R-CNN for 3D Object Detection. TCSVT.


He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In ICCV.


Li, Z.; Wang, F.; and Wang, N. 2021. LiDAR R-CNN: An Efficient and Universal 3D Object Detector. In CVPR.


Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In ICLR.


Pan, X.; Xia, Z.; Song, S.; Li, L. E.; and Huang, G. 2021. 3d object detection with pointformer. In CVPR.


Shi, S.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020c. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. TPAMI.


Wu, S.; Li, X.; and Wang, X. 2020. IoU-aware Single-stage Object Detector for Accurate Localization. IVC.


Yan, Y.; Mao, Y.; and Li, B. 2018. SECOND: Sparsely Embedded Convolutional Detection. Sensors.

Yang, B.; Luo, W.; and Urtasun, R. 2018. PIXOR: Real-time 3D Object Detection from Point Clouds. In CVPR.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In CVPR.

Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; and Jia, J. 2019. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In ICCV.

Yang, Z.; Zhou, Y.; Chen, Z.; and Ngiam, J. 2021. 3D-MAN: 3D Multi-frame Attention Network for Object Detection. In CVPR.


Zhou, Y.; and Tuzel, O. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In CVPR.
