

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 rescore 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 rescore 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

^{*}These authors contributed equally.

[†]Work done while at Horizon Robotics.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

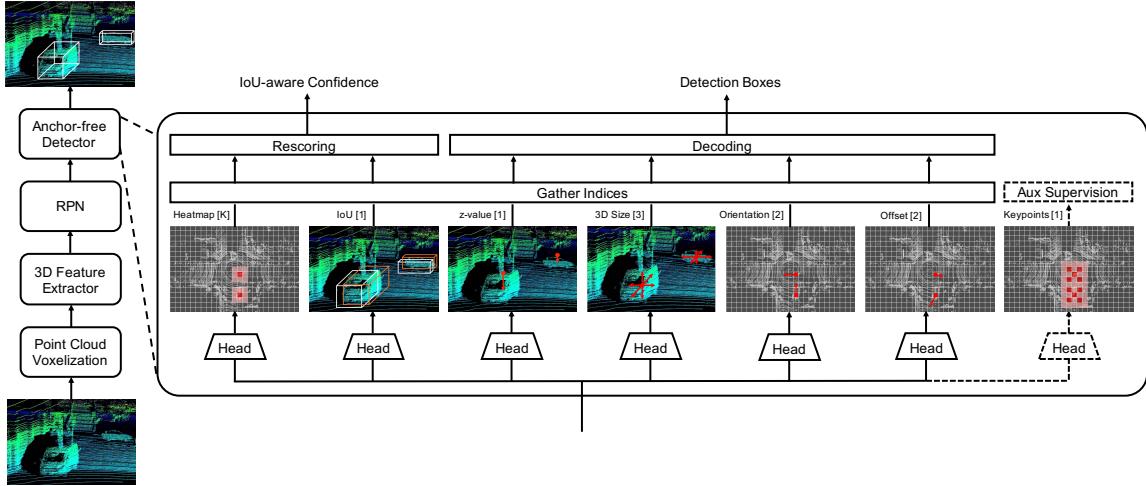


Figure 1: The framework of anchor-free one-stage 3D detection (AFDetV2) system. The whole pipeline consists of the Point Cloud Voxelization, 3D Feature Extractor, backbone and the Anchor-Free Detector. The number in the brackets indicates the output channels in the last convolution layer. K is the number of categories used in the detection. The auxiliary supervision which is turned off at inference is shown in dashed lines. Better viewed in color and zoom in for more details.

refine the positions of the boxes; instead, it rescores the classification 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 rescoreing.

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 rescore 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*¹. 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.

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 problem, Part-A² (Shi et al. 2020c) designed an ROI-aware point cloud pooling operation, while LiDAR R-CNN (Li, Wang,

¹<http://cvpr2021.wad.vision/>, accessed on Dec 5, 2021.

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 keypoints, 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. AFDet (Ge et al. 2020, 2021; Ding et al. 2020; Wang et al. 2020c), 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.

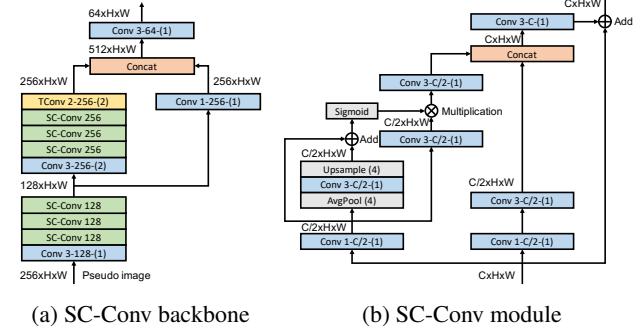


Figure 2: (a) denotes the backbone with the self-calibrated convolution (SC-Conv). “Conv” stands for convolutional layer and “TConv” stands for transposed convolutional layer. The format of the layer setting follows “kernel size-channels-(strides)”, i.e. k - C -(s). “SC-Conv” stands for SC-Conv module and only channels are shown. (b) gives the detailed structure of the SC-Conv module.

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 AFDetV2, 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

| Variants | Sensors | Frames | Ens | Usage | Server | Training Range | | | Inference Range | | | Grid Size | | |
|--------------|---------|--------|-----|----------|---------|-----------------------------|------------------------------|------------------------------|-----------------|------|------|-----------|---|---|
| | | | | Scenario | Latency | x | y | z | x | y | z | x | y | z |
| AFDetV2-Lite | LT | 1 | ✗ | Onboard | 46.90 | $\pm 75.2 \pm 75.2 (-2, 4)$ | $\pm 75.2 \pm 75.2 (-2, 4)$ | $\pm 75.2 \pm 75.2 (-2, 4)$ | 0.1 | 0.1 | 0.15 | | | |
| AFDetV2-Base | LT | 2 | ✗ | Onboard | 55.86 | $\pm 75.2 \pm 75.2 (-2, 4)$ | $\pm 75.2 \pm 75.2 (-2, 4)$ | $\pm 75.2 \pm 75.2 (-2, 4)$ | 0.1 | 0.1 | 0.15 | | | |
| AFDetV2 | LT | 2 | ✗ | Onboard | 60.06 | $\pm 75.2 \pm 73.6 (-2, 4)$ | $\pm 80.0 \pm 76.16 (-2, 4)$ | $\pm 80.0 \pm 76.16 (-2, 4)$ | 0.1 | 0.08 | 0.15 | | | |
| AFDetV2-Ens | L | 2 | ✓ | Offboard | - | $\pm 75.2 \pm 73.6 (-2, 4)$ | $\pm 80.0 \pm 76.16 (-2, 4)$ | $\pm 80.0 \pm 76.16 (-2, 4)$ | 0.1 | 0.08 | 0.15 | | | |

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.

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

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.

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)

| Methods | Reference | # Stages | ALL (3D AP/APH) | | VEH (3D AP/APH) | | PED (3D AP/APH) | | CYC (3D AP/APH) | |
|---------------------------|---------------|----------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | | | L1 | L2 | L1 | L2 | L1 | L2 | L1 | L2 |
| StarNet | arXiv 2019 | Two | - | - | 53.70/- | - | 66.80/- | - | - | - |
| PPBA | ECCV 2020 | Single | - | - | 62.40/- | - | 66.00/- | - | - | - |
| MVF | CoRL 2020 | Single | - | - | 62.93/- | - | 65.33/- | - | - | - |
| CVCNet | NeurIPS 2020a | Single | - | - | 65.20/- | - | - | - | - | - |
| Perspective EdgeConv | CVPR 2021 | Single | - | - | 65.20/- | -/56.70 | 73.90/- | -/59.60 | - | - |
| 3D-MAN | CVPR 2021 | Multi | - | - | 69.03/68.52 | 60.16/59.71 | 71.71/67.74 | 62.58/59.04 | - | - |
| RCD | CoRL 2020 | Two | - | - | 69.59/69.16 | - | - | - | - | - |
| Pillar-based | ECCV 2020d | Single | - | - | 69.80/- | - | 72.51/- | - | - | - |
| [†] SECOND | Sensors 2018 | Single | 67.20/63.05 | 60.97/57.23 | 72.27/71.69 | 63.85/63.33 | 68.70/58.18 | 60.72/51.31 | 60.62/59.28 | 58.34/57.05 |
| [‡] PointPillars | CVPR 2019 | Single | 68.87/63.33 | 62.63/57.53 | 71.60/71.00 | 63.10/62.50 | 70.60/56.70 | 62.90/50.20 | 64.40/62.30 | 61.90/59.90 |
| LiDAR R-CNN | CVPR 2021 | Two | 71.10/66.20 | 64.63/60.10 | 73.50/73.00 | 64.70/64.20 | 71.20/58.70 | 63.10/51.70 | 68.60/66.90 | 66.10/64.40 |
| RangeDet | arXiv 2021 | Single | 71.53/- | - | 72.85/- | - | 75.94/- | - | 65.80/- | - |
| MVF++ | CVPR 2021 | Single | - | - | 74.64/- | - | 78.01/- | - | - | - |
| Voxel Transformer | ICCV 2021b | Two | - | - | 74.95/74.25 | 65.91/65.29 | - | - | - | - |
| RSN | CVPR 2021 | Two | - | - | 75.10/74.60 | 66.00/65.50 | 77.80/72.70 | 68.30/63.70 | - | - |
| H ² 3D R-CNN | TCSVT 2021b | Two | - | - | 75.15/- | 66.14/- | - | - | - | - |
| M3DETR | arXiv 2021 | Two | - | - | 75.71/75.08 | 66.58/66.02 | - | - | - | - |
| Voxel R-CNN | AAAI 2021a | Two | - | - | 75.59/- | 66.59/- | - | - | - | - |
| Pyramid R-CNN | ICCV 2021a | Two | - | - | 76.30/75.68 | 67.23/66.68 | - | - | - | - |
| CenterPoint | CVPR 2021a | Two | - | - | 76.70/76.20 | 68.80/68.30 | 79.00/72.90 | 71.00/65.30 | - | - |
| PV-RCNN | CVPR 2020a | Two | 73.44/69.63 | 66.80/63.33 | 77.51/76.89 | 68.98/68.41 | 75.01/65.65 | 66.04/57.61 | 67.81/66.35 | 65.39/63.98 |
| Part-A ² | TPAMI 2020c | Two | 73.63/70.25 | 66.93/63.84 | 77.05/76.51 | 68.47/67.97 | 75.24/66.87 | 66.18/58.62 | 68.60/67.36 | 66.13/64.93 |
| CT3D | ICCV 2021 | Two | - | - | - | 69.04/- | - | - | - | - |
| PV-RCNN-v2 | arXiv 2021 | Two | 74.81/71.00 | 68.42/64.87 | 78.79/78.21 | 70.26/69.71 | 76.67/67.15 | 68.51/59.72 | 68.98/67.63 | 66.48/65.17 |
| AFDetV2-Lite | Ours | Single | 77.18/74.83 | 70.97/68.77 | 77.64/77.14 | 69.68/69.22 | 80.19/74.62 | 72.16/66.95 | 73.72/72.74 | 71.06/70.12 |

Table 3: The single-frame LiDAR-only non-ensemble 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/L2 which is the official ranking metric. [†]: reported by PV-RCNN++ (2021). [‡]: reported by LiDAR R-CNN (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-scale backbone. To fully explore the potential of the single-stage framework, we apply the Self-Calibrated Convolutions (SC-Conv) (Liu et al. 2020) to the backbone and replace the basic 3×3 convolutional blocks. 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 backbone and SC-Conv block are shown in Fig. 2a and 2b.

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 et al. 2017; Yin, Zhou, and Krahenbuhl 2021a) or RoI pooling (Shi, Wang, and Li 2019; Shi et al. 2020a,c) are required in the second stage network.

Recently, CIA-SSD (Zheng et al. 2021) migrates IoU-prediction head from 2D image (Wu, Li, and Wang 2020) to anchor-based 3D LiDAR detection. Different from CIA-SSD, we adopt an IoU prediction head to our anchor-free network. To incorporate IoU information into confidence score, we recalculate the final confidence score by a simple post-processing function:

$$f = score^{1-\alpha} * iou^\alpha \quad (1)$$

where *score* is the original classification score, *iou* is the predicted IoU and α is the hyperparameter $\in [0, 1]$ that controls the contributions from the classification score and predicted IoU.

After rescoreing, the ranking of the predictions takes both the classification confidence and localization accuracy into account. The rescoreing process will lower the confidence of

| Methods | Sensors | Frames | Ens | Server Latency | ALL (3D AP/APH) | | VEH (3D AP/APH) | | PED (3D AP/APH) | | CYC (3D AP/APH) | |
|--|---------|--------|-----|----------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | | | | | L1 | L2 | L1 | L2 | L1 | L2 | L1 | L2 |
| (a) Single-frame LiDAR-only Non-ensemble Methods | | | | | | | | | | | | |
| StarNet (2019) | - | 1 | X | - | - | - | 61.50/61.00 | 54.90/54.50 | 67.80/59.90 | 61.10/54.00 | - | - |
| PPBA (2020) | - | 1 | X | - | - | - | 67.50/67.00 | 59.60/59.10 | 69.70/61.70 | 63.00/55.80 | - | - |
| [†] PointPillars (2019) | LT | 1 | X | - | - | - | 68.60/68.10 | 60.50/60.10 | 68.00/55.50 | 61.40/50.10 | - | - |
| RCD (2020) | - | 1 | X | - | - | - | 71.97/71.59 | 65.06/64.70 | - | - | - | - |
| Pseudo-Labeling (2021) | - | 1 | X | - | - | - | 74.00/73.60 | - | 69.80/57.90 | - | - | - |
| M3DETR (2021) | - | 1 | X | - | 71.05/67.09 | 65.50/61.92 | 77.75/77.17 | 70.63/70.06 | 68.10/58.87 | 60.57/52.37 | 67.28/65.69 | 65.31/63.75 |
| Light-FMFNet (2021) | L | 1 | X | 62.31 | 71.24/67.26 | 65.88/62.18 | 77.85/77.30 | 70.16/69.65 | 69.52/59.78 | 63.62/54.61 | 66.34/64.69 | 63.87/62.28 |
| HIKVISION.LIDAR (2021) | L | 1 | X | 54.13 | 75.19/72.58 | 69.74/67.29 | 78.63/78.14 | 71.06/70.60 | 76.00/69.90 | 69.82/64.11 | 70.94/69.70 | 68.35/67.15 |
| CenterPoint (2021a) | - | 1 | X | - | - | - | 80.20/79.70 | 72.20/71.80 | 78.30/72.10 | 72.20/66.40 | - | - |
| AFDetV2-Lite (Ours) | LT | 1 | X | 46.90 | 77.56/75.20 | 72.18/69.95 | 80.49/80.03 | 72.98/72.55 | 79.76/74.35 | 73.71/68.61 | 72.43/71.23 | 69.84/68.67 |
| (b) Multi-frame LiDAR-only Non-ensemble Methods | | | | | | | | | | | | |
| 3D-MAN (2021) | L | 15 | X | - | - | - | 78.71/78.28 | 70.37/69.98 | 69.97/65.98 | 63.98/60.26 | - | - |
| RSN (2021) | LT | 3 | X | - | - | - | 80.70/80.30 | 71.90/71.60 | 78.90/75.60 | 70.70/67.80 | - | - |
| X_Autonomous3D (2021) | L | 2 | X | 68.42 | 77.54/75.61 | 72.29/70.46 | 81.49/81.02 | 74.04/73.60 | 78.17/73.93 | 72.29/68.27 | 72.96/71.88 | 70.55/69.50 |
| CenterPoint (2021a) | L | 2 | X | - | 78.71/77.18 | 73.38/71.93 | 81.05/80.59 | 73.42/72.99 | 80.47/77.28 | 74.56/71.52 | 74.60/73.68 | 72.17/71.28 |
| AFDetV2-Base (Ours) | LT | 2 | X | 55.86 | 79.24/77.67 | 74.06/72.57 | 81.27/80.82 | 73.89/73.46 | 81.08/77.87 | 75.34/72.29 | 75.35/74.33 | 72.96/71.97 |
| Pyramid R-CNN (2021a) | L | 2 | X | - | - | - | 81.77/81.32 | 74.87/74.43 | - | - | - | - |
| CenterPoint++ (2021b) | LT | 3 | X | 57.12 | 79.41/77.96 | 74.22/72.82 | 82.78/82.33 | 75.47/75.05 | 81.07/78.21 | 75.13/72.41 | 74.40/73.33 | 72.04/71.01 |
| AFDetV2 (Ours) | LT | 2 | X | 60.06 | 79.77/78.21 | 74.60/73.12 | 81.65/81.22 | 74.30/73.89 | 81.26/78.05 | 75.47/72.41 | 76.41/75.37 | 74.05/73.04 |
| (c) Ensembled Methods | | | | | | | | | | | | |
| TS-LidarDet (2020a) | L | 1 | ✓ | - | 74.87/71.05 | 69.10/65.53 | 80.75/80.18 | 72.65/72.12 | 74.45/65.01 | 68.10/59.32 | 69.42/67.97 | 66.55/65.16 |
| RSN Ens (2021) | LT | 3 | ✓ | - | - | - | 81.38/80.97 | 72.80/72.43 | 82.41/77.98 | 74.75/70.68 | - | - |
| PV-RCNN Ens (2020b) | L | 2 | ✓ | - | 78.82/76.90 | 73.35/71.52 | 81.06/80.57 | 73.69/73.23 | 80.31/76.28 | 73.98/70.16 | 75.10/73.84 | 72.38/71.16 |
| 3DAL (2021) | L | ~200 | ✓ | - | - | - | 85.84/85.46 | 77.24/76.91 | - | - | - | - |
| HorizonLiDAR3D (2020) | CL | 5 | ✓ | - | 83.28/81.85 | 78.49/77.11 | 85.09/84.68 | 78.23/77.83 | 85.03/82.10 | 79.32/76.50 | 79.73/78.78 | 77.91/76.98 |
| AFDetV2-Ens (Ours) | L | 2 | ✓ | - | 84.07/82.63 | 79.04/77.64 | 85.80/85.41 | 78.71/78.34 | 85.22/82.16 | 79.71/76.75 | 81.20/80.30 | 78.70/77.83 |

Table 4: The 3D AP/APH performance comparison on the Waymo Open Dataset *test* set. “L”, “LT” and “CL” mean “all LiDARs”, “top-LiDAR only” and “camera and all LiDARs”, respectively. “ALL” stands for the mean of all three classes. The table is mainly sorted by ALL APH/L2 which is the official ranking metric. “Ens” is short for ensemble. “Server Latency” is the latency measured by the official testing server in milliseconds. The latency optimization is allowed by the Challenge Sponsor. The table is split into three major rows: (a) single-frame LiDAR-only non-ensemble methods; (b) multi-frame LiDAR-only non-ensemble methods; (c) ensemble methods. Our models consistently outperform previous state-of-the-art methods under different settings. AFDetV2 and AFDetV2-Base are the two award-winning entries. [†]: reported by RSN (2021).

the predictions with higher classification scores but worse localization accuracy, and vice versa. Our single-stage network runs much faster than most existing two-stage LiDAR detectors while surpassing their detection results, as shown in the leader board in Tab. 4.

Keypoint auxiliary supervision We devise a keypoint prediction sub-head as auxiliary supervision in the detection head inspired by (Wang et al. 2020b). We add another heatmap that predicts 4 corners and the center of every object in BEV during training. We draw the 5 keypoints of each object at the same keypoint heatmap but with a halved radius. During inference, the keypoint prediction sub-head is disabled thus does not influence the inference speed.

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 $(2 * iou - 0.5) \in [-1, 1]$ for the IoU-aware head, where 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:

$$\mathcal{L} = \mathcal{L}_{heat} + \lambda_{off}\mathcal{L}_{off} + \lambda_z\mathcal{L}_z + \lambda_{size}\mathcal{L}_{size} + \lambda_{ori}\mathcal{L}_{ori} + \lambda_{iou}\mathcal{L}_{iou} + \lambda_{kps}\mathcal{L}_{kps} \quad (2)$$

where λ 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 detection score (NDS) is the main ranking metric for this dataset. We also report Mean Average Precision (mAP).

Waymo Open Dataset The Waymo Open Dataset (Sun et al. 2020) contains 798 *training* sequences, 202 *val* sequences, and 150 *test* sequences for VEHICLE, PEDESTRIAN, and CYCLIST detection. Boxes having more than five LiDAR points and not marked as LEVEL_2 (L2) are classified as LEVEL_1 (L1). Rest boxes that enclose at least one LiDAR point are classified as L2. All L1 and L2 boxes are considered in the L2 metric.

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,

| Keypoint | SC-Conv | IoU | ALL | VEH | PED | CYC |
|----------|---------|-----|--------------|--------------|--------------|--------------|
| ✓ | | | 64.65 | 64.96 | 61.82 | 67.16 |
| | ✓ | | 64.77 | 65.13 | 61.72 | 67.47 |
| | | ✓ | 65.20 | 65.24 | 62.21 | 68.14 |
| ✓ | ✓ | | 67.77 | 68.11 | 66.50 | 68.69 |
| ✓ | ✓ | ✓ | 65.58 | 65.89 | 62.56 | 68.28 |
| ✓ | ✓ | ✓ | 68.77 | 69.22 | 66.95 | 70.12 |

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

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 $[\pm 54m, \pm 54m, (-5m, 3m)]$ 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 3×10^{-3} , 10, and $[0.95, 0.85]$. The weight decay is fixed to 0.01. We set all λ 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 α in the IoU rescore 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 fintune 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.

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 ab-

| Model | Frames | ALL | VEH | PED | CYC |
|---------------------------|--------|--------------|--------------|--------------|--------------|
| †PointPillars (2019) | 1 | - | 60.10 | 50.10 | - |
| CenterPoint-PP (2021a) | 2 | 60.30 | 65.50 | 55.10 | 60.20 |
| CenterPoint-PP-2stage | 2 | 61.40 | 66.70 | 55.90 | 61.70 |
| AFDetV2-PP-w/o IoU (Ours) | 2 | 63.15 | 67.29 | 61.84 | 60.34 |
| AFDetV2-PP (Ours) | 2 | 64.57 | 67.43 | 64.02 | 62.26 |

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. †: reported by RSN (2021).

lation 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 $[0^\circ, \pm 22.5^\circ, \pm 45^\circ, \pm 135^\circ, \pm 157.5^\circ, 180^\circ]$ 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 sever 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 leaderboard¹.

¹<https://www.nuscenes.org/object-detection?externalData=no&mapData=no&modalities=Lidar>, accessed on Dec 5, 2021.

| 2S-box | 2S-score | IoU | ALL | VEH | PED | CYC |
|--------|----------|-----|--------------|--------------|--------------|--------------|
| ✓ | | | 68.81 | 69.05 | 67.39 | 70.00 |
| ✓ | ✓ | | 68.79 | 68.78 | 67.55 | 70.04 |
| ✓ | ✓ | ✓ | 70.42 | 69.63 | 70.62 | 71.01 |
| ✓ | ✓ | ✓ | 71.01 | 70.10 | 71.71 | 71.20 |
| | | ✓ | 71.10 | 70.73 | 71.24 | 71.33 |

Table 7: The analysis of each component of the second stage and the comparison with our single-stage method. “2S-box” and “2S-score” 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.

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/L2 which is specified as the official ranking metric on the dataset testing server. 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 68.77 APH/L2 for the mean of all three classes, surpassing prior art (Shi et al. 2021) by 3.9%.

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 major rows. The upper row is for (a) single-frame LiDAR-only non-ensemble methods; the middle row is for (b) multi-frame LiDAR-only non-ensemble methods; and the bottom row is for (c) ensemble methods. The upper row and middle row aim at onboard real-time scenario; the bottom row is for offboard use cases.

For the upper row, 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 2.6 ALL APH/L2.

The middle row is for multi-frame LiDAR-only non-ensemble models. As we can see our AFDetV2 surpasses all the other methods under the ALL APH/L2 metric, ranking the 1st on the official Real-Time 3D Detection leaderboard¹.

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

¹<https://waymo.com/open/challenges/2021/real-time-3d-prediction/>, accessed on Dec 5, 2021.

| 2S-box | 2S-score | SC-Conv | IoU | ALL | VEH | PED | CYC |
|--------|----------|---------|-----|--------------|--------------|--------------|--------------|
| ✓ | | | | 53.23 | 60.71 | 45.93 | 53.04 |
| ✓ | ✓ | | | 54.55 | 64.98 | 45.46 | 53.22 |
| | | ✓ | | 58.84 | 65.62 | 48.98 | 61.93 |
| ✓ | | ✓ | | 57.14 | 64.54 | 49.95 | 56.94 |
| ✓ | ✓ | ✓ | | 57.69 | 67.46 | 49.39 | 56.23 |
| ✓ | ✓ | ✓ | ✓ | 58.78 | 65.61 | 49.02 | 61.70 |
| ✓ | ✓ | ✓ | ✓ | 59.56 | 65.25 | 53.29 | 60.15 |
| | | ✓ | ✓ | 59.88 | 65.36 | 53.64 | 60.64 |

Table 8: Ablation study of the second-stage components for Voxel R-CNN (2021a) with the AFDetV2 components. All the experiments were done under the framework OpenPCDet (2020), with the default settings of Voxel R-CNN. All models were trained on 1/5 subset of the *training* set and validated on 1/5 subset of *val* set. All results are in 3D APH/L2. The top row shows the contributions of the box and score refinements in the second stage. It can be seen that the score refinement was the main factor of the improvement. The second row shows that by replacing the convolution blocks in the backbone with the SC-Conv block, the overall APH was improved significantly, even without the second stage. The last row shows that the second stage is unnecessary when we exploit the SC-Conv and IoU branch in the first stage.

detection. We list all published ensemble models in the bottom row of the Tab. 4. Our AFDetV2-Ens outperforms all published prior arts. It also ranks the 1st on the official Non-Real-Time 3D Detection leaderboard². While only LiDAR is used as input for AFDetV2-Ens, it still outperforms HorizonLiDAR3D which uses both camera and LiDAR input. Our AFDetV2-Ens only takes 2 frames as input. We expect that adding more frames, e.g. 3DAL (Qi et al. 2021) uses ~200 frames, can further improve the detection accuracy.

In Fig. 3, we show some visualization results of our AFDetV2 on Waymo Open Dataset *test* set.

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 is given the title of “Most Efficient Model”.

Our entry AFDetV2 won the 1st place in the Real-Time 3D Detection of WOD Challenge 2021. Our AFDetV2-Base is a faster one and is only 0.55% lower than AFDetV2. AFDet-Base was entitled the “Most Efficient Model” by the WOD Challenge Sponsor, showing a superior computational efficiency.

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

²<https://waymo.com/open/challenges/2020/3d-detection/>, accessed on Dec 5, 2021.

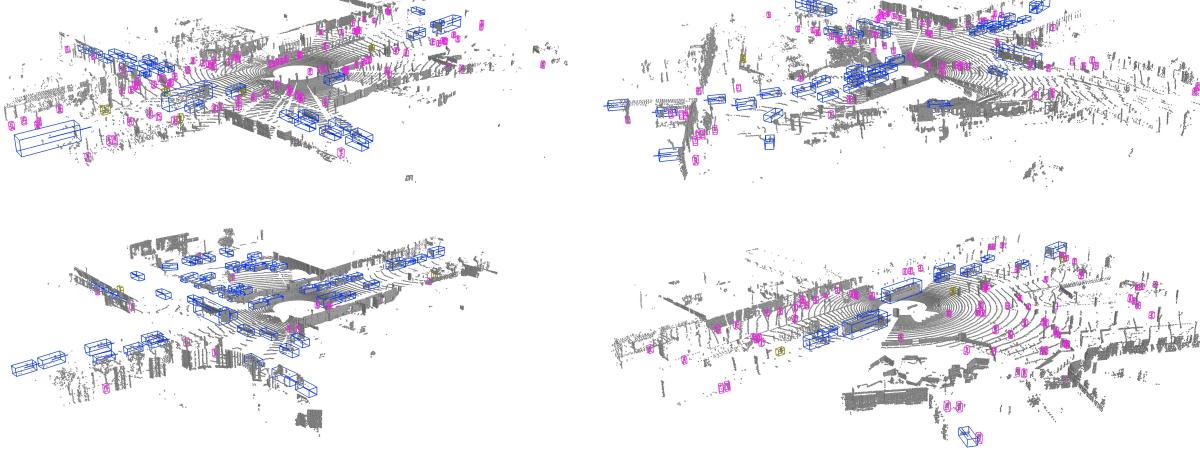


Figure 3: The detection results of AFDetV2 on Waymo Open Dataset *test* set, only bounding boxes with score larger than 0.50 are visualized. The bounding boxes of VEHICLE, PEDESTRIAN and CYCLIST are in the color blue, magenta and olive, respectively. Additional NMS is conducted for better visualization. Better viewed in color and zoom in for more details.

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 (Liu et al. 2020) in backbone led to an increment of 0.55 ALL APH/L2, while adding IoU rescoring enhanced the model with a considerable 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.

Analysis of two-stage vs. single-stage We conduct in-depth analysis over each component of the R-CNN style two-stage detector. In Tab. 7, we migrate the second stage network proposed by CenterPoint (2021a) to AFDetV2. We feed five ROI features to the second stage for each first-stage proposal. The score and box are refined at the second stage by multi-layer perceptrons. The second stage box refinement alone would not bring performance improvement. The second stage box refinement and the second confidence score 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.

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 SEC-OND (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 in the second stage would bring significant improvement (+5.61 ALL APH/L2) over the baseline. In the second row, when we enhance its first stage by SC-Conv backbone, the box refinement in the second stage alone shows only minor improvement, which means bounding box regression in the first stage can achieve enough precision. In the last row, compared to the original first stage model, our enhanced single-stage model with SC-Conv and IoU rescoring shows higher improvement (+6.65 ALL APH/L2) than adding the second stage (+5.61 ALL APH/L2). Also, an extra second stage is useless to the overall performance, which is similar to our findings in Tab. 7. Once again, we prove that the second stage is unnecessary if we strengthen and fully utilize the first stage by our proposed methods.

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

- Bewley, A.; Sun, P.; Mensink, T.; Anguelov, D.; and Sminchisescu, C. 2020. Range Conditioned Dilated Convolutions for Scale Invariant 3D Object Detection. In *CoRL*.
- Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liang, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*.
- Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving into High Quality Object Detection. In *CVPR*.
- Caine, B.; Roelofs, R.; Vasudevan, V.; Ngiam, J.; Chai, Y.; Chen, Z.; and Shlens, J. 2021. Pseudo-labeling for Scalable 3D Object Detection. arXiv:2103.02093.
- 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*.
- Chen, Q.; Sun, L.; Cheung, E.; and Yuille, A. L. 2020a. Every View Counts: Cross-View Consistency in 3D Object Detection with Hybrid-Cylindrical-Spherical Voxelization. In *NeurIPS*.
- Chen, Q.; Sun, L.; Wang, Z.; Jia, K.; and Yuille, A. 2020b. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *ECCV*.
- Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-View 3D Object Detection Network for Autonomous Driving. In *CVPR*.
- Cheng, S.; Leng, Z.; Cubuk, E. D.; Zoph, B.; Bai, C.; Ngiam, J.; Song, Y.; Caine, B.; Vasudevan, V.; Li, C.; et al. 2020. Improving 3d Object Detection Through Progressive Population based Augmentation. In *ECCV*.
- Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021a. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In *AAAI*.
- 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*.
- Ding, Z.; Hu, Y.; Ge, R.; Huang, L.; Chen, S.; Wang, Y.; and Liao, J. 2020. 1st Place Solution for Waymo Open Dataset Challenge—3D Detection and Domain Adaptation. arXiv:2006.15505.
- Fan, L.; Xiong, X.; Wang, F.; Wang, N.; and Zhang, Z. 2021. Rangedet: In Defense of Range View for LiDAR-Based 3D Object Detection. arXiv:2103.10039.
- Ge, R.; Ding, Z.; Hu, Y.; Shao, W.; Huang, L.; Li, K.; and Liu, Q. 2021. Real-Time Anchor-Free Single-Stage 3D Detection with IoU-Awareness. arXiv:2107.14342.
- Ge, R.; Ding, Z.; Hu, Y.; Wang, Y.; Chen, S.; Huang, L.; and Li, Y. 2020. AFDet: Anchor Free One Stage 3D Object Detection. In *CVPR Workshops*.
- Girshick, R. 2015. Fast r-cnn. In *ICCV*.
- Graham, B.; and van der Maaten, L. 2017. Submanifold Sparse Convolutional Networks. arXiv:1706.01307.
- Guan, T.; Wang, J.; Lan, S.; Chandra, R.; Wu, Z.; Davis, L.; and Manocha, D. 2021. M3DeTR: Multi-representation, Multi-scale, Mutual-relation 3D Object Detection with Transformers. arXiv:2104.11896.
- Gugger, S. 2018. The 1cycle policy. <https://sgugger.github.io/the-1cycle-policy.html>. Accessed: 2021-12-05.
- He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; and Zhang, L. 2020. Structure Aware Single-stage 3D Object Detection from Point Cloud. In *CVPR*.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask R-CNN. In *ICCV*.
- Hu, P.; Ziglar, J.; Held, D.; and Ramanan, D. 2020. What you see is what you get: Exploiting visibility for 3d object detection. In *CVPR*.
- Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*.
- Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging Weights Leads to Wider Optima and Better Generalization. In *UAI*.
- Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; and Jiang, Y. 2018. Acquisition of Localization Confidence for Accurate Object Detection. In *ECCV*.
- Kuang, H.; Wang, B.; An, J.; Zhang, M.; and Zhang, Z. 2020. Voxel-FPN: Multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds. *Sensors*.
- Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *CVPR*.
- Law, H.; and Deng, J. 2018. Cornernet: Detecting Objects as Paired Keypoints. In *ECCV*.
- Li, Z.; Wang, F.; and Wang, N. 2021. LiDAR R-CNN: An Efficient and Universal 3D Object Detector. In *CVPR*.
- Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *ICCV*.
- Liu, H.; Zheng, R.; Peng, J.; and Tian, L. 2021. 3rd Place Solution of Waymo Open Dataset Challenge 2021 Real-time 3D Detection Track. <https://bit.ly/3DsdkUr>. Accessed: 2021-12-05.
- Liu, J.-J.; Hou, Q.; Cheng, M.-M.; Wang, C.; and Feng, J. 2020. Improving Convolutional Networks with Self-Calibrated Convolutions. In *CVPR*.
- Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *ICLR*.
- Mao, J.; Niu, M.; Bai, H.; Liang, X.; Xu, H.; and Xu, C. 2021a. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In *ICCV*.
- Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; and Xu, C. 2021b. Voxel transformer for 3d object detection. In *ICCV*.
- Meyer, G. P.; Laddha, A.; Kee, E.; Vallespi-Gonzalez, C.; and Wellington, C. K. 2019. LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. In *CVPR*.

- Murhij, Y.; and Yudin, D. 2021. Real-time 3D Object Detection using Feature Map Flow. arXiv:2106.14101.
- Ngiam, J.; Caine, B.; Han, W.; Yang, B.; Chai, Y.; Sun, P.; Zhou, Y.; Yi, X.; Alsharif, O.; Nguyen, P.; et al. 2019. Starnet: Targeted Computation for Object Detection in Point Clouds. arXiv:1908.11069.
- Noh, J.; Lee, S.; and Ham, B. 2021. HVPR: Hybrid Voxel-Point Representation for Single-stage 3D Object Detection. In *CVPR*.
- OpenPCDet Development Team. 2020. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds. <https://github.com/open-mmlab/OpenPCDet>. Accessed: 2021-12-05.
- Pan, X.; Xia, Z.; Song, S.; Li, L. E.; and Huang, G. 2021. 3d object detection with pointformer. In *CVPR*.
- Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. In *ICCV*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Point-Net++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*.
- Qi, C. R.; Zhou, Y.; Najibi, M.; Sun, P.; Vo, K.; Deng, B.; and Anguelov, D. 2021. Offboard 3D Object Detection from Point Cloud Sequences. In *CVPR*.
- Ren, S.; He, K.; Girshick, R.; and Sun, J. 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *TPAMI*.
- Sheng, H.; Cai, S.; Liu, Y.; Deng, B.; Huang, J.; Hua, X.-S.; and Zhao, M.-J. 2021. Improving 3d object detection with channel-wise transformer. In *ICCV*.
- Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020a. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *CVPR*.
- Shi, S.; Guo, C.; Yang, J.; and Li, H. 2020b. PV-RCNN: The Top-Performing LiDAR-only Solutions for 3D Detection/3D Tracking/Domain Adaptation of Waymo Open Dataset Challenges. arXiv:2008.12599.
- Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; and Li, H. 2021. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. arXiv:2102.00463.
- Shi, S.; Wang, X.; and Li, H. 2019. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. 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*.
- Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *CVPR*.
- Sun, P.; Wang, W.; Chai, Y.; Elsayed, G.; Bewley, A.; Zhang, X.; Sminchisescu, C.; and Anguelov, D. 2021. RSN: Range Sparse Net for Efficient, Accurate LiDAR 3D Object Detection. In *CVPR*.
- Wang, F.; Li, Z.; Yan, Y.; and Wang, N. 2020a. Multi-Stage Refinement Network for Point Cloud 3D Object Detection. <https://bit.ly/31zNIYO>. Accessed: 2021-12-05.
- Wang, G.; Tian, B.; Ai, Y.; Xu, T.; Chen, L.; and Cao, D. 2020b. CenterNet3D: An Anchor free Object Detector for Autonomous Driving. arXiv:2007.07214.
- Wang, Y.; Chen, S.; Huang, L.; Ge, R.; Hu, Y.; Ding, Z.; and Liao, J. 2020c. 1st Place Solutions for Waymo Open Dataset Challenges—2D and 3D Tracking. arXiv:2006.15506.
- Wang, Y.; Fathi, A.; Kundu, A.; Ross, D. A.; Pantofaru, C.; Funkhouser, T.; and Solomon, J. 2020d. Pillar-based Object Detection for Autonomous Driving. In *ECCV*.
- Wu, S.; Li, X.; and Wang, X. 2020. IoU-aware Single-stage Object Detector for Accurate Localization. *IVC*.
- Xu, J.; Tang, X.; Dou, J.; Shu, X.; and Zhu, Y. 2021. CenterAtt: Fast 2-stage Center Attention Network. arXiv:2106.10493.
- 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*.
- Yin, J.; Shen, J.; Guan, C.; Zhou, D.; and Yang, R. 2020. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *CVPR*.
- Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021a. Center-Based 3D Object Detection and Tracking. In *CVPR*.
- Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021b. CenterPoint++ Submission to the Waymo Real-time 3D Detection Challenge. <https://bit.ly/32T4XVp>. Accessed: 2021-12-05.
- Zhang, H.; Wang, Y.; Dayoub, F.; and Sünderhauf, N. 2021. VarifocalNet: An IoU-aware Dense Object Detector. In *CVPR*.
- Zheng, W.; Tang, W.; Chen, S.; Jiang, L.; and Fu, C.-W. 2021. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. In *AAAI*.
- Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; and Vasudevan, V. 2020. End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds. In *CoRL*.
- Zhou, Y.; and Tuzel, O. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*.
- Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; and Yu, G. 2019. Class-balanced Grouping and Sampling for Point Cloud 3D object Detection. arXiv:1908.09492.
- Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Li, W.; Ma, Y.; Li, H.; Yang, R.; and Lin, D. 2021. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *TPAMI*.