Anchor-free 3D Single Stage Detector with Mask-Guided Attention for Point Cloud

Jiale Li
jialei@zju.edu.cn
Zhejiang University
Hangzhou, China

Ling Shao
ling.shao@ieee.org
Inception Institute of Artificial Intelligence
Abu Dhabi, United Arab Emirates

Hang Dai*
hang.dai@mbzuai.ac.ae
Mohamed bin Zayed University of Artificial Intelligence
Abu Dhabi, United Arab Emirates

Yong Ding*
dingy@vlsi.zju.edu.cn
Zhejiang University
Hangzhou, China

1 INTRODUCTION

Ongoing development of various applications such as autonomous driving [5, 11, 32] and robotics [41] has led to the increasing demand for 3D object detectors that predict the class label, 3D bounding box and detection confidence score for each instance in a given 3D scene. In recent years, the 2D object detection methods have made great breakthroughs [6, 15, 22, 49, 55, 59], far ahead of the 3D object detection methods, in particular, the anchor-free detection methods. As one of the vital media for 3D space modeling and 3D visual processing, point clouds have received increasing attention from the multimedia community [10, 17, 53, 54]. The non-uniform and sparse points in the point cloud are much different from the pixels uniformly and regularly distributed in the image. So it is difficult to directly leverage the advanced 2D object detection methods for 3D object detection.

Although we can convert the point cloud into an image-like 2D data structure at the input level [3, 20, 38, 48], this causes the issue of losing important information such as the 3D shape of the object in the data transformation. Thus, we propose a lossless feature-level 3D-to-2D transformation. The voxel-based methods [58, 67] are a family of methods that apply the sparse 3D convolution on the voxelized point cloud in 3D space to learn the structured 3D feature volumes, which explicitly explore the 3D shape information. Since there is no perspective distortion and overlaps among objects in the Bird’s Eye View (BEV) [3, 20], we can losslessly convert the structured sparse 3D feature volumes into the sparse 2D feature maps by flattening the features in the height dimension into the channel dimension [14, 42].

Although the sparse 2D feature maps can be generated by voxel-based feature learning, it is still far away from regressing the bounding box without anchors [49, 64]. Only the partial surface of the object is captured in the point cloud acquisition process, and most regions in the point cloud are empty. Thus, the central area of the object is also empty in the sparse 2D feature maps, as shown in Fig. 1. However, the 2D approaches [49, 63, 64] demonstrate that the central area of the object is very important in locating the object without anchors. To overcome this, we propose an Attentive Deformable Feature Adaptation (ADFA) module to make the sparse 2D feature maps fit to dense, especially in the object regions. The ADFA module incorporates the feature maps on three scales with the small receptive fields on the high-resolution features and the large receptive fields on the low-resolution features. At each scale, we employ the deformable convolution to adaptively adjust the...
Figure 1: Top: the RGB image and the occupancy grids of the sparse 3D feature volume in the BEV. The grid with the black pixel is empty and the red dot represents the center point of an object. Object instances with the empty central areas are highlighted by yellow boxes. Bottom: the input feature map, the output feature map of the ADFA module and the predicted mask-guided attention map. Note that the output feature map of the ADFA is much denser.

receptive fields so that the center of the object can better perceive the entire shape information of the object. The multi-scale feature maps are further penalized with supervised mask-guided attention for highlighting the object from the complex background.

Different from most of the existing 3D detectors [42, 58, 61], we propose a simple and effective voxel-based 3D single-stage object detector in an anchor-free manner, which detects 3D objects as the centers of the objects in the BEV. The anchor-based 3D detectors are very popular. To handle the variable objects, they tile numerous anchor boxes of predefined locations, sizes, and even orientations in 3D space. The large searching space and the numerous anchor box attributes magnify the issues caused by the imbalance of positive and negative samples. And the anchor box attributes need to be defined manually. However, only a few methods focus on the challenging anchor-free 3D object detection [2].

The classification score from the detection head is usually used as the detection confidence assigned to the best prediction results. However, the classification score is not positively related to the location accuracy of the predicted bounding boxes [18]. Thus, we propose a novel detection confidence re-calibration scheme to improve the correlation. Since the 3D Intersection-over-Union (IoU) reflects the location accuracy, a confidence estimation head is designed in parallel with other detection heads using the 3D IoUs of the predicted bounding boxes as the training target. In the inference, the predicted IoU is used to re-calibrate the classification score based detection confidence. The experimental results show that the re-calibrated confidence improves the correlation between the detection confidence and the location accuracy, and significantly enhances the 3D object detection performance.

2 RELATED WORK

Anchor-based detector. As predefined anchors ease the detection process, both the 2D [15, 40] and 3D [14, 33, 42, 45, 61] detectors are dominated by anchor-based methods. They first evenly place numerous predefined anchors on the detection space as all potential candidates. Then, they detect objects by predicting the classes and refining the positions of anchors in one [28, 29, 67] or several [1, 27, 44] times. The complex anchor boxes with different attributes need to be manually designed and tuned according to the dataset.

Anchor-free detector. The 2D anchor-free approaches can be grouped into two categories. One way is to represent the object as the center point or central area. In [39, 64], only the grid cell that contains the object center is responsible for detecting this object. With the center sampling strategy [63], FCOS [49] can be improved by regressing the box from the central area instead of the entire area inside the object bounding box, which indicates that the central area of the object can produce better predictions. The other way is to generate an axis-aligned 2D bounding box by grouping multiple defined keypoints [8, 22] or self-learned points [59, 65]. However, it is not feasible to transform multiple keypoints into a rotated 3D bounding box without other constraints. Recently, the anchor-free 3D detector HotSpotNet [2] defines hotspots as the non-empty voxels that only lie on the surface of the object. The locations of the non-empty voxels usually vary due to the perspective issues and lack consistency compared with the centers of the objects.

Structured representations. Many methods convert unstructured point clouds to regular grids. MV3D [3] and AVOD [20] project point clouds as the 2D images in the BEV or Front View (FV). The projected 2D images can be applied to 2D convolutional networks at the cost of information loss in the projection process. VoxeNet [67] evenly divides a point cloud into small 3D voxels in XYZ directions for 3D convolutional processing. PointPillars [21] roughly generate pillars along the height direction and the following works [24, 30, 62] develop better strategies for expressive feature learning. Considering that most of the voxels are empty, SECOND [58] introduces the sparse 3D convolution [12, 13] for efficient voxel processing. With sparse 3D convolution, these voxel-based methods [14, 42, 44] are efficient for accurate 3D object detection using the structured point cloud features.

Unstructured representations. The PointNet [36] and PointNet++ [37] introduce a new way of learning point-wise features directly on the point cloud without any representational transforms. F-PointNet [35] first employs [37] to extract the local point cloud features in the frustum proposals generated by the image. To avoid perspective distortion and object occlusion in the image, PointRCNN [43] and STD [61] directly generate a 3D object proposal on each point of the entire point cloud [25]. However, the grouping of points on a large scale leads to computational costs. 3DSSD [60] limited the proposals in a sampled point set, but objects may not be retained in the uneven down-sampling process. PV-RCNN [42] proposes a Voxel Set Abstraction operation to aggregate the voxel-wise features in some sampled keypoints. The two-stage methods STD and PV-RCNN both learn the IoUs of the proposal boxes to guided the NMS with detection performance improvements. Besides, Point-GNN [45] directly builds a graph on the entire point cloud, and designs a graph neural network (GNN) [56] to learn vertex-wise features with local relationships, and conducts 3D object detection on vertexes. The flexible point cloud feature learning based on the points and graphs still does not change the unordered and sparse properties of point clouds.
When it is projected into the BEV, the 3D feature volume can be reshaped to the sparse 3D feature volume. Thus, the 3D feature volume can be reshaped into the sparse 2D feature map in the BEV and adapted to dense by the ADFA module. Finally, three detection heads perform the center point classification, the 3D bounding box regression, and the IoU-based detection confidence estimation, respectively. The "Obj" is short for "Object" and the feature channels are not shown in this figure for clarity. Each stage can be clearly distinguished in different series of colors.

3 METHOD

The overall architecture is presented in Fig. 2, including three main components: 1) ResVoxelNet backbone that efficiently encodes the point cloud into the sparse 3D feature volume; 2) ADFA module that conducts feature enhancement on the reshaped 2D feature map; 3) our anchor-free detection head.

3.1 ResVoxelNet: 3D Convolutional Backbone

Following [14, 42, 58], a point cloud is uniformly divided into 3D voxels with spatial resolution of $(L_0, W_0, H_0)$ by a quantization step $d = (d_x, d_y, d_z)$. The feature of each non-empty voxel is initialized to the mean of the commonly used point-wise features (e.g., 3D coordinates and reflection intensity) of the points inside. The ResVoxelNet is used for the hierarchical voxel-based feature extraction, which is the refined VoxelNet [67] with residual blocks [16]. The network employs a series of $3 \times 3 \times 3$ 3D sparse convolutional blocks to gradually convert the point clouds into feature volumes with $1 \times 1$, $2 \times 2$, $4 \times 4$, $8 \times 8$ downsampled sizes. The first layer of each block downsamples the feature volumes implemented by a normal sparse convolution [12], followed by a sequence of sparse residual blocks. The sparse residual block contains an identity connection and a parallel residual connection, which operates on the sparse tensors. The residual connection is composed of two $3 \times 3 \times 3$ submanifold sparse convolutions [13] with stride 1 for learning the residual features.

3.2 Attentive Deformable Feature Adaptation

Feature-level 3D-to-2D transformation. From ResVoxelNet, we can generate 3D feature volume in a shape as $(C_1, L, W, H)$. It takes too much computational cost when it is processed further or directly followed by a detection head. Considering that the object still maintains its original physical size without distortion and overlaps when it is projected into the BEV, the 3D feature volume can be stacked on the height dimension to form a 2D projection in the BEV following [14, 42, 58]. Thus, the 3D feature volume can be reshaped as $(C_1 \times H, L, W)$. Since the features of the height dimension are merged into the channel dimension, the height information is still retained. Then, we can employ the 2D convolutional operations to handle the feature embeddings.

Deformable convolutional tower. The deformable convolutional tower aggregates the features at different scales and adaptively adjusts the receptive fields with the deformable convolution [68]. As shown in Fig. 3, the convolutional tower contains three levels that share the same structure. The first layer of each level applies a $3 \times 3$ convolutional layer with stride $s_1$ to downsample the feature map, followed by several $3 \times 3$ convolutional layers with the stride of 1. We then use a $3 \times 3$ deformable convolutional layer to adaptively adjust the receptive fields at each position on the feature map. Next, the feature maps from the three branches are upsampled to the desired resolution and then concatenated together. Before using another deformable convolutional layer to further enhance the feature learning, we employ another $3 \times 3$ convolutional layer with the stride of 1 to reduce the channels for computational efficiency. Each convolutional layer is followed by the BatchNorm and ReLU operations. We set the three-level down-sampling stride $s_1$ to 1, 2, 2, and employ three transpose convolutions with strides of 1, 2, 4 to upsample the features.
Learning with mask-guided attention. The point cloud object detection is sensitive to noise [30]. Since noise is inevitable in the real world, it is necessary to introduce attention to strengthen the clues from objects and weaken the clues from non-objects for robust feature learning. We introduce a supervised mask-guided attention mechanism to highlight the object pixels from the complex background. As shown in Fig. 4, the feature map $F$ is fed into a sub-network $\phi_3$ with the sigmoid function $\sigma$ to predict a class agnostic segmentation mask distributed between values of 0 and 1, which can serve as the attention map $S \in [0, 1]^{L \times W}$. The foreground pixels on the objects and the background pixels are expected to have a value close to 1 and 0, respectively. Hence, the objects can be highlighted from the background as
$$S = \sigma_3(F), \quad G = (1 + S)F.$$
An identity connection is applied in Eq. (1) by adding $F$, which aims to preserve the background context information.

The sub-network $\phi_3$ is composed of a $3 \times 3$ convolutional layer followed by the BatchNorm and ReLU operations for feature embedding and a $1 \times 1$ convolutional layer for the output. During training, a target segmentation mask can be generated from the provided ground truth boxes by checking whether the pixel is in the ground truth box in the BEV. The focal loss [28] with default settings is employed to handle the unbalanced number of the foreground and background pixels with the loss term $L_{sem}$.

3.3 Anchor-free Detection Head

The proposed model can learn a dense mask-guided 2D feature map $G$ in $C_2 \times L \times W$ dimension from the point cloud. The height information is retained in the 2D feature map by flatting the features from the height dimension into the channel dimension. Inspired by [64], we construct a new single-stage anchor-free 3D detection head. The $K$-class object detection task is decomposed into three sub-tasks: $K$-class center point classification in the BEV, 3D bounding box regression based on the center points, and estimation of the IoU-based detection confidence. The third part of Fig. 2 shows three parallel heads in a structure similar to the sub-network $\phi_3$.

Center point classification. The center point classification head treats the objects as center points. Given the 2D feature map $G \in \mathbb{R}^{C_2 \times L \times W}$ in the BEV, a heatmap $\hat{P} \in [0, 1]^{K \times L \times W}$ is predicted to represent the probability of the center point of each object. The local peaks of the $k$-th channel can be detected as the center points representing the objects of class $k$. And the center point can be directly decoded as the object’s center coordinates $(x, y)$ in the BEV.

The target heatmap $P \in [0, 1]^{K \times L \times W}$ can be generated following [22, 64]. For each ground truth bounding box of class $k$, the pixel index $\hat{p} \in \mathbb{R}^2$ of the center point $p \in \mathbb{R}^2$ on the feature map can be calculated as $\hat{p} = \lfloor \frac{p}{R} \rfloor$, where $R$ is the output stride. Then a 2D Gaussian $\exp(-\frac{(x-x_\hat{p})^2+(y-y_\hat{p})^2}{2\sigma_p^2})$ is placed around $\hat{p}$ to form the target heatmap, where $\sigma_p$ is a box-size adaptive standard deviation [22]. Only the pixels $\hat{p}$ at center points are positive, and all other pixels are negative. We use a variant of focal loss to supervise the center classification task as:
$$L_{cls} = -\frac{1}{N} \sum_{i} \left\{ \begin{array}{ll} (1 - \hat{P}_i)^{\alpha} \log(\hat{P}_i) & \text{if } \hat{P}_i = 1, \\ (1 - \hat{P}_i)^{\beta} \hat{P}_i^{\gamma} \log(1 - \hat{P}_i) & \text{otherwise,} \end{array} \right.$$  
where $i$ indexes the pixel location on the heatmap, $N$ is the number of objects, and the hyperparameters $\alpha$ and $\beta$ are selected as $\alpha = 2$ and $\beta = 4$ [22]. Please refer to Fig. 7.(h) - (j) for a clear visualization.

3D Bounding box regression. The 3D object detection locates an object as $(x, y, z, l, w, h, \theta)$, where $(x, y, z)$, $(l, w, h)$, and $\theta$ are denoted as the 3D position coordinates, 3D bounding box size, and the rotation angle of the object.

Although the $(x, y)$ can be decoded from the integer coordinates of the detected center point from the 2D feature map, we can predict an offset $(\Delta x, \Delta y)$ to compensate for the difference between the actual floating-point coordinates and the integer coordinates of the center point. Since the height information is still retained in the 2D feature map, the coordinate $z$ and the size $(l, w, h)$ are directly regressed. The predictions of the offset $(\Delta x, \Delta y)$, the coordinate $z$ and the size $(l, w, h)$ are all trained with the L1 loss:
$$L_{11} = \frac{1}{N} \sum_{\hat{p}} |\hat{A}_p - A_p|,$$
where only the center point locations $\hat{p}$ on the prediction map $\hat{A}$ and that on the target map $A$ are computed for the loss. All other locations are ignored.

To ease the regression of the rotation angle without any predefined oriented anchors, we divide $(0, 2\pi)$ into $b$ bins and use the bin-res loss [43] to conduct the supervision. The model first roughly classifies which bin the $\theta$ is in, and then finely regresses the residuals in this bin. For each center point location $\hat{p}$, the rotation angle loss $L_{rot, \hat{p}}$ can be composed of the bin classification loss $L_{rot, \bin}$ and the residual regression loss $L_{rot, \res}$
$$L_{rot, \hat{p}} = L_{rot, \bin} + L_{rot, \res},$$
where $\hat{r}_{\bin}$ and $\hat{r}_{\res}$ are predicted bin classes and residuals while $r_{\bin}$ and $r_{\res}$ are their targets. The total rotation angle loss of $N$ objects can be formulated as
$$L_{rot} = \frac{1}{N} \sum_{\hat{p}} L_{rot, \hat{p}}.$$  
Moreover, a corner loss is used to supervise all items related to the 3D bounding box as a whole. The corner loss $L_{\text{corner}}$ is defined as the L2 distance between the eight corners of the predicted bounding
box and the assigned ground truth as

\[ L_{\text{corner}} = \frac{1}{N} \sum_{\hat{p}} \sum_{k=1}^{8} \| \hat{B}_{\hat{p}, k} - B_{\hat{p}, k} \|, \]

where \( \hat{p} \) indicates that the \( L_{\text{corner}} \) only considers the bounding boxes regressed from the center point locations. Therefore, the bounding box prediction is supervised by

\[ L_{\text{box}} = y_1 L_{\text{offset}} + y_2 L_{z\text{-coord}} + y_3 L_{\text{size}} + y_4 L_{\text{rot}} + y_5 L_{\text{corner}}, \]

where \( y_1 \sim y_5 \) are the constant coefficients to balance each loss.

\textbf{IoU-based detection confidence recalibration.} To detect an object with a confidence score, we can directly use the local peak value on the center point classification heatmap as detection confidence of the object [64]. The model can achieve better detection performance when the detection confidence is positively correlated with the IoU of the predicted bounding box [18, 25]. However, the local peak values on the classification heatmap are all close to 1 and have a low positive correlation with the actual IoUs of the predicted bounding boxes. To improve the correlation, a parallel confidence estimation head is designed to predict IoU-based confidence. The detection confidence is re-calibrated from the classification score to the IoU-based confidence.

The training samples and targets for the confidence estimation head are obtained automatically during the training of the network. The top \( M \) local peaks on the predicted classification heatmap are utilized to select the corresponding \( M \) predicted bounding boxes as the confidence training samples. \( M \) is a hyperparameter we chose to balance the training samples with IoUs greater than zero and equal to zero. For the \( m \)-th bounding box, we calculate the 3D IoUs between the \( m \)-th bounding box and all ground truth bounding boxes, and then assign the largest one as its actual IoU denoted as \( \text{IoU}_m \). We follow [42] to normalize each confidence training target \( c_m \) between [0, 1] as

\[ c_m = \min(1, \max(0, 2\text{IoU}_m - 0.5)). \]

The confidence estimation head is then trained to minimize the cross-entropy loss computed from the predicted confidence \( \hat{c}_m \) and its target \( c_m \) as

\[ L_{\text{iou}} = - \frac{1}{M} \sum_{m} (c_m \text{log}(\hat{c}_m) + (1 - c_m)\text{log}(1 - \hat{c}_m)). \]

In the inference stage, the confidence \( \hat{c} \) predicted by the network can be directly assigned to the corresponding predicted bounding box as the detection confidence score.

The two-stage methods PV-RCNN [42] and STD [61] also learn the IoUs of the proposal boxes to guide the NMS. Note that there are two main differences in our method: 1) PV-RCNN and STD require the actual IoUs of the proposal boxes as the training target for their confidence branches in the second stage. However, there are no proposal boxes or anchor boxes in our one-stage anchor-free methods. Instead, we use IoUs of the predicted boxes as training targets for IoU prediction in a parallel regression head. 2) PV-RCNN and STD use the predicted IoUs for refined boxes de-redundancy in NMS, while we only re-calibrate the detection confidence using the predicted IoUs without NMS.

\textbf{Training loss.} Our network is trained end-to-end with a multitask loss \( L_{\text{total}} \) as Eq.(10), which is equally contributed by the center point classification loss \( L_{\text{cls}} \), the 3D bounding box regression loss \( L_{\text{box}} \), the IoU-based confidence training loss \( L_{\text{iou}} \), and the mask-guided attention training loss \( L_{\text{sem}} \).

\[ L_{\text{total}} = L_{\text{cls}} + L_{\text{box}} + L_{\text{iou}} + L_{\text{sem}}. \]

\section{EXPERIMENT}

\subsection{Datasets}

\textbf{KITTI Dataset.} KITTI 3D object detection benchmark [19] is widely used for 3D object detection evaluation. The KITTI dataset [11] provides 7481 training samples and 7518 testing samples with objects of three classes: car, pedestrian, and cyclist. The training samples are generally divided into the \textit{train} split set (3712 samples) and the \textit{val} split set (3769 samples) for experiments [3, 43, 67]. For each class, the detection performance is evaluated as the Average Precision (AP) based on three difficulty levels: easy, moderate, and hard. Following the official evaluation protocol, we calculate the AP\textsubscript{3D} with 40 recall positions as the evaluation metric.

\textbf{Waymo Open Dataset.} The newly released Waymo Open Dataset [47] is the currently largest public dataset for autonomous driving. Waymo contains ~158K point cloud training samples and ~40K point cloud validation samples. The objects in Waymo are divided into two levels: LEVEL\textsubscript{1} for boxes with more than five LiDAR points, and LEVEL\textsubscript{2} for boxes with at least one LiDAR point. The official evaluation tool provides metrics of AP\textsubscript{3D} and AP\textsubscript{BEV} for the 3D and BEV detection.

\subsection{Implement Details}

\textbf{Network architecture.} Since the objects in KITTI are only annotated in the camera Field of View (FOV), the quantization step \( d \) is set as (0.05, 0.05, 0.1) meters to voxelize the KITTI point clouds within the range of \([0, 70.4] \times [-40, 40] \times [-3, 1]\) meters along the \( X, Y, Z \) axis, resulting in an initial spatial resolution \((L_0, W_0, H_0)\) of (1408, 1600, 40). For the full 360°-field detection in Waymo, we voxelize the Waymo point clouds within the range of \([-75.7, 75.7] \times [-75.2, 75.2] \times [-2, 4]\) meters with the quantization step \( d \) as (0.1, 0.1, 0.15) along the \( X, Y, Z \) axis, resulting in an initial spatial resolution \((L_0, W_0, H_0)\) of (1504, 1504, 40). The ResVoxelNet downsamples the feature volumes with \( 1 \times 2 \times 4 \) \( \times 8 \times 8 \) while increasing the feature dimension as 16, 32, 64, 128. The ADFA module maintains the \( 8 \times 8 \) downsampled resolution \((L, W)\) of (176, 200) or (188, 188) with a feature dimension \( C_2 \) of 256 for KITTI and Waymo models. Each head is implemented by two convolutions with stride of 1: a \( 3 \times 3 \) convolution with 128 channels, and a \( 1 \times 1 \) convolution with different numbers of channels for different output values.

\textbf{Training details.} Our model is trained from scratch in an end-to-end manner with the AdamW [31] optimizer and one-cycle policy [46] with LR max 0.01, division factor 10, momentum ranges from 0.95 to 0.85, weight decay 0.01. For KITTI, the model is trained for 50 epochs with batch size 8 on a Tesla V100 GPU. For Waymo, the model is trained for 36 epochs with batch size 36 on 4 Tesla V100 GPUs. To avoid overfitting during the training process, four commonly used data augmentation strategies are used to avoid overfitting; ground truth sampling [58], random flipping along the \( X \) axis, global scaling with a random scaling factor in \([0.95, 1.05]\), global rotation around the \( Z \) axis with a random angle in \([-\frac{\pi}{4}, \frac{\pi}{4}]\).
Table 1: Performance comparison on car class of the KITTI test set. Methods are divided into two types: two-stage and single-stage. Blue numbers are results for the best detection performance. Bold numbers denote the top-2 results for single-stage point cloud detectors.

| Method          | Sensor            | Anchor | AP<sub>3D</sub> | Easy | Mod. | Hard |
|-----------------|-------------------|--------|-----------------|------|------|------|
| MV3D[3]         | LiDAR + RGB      | ✓      | 74.97           | 63.63| 54.00|
| F-PointNet[35]  | LiDAR + RGB      | ✓      | 82.19           | 69.79| 60.59|
| PointPainting[50]| LiDAR + RGB     | ✓      | 82.11           | 71.70| 67.08|
| AVOD[20]        | LiDAR + RGB      | ✓      | 83.07           | 71.76| 65.73|
| PI-RCNN[57]     | LiDAR + RGB      | ✓      | 84.37           | 74.82| 70.03|
| CLOCs_PointCas[34]| LiDAR + RGB   | ✓      | 87.50           | 76.68| 71.20|
| MMF[26]         | LiDAR + RGB      | ✓      | 88.40           | 77.43| 70.22|
| PointRCNN[43]   | LiDAR only       |        | 86.96           | 75.64| 70.70|
| FastPointRCNN[4]| LiDAR only       |        | 85.29           | 77.40| 70.24|
| STD[61]         | LiDAR only       |        | 87.95           | 79.71| 75.09|
| Part-A2[44]     | LiDAR only       |        | 87.81           | 78.49| 73.51|
| PV-RCNN[42]     | LiDAR only       | ✓      | 90.25           | 81.43| 76.82|

VoxNet[67]  | LiDAR only       | ✓      | 78.82           | 64.17| 57.51|
| SECOND[58]    | LiDAR only       | ✓      | 83.34           | 72.55| 65.82|
| PointPillars[21]| LiDAR only     | ✓      | 82.58           | 74.31| 68.99|
| SCNet[52]     | LiDAR only       | ✓      | 83.34           | 73.17| 67.93|
| TA3Net[60]    | LiDAR only       | ✓      | 84.39           | 75.94| 68.82|
| Associate-3Ddet[7]| LiDAR only    | ✓      | 85.99           | 77.40| 70.53|
| Point-GNN[45] | LiDAR only       | ✓      | 88.33           | 79.47| 72.29|
| HotSpotNet[2] | LiDAR only       | ✓      | 89.60           | 78.31| 73.54|
| Ours           | LiDAR only       | ✓      | 88.16           | 79.68| 72.39|

Table 2: AP<sub>3D</sub> comparison on cyclist and pedestrian classes of the KITTI test set. Methods are divided into two types: two-stage and single-stage. Blue numbers are results for the best detection performance. Bold numbers denote the top-2 results for single-stage point cloud detectors.

| Method          | Anchor | AP<sub>3D</sub> | Easy | Mod. | Hard |
|-----------------|--------|-----------------|------|------|------|
| AVOD-FPN[20]    | ✓      | 63.76           | 50.53| 44.93| 50.46| 42.27| 39.04|
| F-PointNet[35]  | ✓      | 72.27           | 56.12| 49.01| 50.53| 42.15| 38.08|
| STD[61]         | ✓      | 78.69           | 61.59| 55.30| 53.29| 42.47| 38.35|
| PointPainting[50]| ✓      | 77.63           | 63.78| 55.89| 50.32| 40.97| 37.87|
| PV-RCNN[42]     | ✓      | 78.60           | 63.71| 57.65| 50.65| 43.09| 39.65|

SCNet[52]  | ✓      | 67.98           | 50.79| 45.15| 47.83| 38.66| 35.70|
| PointPillars[21]| ✓      | 77.10           | 58.65| 51.92| 51.45| 41.92| 38.89|
| SemanticVoxel[9]| ✓      | 75.70           | 59.44| 52.53| 53.72| 44.34| 40.49|
| TANet[30]      | ✓      | 80.64           | 63.43| 55.15| 50.65| 43.09| 39.65|

Ours  | ✓      | 80.64           | 63.43| 55.15| 50.65| 43.09| 39.65|

4.3 Results on KITTI Dataset

Fig. 5 shows the qualitative results of our method on the KITTI test set. Tab. 1 and Tab. 2 present the performance comparison between our anchor-free single-stage method and the most existing state-of-the-art methods on the KITTI test set. For the main considered car class shown in Tab. 1, our method makes AP<sub>3D</sub> margins of 0.56%, 1.37% on the easy and moderate levels compared with the latest anchor-free method HotSpotNet [2], which shows that the object center is more robust than the various hotspots lying on the object surface. For AP<sub>3D</sub> on the three levels, our method outperforms the state-of-the-art anchor-based single-stage point cloud detectors. Our method still outperforms the most two-stage methods, and achieves better performance with the STD [61] and Part-A2 [44] on the easy level. Our method also has significant improvement when compared to MV3D [3] and AVOD [20], indicating that the feature level 3D-to-2D transformation effectively avoids the information loss caused by the 3D-to-2D data conversion at the input level.

In addition, our method performs well on small objects like pedestrians, as shown in Tab. 2. Most methods do not provide results for cyclists and pedestrians, due to the difficulty of detecting small objects. For cyclists, our method not only outperforms the single-stage detectors, but also achieves better performance against the two-stage anchor-based PV-RCNN [42] on the easy level. The results show the effectiveness of our method.

4.4 Results on Waymo Open Dataset

We evaluate the performance on the Waymo Open Dataset for both LEVEL_1 and LEVEL_2, and compare the AP<sub>3D</sub> and AP<sub>BEV</sub> with the top-performing methods. Tab. 3 shows that we achieve better performance than the existing methods with 71.98% AP<sub>3D</sub> and 87.15% AP<sub>BEV</sub> on the most commonly used LEVEL_1. Besides, our method outperforms PV-RCNN on the BEV of LEVEL_2. The improvements on the AP<sub>BEV</sub> demonstrate the effectiveness of our method for detecting the objects as the center points in the BEV.

4.5 Ablation Study

Extensive ablation experiments are conducted to analyze each individual component of our methods. Following the existing works [42, 43, 45, 55, 58, 61], we train the models on the KITTI trian split set with the same training settings and adopt the AP<sub>3D</sub> on the top-100 local peaks on each channel of the center point classification heatmap. The detection results at the top 50 or 200 local peaks are kept as candidates for KITTI and Waymo since the KITTI point cloud contains more object instances than KITTI. A threshold <i>δ</i> of 0.5 is then applied to filter the negative detection results.

Inference details. For the inference, the Non-maximum suppression (NMS) post-processing process is no longer needed [64]. A 3 × 3 max-pooling operation is applied to extract the local peaks on each channel of the center point classification heatmap. The detection results at the top 50 or 200 local peaks are kept as candidates for KITTI and Waymo since the KITTI point cloud contains more object instances than KITTI. A threshold <i>δ</i> of 0.5 is then applied to filter the negative detection results.
Figure 5: Qualitative results of our method on the KITTI test set. We show the predicted 3D bounding boxes of cars (blue), cyclists (green) and pedestrians (yellow) on both the image and the point cloud. Best viewed with color and zooming in.

Figure 6: Latency comparison: AP\textsubscript{3D} on moderate car class of KITTI test set (top consideration in KITTI) vs. inference time in milliseconds.

Table 3: Performance comparison on the Waymo validation for vehicle detection. The top-1 results are in bold.

| Difficulty | Method          | Stage | Overall | [0, 30)m | [30, 50)m | [50, Inf)m |
|------------|-----------------|-------|---------|----------|-----------|-----------|
| LEVEL_1    | PV-RCNN[42]     | 2     | 65.36   | 91.58    | 65.13     | 36.46     |
|            | Ours            | 1     | 74.78   | 94.39    | 80.87     | 59.16     |
| LEVEL_2    | PV-RCNN[42]     | 2     | 70.30   | 91.92    | 69.21     | 42.17     |
|            | Ours            | 1     | 71.98   | 91.85    | 70.58     | 48.04     |

Table 4: Effectiveness (AP\textsubscript{3D}) of each individual component in the ADFA module. “Conv.”, “DCN”, and “Att.” denote 2D convolutional tower, deformable convolution, and mask-guided attention, respectively.

| Supervision Reweighting | KITTI | Waymo |
|-------------------------|-------|-------|
|                        | Easy  | Mod.  | Hard | LEVEL_1 | LEVEL_2 |
| ×                       | 90.87 | 78.55 | 75.92 | 70.29    | 61.20    |
| √                       | 89.81 | 80.97 | 78.23 | 71.17    | 62.13    |
| √ √                     | 90.27 | 81.14 | 78.40 | 71.42    | 64.24    |
| √ √ √                   | 91.86 | 83.29 | 78.51 | 71.98    | 64.78    |

The importance of the ADFA module. When adding a vanilla convolutional tower combined with multiple receptive fields, the performance becomes better, especially for the moderate and hard levels on KITTI. After the receptive fields are adaptively adjusted by the deformable convolutions and the feature maps are enhanced by the mask-guided attention, the performance becomes better on both KITTI and Waymo. Compared with the input feature map in Fig. 7.(c), the output feature map in Fig. 7.(d) becomes much denser and more focused on the entire object. With the dense mask-guided feature map provided by the ADFA module, our model achieves the best performance as shown in the last row of Tab. 4.

Supervised mask-guided attention. Tab. 5 validates the effectiveness of supervised mask-guided attention. In multi-task learning, the related auxiliary tasks usually facilitate the learning of the main task. As shown in the second row of Tab. 5, the mask-guided supervision contributes to the improvement on KITTI and Waymo. When we further use the mask-guided attention map to penalize the feature map, our method achieves much higher AP\textsubscript{3D}. Fig. 7.(f) illustrates the predicted mask-guided attention map.

Detection confidence recalibration. We investigate the importance of the IoU-based detection confidence recalibration in three models with different settings shown in Tab. 6 and Tab. 7. From model A to model B, the joint learning of the auxiliary IoU-based confidence estimation task brings the slight improvements of 0.72%, 0.61%, 0.45% AP\textsubscript{3D} on KITTI and 0.56%, 0.53% AP\textsubscript{3D} on Waymo. The comparison between the model B and C with the same trained network demonstrates that the proposed IoU-based detection confidence recalibration scheme significantly improves the model B with 2.90%, 2.17%, 1.50% AP\textsubscript{3D} on KITTI and 1.94%, 1.71% AP\textsubscript{3D} on Waymo, respectively. Moreover, we calculate the correlation between the confidence of the detection results and their actual 3D IoUs to analyze how the confidence recalibration mechanism works.
Figure 7: Visualization of the ADFA module and center point classification. (a) RGB image. (b) Occupancy grids of the sparse 3D feature volume in the BEV. (c) Input feature map of ADFA module. (d) Output feature map of ADFA module. (e) Occupancy grids with red points of object centers. (f) Predicted mask-guided attention map. (g) Binary segmentation mask. (h) Target and (i) predicted heatmap of center point classification. (j) The local peaks from the predicted heatmap and the target heatmap are colored in blue and red, and become purple if they overlap.

Table 6: Effectiveness of detection confidence recalibration on KITTI. "Head\textsubscript{iou}" and "Recalib." denote IoU-based detection confidence estimation head and recalibration.

| Setting | Head\textsubscript{iou} Recalib. | PLCC | SRCC | KITTI Easy Mod. Hard |
|---------|-------------------------------|------|------|---------------------|
| A       | × ×                           | 0.478| 0.552| 88.24 80.51 76.56   |
| B       | √ ×                           | 0.485| 0.556| 88.96 81.12 77.01   |
| C       | √ √                           | 0.547| 0.649| 91.86 83.29 78.51   |

The highest Pearson Linear Correlation Coefficient (PLCC) and Spearman Rank-order Correlation Coefficient (SRCC) in the third rows of Tab. 6 and Tab. 7 indicates that the recalibrated confidence becomes more relevant to the detection accuracy.

Hyperparameter $M$. As shown in Tab. 8, the performance is relatively stable with the $M$ around 24 on KITTI but drops as the $M$ becomes larger or smaller. Generally, each scene of the KITTI dataset does not contain many object instances. A large $M$ may include too many negative detection results with IoUs of zero, which causes that the IoU training loss on the positive detection results is overwhelmed by that on the negative detection results. A small $M$ may include many training samples with positive IoUs, which is not conducive to the IoU-based confidence estimation head for distinguishing the false positives with IoUs of zero during the inference. According to the highest mean AP\textsubscript{3D} shown in Tab. 8 and Tab. 9, we choose $M$ as 24 and 104 for KITTI and Waymo, respectively. The reason why the $M$ on Waymo is larger than KITTI is that the full 360°-field scene in Waymo contains more object instances than the camera FOV in KITTI.

5 CONCLUSION

In this work, we propose a novel anchor-free single-stage 3D detection framework. The proposed ADFA module enables the center of the object to better perceive the whole object with the adaptive receptive fields and the supervised mask-guided attention. We design an anchor-free 3D detection head to get rid of hand-crafted anchors and NMS for box de-redundancy without sacrificing the detection accuracy. We propose an IoU-based detection confidence re-calibration scheme to improve the correlation between the detection confidence and the detection accuracy, which significantly improves the 3D object detection performance. Experimental results on the widely used KITTI dataset and the newly released large-scale Waymo Open Dataset demonstrate that our method achieves better performance than the anchor-based 3D object detectors, and each individual component in our method is effective with significant performance gains.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China under Grant 2018YFE0183900.
