ACDet: Attentive Cross-view Fusion for LiDAR-based 3D Object Detection

Jiaolong Xu Guojun Wang Xiao Zhang Guowei Wan*
Baidu Apollo Autonomous Driving Platform
{xujiaolong, wangguojun01, zhangxiao23, wanguowei}@baidu.com

Abstract

Recent works on 3D object detection take the range image as input, which have achieved comparable performance with bird's eye view (BEV) based methods. Compared to BEV, range view provides dense and compact observations which allows for more popular feature encoders. To leverage complementary information of range view and BEV, we present ACDet - a novel single-stage multi-view fusion method. Rather than fusing point-level features from range view and BEV at early stage, the key contribution is that we introduce an attentive cross-view fusion module based on transformer to fuse higher level features, and further adopt a supervised foreground mask learned from BEV features to enhance the fused features. Notably, a geometric-attention kernel is proposed to enhance features extracted from range image. Finally, we design an anchor-free detection head with optimized label assignment strategy, and its performance exceeds the existing anchor-based and anchor-free 3D detection heads by a large margin. We evaluate our ACDet model extensively on the KITTI dataset and Waymo Open Dataset (WOD). ACDet outperforms most of single-stage models on KITTI dataset in terms of multi-class 3D and BEV mean average precision. ACDet also outperforms both range-view and multi-view fusion methods on WOD.

1. Introduction

3D object detection from LiDAR sensors is one of the core modules of autonomous vehicles, and has always been a hot research topic in both academia and industry. Existing methods can be summarized into three categories according to data representations: point-based, voxel-based, and range-view-based methods. Point-based methods [24, 29, 44] extract discriminative point features by PointNet [25] or its variants [26, 38], while usually suffering from the high computation cost and memory demand. The most representative voxel-based method VoxelNet [49] divides point clouds into regular grids, which makes it convenient to apply 3D convolutions. To alleviate the high computational cost of 3D convolutions, SECOND [42] introduces sparse 3D convolutions, while PointPillars [13] compresses the representation into 2D grids and directly applies 2D convolutions. Those voxel-based methods inherently suffer from quantization error. Therefore, recent methods [28, 22] incorporate point features to improve the performance. Compared to point-based and voxel-based representations, range view provides dense and compact observations, and is widely applied in 3D semantic segmentation [20, 5, 41]. It is gratifying that recent works [18, 1, 15, 16, 7, 2] have promoted the use of range-view-based representation in 3D object detection.

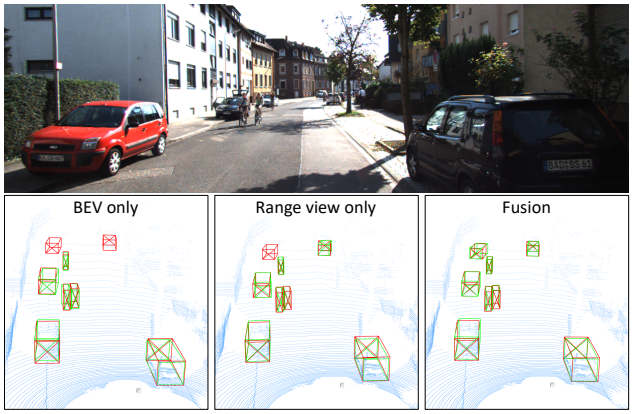

Figure 1: An example of detection results (in green) from BEV only, range view only, and attentive cross-view fusion settings, compared with ground truth (in red). Our attentive cross-view fusion of BEV and range view features could recall those distant objects with few points.

Most range-view-based methods ignore the depth discontinuity between pixels, and directly apply standard 2D convolutions on the range image. The adjacent pixels on range image may be far away in 3D scene, especially the boundary area of objects. Directly applying standard 2D convolutions on the boundary pixels will produce similar features, but in fact they should be distinctly different. To address this issue, recent works [7, 2] enhance features from range image by utilizing the underlying 3D structures. Inspired by these works, we propose a geometric-attention kernel, which takes into account the spatial attention among

*Corresponding author

neighbors and further strengthens the output features by concatenating geometric features. Moreover, to alleviate the computational issue, we introduce a mask-based scheme to avoid unnecessary computation, which further speeds up the kernel operation.

Objects on range view are easily overlapped with each other, which makes it difficult to directly generate proposals from range view. To solve this occlusion issue, recent works [15, 16] project point-wise features extracted from range view to BEV and regress bounding boxes from BEV feature maps. Since the range-view features have been converted to BEV, it is natural to integrate features extracted from BEV. A recent study [21] on multi-modal fusion demonstrates that the use of an independent module to fuse different modalities at deep layers would achieve optimal performance. Our work shares a similar finding with [21] that fusing different modalities at deep layers can achieve a better performance than the fusion of low-level features, since the high-level semantic features from different views contain more complementary information. The early-stage fusion usually prevents the model from learning a rich semantic representation from each specific view. Based on this observation, we propose an attentive cross-view fusion module based on transformer [34] to fuse the deepest feature maps from range view and BEV. Furthermore, a supervised foreground mask learned from the deepest BEV feature map is adopted to further enhance the fused features. The foreground mask is supervised by pixel-level Gaussian distributions of ground truth bounding boxes. It not only enhances the fused features, but also guides the BEV encoder to learn richer contextual and semantic features, thus assisting the transformer to improve the fusion efficacy. We conduct a series of comparative experiments on different fusion strategies to demonstrate that our fusion strategy is superior to others.

To obtain high-quality 3D bounding boxes, the detection head plays a crucial role in a single-stage model. Anchor-based head is still the mainstream of 3D object detection [13, 42], due to its high accuracy. However, its performance highly depends on the manually designed parameters, which limits its generalization. Meanwhile, anchor-free head is becoming more and more popular due to its simplicity and comparable performance [46, 3, 8]. In this work, we design a novel anchor-free detection head, which is composed of decoupled classification and localization branches. Such design has been proven to be effective by FCOS [33] on 2D object detection and recently by [36] on monocular 3D object detection. Instead of commonly used classification score, we propose a class-agnostic objectness score in association with IoU (intersection over union) prediction to represent the confidence of the bounding box. Another rarely noticed issue is the assignment of ground truth and predictions, which can greatly influence the convergence and performance, especially for multi-class 3D object detection. To address this issue, we propose an adaptive center radius strategy in conjunction with Simplified Optimal Transport Assignment (SimOTA) [9]. In order to demonstrate the superiority of our detection head, we conduct an experiment by replacing the detection heads of popular detectors with ours. The results show that our detection head can significantly boost the detection performance.

To summarize, our main contributions are:

- An attentive cross-view fusion module based on transformer and supervised foreground mask, which attentively fuses the complementary information from range view and BEV.
- A geometric-attention kernel, which strengthens the feature learning in range view by aggregating neighboring features with spatially attentive weights.
- An anchor-free detection head, which integrates advanced label assignment strategy and IoU-aware classification loss to predict 3D bounding boxes with better accuracy and higher quality.

2. Related Work

2.1. Data Representations

For LiDAR-based 3D object detection, there are three major data representations.

Point-based. Point-based methods leverage raw point cloud directly, which extract point-based features by popular feature learning networks, *e.g.*, PointNet++[26], DGCNN[38], *etc*. F-PointNet [24] uses PointNet [25] to encode point features in frustums of 2D ROIs, and then regresses 3D bounding boxes. PointRCNN [29] takes a two-stage pipeline, which directly generates 3D proposals from raw point cloud and refines boxes and confidence of proposals by local spatial features. 3DSSD [44] adopts a single-stage pipeline which introduces a fusion sampling strategy to remove the time-consuming feature propagation layers and directly regresses 3D bounding boxes. Other works combine point and voxel features to enhance the 3D structure information thus improve the detection accuracy, *e.g.*, PV-RCNN [28], SA-SSD [12] and Voxel-RCNN [6].

Voxel-based. Voxel-based methods divide raw point cloud into 3D voxel grids and apply 3D convolutions to extract features from voxel representation. VoxelNet [49] uses PointNet [25] to learn point features inside each voxel and then applies 3D convolutions to extract voxel features. SECOND [42] accelerates VoxelNet by sparse 3D convolutions. PointPillars [13] extracts features in pillars and treats them as a BEV pseudo-image, which allows the exploitation of 2D convolutions. PIXOR[43] directly encodes handcrafted BEV features, followed by 2D convolutions.

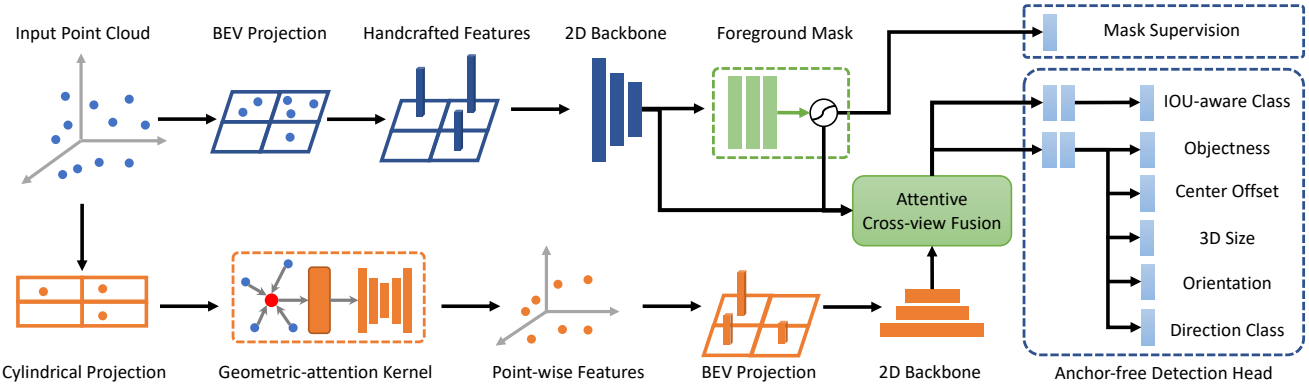

Figure 2: Overview of the proposed network. The input point cloud is first projected to bird's eye view (BEV) and cylindrical view (CYV) [37] respectively. Then, BEV and CYV features are extracted by two separate streams. Third, the CYV features are projected to BEV and fused with BEV features by an attentive cross-view fusion module based on transformer and supervised foreground mask learned from BEV features. Finally, an anchor-free detection head takes the fused features as input and predicts 3D bounding boxes and classes.

Range-view-based. Range-view-based methods transform raw point cloud to 2D range image in perspective view by spherical or cylindrical projection [37]. LaserNet [18] performs 2D convolutions on raw range images and then regresses 3D boxes directly. RCD [1] introduces a range-conditioned dilation layer to augment 2D convolutions. RangeRCNN [15] and RangeIoUDet [16] apply an encoder-decoder full convolution network to learn point-wise features from range image, and then project the point features to BEV, followed by a region proposal network to predict 3D bounding boxes. To overcome the depth discontinuity of range image, a meta-kernel convolution is proposed in [7] to aggregate 3D geometric information from point cloud. Similarly, Edge-Conv kernel [39] is adopted in [2]. We propose a geometric-attention kernel to strengthen features extracted from range image and speed up the computation by masking out empty pixels.

2.2. Multi-view Fusion

To fully encode the scene, features from different modalities are often fused together. Multi-view features could be from different sensors, *e.g.*, LiDAR, camera, *etc.*, or from different views of the same sensor, *e.g.*, point view, voxel view and range view of point cloud. MV3D [4] fuses features from frontal view, BEV, and camera view to improve 3D object detection. CLOCs [23] combines object proposals from camera- and LiDAR-based object detectors. With only LiDAR sensor, PV-RCNN [28] fuses point and voxel information to generate high-quality 3D proposals. Further, PVGNet [19] extracts point, voxel and BEV features separately, and then fuses them to point-wise features. MVF [48] and Pillar-OD [37] first apply voxelization on BEV and range view to extract point features, and then perform cross-view fusion by point-wise concatenation. We follow MVF [48] and Pillar-OD [37] to fuse BEV and range view features. Rather than fusing point-wise features at early stage or proposals at late stage as previous works, we perform feature fusion at the deepest level of BEV and range view feature maps. Specifically, instead of simple concatenation, we apply cross-view transformer [34] for better leveraging complementary information. Furthermore, we incorporate a supervised foreground mask learned from BEV features to enhance the fused features in residual attention way [35].

2.3. Anchor-free Detection Head

Anchor-based head can provide high quality proposals with prior knowledge, which is widely used in 3D object detectors [13, 42, 16]. Recently, anchor-free head is becoming more and more popular due to its comparable performance and less domain-dependent design parameters. HotspotNet [3] defines hot spots as the non-empty voxels that only lie on the surface of the object. Center-based head [46, 8] predicts objects on 2D Gaussian heatmap without prior anchors. Our anchor-free detection head adopts several delicate designs from both 2D and 3D object detection fields. We follow the decoupled design of classification and regression in FCOS [33], and incorporate IoU-aware class prediction proposed in [47]. To better assign ground truth to predictions during training, we propose an adaptive center radius setting strategy and employ Simplified Optimal Transport Assignment (SimOTA) [9], which boosts multi-class detection performance.

3. Method

As shown in Fig. 2, the entire network is mainly composed of four modules: a cylindrical-view features extractor (in orange), a bird's eye view features extractor (in dark blue), an attentive cross-view fusion module (in green), and an anchor-free detection head (in light blue). The input point cloud is first projected to the bird's eye view (BEV) and the cylindrical view (CYV) respectively. The

BEV features are extracted by a 2D convolution backbone with handcrafted features as input. The points on the CYV are fed into a geometric-attention kernel to obtain the features that convey 3D geometric information, followed by an encoder-decoder 2D full convolution network to extract point-wise features. The point-wise features are then projected to BEV and further encoded by a 2D convolution backbone. Finally, CYV features and BEV features are fused by an attentive cross-view fusion module and the fusion results are sent to an anchor-free detection head for 3D object classification and localization.

3.1. Cylindrical-view Projection

To obtain 2D range image, we can apply either spherical or cylindrical projection on 3D point cloud. Given a Cartesian point $p_i = (x_i, y_i, z_i)$, its spherical coordinate $(\varphi_i, \theta_i, d_i)$ is defined as: $\varphi_i = \arctan(\frac{y_i}{x_i}), \theta_i = \arccos(\frac{z_i}{d_i}), d_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$, and the cylindrical coordinate (φ_i, z_i, ρ_i) is given by $\varphi_i = \arctan(\frac{y_i}{x_i}), z_i = z_i, \rho_i = \sqrt{x_i^2 + y_i^2}$. As studied in [37], the spherical projection causes distortion in the Z-axis and objects in spherical view are no longer in their physical scales, *e.g.*, the distant objects become small. The cylindrical projection mitigates this scale issue. We follow [37] to use cylindrical projection in our method. In the following, the range view refers to cylindrical view, unless stated otherwise.

3.2. Geometric-attention Kernel

Directly applying standard 2D convolutions on range image always leads to inferior performance on 3D object detection. The main reason is that standard 2D convolution ignores the 3D geometric information when aggregating neighborhood information on range image. To address this issue, a special convolution named meta-kernel is proposed in [7] to use 3D geometric information to strengthen 2D range view representation. In this work, we propose an attentive geometric-aware kernel, named *geometric-attention kernel*, which aggregates neighboring features by considering the spatial attention weights among neighbors, and further strengthens the output features by concatenating geometric features. Furthermore, we incorporate a binary mask to skip the computation of empty pixels on range image, which significantly speeds up the kernel operation.

The proposed geometric-attention kernel is illustrated in Fig. 3. It takes inputs in the form of a feature map $\mathbf{F} \in \mathbb{R}^{H \times W \times D_{in}}$, a per-pixel Cartesian coordinates map of $\mathbf{P} \in \mathbb{R}^{H \times W \times 3}$, and a binary mask $\mathbf{M} \in \mathbb{R}^{H \times W}$ which indicates the occupancy of each pixel by projected points. The output of the kernel is a new feature map $\mathbf{F'} \in \mathbb{R}^{H \times W \times D_{out}}$. For each location $\mathbf{p}_i = \{x_i, y_i, z_i\}$ in the Cartesian coordinates map of $\mathbf{P}$, whether occupied or not, a 3×3 sampling operation is performed to sample its nine neighbors $\mathcal{N}$ using $im2col$ operation. The

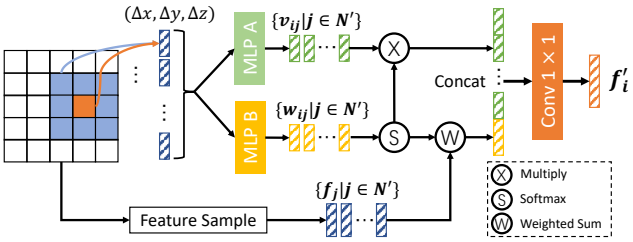

Figure 3: Illustration of geometric-attention kernel convolution.

sampled locations are further filtered by the binary mask to generate valid neighbors $\mathcal{N}'$. The relative coordinates $\{\Delta\mathbf{p}_{ij} = \{x_i - x_j, y_i - y_j, z_i - z_j\} | j \in \mathcal{N}'\}$ between $\mathbf{p}_i$ and its neighbors are calculated. Meanwhile, the sampling process is also applied to the feature map $\mathbf{F}$ to get the corresponding feature vectors $\{\mathbf{f}_j | j \in \mathcal{N}'\}$. A shared MLP (Multi-Layer Perceptron) takes the relative coordinates $\{\Delta\mathbf{p}_{ij}\}$ as input to generate 32-dimension feature vectors $\{\mathbf{v}_{ij} | j \in \mathcal{N}'\}$ to represent the position embeddings of neighborhood. Another shared MLP also takes the relative coordinates as input, but generates 1-dimension weights $\{\mathbf{w}_{ij} | j \in \mathcal{N}'\}$ normalized by *Softmax* operation, which represent the spatial attention weights of the neighbors. The aggregated feature and weighted position embeddings are calculated by *Weighted Sum* and *Multiply* operations respectively, denoted as $\sum_{j \in \mathcal{N}'} \mathbf{w}_{ij} \cdot \mathbf{f}_j$ and $\{\mathbf{w}_{ij} . \mathbf{v}_{ij} | j \in \mathcal{N}'\}$. Finally, the aggregated feature and weighted position embeddings are concatenated together, and passed through a fully connected layer (*i.e.*, 1×1 convolution) to obtain the output feature f'_i. The main difference to meta-kernel [7] is that we learn attentive weights for neighboring features before aggregation and concatenate geometric features to enhance the output features, while meta-kernel applies elemental-wise multiplication to the weights and neighboring features and then concatenate them. Furthermore, we use a binary mask to speed up the computation. As it is shown in experimental results (Table 1), our geometric-attention kernel is proven to be superior on both accuracy and efficiency.

3.3. Attentive Cross-view Fusion

In this section, we first briefly describe the CYV and BEV features extraction, and then we elaborate the details of the proposed attentive cross-view fusion module.

CYV Features Extraction. As shown in Fig. 2, we feed the cylindrically projected points into the geometric-attention kernel to generate features containing geometric information. After that, we apply an Unet-like [27] encoder-decoder 2D FCN with dilated convolutions as in [16] to extract multi-scale features in range view. The point-wise features can be easily recovered by using the projection indices in range image. The point-wise features are then projected to BEV, followed by a 2D convolution backbone to extract multi-scale features in BEV. To mark the origin of this

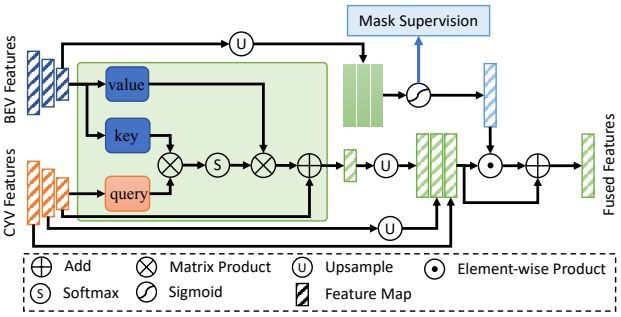

Figure 4: Illustration of attentive cross-view fusion module.

multi-scale features, we denote it as $\{\mathbf{F}^{(l)}_{cyv}|_{l=0,1,2}\}$, where l represents the index of the feature layer.

BEV Features Extraction. As shown in Fig. 2, for BEV features extraction, we use simple handcrafted features as input instead of learned point features [37, 48] or pillar features [13]. It is mainly considered from two aspects. On one hand, since we take CYV features as our main features, rich raw information contained in handcrafted features of BEV can best complement the loss of CYV information. On the other hand, applying handcrafted features is more efficient in computation and memory, which is more conducive to deployment on autonomous vehicles. Specifically, we include maximum height, mean intensity and density of grids as handcrafted features. After forming handcrafted features, we apply a group of 2D convolution layers to extract multi-scale BEV features, denoted as $\{\mathbf{F}^{(l)}_{bev}|_{l=0,1,2}\}$.

Cross-view Fusion. Until now, we have two kinds of multi-scale features in BEV, $\mathbf{F}^{l}_{cyv}$ and $\mathbf{F}^{l}_{bev}$, which are extracted independently from range view and BEV with different neighborhood gathering strategies. Previous works [48, 37] fuse point-wise features from multi-view by simple concatenation, project them to BEV and then apply 2D backbone to achieve the fusion of multi-view features. There are two drawbacks to this fusion strategy. The first is that early-stage fusion of features from shallow layers may impair the learning of rich semantic information in a specific view. The second is that simple concatenation cannot selectively fuse complementary information, and even introduces extra noise, which may reduce the performance. Therefore, we fuse multi-view features at deep layers by an attentive cross-view fusion module. Specifically, the fusion module is based on transformer to attentively fuse the deepest feature maps from range view and BEV. Furthermore, we incorporate with a supervised foreground mask learned from the deepest BEV feature map to further enhance the fused features. The results (Table 2) show that our fusion method outperforms simple concatenation by a large margin.

As shown in Fig. 4, only the deepest features $\mathbf{F}^{(2)}_{cyv} \in \mathbb{R}^{H_2 \times W_2 \times C1}$ and $\mathbf{F}^{(2)}_{bev} \in \mathbb{R}^{H_2 \times W_2 \times C2}$ of CYV and BEV features are fused by a transformer network. CYV features $\mathbf{F}^{(2)}_{cyv}$ and BEV features $\mathbf{F}^{(2)}_{bev}$ are first fed into 1×1 convolution to form query matrix $\mathbf{Q}$, key matrix $\mathbf{K}$ and value matrix $\mathbf{V}$ with the same channel dimension. We reshape the query, key, and value matrices into feature vectors, denoted as $\mathbf{Q}_c \in \mathbb{R}^{N \times C}$, $\mathbf{K}_b \in \mathbb{R}^{N \times C}$, and $\mathbf{V}_b \in \mathbb{R}^{N \times C}$ respectively, where $N = H_2 \times W_2$ is the number of pixels. The similarity between the query and key feature vectors is calculated by matrix product operation, and normalized into attention weights $\mathbf{W} \in \mathbb{R}^{N \times N}$ via a softmax operation. Then, the weights $\mathbf{W}$ and the value $\mathbf{V}_b$ are used to form the interaction term by matrix product operation. The fusion features $\mathbf{F}^{(2)}_{fusion}$ is calculated as:

$$\mathbf{F}^{(2)}_{fusion} = softmax(\mathbf{Q}_c \otimes (\mathbf{K}_b)^T) \otimes \mathbf{V}_b \oplus \mathbf{Q}_c. \quad (1)$$

After reshaping the fusion features $\mathbf{F}^{(2)}_{fusion}$ back to its original matrix dimension, we upsample and concatenate it with shallow CYV features to produce multi-scale features:

$$\mathbf{F}_{fusion} = concat(\mathbf{F}^{(0)}_{cyv},\ U_{\times 2}(\mathbf{F}^{(1)}_{cyv}),\ U_{\times 4}(\mathbf{F}^{(2)}_{fusion})). \quad (2)$$

The deepest BEV feature map is fed into a mask attention module. The mask attention module Φ_m is composed of a 3×3 convolution layer followed by the BatchNorm and ReLU for feature embedding, and a 1×1 convolution layer and sigmoid operation for generating a 1-dimension mask. During the training, the focal loss [17] with default settings is employed to handle the unbalanced number of the foreground and background pixels. The enhanced feature map by foreground mask can be calculated following the residual-attention style in [35]:

$$\mathbf{F}_{enhanced} = (1 + \Phi_m(\mathbf{U}_{\times 4}(F^{(2)}_{bev}))) \odot \mathbf{F}_{fusion}. \quad (3)$$

3.4. Foreground Mask Supervision

3D object detection from point cloud is known to be sensitive to noises and sparsity, especially for the far away and small objects which contain very few points. Recent work [14] directly introduces a supervised mask-guided attention mechanism to highlight the object pixels from complex background. The mask is generated by projecting the bounding boxes of objects into the BEV feature map and assigning pixels inside projected box as 1, otherwise 0. We argue that the attention mask should focus on enhancing non-empty pixels to reduce noisy clues, rather than inferring the underlying structure of objects from empty pixels. Therefore, we only consider the non-empty pixels inside the projected boxes as foreground and skip empty ones. We adopt a soft assignment instead of 0-1 hard assignment in [14]. A Gaussian distribution centered at each foreground pixel is applied to enlarge the foreground features. The soft assignment can provide dense supervision for nearby pixels with decreasing weights. Specifically, we set the Gaussian radius to $\sigma = \max(0.5 * \min(w, l), \tau)$, where w and l are the pixel width and length of the projected box respectively, and $\tau = 0.5$ is the minimal radius.

3.5. Anchor-free Detection Head

Recent works [46, 8] show that anchor-free detectors can achieve comparable or even better performance to anchor-based detectors. Anchor-free detection head requires much fewer design parameters and has better generalization ability due to no assumption on domain-specific anchors. As illustrated in Fig. 2, our anchor-free detection head adopts a decoupled design with two branches for classification and localization respectively, which is widely used in both 2D object [33, 9] and 3D object detection [13] as it can better solve the conflict between classification and localization tasks. The classification branch predicts the IoU between the predicted and ground truth bounding box, and the regression branch contains five sub-branches, *i.e.*, objectness, 3D center offset, 3D size, orientation and direction class.

For training, the similarity between predicted and ground truth boxes needs to be measured. Sampling multiple locations as *positive samples* within ground truth boxes can alleviate the extreme imbalance of positive / negative samples. FCOS [33] applies a simple strategy called "center sampling" that only samples the central portion of ground truth boxes as positive samples, *e.g.* 3×3 area. YOLOX [9] adopts a similar strategy by setting a center radius hyperparameter. We argue that a fixed center radius may be sub-optimal because objects have large variance in size. For example, a small center radius suitable for pedestrian may impair predictions for cars, and thus decreases the detection accuracy of car category. Therefore, we propose an adaptive strategy which adjusts the center radius according to the object size. Specifically, the center radius of a ground truth box is set to half the short side of the rectangle which is the projection of the ground truth box to BEV.

After selecting candidate positive samples through the adaptive center radius, an important issue is about how to optimally assign ground truth to each sample, *a.k.a*, label assignment problem. The advanced label assignment technique has made great progress in 2D object detection community. We adopt the recently proposed Simplified Optimal Transport Assignment (SimOTA) [9], for its superior performance and simplicity. SimOTA uses the weighted sum of cross entropy loss and IoU loss to calculate the cost matrix between candidate samples and ground truth. For each ground truth, the top k candidates with minimal cost are selected and assigned as final positive samples. We set $k = 10$ in this work. In Sect. 4.1, we conduct detailed ablation experiments to our head design, and show its superior performance over existing anchor-based and anchor-free head.

3.6. Loss Functions

We utilize the multi-task loss function for jointly optimizing the classification, box localization, objectness and direction classification tasks. For classification, we use the IoU-aware classification loss to couple the class prediction and bounding box quality prediction, which is defined as:

$$L_{cls} = -q \log(p) + (1 - q) \log(1 - p), \tag{4}$$

where p is the predicted score, and q is the IoU between the predicted and ground truth bounding box.

The box localization loss is applied by a smooth L_1 loss:

$$L_{loc} = \sum_{b \in (\Delta x, \Delta y, \Delta z, w, l, h, \theta)} SmoothL_1(\Delta b). \tag{5}$$

For objectness prediction, we apply a binary cross entropy loss, denoted as L_{obj}. To improve orientation prediction accuracy, we follow [42] and add a 2-bin direction classification task which is used to judge whether the heading angle falls in $[0, \pi)$ or $[\pi, 2\pi)$. Finally, the total loss is written as:

$$L = \frac{1}{N_{pos}} (\lambda_{cls} L_{cls} + \lambda_{loc} L_{loc} + \lambda_{obj} L_{obj} + \lambda_{dir} L_{dir}), \tag{6}$$

where N_{pos} is the number of positive samples and $\lambda_{cls} = \lambda_{loc} = \lambda_{obj} = \lambda_{dir} = 1.0$.

4. Experimental Results

We evaluate our ACDet on both the KITTI dataset [10] and Waymo Open Dataset (WOD) [32]. The KITTI dataset contains 7481 training samples and 7518 testing samples. We follow general settings and split the 7481 training samples into 3712 training samples and 3769 validation samples. We report average precision on 3 classes with 40 recall positions (R40), which is based on an IoU threshold of 0.7 for cars and 0.5 for pedestrians and cyclists. For Waymo Open Dataset, we use a quarter of the dataset for experiments. We first conduct extensive experiments to ablate the key modules of the proposed model on the KITTI dataset. Next, we compare our method with state-of-the-art methods on both datasets. Code is publicly available at https://github.com/Jiaolong/acdet.

4.1. Ablation Study

We first analyze individual modules of our proposed method by comparing different design choices. Then, we verify the contribution of each module in the network. All models are trained on the *train* split and evaluated on the *val* split for 3 classes of the KITTI dataset.

Effect of Geometric-attention Kernel. We validate the effectiveness of geometric-attention kernel by comparing with standard 2D convolution, Meta-kernel [7] and EdgeConv kernel [2]. To eliminate the interference of other factors, we conduct this experiment on range image only without cross-view fusion and foreground mask enhancement. The 3D average precisions on 3 classes of these methods are shown in Table 1. The standard 2D convolution performs worst due to the lack of 3D geometric information.

Our geometric-attention kernel outperforms the other two specific kernels at all levels of difficulty. In the last column, we compare the inference time of these kernels on 2560×64 size of range image.

Method	3D mAP Easy	Moderate	Hard	Time (ms)
2D Convolution	78.46	66.92	62.48	**1.11**
Meta-kernel [7]	80.37	68.26	64.71	13.44
Edge-Conv. [38]	79.31	66.72	63.13	19.49
Ours	**82.36**	**68.66**	**65.51**	6.47

Table 1: Performance of different kernels on KITTI *val* split.

Effect of Cross-View Fusion. As our attentive cross-view fusion module is composed of feature fusion and foreground mask supervision, we verify several feature fusion strategies with and without foreground mask enhancement. With or without mask, we all carry out experiments with range image only (*CYV only*) as the baseline reference. Under the setting without mask, we conduct three fusion strategies. The first is *early fusion*, which simply concatenates features in each cell at BEV from two views without carrying out 2D convolution backbone. The second is *concat. M*, which concatenates BEV features $\{\mathbf{F}^{(l)}_{bev}|_{l=0,1,2}\}$ and range view features $\{\mathbf{F}^{(l)}_{cyv}|_{l=0,1,2}\}$ at multiple scales. The third is *transformer*, which performs cross-view transformer at the deepest feature maps of BEV and range view. As shown in Table 2, *transformer* achieves the best performance, while *early fusion* and *concat. M* are worse than *CYV only*.

Under the setting with mask, we also conduct three fusion strategies. The first is *concat. D*, which concatenates the deepest feature maps from BEV and range view. The second is *add. D*, which adds up the deepest feature maps from BEV and range view. The third is *transformer*, which is the same as the setting without mask. As shown in Table 2, our *transformer* outperforms other fusion strategies and improves the baseline *CYV only* on moderate level for almost 3 percentage points. Compared to *transformer* without mask, it is clear that mask supervision can significantly boost cross-view transformer to attentively fuse complementary information from both views.

Effect of Detection Head. For the sake of clarity, the models under this ablation do not include BEV features fusion. We first verify the effectiveness of the adaptive center radius strategy by comparing with fixed center radius for all classes, and class specific center radius. For fixed center radius, we apply 0.3, 0.5 and 1.0 respectively. The class specific radius for car, pedestrian and cyclist are set as $1.6, 0.6, 0.6$ respectively. As shown in Table 3, we can see that the proposed adaptive center radius obtained best accuracy, outperforming both fixed and class specific center radius at all levels of difficulty. We further replace our detection head with the widely used anchor-based SSD head [13]

Method	3D mAP Easy	Moderate	Hard
w/o mask:			
CYV only	81.25	68.31	64.91
Early fusion	77.80	65.65	61.72
Concat. M	80.64	67.34	63.47
Transformer	81.98	68.96	65.23
w/ mask:			
CYV only	81.65	68.40	64.84
Concat. D	80.97	69.08	65.05
Add. D	80.64	68.19	64.22
Transformer (ours)	**83.51**	**71.28**	**67.31**

Table 2: 3D mAP on 3 classes with different fusion strategies.

Method	3D mAP Easy	Moderate	Hard
Radius = 0.3	80.77	67.87	63.41
Radius = 0.5	77.91	66.34	62.63
Radius = 1.0	78.95	66.83	63.39
Class specific radius	78.67	66.30	62.59
Adaptive radius (ours)	**82.36**	**68.66**	**65.51**
with SSD[13] head	76.3	65.34	62.06
with CenterPoint[46] head	71.21	60.55	57.47

Table 3: 3D mAP on 3 classes with different head settings.

	Kernel	Head	Fusion	3D mAP Easy	Moderate	Hard
A1				75.83	64.79	61.40
A2	✓			76.30	65.34	62.06
A3	✓	✓		82.36	68.66	65.51
A4	✓	✓	✓	**83.51**	**71.28**	**67.31**

Table 4: Ablation of our key modules on 3 classes detection.

and state-of-the-art anchor-free CenterPoint head [1] [46]. As shown in Table 3, the accuracy drops more than 2 points and 7 points on moderate difficulty by replacing with SSD head and CenterPoint head respectively.

In order to investigate our detection head more objectively, we replace the detection heads of PointPillars [13] and SECOND [42] with ours to verify its effectiveness. Encouragingly, our detection head significantly improves the accuracy at all levels of difficulty, as shown in Fig. 5. Especially for SECOND detector, the moderate 3D mAP is improved about 5 percentage points, *i.e.*, 63.79 vs 69.07.

Contribution of Each Module. To investigate the contribution of each module, we conduct ablation experiments on the entire pipeline. The results are shown in Table 4, where *A1* is a range image model with SSD detection head. *A2* includes geometric-attention kernel, which improves *A1* about 1 percentage point on 3D average precision. *A3* replaces SSD head with the proposed anchor-free head, which boosts 3D accuracy at moderate level by 3.3 percentage

[1] https://github.com/tianweiy/CenterPoint-KITTI

Method	View	Car 3D/BEV AP			Pedestrian 3D/BEV AP			Cyclist 3D/BEV AP			Overall mAP
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Moderate
VoxelNet [49]	BEV	77.47/89.35	65.11/79.26	57.73/77.39	39.48/46.13	33.69/40.74	31.51/38.11	61.22/66.70	48.36/54.76	44.37/50.55	49.05/58.25
SECOND [42]	BEV	83.13/88.07	73.66/79.37	66.20/77.95	51.07/55.10	42.56/46.27	37.29/44.76	70.51/73.67	53.85/56.04	46.90/48.78	56.69/60.56
PointPillars [13]	BEV	82.58/90.07	74.31/86.56	68.99/82.81	51.45/57.60	41.92/48.64	38.89/45.78	77.10/79.90	58.65/62.73	51.92/55.58	58.29/65.68
Point-RCNN [29]	PV	86.96/92.13	75.64/87.39	70.70/82.72	47.98/54.77	39.37/46.13	36.01/42.84	74.96/82.56	58.82/67.24	52.53/60.28	57.94/66.92
Point-GNN [31]	PV	88.33/93.11	79.47/89.17	72.29/83.90	51.92/55.36	43.77/47.07	40.14/44.61	78.60/81.17	63.48/67.28	57.08/59.67	62.24/67.84
STD [45]	PV	87.95/94.74	79.71/89.19	75.09/**86.42**	53.29/60.02	42.47/48.72	38.35/44.55	78.69/81.36	61.59/67.23	55.30/59.35	61.26/68.38
3D-SSD [44]	PV	88.36/92.66	79.57/89.02	74.55/85.86	**54.64/60.54**	**44.27**/49.94	40.23/45.73	82.48/85.04	64.10/67.62	56.90/61.14	62.65/68.86
VoxSet [11]	PV	88.53/92.70	**82.06**/89.07	**77.46**/86.29	-	-	-	-	-	-	-
Part-A^2 [30]	BEV + PV	87.81/91.70	78.49/87.79	73.51/84.61	53.10/59.04	43.35/49.81	40.06/45.92	79.17/83.43	63.52/68.73	56.93/61.85	61.79/68.78
PV-RCNN [28]	BEV + PV	**90.25/94.98**	81.43/**90.65**	76.82/86.14	52.17/59.86	43.29/**50.57**	40.29/**46.74**	78.60/82.49	63.71/68.89	57.65/62.41	62.81/70.04
F-PointNet [24]	PV + IM	82.19/91.17	69.79/84.67	60.59/74.77	50.53/57.13	42.15/49.57	38.08/45.48	72.27/77.26	56.12/61.37	49.01/53.78	56.02/65.20
F-ConvNet [40]	PV + IM	87.36/91.51	76.39/85.84	66.69/76.11	52.16/57.04	43.38/48.96	38.80/44.33	81.98/84.16	65.07/68.88	56.54/60.05	61.61/67.89
LaserNet [18]	SPV	-/79.19	-/74.52	-/68.45	-	-	-	-	-	-	-
RCD [1]	SPV + PV	70.54/82.26	60.56/75.83	55.58/69.91	-	-	-	-	-	-	-
RangeIoUDet [16]	SPV + PV	88.60/92.28	79.80/88.59	76.76/85.83	-	-	-	83.12/85.99	**67.77**/71.49	**60.26**/63.62	-
ACDet (ours)	BEV + CYV	88.47/92.87	78.85/89.21	73.86/85.80	53.41/58.35	44.79/49.82	**41.96**/47.17	**83.80/87.76**	66.61/**71.48**	59.99/**64.69**	**63.42/70.17**

Table 5: Results of car, pedestrian and cyclist evaluated on the KITTI *test* set with 40 recall positions (R40). The best and second best results are marked in bold and underline, respectively. PV: Point View. SPV: Spherical Projection View. IM: Image View.

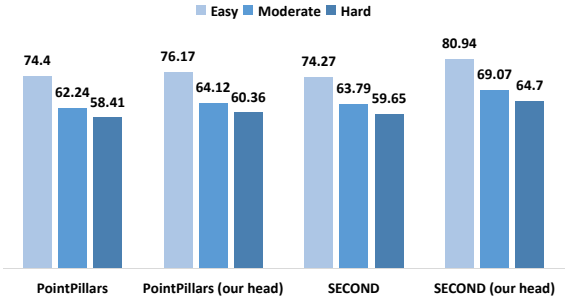

Figure 5: Results of applying our head in popular detectors.

points. *A4* includes cross-attention fusion module, which further improves *A3* by 2.6 percentage points on 3D average precision at moderate level.

4.2. Comparison with State-of-the-Art

Table 5 shows the performance of the proposed method and state-of-the-art methods on the KITTI test set. We divide comparison methods into several groups according to their data representations. On overall performance, our method achieves best and second best on 3D and BEV mean average precision (mAP) respectively. For car category, the 3D average precision (AP) of our model is close to PV based methods but is slightly outperformed by BEV + PV based methods, which commonly adopt a second stage refinement to obtain extra accuracy gain. Our method achieves best BEV accuracy at moderate level among the range-view based methods. For pedestrian and cyclist detection, our method achieves top accuracy on both 3D and BEV AP, especially at moderate and hard levels. For example, our model obtains best 3D AP for pedestrian detection at hard level, and best BEV AP for cyclist detection at all difficult levels. As pedestrian and cyclist are very tiny objects in BEV, this verifies the superiority of the proposed multi-view fusion strategy.

We also conduct experiments on Waymo Open Dataset (WOD) in Table 6 to compare with range-view and multi-view fusion based methods. Our method outperforms all range-view based methods (first and second block) by a large margin. It is worth noting that PPC [2] and RangeDet[7] are Edge-Conv and meta-kernel based methods respectively, which further verifies the effectiveness of proposed geometric-attention kernel. Compared to multi-view fusion methods, we outperform the *early* fusion based method MVF by a large margin, and attains comparable or better accuracy than PillarOD under all distance thresholds.

Method	All	0-30m	30-50m	50-75m
LaserNet [18]	52.11	70.94	52.91	29.62
RCD [1]	68.95	87.22	66.53	44.53
PPC + EdgeConv [2]	65.20	-	-	-
RangeDet†[7]	67.37	85.91	62.61	42.77
MVF [48]	62.93	86.30	60.20	36.02
PillarOD [37]	69.80	**88.53**	66.50	42.93
ACDet† (ours)	**70.25**	88.43	**68.04**	**46.18**

Table 6: LEVEL_1 AP of vehicle detection on WOD *val* split. †: trained with uniformly sampled 25% training data.

5. Conclusion

We have presented ACDet - a novel single-stage multi-view fusion framework which consists of a geometric-attention kernel, a transformer based cross-view fusion module and an anchor-free detection head. ACDet addresses the challenge of detecting small objects with sparse LiDAR points, and improves the overall accuracy of multi-class 3D object detection. The experimental results on KITTI and Waymo Open Dataset demonstrate the effectiveness and generalization of the proposed method. Importantly, the novel anchor-free detection head can be applied to different 3D detectors to boost the detection accuracy. The inference speed is about 20 FPS on a single RTX 3090 GPU, which can be improved in future work.

References

[1] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range Conditioned Dilated Convolutions for Scale Invariant 3D Object Detection. In *Conference on Robot Learning*, 2020. 4321, 4323, 4328

[2] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiyue Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Dragomir Anguelov. To the Point: Efficient 3D Object Detection in the Range Image With Graph Convolution Kernels. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021. 4321, 4323, 4326, 4328

[3] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan L. Yuille. Object as Hotspots: An Anchor-Free 3D Object Detection Approach via Firing of Hotspots. In *European Conf. on Computer Vision*, pages 68–84, 2020. 4322, 4323

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-View 3D Object Detection Network for Autonomous Driving. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 4323

[5] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds. In *International Symposium on Visual Computing*, pages 207–222, 2020. 4321

[6] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In *AAAI*, pages 1201–1209, 2020. 4322

[7] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. RangeDet:In Defense of Range View for LiDAR-based 3D Object Detection. In *Int. Conf. on Computer Vision*, 2021. 4321, 4323, 4324, 4326, 4327, 4328

[8] Runzhou Ge, Zhuangzhuang Ding, Yihan Hu, Yu Wang, Sijia Chen, Li Huang, and Yuan Li. AFDet: Anchor Free One Stage 3D Object Detection. *arXiv:2006.12671*, 2020. 4322, 4323, 4326

[9] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO Series in 2021. *arXiv:2107.08430*, 2021. 4322, 4323, 4326

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2012. 4326

[11] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2022. 4328

[12] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure Aware Single-Stage 3D Object Detection From Point Cloud. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 4322

[13] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019. 4321, 4322, 4323, 4325, 4326, 4327, 4328

[14] Jiale Li, Hang Dai, Ling Shao, and Yong Ding. Anchor-free 3D Single Stage Detector with Mask-Guided Attention for Point Cloud. In *ACM International Conference on Multimedia*, 2021. 4325

[15] Zhidong Liang, Ming Zhang, Zehan Zhang, Xian Zhao, and Shiliang Pu. RangeRCNN: Towards Fast and Accurate 3D Object Detection with Range Image Representation. *arXiv:2009.00206*, 2020. 4321, 4322, 4323

[16] Zhidong Liang, Zehan Zhang, Ming Zhang, Xian Zhao, and Shiliang Pu. RangeIoUDet: Range Image Based Real-Time 3D Object Detector Optimized by Intersection Over Union. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 7140–7149, 2021. 4321, 4322, 4323, 4324, 4328

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *Int. Conf. on Computer Vision*, pages 2980–2988, 2017. 4325

[18] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 12677–12686, 2019. 4321, 4323, 4328

[19] Zhenwei Miao, Jikai Chen, Hongyu Pan, Ruiwen Zhang, Kaixuan Liu, Peihan Hao, Jun Zhu, Yang Wang, and Xin Zhan. PVGNet: A Bottom-Up One-Stage 3D Object Detector with Integrated Multi-Level Features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021. 4323

[20] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. RangeNet++: Fast and Accurate LiDAR Semantic Segmentation. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 4213–4220, 2019. 4321

[21] Arsha Nagrani, Shan Yang, Anurag Arnab, Cordelia Schmid, and Chen Sun. Attention Bottlenecks for Multimodal Fusion. In *arXiv:2107.00135*, 2021. 4322

[22] Jongyoun Noh, Sanghoon Lee, and Bumsub Ham. HVPR: Hybrid Voxel-Point Representation for Single-Stage 3D Object Detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 14605–14614, June 2021. 4321

[23] Su Pang, Daniel Morris, and Hayder Radha. CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 10386–10393, 2020. 4323

[24] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 918–927, 2018. 4321, 4322, 4328

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 4321, 4322

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, 2017. 4321, 4322

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 4324

[28] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 4321, 4322, 4323, 4328

[29] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 4321, 4322, 4328

[30] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020. 4328

[31] Weijing Shi and Raj Rajkumar. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2020. 4328

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 4326

[33] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully Convolutional One-Stage Object Detection. In *Int. Conf. on Computer Vision*, 2019. 4322, 4323, 4326

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 2017. 4322, 4323

[35] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual Attention Network for Image Classification. *arXiv:1704.06904*, 2017. 4323, 4325

[36] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection. *arXiv: 2104.10956*, 2021. 4322

[37] Yue Wang, Alireza Fathi, Abhijit Kundu, David Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based Object Detection for Autonomous Driving. In *European Conf. on Computer Vision*, 2020. 4323, 4324, 4325, 4328

[38] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (TOG)*, 2019. 4321, 4322, 4327

[39] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5):146, 2019. 4323

[40] Zhixin Wang and Kui Jia. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1742–1749, 2019. 4328

[41] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. SqueezeSegV3: Spatially-Adaptive Convolution for Efficient Point-Cloud Segmentation. *arXiv:2004.01803*, 2020. 4321

[42] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, 2018. 4321, 4322, 4323, 4326, 4327, 4328

[43] Bin Yang, Wenjie Luo, and Raquel Urtasun. PIXOR: Real-time 3D Object Detection from Point Clouds. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 4322

[44] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3DSSD: Point-based 3D Single Stage Object Detector. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 4321, 4322, 4328

[45] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In *Int. Conf. on Computer Vision*, pages 1951–1960, 2019. 4328

[46] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-Based 3D Object Detection and Tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021. 4322, 4323, 4326, 4327

[47] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. In *AAAI*, pages 3555–3562, 2020. 4323

[48] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds. In *CoRL*, pages 923–932, 2020. 4323, 4325, 4328

[49] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 4321, 4322, 4328