Enhancing Multi-modal Features Using Local Self-attention for 3D Object Detection

Hao Li[1], Zehan Zhang[1,2], Xian Zhao[1], Yulong Wang[1], Yuxi Shen[1], Shiliang Pu[1⋆], and Hui Mao[1]

[1] Hikvision Research Institute
[2] Shanghai Jiao Tong University
{lihao85,zhangzehan,zhaoxian,wangyulong13,
shenyuxi,pushiliang.hri,maohui}@hikvision.com

Abstract. LiDAR and Camera sensors have complementary properties: LiDAR senses accurate positioning, while camera provides rich texture and color information. Fusing these two modalities can intuitively improve the performance of 3D detection. Most multi-modal fusion methods use networks to extract features of LiDAR and camera modality respectively, then simply add or concancate them together. We argue that these two kinds of signals are completely different, so it is not proper to combine these two heterogeneous features directly. In this paper, we propose EMMF-Det to do multi-modal fusion leveraging range and camera images. EMMF-Det uses self-attention mechanism to do feature reweighting on these two modalities interactively, which can enchance the features with color, texture and localiztion information provided by LiDAR and camera signals. On the Waymo Open Dataset, EMMF-Det acheives the state-of-the-art performance. Besides this, evaluation on self-built dataset further proves the effectiveness of our method.

Keywords: 3D detection, Self-attention, Range images, Multi-modal fusion

1 Introduction

Recently, 3D object detection has attracted more and more attention due to its importance in autonomous driving. In the 3D object detection task, the network predicts 3D objects using multiple kinds of signals. Among them, LiDAR signal is sparse and provides accurate location information of objects. Different from LiDAR signal, camera signal is compact and provides rich texture and color information. It is intuitive to leverage these signals to improve 3D detection performance.

From the Fig.1, we can observe that the road conditions are complicated in real world. Even human eyes cannot identify objects from background only using LiDAR representation. If relying too much on LiDAR signal, the model will produce a number of wrong detection results. To take the Human Visual System's

[⋆] Corresponding author.

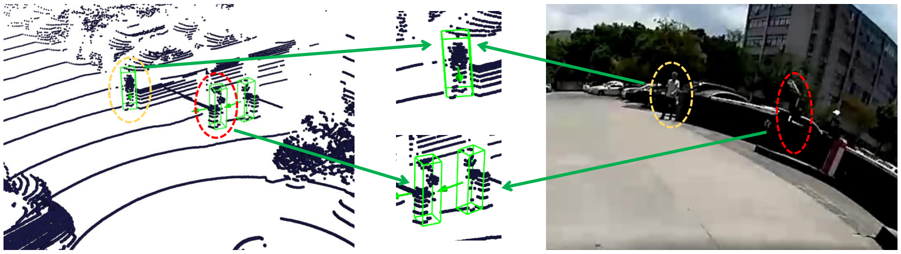

Fig. 1. Visualization of some instances in autonomous driving scenes. The yellow circle is a person while the red one is actually a camera mounted on a pole. It is difficult for even the human being to distinguish the category of objects only relying on LiDAR representation. This is mainly because that only the shape and relative position of objects is not enough to distinguish them from background. Therefore, color and texture information are very important for the 3D detection tasks.

characteristics into account, texture and color information play a important role in recognizing and locating objects. This raises an interesting question: *Can we integrate the texture and color information from camera images with LiDAR points to make LiDAR representation of objects more differentiated?* The answer is yes, however, it is a non-trival work for two reasons. In the one respect, camera image and LiDAR representation are two heterogenous signals that have different characteristics. There exists a big domain gap between them. In the other respect, camera images record the color, texture information that are sensitive to occlusion. Previous works mostly use 2D/3D CNN to extract LiDAR features and camera features and concatenate them together on the bird's-eye view (BEV) or the point-wise view. However, these methods only establish a simple correspondence between LiDAR features and camera image features and concatenate the multi-modal features naively. The two main issues mentioned above remain unsolved. The success of transformer in Natural Language Processing (NLP) attracts the attention of the vision community. Transformer establishes the topology graph between discrete tokens and aggregates the features according to the attention coefficients. Benefiting from the self-attention mechanism, transformer is a suitable alternative to fuse multi-modal features that have a large domain gap. However, directly calculating self-attention on the whole 3D scene is computationally demanding. Therefore, we design a fusion method using local self-attention to enhance the LiDAR features with camera feature. The other advantage of local self-attention is that it can alleviate the point misalignment caused by occlusion. And we call the resulting 3D detector EMMF-Det.

EMMF-Det uses range images and camera images as input. This is mainly because range image is a compact and regular LiDAR signal that can be processed by 2D convolutions like camera image. To process the range image, we apply a 2D encoder-decorder network to extract high-level point-wise feature.

Prior works also have the similar idea to process range images for 3D detection [24, 10] and segmentation [29] tasks. For the camera images, we adopt the pretrained UniverseNet-50 [39] followed by a feature pyramid network(FPN) [25] to generate a group of camera image features that consists of high-level features at different scales. After obtaining the LiDAR features and camera features, we use the self-attention mechanism of transformers to integrate the LiDAR features and camera feature. To be specific, we first divide the whole 3D scene into local regions and calculate the local attention matrices for LiDAR and camera modalities. Then we swap the attention matrices and use camera(LiDAR) attention matrix to do feature-reweighting on the LiDAR(camera) features. In this way, color, texture information from local region are utilized to enhance the LiDAR feature.

Besides the multi-modal fusion module design, we find that the performances of most existing multi-modal fusion methods are limited by the lack of data augmentations used in LiDAR-only methods, especially copy paste. In the proposed EMMF-Det, we introduce two practical data augmentation strategies during training to complement this shortage. One is the multi-modal copy-paste that adds some groundtruths to range images and camera images simultaneously. Besides, we find that laser has poor reflectivity on transparent materials or objects with black color. Hence we randomly corrupt the points in range images during training, to enable LiDAR features more robust.

In summary, our contributions are summarized as follow:

- We propose a multi-modal feature fusion framework EMMF-Det, which uses a local transformer fusion module to do feature re-weighting on the multi-modal features locally.
- We explore the characteristics of LiDAR signal and introduce two effective multi-modal data augmentation strategies to improve the 3D detection performance.
- We evaluate our method on the Waymo Open Dataset and self-built dataset. EMMF-Det achieves state-of-the-art results in range-view-based methods on the Waymo Open Dataset, and the experiment results on self-built dataset further prove the effectiveness of our method.

2 Related work

2.1 Lidar-only 3D Object Detection

Lidar-only 3D object detection methods aim to predict 3D object boxes using LiDAR returns. They can be divided into voxel-based, point-based, and range-view-based methods based on different representations. Voxel-based methods [63, 51, 18, 21, 53, 48, 17, 38, 13, 62, 48, 58] use voxelization to encode unordered points into voxels, which can be processed by 3D convolutions. In these methods, VoxelNet [63] is an end-to-end 3D detection network, which uses voxels as input. Second [51] uses sparse 3D convolution to accelerate VoxelNet. PointPillar [18] uses pillars that can be seen as a variant of voxels to encode points. Point-based [37, 50, 36, 52, 13] methods take raw point cloud as input. Among these methods, PointRCNN [37] is a two-stage 3D detection framework using PointNet++ [34] to extract features from points. PV-RCNN [36] uses a RoIgrid pooling layer to improve the quality of 3D proposals generated by the voxel CNN. Recently, some methods [41, 24, 10, 23, 1] use range-view as input to extract 3D features. RangeIoUDet [24] uses 2D convolution to process range images and inherits the RPN module from PointPillars. In [1], the authors propose a range conditioned pyramid to alleviate the scale variation in range images. [41] designs a Range Sparse Net using sparse convolutions to process the foreground features. However, during training, RSN needs to train each category separately, which makes it impractical.

Voxel-based and Point-based methods use 3D convolution layers as the backbone to achieve high performance. But 3D convolution is difficult to optimize on practical chips. This is the reason why most onboard algorithms are still dominated by methods like pointpillars, which use 2D convolutional layers as the backbone. Our approach EMMF-Det achieves SOTA performance only using 2D convolutional layers, which is efficient and easy to deploy. Compared with other range-view-based methods, our method also has obvious advantages on performance.

2.2 Multi-modal fusion 3D Object Detection

Multi-modal fusion 3D object detection uses multiple modalities information to predict the 3D object bounding boxes. Existing multi-modal fusion methods can be divided into early and late fusion. As for late fusion methods, F-ConvNet [50] and F-PointNet [33] directly do 3D detection in frustum, which is projected by 2D bounding box. AVOD [16] and MV3D [6] perform multi-modal fusion for 3D proposals using the ROI-pooling layer. MVP [55] proposes to use 2D detection to generate dense 3D virtual points to augment an otherwise sparse 3D pointcloud. In summary, late-fusion methods fuse the 2D detection or segmentation results with the 3D detector, relying heavily on the 2D tasks. If the 2D models fails during the inference, it will hurt the performance seriously. Instead of late fusion, more and more researchers focus on early fusion, leveraging 2D and 3D features simultaneously. Existing early fusion methods can be divided into two categories according to the form of feature alignment. One family of early fusion methods [56, 22, 56] construct camera feature maps and LiDAR feature maps separately and concatenate the multi-modal features on the BEV). However, it is not proper to do fusion on the BEV directly because the field of view (FOV) of cameras is totally different from BEV. So the cross-view transformation will cause feature blurring during the fusion, so 3D-CVF [56] proposes the auto-calibration, which projects multi-modal features to a smooth BEV map. The other family [40, 45, 44] fuses the multi-modal features by the point-wise manner, which uses each point in the point cloud as the medium to concatenate the multi-modal features point by point. MVX-Net [40] extracts the multi-modal features separately and fuses them in point cloud coordinates system. Pointpainting concatenates the 2D segmentation scores with original LiDAR to enrich the information. EPNet [15] proposes an adaptive fusion module to combine LiDAR features and RGB features point by point. However, it is not easy to combine these two heterogenous signals together without loss. Our proposed EMMF-Det utilizes self-attention to aggregate the multi-modal features by applying feature re-weighting in the local region to improve the performance of LiDAR-only 3D detction methods.

2.3 Transformer in 3D Vision

Recently, the success of transformer [43] in NLP attracts the attention of vision community and inspires a lot of works to investigate the power of attention in visual recognition [8, 26, 42, 59]. Recently, several works have investigated using transformer in 3D perception tasks. In [60, 12], authors design self-attention networks for scene segmentation, object part segmentation and object classification. [2, 35, 27, 31] introduce the self-attention layers to learn point cloud representation, aiming to improve the performance of existing 3D detection models. In [30], 3DETR is proposed to use an end-to-end transformer framework for 3D object detection, which can be seen as a variant of DETR [3]. Transfuser [32] proposes to integrate image and LiDAR representation using transformers in strategy prediction task. In [49], DETR3D is proposed to use multi-view images to predict 3D bounding boxes. SST [9] uses a single-stride sparse transformer module to process point clouds and achieves the SOTA performance on the open dataset. To our best knowledge, our proposed method, EMMF-Det, is the first work to use transformer to fuse multi-modal features in the 3D detection task.

3 Methods

This section presents the details of our proposed method EMMF-Det, which is a multi-modal fusion 3D detector. EMMF-Det consists of four components. In section 3.1, we design an one-stage 3D detector as our baseline; In section 3.2, we describe the details of processing 2D features; In section 3.3, a novel fusion strategy based on transformer is introduced for multi-modal fusion.

3.1 LiDAR Detector Pipeline

Range image is a compact, regular LiDAR signal that can be processed by 2D convolution like camera images. Inspired by some prior range-based detection and segmentation works [23, 29, 24], we apply a 2D encoder-decorder network to extract high-level point-wise features using range images. The encoder-decorder network is a symmetric architecture, which has four down-sampling blocks and four up-sampling blocks. Benefitting from the symmetric architecture and correspondence between range image and LiDAR points, we can obtain point-wise LiDAR high-dimensional features, which are suitable for our multi-modal fusion framework. The projection from LiDAR points to range images can be calculated by eq.1.

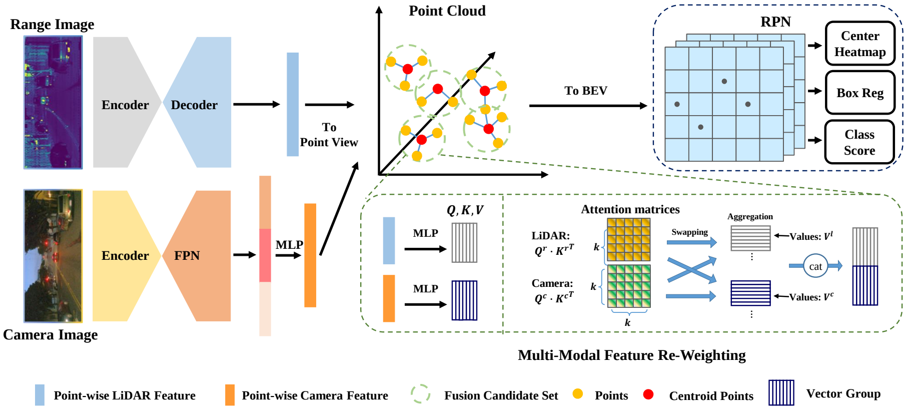

Fig. 2. The overall architecture of our proposed EMMF-Det. In the left part, we use a 2D FCN to extract point-wise LiDAR features from range images, and a 2D detection network with FPN to extract camera image features at three scales. Then we project the LiDAR features and camera features onto point cloud and perform multi-modal fusion using local self-attention. The details of multi-modal fusion module are described in section 3.3. Finally, we scatter the point-wise features to a BEV map and use an anchor-free detector to predict the 3D bounding boxes.

$$\begin{pmatrix} u_r \\ v_r \end{pmatrix} = \begin{pmatrix} \frac{1}{2}[1 - arctan(y_l, x_l)\pi^{-1}] \times w \\ [1 - (arcsin(z_l, r) + f_{down})f^{-1}] \times h \end{pmatrix} \tag{1}$$

where (x_l, y_l, z_l) represents the point coordinates in the LiDAR coordinate system. (u_r, v_r) is the pixel coordinates in the range image. $r = \sqrt{x_l^2 + y_l^2 + z_l^2}$ is the range value. w and h symbolize the width and height of the range image. $f = f_{up} + f_{down}$ is the vertical field-of-view of the LiDAR sensor, where f_{up} and f_{down} represent the distance below and above the ground plane respectively.

Given N points, we can obtain the point-wise features f_i^l with high dimensions (64 dims). The pointwise features can be projected to the x-y plane to generate the BEV features. Similar with other anchor-free 3D detection methods [46, 47, 54], we design a 2D FCN to process the BEV features to predict a keypoint heatmap. Each peak in the heatmap represents a ground truth center. As for the regression head for size, rotation and location of objects, we use the features stored at the peak to regress the location refinement $o \in \mathbb{R}^2$ in map view, height-above-ground $h_g \in \mathbb{R}$, 3D box size $s \in \mathbb{R}$ and a yaw rotation angle $\{\sin(x), \cos(x)\} \in \mathbb{R}^2$. Using these information, we can predict the full state information of 3D objects.

The above range-view-based model is a single-stage 3D detector that only uses 2D convolutions and achieves superior performance compared to other range-view-based methods. However, only using the location information encoded by LiDAR features is not enough to compete with SOTA methods. So we use the attention mechanism to fuse camera images and range images to further improve the performance.

3.2 Camera Feature Pipeline

In parallel to the 3D detection pipeline, we introduce an existing 2D detector and use the 2D features generated by FPN to incorporate with the LiDAR feature. Here we choose the pretrained UniverseNet-50 [39] followed by a feature pyramid network (FPN) [25] to generate a group of RGB features that consists of high-level features at different scales (8×, 16×, 32× downsampling from the original size) for each image.

To fuse the LiDAR features and camera features, we project the LiDAR point onto the image at first by the equation [28, 11]:

$$\alpha[u_c, v_c, 1]^T = K(Rp + t) \tag{2}$$

where (u_c, v_c) is the pixel coordinate in the camera image and p denotes the point coordinates (x_l, y_l, z_l). K is the intrinsic calibration matrix of camera. R and t are the rotation and translation vector from LiDAR coordinates to the camera coordinates. However, the high-level features we obtain have different resolution from the original image. So we need to divide projected pixels coordinates by the downsampling scale factor and round them to integer. Through this way, we can collect the three scales point-wise camera features for N points. To align with the LiDAR point features, we concatenate the three scales camera features together and use a MLP to reduce the dimension to 64, so the point-wise camera features $f_i^c \in \mathbb{R}^{1\times 64}, i \in \{1, 2, ..., N\}$ can be obtained.

3.3 Feature Re-Weighting using Self-attention

Once obtaining features $(f_i^l, f_i^c), i \in \{1, 2, ..., N\}$ for N points from two modalities, most multi-modal fusion methods will concatenate or add them together. We argue that it is not very proper to do this. LiDAR signals and camera signals are heterogeneous signals that have a large domain gap. By the characteristics of Human Visual System, color and texture information can help to distinguish the objects from background, while range information helps to locate the objects. So our key idea is to exploit the self-attention mechanism from transformers to incorporate the texture, color information with range information to improve the 3D detection performance.

Fusion Candidate Sets. Transformer takes a sequence consisting of discrete tokens as input . Unlike the tokens used in Natural Language Processing (NLP), we use each LiDAR point as one token in 3D detection. However, directly calculating self-attention on the whole scene in 3D vision tasks will takes up a lot of memory. So we propose to split the whole point cloud into small point sets, defined as fusion candidate sets, and compute the self-attention in the local region. The advantages are: first, caculating self-attention in the local region can reduce the computational complexity; second, local texture features and colors are more meaningful to distinguish the boundary of objects.

One remaining issues is that how to generate local partition of a point cloud. Given N input points $\{p_1, p_2, ..., p_N\}$ with $p_i \in \mathbb{R}^{1\times3}$, we first use farthest point sampling (FPS) [34] to choose N' points $\{p_{c_1}, p_{c_2}, ..., p_{c_{N'}}\}$ with $p_{c_i} \in \mathbb{R}^{1\times3}$ as the set of centroids. For centroid point p_{c_i}, we apply k-nearest neighbor algorithm to select k points in the neignborhood of p_{c_i} and denote these points as set $\mathbf{\Omega}_i$. Through this, we have divided the point cloud into N' partitions, which are k-nearest neighbor sets.

Feature Re-Weighting Layer. Following [43], we denote the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ as queries, keys and values separately. Given the LiDAR feature set $\{f_1^l, f_2^l, ..., f_n^l\}$, $f_i^l \in \mathbb{R}^{1\times64}$ and camera feature set $\{f_1^c, f_2^c, ..., f_n^c\}$, $f_i^c \in \mathbb{R}^{1\times64}$ for k input points. $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ for fusion candidate set $\mathbf{\Omega}_i$ can be generated by the linear transformations of the multi-modal points features . Then the attention matrices of $\mathbf{\Omega}_i$ can be caculated via matrix dot-product operations:

$$\begin{gathered}\mathbf{A}_i^{mod} = (\alpha_i^{mod})_{m,n} = \mathbf{Q}_i^{mod} \cdot \mathbf{K}_i^{mod\top} \\ m, n \in 1, ..., k, \ mod \in \{l, c\}\end{gathered} \tag{3}$$

where $\mathbf{A}_i^{mod}$ is the attention matrix of k points. l and c denote the LiDAR modality and camera modality separately.

For points set $\mathbf{\Omega}_i$, we calculate the attention matrices $(\mathbf{A}_i^l, \mathbf{A}_i^c)$ in LiDAR and camera modality separately. α_i^l in matrix $\mathbf{A}_i^l$ represents the attention coefficients that are mostly determined by location information in LiDAR signal. α_i^c in matrix $\mathbf{A}_i^c$ represents the attention coefficients that are mostly determined by texture, color information in camera signal. These two attention matrices $(\mathbf{A}_i^l, \mathbf{A}_i^c)$ help to build the topology graph for fusion candidate set $\mathbf{\Omega}_i$. After obtaining attention matrices $(\mathbf{A}_i^l, \mathbf{A}_i^c)$, we propose to do feature re-weighting on the LiDAR point features by aggregating the value vector $\mathbf{V}_i^l$ with attention matrix $\mathbf{A}_i^c$. By this way, the correlation of LiDAR points in local region can be enhanced by the texture and color information provided by camera image. Coupled with the attention matrix $\mathbf{A}_i^l$ from its own modality, we can get the final output features $\mathbf{F}_i^l$ for LiDAR modality. In turn, we can also use attention matrix $\mathbf{A}_i^l$ from LiDAR modality to enhance the camera features $\mathbf{V}_i^c$. The whole process be described as follows:

$$\mathbf{F}_i^{mod} = \mathrm{softmax}(\frac{\mathbf{A}_i^l/2 + \mathbf{A}_i^c/2}{\sqrt{d_k}}) \cdot \mathbf{V}_i^{mod} \tag{4}$$

where the d_k is the dimension of querys and keys. We adapt this equation from [43], which proposes the transformer for the first time. The attention matrices $(\mathbf{A}_i^l, \mathbf{A}_i^c)$ from two modalities represent the attention coefficients among points in the fusion candidate sets. The operation of attention swapping introduces attention matrix from one modality to do feature re-weighting on the other modality.

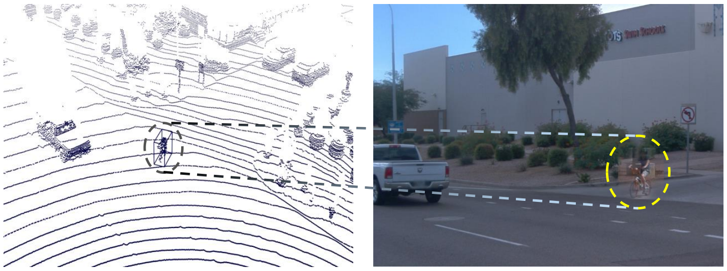

Fig. 3. Examples of multi-modalities copy paste data augmentation. The left one is the visualization of LiDAR point. The right one is the visualization of camera images. The cyclist in the yellow is pasted on by eq.5 and eq.6.

After processing with two feature re-weighting layers, we can get the output features $\{\tilde{\mathbf{F}}^l_i, \tilde{\mathbf{F}}^c_i\}, i \in \{1, ..., N'\}$ for set $\boldsymbol{\Omega}_i$. The point-wise features in set $\tilde{\mathbf{F}}^{mod}_i$ are fused in the local region. Then we pick out the point-wise features from two modalities and concatenate them point-by-point. Note that, some points may appear multiple times in different sets, so we use average pooling to de-duplicate the features of these points. Finally, we use the point-wise features after fusion as the input to the RPN head.

Our proposed feature re-weighting layer using local self-attention enhances the LiDAR features with texture and color information provided by camera images. The enhanced features improve the performance of range-view-based detector in section 3.1, and help it to achieve state-of-the-art performance.

3.4 Data Augmentation

Multi-modalities Copy Paste. In LiDAR-only methods, GT-Paste augmentation is a widely used data augmentation strategy. In EMMF-Det, we extend the copy-paste strategy into multi-modalities. First, we generate a database containing the labels of all ground truths and their associated point cloud data, image patches. Then during training, we randomly select some ground truths from this database and introduce them into the current training scene, both in LiDAR points and RGB images. To avoid physically impossible outcomes, we perform a collision test in both LiDAR points and camera images, and remove any ground truths that collide with other objects. For the camera modality, if we directly replace the corresponding area of original training image I_i with associated image patch p from I_j, generated image $\tilde{I}_i$ will look very different from authentic images in terms of co-occurrences of color or layout. We propose to use a linear combination of pixels instead of "copy-paste". The linear combination of pixels [20, 19, 57] can be described as follow:

$$\tilde{I}_i = \mathbf{M}_p \odot I_j + (\mathbf{1} - \mathbf{M}_p) \odot I_i \tag{5}$$

where $\mathbf{M}_p$ is a mask with the same size as the image I_j. The value of p area in mask $\mathbf{M}_p$ obeys 2D Gaussian distribution in eq.6, and other area in $\mathbf{M}_p$ is filled with zero values.

$$\mathbf{M}_p = \begin{cases} e^{-\frac{(x-x_0)^2}{(w/2)^2} - \frac{(y-y_0)^2}{(h/2)^2}}, & (x,y) \in p \\ 0, & otherwise \end{cases} \tag{6}$$

where (x, y) is the coordinates in image coordinates system. w and h are the width and height of patch p. In the Fig.3, we present some augmented examples.

Noise jitter for Range Image. According to the properties of LiDAR sensor, the laser has poor reflectivity when it encounters transparent materials or objects with black color. To address this issue, we propose a data augmentation method for corrupting range images, which has been used in the 2D vision community [5, 7] and has not been explored in 3D vision. EMMF-Det uses range images as input. We randomly generate some patches of variable size on the range image and replace the point cloud data in these patches with zero values. By adding data corruption, we train a network to learn robust representation for LiDAR signals.

4 Experiments

Here we first describe the implementation details of EMMF-Det in section 4.1. Then we compare with the SOTA methods on Waymo Open Dataset and our actual operation scenario dataset in section 4.2 and section 4.3. We also conduct extensive ablation studies to analyze the effectiveness of different components in section 4.4.

4.1 Implementation details

Network Details. The overall framework is shown in Fig.2. The 2D FCN part in section 3.1 uses four blocks to downsample the features by a factor of 8 and another four blocks to recover the features to the original size. All blocks apply a series of dilated convolutions to extract multi-scale features. For the anchor-free head, we follow the settings in CenterPoint [54]. In the feature re-weighting layer, we use the hyperparameters $k = 8$ to search candidate fusion set $\mathbf{\Omega}$ for Waymo Open Dataset and self-built dataset. The choice of N_c is mainly determined by k and the number of points in the point cloud. Here we empirically set $10,240$ for Waymo Open Dataset and $2,048$ for self-built dataset when k equals 8.

Training and Inference Details. We train the network for 30 epochs on 8 Tesla V100 GPUs using the ADAM optimizer with batch size 16. For the learning rate, we use a one-cycle learning rate policy with max learning rate 0.003, weight decay 0.01, and momentum 0.85 to 0.95. During the inference stage, we keep the top 500 predictions and use NMS with IoU threshold 0.5 and score threshold 0.1.

Table 1. Performance comparison on Waymo validation set. The AP/APH results for Level 1 and Level 2 are shown in the table below. Note that all methods we compared with only use one frame LiDAR signal as input. † : implemented by ourselves using open source code. $\star$: from [9]. ‡ : design two model for vehicle and pedestrian seperately. $*$: use IoU prediction module in [61]. Some papers only provide evaluation results on one category, so we use '−' instead of those that are not provided. The best result is marked in red, and the second result is marked in blue.

Method	Input	Vehicle (AP/APH)				Ped (AP/APH)				All (AP/APH)			
		L1		L2		L1		L2		L1		L2	
Two-stage:													
PV-RCNN† [36]	Point Cloud	75.2	74.6	66.4	65.9	65.9	58.1	58.2	51.1	70.6	66.4	62.3	58.5
Single-stage:													
MVF [62]	Point Cloud	62.9	-	-	-	65.3	-	-	-	64.1	-	-	-
PointPillars [18]	Point Cloud	56.5	-	-	-	59.3	-	-	-	57.9	-	-	-
Pillar-based [48]	Point Cloud	69.8	-	-	-	72.5	-	-	-	71.2	-	-	-
SECOND† [51]	Point Cloud	70.3	69.8	62.6	62.1	62.1	52.9	54.0	45.9	66.2	61.4	58.3	54.0
CenterPoint-pillar* [54]	Point Cloud	74.6	74.1	66.3	65.8	71.5	59.6	64.1	53.2	73.1	66.9	65.2	59.5
CenterPoint-voxel* [54]	Point Cloud	74.8	74.2	66.7	66.2	75.8	69.6	68.3	62.4	75.3	71.9	67.5	64.3
SST [9]	Point Cloud	74.2	73.8	65.5	65.1	78.7	69.5	70.0	61.6	76.5	71.7	67.8	63.4
AFDetV2-Lite* [14]	Point Cloud	77.6	77.1	69.7	69.2	80.2	74.6	72.2	67.00	78.9	75.9	71.0	68.1
PointAugmenting [45]	Point Cloud, Cam	67.4	-	62.7	-	75.4	-	70.6	-	71.4	-	66.7	-
RCS [1]	Range	69.6	69.2	-	-	-	-	-	-	-	-	-	-
RSN ‡ [41]	Range	74.6	-	-	65.5	77.8	-	-	63.7	76.2	-	-	64.6
RangeDet [10]	Range	72.9	-	-	-	75.9	-	-	-	74.4	-	-	-
RangeIoUDet [24]	Range	72.2	-	-	-	60.4	-	-	-	66.3	-	-	-
PPC-EdgeConv [4]	Range	65.2	-	-	56.7	73.9	-	-	59.6	69.6	-	-	58.2
PPC-EdgeConv-Cam [4]	Range, Cam	-	-	-	-	75.5	-	-	61.5	-	-	-	-
EMMF-Det	Range, Cam	76.2	75.5	67.8	67.3	77.5	69.1	69.8	61.9	76.9	72.3	68.8	64.6
EMMF-Det*	Range, Cam	77.1	76.7	69.1	68.4	80.5	74.7	72.6	65.9	78.8	75.7	70.9	67.2

4.2 3D Detection on Waymo Dataset

Waymo dataset is one of the most popular 3D detection datasets, which contains totally 798 scenes for training and 202 scenes for validation. Each pair of sample consists of one frame of LiDAR return and five frames of camera images. These five camera images only cover the 250 degree range in total compared with the LiDAR which covers 360 degree range. To address this issue, we use LiDAR-only model to complete the prediction result.

Waymo uses (Average precision) AP and APH which incorporates heading information as the metric. We compare our method with many SOTA 3D detection methods, including the most popular one-stage 3D detector CenterPoint[54], two-stage 3D detector PV-RCNN [36], transformer based method SST [9], the latest multimodal detection method pointaugmenting [45] and some methods [1, 10, 24, 4] using range images as input. For fair comparison, the results provided in this table only use single frame LiDAR as input. Table.1 shows the performance of our method on the validation set. Since some works only train their models on vehicle or pedstrian and the results reported in their papers arev not complete, we use their open-source codes to reproduce on all categories. The reproduced accuracy of PV-RCNN is 4.0 points better than results reported in

Table 2. Performance comparison on self-built dataset. † : implemented by ourselves using open source code. The best result is marked in red, and the second result is marked in blue.

Method	Model	Car		Bus		Pedestrian		Cyclist		Tricycle		Mean	
		Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard	Easy	Hard
Two-stage:													
PV-RCNN† [6]	3D conv	98.6	95.3	84.2	72.9	53.6	45.9	77.3	72.3	73.3	61.5	77.3	69.6
Single-stage:													
MVX-Net† [40]	3D conv	96.9	96.4	72.1	63.8	56.2	46.8	79.6	73.8	64.7	58.7	73.9	67.9
SECOND† [51]	3D conv	98.5	95.3	83.8	72.8	50.0	42.9	76.6	70.1	64.8	57.3	71.3	67.7
CenterPoint† [54]	3D conv	99.0	95.1	85.1	76.1	53.1	44.3	79.7	72.2	68.4	64.7	77.1	70.5
PointPillars† [18]	2D conv	96.4	95.2	80.3	66.4	47.6	39.6	71.3	64.9	58.9	51.0	70.9	63.4
RangeIoUDet† [24]	2D conv	98.9	96.2	82.7	70.0	52.3	44.2	76.0	69.6	66.7	59.5	75.3	67.9
EMMF-Det	2D conv	99.4	96.7	87.2	78.4	58.9	51.3	82.9	76.8	72.3	66.1	80.1	73.9

original paper. CenterPoint only reports the APH on Level2 difficulty, so we use the results of the latest published paper SST [9] as a reference. Compared with RSN [41], our method performs better on vehicle but worse on pedestrian's APH. However, it designs two networks and trains two categories seperately, while we only uses one network. Inspired by AFDetV2 [14], we integrate the IoU module[61] with EMMF-Det and achieves 77.1 L1 AP on vehicle and 80.5 L1 AP on pedestrian. Compared with all other 3D detection methods, it can be observed that EMMF-Det outperforms on the vehicle class. And compared with other range-view-based methods, the performance of EMMF-Det is significantly ahead.

4.3 3D Detection on Self-Built Lidar-Vision Dataset

We also evaluate our proposed method on the self-built dataset. This dataset is collected by a HESAI Pandar40 LiDAR sensor and a front-view camera. There are $18,000$ samples collected for training, $3,000$ samples for validation and $3,000$ samples for testing in total. We label these samples with five categories including: car, bus, pedestrian, cyclist and tricycle. The 3D detection performance is measured by the average precision (AP) with 40 recall positions. The IoU threshold for vehicle category is set as 0.7. For other non-vehicle categiories, the IoU threshold is set as 0.5. Furthermore, we use the pixel height of the box on the camera image to measure the difficulty of bounding boxes.

Table.2 shows the performance of our method on the test set of self-built dataset. We adopt the open-source code to reproduce the SOTA methods on our dataset. For each method, we adjust the parameters based on our dataset and report the best accuracies. From the Table.2, we can observe that our model boost the performance on all categories significantly. The results can prove that our proposed EMMF-Det still has a good performance even if the LiDAR sensor is changed.

Table 3. Quantitative performance comparison on Waymo validation set of using Multi-Modal Feature Re-Weighting. (averaged by 5 trials)

Method	Vehicle (AP/APH)				Ped (AP/APH)				All (AP/APH)			
	L1		L2		L1		L2		L1		L2	
Range-Based-Method	70.7	70.1	62.3	61.8	74.7	65.5	66.2	58.4	72.7	67.8	64.3	60.1
+ Naive Fusion	71.1	71.0	63.1	62.6	75.4	66.1	66.5	59.1	73.3	68.5	64.8	60.8
+ Feature Re-Weighting	**72.2**	**71.8**	**65.0**	**64.7**	**76.3**	**67.1**	**67.8**	**60.7**	**74.3**	**69.5**	**66.4**	**62.7**

Table 4. Quantitative performance comparison of differnet choices of k on Waymo validation set. (averaged by 5 trials)

Method	Vehicle (AP/APH)				Ped (AP/APH)				All (AP/APH)			
	L1		L2		L1		L2		L1		L2	
$k = 8$	**72.2**	**71.8**	**65.0**	**64.7**	**76.3**	67.1	67.8	**60.7**	**74.3**	**69.5**	**66.4**	**62.7**
$k = 16$	71.6	71.4	64.2	63.6	76.1	**67.2**	**68.0**	60.5	73.9	69.3	66.1	62.1

4.4 Ablation Studies

In this section, we present ablation studies on Waymo Open Dataset to better understand how each component affects the performance. Unless specified, we train the model for 30 epochs with 20% training samples and evaluate on the entire validation set for the reason that training with whole Waymo dataset is computationally demanding.

Effectiveness of Feature Re-Weighting. In Table.3, we compare the performance with model without feature re-weighting. 'Range-based-method' denotes the 3D detector only using range images as input. 'naive fusion' denotes the method directly concatenating the multi-modal features together to improve the range image features with camera image features. 'Feature re-weighting' denotes the method using self-attention to do feature re-weighting mentioned in section 3.3. It can be observed that our proposed feature re-weighting is a effective fusion method for range and camera images.

Effectiveness of *k* in Fusion Candiate Set. For the hyperparameter k in fusion candidate set, we refer to the *ball query* operation in pointnet++ [34] and empirically try several choices of k. The comparisons are illustrated in Table.4. When k is set as 8, the model achieves the best performance. If we set k too large, the performance becomes poor. This is partly because redundant points in the set $\mathbf{\Omega}$ will have a negative impact on the localization of objects. However, the improvment is robust to the hyperparameter k. Whether K is set as 8 or 16, the performance is still better than directly concancating multi-modal features.

Table 5. Comparison of using different augmentation strategies. * denotes using 'linear combination of pixels' instead of 'copy paste' in camera modality.(averaged by 5 trials)

Method	Vehicle (AP/APH)		Ped (AP/APH)		All (AP/APH)	
	L1	L2	L1	L2	L1	L2
+ wo copy-paste	70.5 70.2	62.8 61.7	72.1 63.9	65.1 58.0	71.3 67.1	64.0 59.9
+ multi-modal copy-paste	71.2 70.8	63.3 62.8	74.6 65.0	66.2 59.6	73.0 67.8	64.8 61.2
+ multi-modal copy-paste*	71.8 71.2	63.8 63.2	74.7 65.5	66.8 60.0	73.3 68.4	65.3 61.6
+ noise jitter	**72.2 71.8**	**65.0 64.7**	**76.3 67.1**	**67.8 60.7**	**74.3 69.5**	**66.4 62.7**

Effectiveness of Data Augmentations. To further improve our proposed EMMF-Det, we introduce two data augmentation strategies. One is multi-modal copy-paste, and the other is noise jitter for range images. In the Table.5, 'multi-modal copy-paste' means directly pasting the image patches on the original images in camera modality, while 'multi-modal copy-paste*' denotes using linear combination of pixels instead. It can be observed that multi-modal copy-paste strategy improve the performance over the baseline. If we use linear combination of pixels, the performance can be further improved. Noise jitter for range images enables the model to learn robust features. The improvement in Table.5 further supports our motivation of introducing certain disturbances during training.

5 Conclusion and Discussion

In this paper, we propose the EMMF-Det, which is a novel multi-modal fusion 3D detector using range images and camera images as input. EMMF-Det uses a one-stage anchor-free 3D detector as the baseline, and fuse the LiDAR features with the camera features to further improve the performance. To enhance the LiDAR features with the texture, color information provided by camera images, multi-modal feature re-weighting layer is designed for local fusion within the points/pixels neighborhood. Furthermore, we introduce two effective data augmentation strategies including multi-modal copy-paste and noise jitter for range images based on the properties of LiDAR and camera signal. Experiments evaluated on several 3D detection benchmarks demonstrate the effectiveness of our method.

Limitation of Range-view-based Methods. EMMF-Det uses range images as input, which are raw signal scaned by mechanical LiDAR. The results and conclusions are under the premise that Waymo and self-built dataset both are collected by mechanical LiDAR. If the dataset is collected by solid-state LiDAR, there will be information loss during the transformation from point cloud to range image. We will explore and try to solve this problem in future work.

Acknowledgement: Supported by National key R&D Program of China (Grant No.2020AAA010400x)

References

1. Bewley, A., Sun, P., Mensink, T., Anguelov, D., Sminchisescu, C.: Range conditioned dilated convolutions for scale invariant 3d object detection. arXiv preprint arXiv:2005.09927 (2020)
2. Bhattacharyya, P., Huang, C., Czarnecki, K.: Self-attention based context-aware 3d object detection. arXiv e-prints pp. arXiv–2101 (2021)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
4. Chai, Y., Sun, P., Ngiam, J., Wang, W., Caine, B., Vasudevan, V., Zhang, X., Anguelov, D.: To the point: Efficient 3d object detection in the range image with graph convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2021)
5. Chen, P., Liu, S., Zhao, H., Jia, J.: Gridmask data augmentation. arXiv preprint arXiv:2001.04086 (2020)
6. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
7. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3d object detector with sparse transformer. arXiv preprint arXiv:2112.06375 (2021)
10. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: Rangedet: In defense of range view for lidar-based 3d object detection. arXiv preprint arXiv:2103.10039 (2021)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
12. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)
13. He, C., Zeng, H., Huang, J., Hua, X.S., Zhang, L.: Structure aware single-stage 3d object detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11873–11882 (2020)
14. Hu, Y., Ding, Z., Ge, R., Shao, W., Huang, L., Li, K., Liu, Q.: Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 969–979 (2022)
15. Huang, T., Liu, Z., Chen, X., Bai, X.: Epnet: Enhancing point features with image semantics for 3d object detection. In: European Conference on Computer Vision. pp. 35–52. Springer (2020)
16. Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L.: Joint 3d proposal generation and object detection from view aggregation. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–8. IEEE (2018)
17. Kuang, H., Wang, B., An, J., Zhang, M., Zhang, Z.: Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds. Sensors **20**(3), 704 (2020)

18. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
19. Li, H., Zhang, X., Sun, R., Xiong, H., Tian, Q.: Center-wise local image mixture for contrastive representation learning. arXiv preprint arXiv:2011.02697 (2020)
20. Li, H., Zhang, X., Tian, Q., Xiong, H.: Attribute mix: Semantic data augmentation for fine grained recognition. In: 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP). pp. 243–246. IEEE (2020)
21. Li, P., Shi, J., Shen, S.: Joint spatial-temporal optimization for stereo 3d object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6877–6886 (2020)
22. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: Multi-task multi-sensor fusion for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7345–7353 (2019)
23. Liang, Z., Zhang, M., Zhang, Z., Zhao, X., Pu, S.: Rangercnn: Towards fast and accurate 3d object detection with range image representation. arXiv preprint arXiv:2009.00206 (2020)
24. Liang, Z., Zhang, Z., Zhang, M., Zhao, X., Pu, S.: Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7140–7149 (2021)
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
27. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3164–3173 (2021)
28. Meyer, G.P., Charland, J., Hegde, D., Laddha, A., Vallespi-Gonzalez, C.: Sensor fusion for joint 3d object detection and semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
29. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4213–4220. IEEE (2019)
30. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2906–2917 (2021)
31. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3d object detection with pointformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7463–7472 (2021)
32. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7077–7087 (2021)
33. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 918–927 (2018)

34. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
35. Sheng, H., Cai, S., Liu, Y., Deng, B., Huang, J., Hua, X.S., Zhao, M.J.: Improving 3d object detection with channel-wise transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2743–2752 (2021)
36. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)
37. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
38. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE transactions on pattern analysis and machine intelligence (2020)
39. Shinya, Y.: USB: Universal-scale object detection benchmark. arXiv:2103.14027 (2021)
40. Sindagi, V.A., Zhou, Y., Tuzel, O.: Mvx-net: Multimodal voxelnet for 3d object detection. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7276–7282. IEEE (2019)
41. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2021)
42. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
44. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4604–4612 (2020)
45. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11794–11803 (2021)
46. Wang, G., Tian, B., Ai, Y., Xu, T., Chen, L., Cao, D.: Centernet3d: An anchor free object detector for autonomous driving. arXiv preprint arXiv:2007.07214 (2020)
47. Wang, Q., Chen, J., Deng, J., Zhang, X.: 3d-centernet: 3d object detection network for point clouds with center estimation priority. Pattern Recognition **115**, 107884 (2021)
48. Wang, Y., Fathi, A., Kundu, A., Ross, D.A., Pantofaru, C., Funkhouser, T., Solomon, J.: Pillar-based object detection for autonomous driving. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. pp. 18–34. Springer (2020)
49. Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., , Solomon, J.M.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: The Conference on Robot Learning (CoRL) (2021)
50. Wang, Z., Jia, K.: Frustum convnet: Sliding frustums to aggregate local pointwise features for amodal 3d object detection. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1742–1749. IEEE (2019)

51. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
52. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1951–1960 (2019)
53. Ye, M., Xu, S., Cao, T.: Hvnet: Hybrid voxel network for lidar based 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1631–1640 (2020)
54. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11784–11793 (2021)
55. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. NeurIPS (2021)
56. Yoo, J.H., Kim, Y., Kim, J., Choi, J.W.: 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16. pp. 720–736. Springer (2020)
57. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
58. Zhang, Z., Ji, Y., Cui, W., Wang, Y., Li, H., Zhao, X., Li, D., Tang, S., Yang, M., Tan, W., et al.: Atf-3d: Semi-supervised 3d object detection with adaptive thresholds filtering based on confidence and distance. IEEE Robotics and Automation Letters (2022)
59. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10076–10085 (2020)
60. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268 (2021)
61. Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.W.: Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 3555–3562 (2021)
62. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: Conference on Robot Learning. pp. 923–932. PMLR (2020)
63. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)