Multi-Modal Fusion Based on Depth Adaptive Mechanism for 3D Object Detection

Zhanwen Liu, *Member, IEEE*, Juanru Cheng, Jin Fan, Shan Lin, Yang Wang, *Member, IEEE*, and Xiangmo Zhao, *Member, IEEE*

***Abstract*—Lidars and cameras are critical sensors for 3D object detection in autonomous driving. Despite the increasing popularity of sensor fusion in this field, accurate and robust fusion methods are still under exploration due to non-homogenous representations. In this paper, we find that the complementary roles of point clouds and images vary with depth. An important reason is that the point cloud appearance changes significantly with increasing distance from the Lidar, while the image's edge, color, and texture information are not sensitive to depth. To address this, we propose a fusion module based on the Depth Attention Mechanism (DAM), which mainly consists of two operations: gated feature generation and point cloud division. The former adaptively learns the importance of bimodal features without additional annotations, while the latter divides point clouds to achieve differential fusion of multi-modal features at different depths. This fusion module can enhance the representation ability of original features for different point sets and provide more comprehensive features by using the dual splicing strategy of concatenation and index connection. Additionally, considering point density as a feature and its negative correlation with depth, we build an Adaptive Threshold Generation Network (ATGN) to generate the depth threshold by extracting density information, which can divide point clouds more reasonably. Experiments on the KITTI dataset demonstrate the effectiveness and competitiveness of our proposed models.**

***Index Terms*—Deep learning, 3D object detection, sensor fusion, Lidar sensor, camera sensor.**

I. Introduction

3D OBJECT detection is an essential task in the field of autonomous driving and plays an indispensable role in the downstream tasks of the perception pipeline, such as safe driving and route planning. Due to these academic and industrial values, 3D object detection has drawn much research attention in recent years [1], [2], [3], [4], [5], [6].

Image-based 3D object detection is prone to false positives due to the lack of depth information, and its performance in identifying and localizing objects is limited by 2D data formats. On the other hand, Lidar provides a highly accurate range view, making it ideal for 3D scene description. However, due to the non-uniform density and orderless nature of 3D point clouds, Lidar-based detection does not perform well on distant objects [7], [8]. In addition, Lidar has difficulty in distinguishing objects with similar geometric structures due to the lack of color and texture information. To address these issues, more and more studies are turning to multi-modal fusion and leveraging the complementary roles of point clouds and images to achieve more accurate and robust detection results.

Previous multi-modal methods can be roughly grouped into two categories. One line of work [9], [10], [11] relies on existing 2D detectors, utilizing frustums to generate proposals. However, the frustums limit the 3D search space for estimating bounding boxes, and the performance of object detection is restricted by 2D detectors due to the cascaded design of the network. Another line of work [12], [13], [14], [15] employs a more focused 3D approach. These methods connect features in the middle of 2D image convolutional networks onto 3D voxels or point features to enrich 3D features. Despite the impressive improvements, these fusion methods still suffer from two problems. Firstly, they simply fuse 3D and 2D features via element-wise addition or concatenation, leading to a decrease in performance when dealing with low-quality image features, *e.g.*, images in poor illumination conditions. Secondly, due to the differences between point clouds and cameras, which capture sparse and continuous features and dense features in discrete states respectively, it is still worth exploring how to align 2D and 3D features at the feature level.

Our work reveals that point clouds lose the spatial shape of objects as sparsity increases, while the properties of objects (*i.e.*, color, texture and edge information) in the 2D image domain remain relatively unchanged with an increase in depth, as shown in Fig. 1. To further illustrate this phenomenon, we conduct two statistical experiments using images and Lidar data from the KITTI dataset [16], as shown in Fig. 2. Firstly, we get ground truth on point clouds and randomly generate several offsets to obtain background bounding boxes with the same depth as ground truth. Then, we cut out the point clouds inside each 3D box (*i.e.*, ground truth, and background bounding boxes) and count

Manuscript received 19 October 2022; revised 31 January 2023; accepted 13 April 2023. Date of publication 26 April 2023; date of current version 21 February 2025. This work was supported in part by the National Natural Science Foundation of China (General Program) under Grant 52172302, in part by Two Chains Integration Key Special Program under Grant 2023-LL-QY-24, in part by the Shannxi Province Traffic Science and Technology Program under Grant 21-02X, in part by the National Key R&D Program of China under Grant 2022YFC3803700, and in part by the National Natural Science Foundation of China (NSFC) under Grant 62206262. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Kai Xu. *(Zhanwen Liu and Juanru Cheng contributed equally to this work.) (Corresponding author: Yang Wang.)*

Zhanwen Liu, Juanru Cheng, Jin Fan, Shan Lin, and Xiangmo Zhao are with the School of Information Engineering, Chang'an University, Xi'an 710064, China (e-mail: zwliu@chd.edu.cn; 1289559064@qq.com; 2060140840@qq.com; lichao971204@foxmail.com; xmzhao@chd.edu.cn).

Yang Wang is with the School of Informatics, University of Science and Technology of China, Hefei 230026, China (e-mail: ywang120@ustc.edu.cn).

Digital Object Identifier 10.1109/TMM.2023.3270638

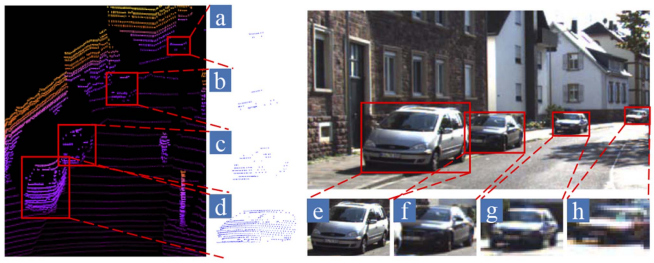


Fig. 1. Near-range points contain complete geometric structure information (c, d), while the far-range objects contain few point clouds and incomplete geometric structures (a, b). In comparison, the corresponding image data always contain dense and complete structure information across different depths of field.

the points. Next, we plot a line graph to show how the number of foreground and background points varies with depth. Likewise, we obtain ground truth on the image and project background bounding boxes of point clouds to the pixel coordinate system. Then we get the pixels inside each 3D box and calculate the RGB values to generate image histograms, which can intuitively reflect the color distribution of objects and backgrounds in the image. Experimental results show that the number of point clouds representing objects decreases sharply with depth, even reaching the same level as the background point clouds. In this case, sparse points will not only lose the geometric structure information of objects but also cause the foreground and background to be indistinct. However, the RGB distribution differs significantly between objects and backgrounds.

Motivated by this, we propose a fusion module based on the Depth Attention Mechanism (DAM) to stimulate the full potential of multi-modal data, which adaptively learns the importance of two heterogeneous features and divides point clouds using depth thresholds. To keep the point cloud sequence unchanged, we apply concatenation within point sets and index connection between point sets to fuse multi-modal features. Due to the sufficient combination of 2D and 3D information, the fusion features are more effective at restoring objects with few points and more discriminative in distinguishing objects with similar geometries. Furthermore, considering the negative correlation between point cloud depth and density, we design an adaptive threshold generation network to divide point clouds more reasonably and guide the efficient completion of the fusion process. The network extracts density information of point clouds as input and relies on MLP [17] abstracting features to adaptively generate the threshold for dividing point clouds. Finally, we design a multi-scale fusion architecture to integrate the two modules, which produces more robust cross-modal features by implementing multi-layer fusion on image and point cloud features of different resolutions.

We evaluate our proposed method on the KITTI dataset [16] to demonstrate its effectiveness and verify the importance of the fusion module based on the Depth Attention Mechanism (DAM) and adaptive threshold generation network through ablation experiments. Qualitative experimental results are also provided to illustrate the effectiveness of our modules.

The key contributions of our work are as follows:

1) We analyze the drop of point cloud number with depth and the distribution of RGB in the image through statistical experiments. And we propose a fusion module based on the Depth Attention Mechanism (DAM) to achieve a differential fusion of image and point cloud features at different depths. Meanwhile, we establish an adaptive threshold generation network based on point cloud density to generate a depth threshold for dividing point clouds in a learnable way.
2) The two modules we proposed have good generalizations. Theoretically, our modules can be built on any point-based detector. In the experiment, we build the proposed modules on three different backbone networks for evaluation, and the results show good improvement.
3) Through quantitative and qualitative experiments on the KITTI dataset, we validate the effectiveness of our proposed modules, with promising detection performance and considerable improvements.

II. Related Work

A. 3D Object Detection Based on Camera Images

Because optical sensors perform well in 2D object detection tasks and have lower device costs, many studies [18], [19], [20], [21], [22] have attempted to expand mature 2D detection pipelines to predict 3D bounding boxes from images. Despite this, most rely on geometrical constraints priors to locate objects due to the lack of depth information. Roddick et al. [20] implement a shift of image domains by mapping image features to orthogonal 3D space, enabling the network to infer the spatial configuration of scenes in a scale-consistent domain. M3D-RPN [19] takes advantage of the 2D and 3D viewing geometry relationships and proposes a depth-aware convolution to improve the understanding of 3D scenes. Chen et al. [1] use a CNN-based object detector to obtain 2D bounding boxes and infer corresponding 3D bounding boxes using semantic, contextual, and shape information. Deep3Dbox [18] solves the direction prediction problem by introducing a novel Multi-Bin loss and enhances the constraint between 2D and 3D bounding boxes by using geometric priors. In addition, some works [23], [24], [25] predict the key points of 3D bounding boxes as an intermediate step to get the object location. Haq et al. [26] use the center regression algorithm to improve heatmap prediction. However, due to the lack of depth information in the image-based methods, accurate 3D object detection has not yet been achieved.

B. 3D Object Detection Based on Point Clouds

Point clouds can capture 3D information better suited to the real world from the environment, so more and more Lidar-based methods [6], [27] have been proposed. Because of the disorder and irregularity of point clouds, many 3D detectors first convert data formats. VoxelNet [4] divides point clouds into voxels and uses stacked voxel feature coding layers to extract voxel features. SECOND [28] is also a voxel-based method and improves computational efficiency by introducing sparse convolution. PointPillars [29] converts point clouds into pseudo images,

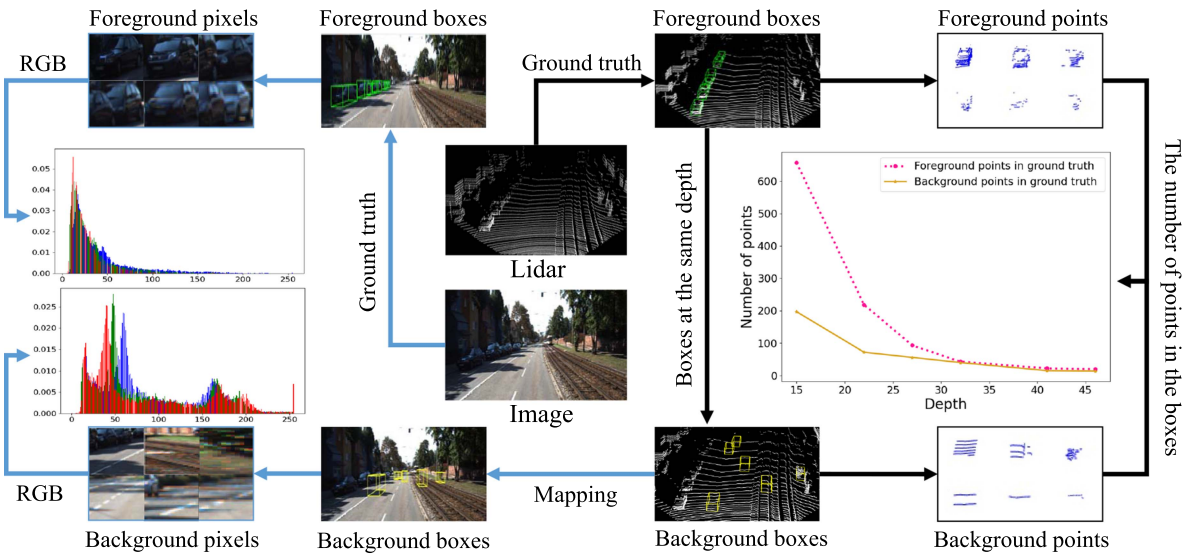


Fig. 2. Statistical experiments on image and Lidar. For Lidar, statistical experiments are conducted on the number of point clouds in the bounding boxes, while for images, statistical experiments are conducted on color information.

freeing the network from time-consuming 3D convolution. CenterNet3D [30] proposes an anchor-free 3D object detection architecture, uses 3D convolution to extract voxel features and proposes a corner attention module to focus more on object boundaries. RangeDet [31] projects raw Lidar data onto regular grids, then uses standard 2D or 3D convolution to calculate BEV characteristics. The performance of the above methods is limited in two aspects: one is the discretization of the grid, and the other is that fine-grained point-level information will be lost in the data conversion process. Other works operate directly on the original point clouds without quantification. PointRCNN [32] uses PointNet++ [33] as the backbone to generate RoIs directly from foreground points, and uses point-level features for bounding box thinning. STD [34] presents a new spherical anchor and implements RoI feature extraction through PointsPool. Point-GNN [35] uses original point clouds to construct a graph and generates predictions using a graph neural network (GNN). However, point clouds have no color and texture information and become sparser with increasing distance from the sensor, which poses challenges for precise object localization.

C. 3D Object Detection Based on Multiple Sensors

The camera-Lidar 3D object detection has received increasing attention due to the complementary effect of point clouds and images. Earlier works [9], [36], [37] use result-level or proposal-level fusion. Qi et al. [9] propose a cascade method, named F-PointNet, that generates 2D proposals from camera images as filters on point clouds to reduce the search range for each object. However, the fusion granularity is too coarse to release the full potential of both modalities. Moreover, cascading methods require additional 2D annotations, and network performance is limited by 2D detectors. MV3D [36] and AVOD [13] try to infer images and BEVs together. They project all RoIs onto BEV and image features and fuse them to predict object bounding boxes. Liang et al. [38] expect to introduce other tasks, such as depth completion, to improve the accuracy of 3D object detection. PointPainting [39] and IPOD [40] project the results of the 2D semantic segmentation task onto 3D point clouds. The difference is that the former directly uses fusion features for 3D object detection, while the latter filters the background points through projection and generates proposals from each foreground point. However, these simple point-level connections ignore the quality of the real data and the contextual relationship between the two modes, resulting in performance degradation when image features or point cloud features are defective. We explore a more robust and effective fusion mechanism where point clouds are divided into different sets according to an adaptively generated threshold. The fusion module independently learns the importance of different modes in each set and uses two splicing strategies to complete the multi-modal feature fusion within and between sets.

III. Methodology

As shown in Fig. 3, the multi-scale fusion architecture consists of four parts: the Lidar and image feature extraction, the adaptive threshold generation network, and the fusion module based on the Depth Attention Mechanism (DAM). In Section III-A, we first introduce the multi-scale fusion architecture. Then, Section III-B introduces how to extract point cloud density information as input to generate a depth threshold adaptively. Finally, Section III-C describes how to achieve dynamic learning of gated weights and complete feature fusion based on depth threshold.

A. Framework Overview

Fig. 3 illustrates the multi-scale fusion architecture. Similar to EPNet [41], in Lidar feature extraction, we use PointNet++[33] as the backbone and build four Set Abstraction (SA) layers

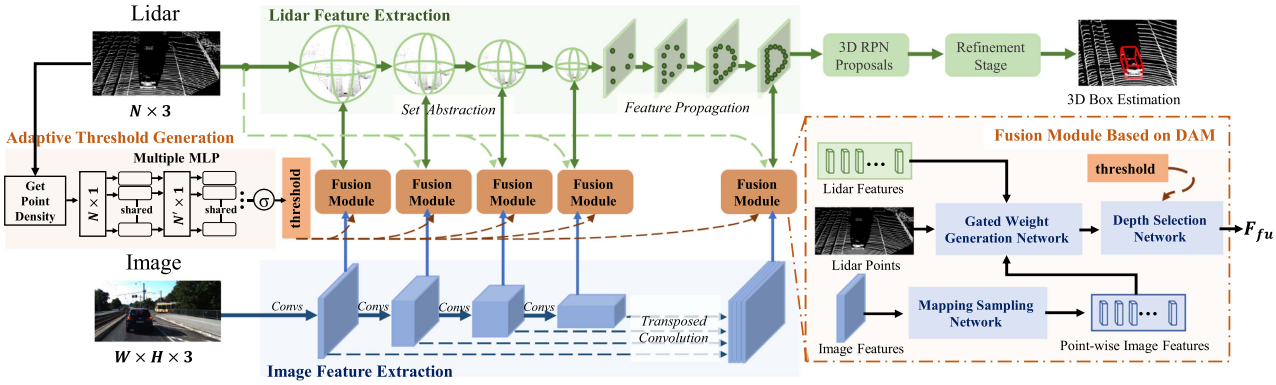


Fig. 3. Multi-scale fusion architecture. There are four parts: Lidar and image feature extraction, adaptive threshold generation network, and fusion module based on the Depth Attention Mechanism (DAM). N represents the number of Lidar points. H and W represent the height and width of the input images, respectively.

for point cloud feature downsampling. The sampling points are 4096, 1024, 256, and 64, respectively. To ensure each point has complete feature representation, four Feature Propagation (FP) layers are used to realize point cloud feature upsampling. In image feature extraction, we build four convolution blocks to match the resolution of point cloud features. Each convolution block contains a BN layer, a ReLU activation function, and two convolution layers where the stride of the second convolution layer is set to 2 for feature downsampling and expanding the receptive field. Transpose convolutions are used to realize upsampling from four image features with different resolutions. The outputs are then concatenated in the feature dimension.

In the feature downsampling stage, we use four fusion modules based on the Depth Attention Mechanism (DAM) to fuse image and point cloud features with the same scale. In the feature upsampling stage, we only use the fusion module on the output of the last FP layer. The adaptive threshold generation network provides a depth threshold for each fusion module to guide the effective implementation of the fusion process. The resulting compact and discriminative fused features are used to generate proposals, which are then modified in the refinement stage to estimate 3D bounding boxes of objects.

B. Adaptive Threshold Generation

The statistical experiments in Fig. 2 demonstrate that the point cloud density decreases significantly as the depth increases. This inhomogeneity is closely related to the natural divergence of Lidar points with distance due to the angular offsets between Lidar lasers. To address this issue, we extract the density information of the point clouds and transfer it to the adaptive threshold generation network to generate a depth threshold in a learnable manner, similar to a small discriminator for dividing point clouds. This allows us to effectively utilize the rich geometric information of near-range point clouds and the ambiguous object geometry information of distant point clouds.

The adaptive threshold generation network is shown in Fig. 3. We use all Lidar points as center points for density calculation. The advantage of selecting all points is that we can get relatively dense density information. Rich and dense input data can guide networks to better express features in learning. We first partition point clouds into spherical neighborhoods centered at all center points, where the radius of areas is artificially set. Then the number of points in each neighborhood is divided by the corresponding volume to obtain volume density for different regions of point clouds. Afterwards, the volume density is fed into multiple MLP [17] for feature abstraction. The sigmoid activation function is used to normalize the output to the range [0,1], and the generated threshold is provided to the fusion module to guide the orderly completion of the fusion process.

C. The Depth Attention Mechanism (DAM)

We argue that the complementary roles of point clouds and images vary with depth. Specifically, near-range point clouds are relatively dense, and the object's geometric information is rich. In this case, image features are more focused on providing color and texture information to assist point clouds in distinguishing objects with similar structures and localizing objects. However, far-range point clouds are too sparse, and a small number of points cannot retain the spatial structure information of the object, thus losing the unique advantages of Lidar sensors. At this point, image features can identify objects from the background more effectively than point clouds, while point cloud features focus on providing depth information to help predict 3D bounding boxes. Therefore, we propose a fusion module based on the Depth Attention Mechanism (DAM). As shown in Fig. 4, it consists of three subnetworks: the mapping sampling network, the gated weight generation network, and the depth selection network. This module can make full use of image and point cloud features, generate gated feature weights for both, and implement a Depth Attention Mechanism (DAM) based on depth threshold to maximize the potential of both modalities and improve detection performance.

More formally, in the mapping sampling network, given the feature map $F_I \in \mathbb{R}^{W \times H \times C}$ extracted from the image backbone and the feature map $F_L \in \mathbb{R}^{N \times C}$ extracted from the Lidar backbone, we first compute the projected point $\tilde{\mathbf{P}}_i(x_i, y_i)$ on the image plane from each point $\mathbf{P}_i(x_i, y_i, z_i)$.

$$\tilde{\mathbf{P}}_\mathbf{i} = R_{in} \times R_{rect} \times T_{velo\_to\_cam} \times \mathbf{P}_i \tag{1}$$

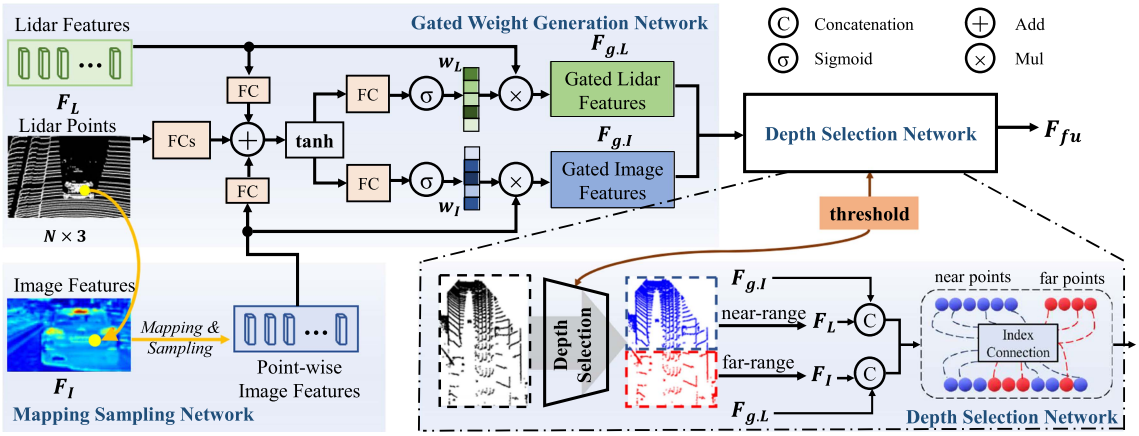


Fig. 4. Fusion module based on the Depth Attention Mechanism (DAM). It consists of a mapping sampling network, a gated weight generation network, and a depth selection network.

Where R_{in} is the camera intrinsic matrix, R_{rect} is the camera rectification matrix, and $T_{velo\_to\_cam}$ represents the projection matrix.

After obtaining the reference point, we adopt bilinear interpolation to obtain point-wise image features, similar as EPNet [41].

The point-wise image features will introduce interference information under adverse factors such as low light. Furthermore, due to the significant difference in point cloud density at different depths, distant objects are covered by only a few points. This vague geometry also introduces confusing information. To achieve efficient fusion, we use a gated mechanism to generate weight matrices for both image features and point cloud features, so that the network can adaptively learn the importance of each modality. Meanwhile, after encoding through multiple Set Abstraction (SA) layers and Feature Propagation (FP) layers, the original depth information in high-dimensional point cloud features is greatly reduced, which is not conducive to adaptively generating weights. One method is to calculate $d^{(p)} = \sqrt{(x^{(p)})^2 + (y^{(p)})^2 + (z^{(p)})^2}$ to compensate for the lost depth information and connect it to the point cloud features. However, this approach not only wastes execution time but also has little effect in encoding since $d^{(p)}$ occupies a small proportion in high-dimensional point cloud features. Another method is to directly use the original point cloud spatial coordinates without additional calculations, as shown in Fig. 4. In the gated weight generation network, we first feed the raw point clouds into several fully connected layers, and the point-wise image features and point cloud features are also passed through a fully connected layer, respectively. Then, we sum the three results of the same channel size and generate two branches through the tanh function, and then compress them into weight maps with a single channel (*i.e.*, w_I and w_L) through another two fully connected layers. We normalize the two weight matrices into the range [0,1] using a sigmoid activation function and multiply them with the point-wise image features and point cloud features, respectively, to generate gated image and gated Lidar features. They are then both fed into the depth selection network for fusion. The formula for generating gated features is described below.

$$F_{g.I} = F_I \times \sigma\left(\boldsymbol{U} tanh\left(\boldsymbol{w}_1 F_{oL} + \boldsymbol{w}_2 F_I + \boldsymbol{w}_3 F_L\right)\right) \quad (2)$$

$$F_{g.L} = F_L \times \sigma\left(\boldsymbol{V} tanh\left(\boldsymbol{w}_1 F_{oL} + \boldsymbol{w}_2 F_I + \boldsymbol{w}_3 F_L\right)\right) \quad (3)$$

where F_{oL} is the raw Lidar data, F_I is the point-wise image features, F_L is the Lidar features, $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{w}_i (i = 1, 2, 3)$ represents learnable parameters in the fusion module, σ represents the sigmoid activation function.

In the depth selection network, we first divide point clouds into two groups by the depth threshold. For the near-range, we directly concatenate Lidar features and gated image features in the channel dimension; for the far-range, we concatenate point-wise image features and gated Lidar features. This coincides with the primary and secondary relationships of semantic and geometric features at different depths. Since point clouds are orderless, instead of using maximum or average pooling to eliminate the influence of disorder like PointNet [17], we use a new feature splicing strategy called index connection, as shown in Fig. 4. The concatenation method changes the order of point clouds, causing unexpected errors in the network during training and learning. Our new splicing method can fuse features between sets while keeping the order of point clouds unchanged. Specifically, we store the index of each point cloud in a matrix and restore the fused features to their corresponding positions during stitching process. This dual-splicing strategy combining concatenation and indexing can be formulated as follows:

$$F_{fu} = \mathcal{C}\left(F_{g.L}{}^{(p_f)} || F_I{}^{(p_f)}, F_L{}^{(p_n)} || F_{g.I}{}^{(p_n)}\right) \quad (4)$$

where F_{fu} represents multi-modal fusion features, $\mathcal{C}(\cdot)$ represents index connection, p_f represents far-range point set, p_n represents near-range point set.

IV. Experiments

We evaluate our models on the KITTI dataset [16]. In this section, we first briefly introduce the dataset in Section IV-A. Then, we provide the implementation details in Section IV-B. Finally,

TABLE I
PERFORMANCE COMPARISON OF 3D DETECTION ON KITTI VAL SET. THE BEST METHODS ARE IN BOLD. THE ABBREVIATIONS 'L' AND 'C+L' DENOTE THE 'LIDAR-ONLY' AND 'CAMERA-LIDAR,' RESPECTIVELY. THE SAME NOTATIONS ARE ALSO USED IN THE FOLLOWING TABLES II AND III. * MEANS THE REPRODUCED RESULTS

Method	Modality	Cars (IoU=0.7)			Pedestrians (IoU=0.5)			Cyclists (IoU=0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
SECOND [28]	L	87.43	76.48	69.10	-	-	-	-	-	-
PointRCNN [32]	L	88.88	78.63	77.38	-	-	-	-	-	-
STD [34]	L	89.70	79.80	79.30	**73.90**	**66.60**	**62.90**	88.50	**72.80**	**67.90**
PointRGCN [44]	L	88.37	78.54	77.60	-	-	-	-	-	-
PV-RCNN [45]	L	92.57	84.83	82.69	64.26	56.67	51.91	88.88	71.95	66.78
Voxel-RCNN [46]	L	92.38	85.29	82.86	-	-	-	-	-	-
CT3D [47]	L	**92.85**	85.82	**83.46**	65.73	58.56	53.04	**91.99**	71.60	67.34
AVOD-FPN [13]	C+L	84.41	74.44	68.65	-	-	-	-	-	-
ContFuse [48]	C+L	86.32	73.25	67.81	-	-	-	-	-	-
PI-RCNN [49]	C+L	87.63	77.87	76.17	-	-	-	-	-	-
F-PointNet [9]	C+L	83.76	70.92	63.65	70.00	61.32	53.59	77.15	56.49	53.37
CLOCs [50]	C+L	92.78	**85.94**	83.25	-	-	-	-	-	-
EPNet [41]	C+L	92.28	82.59	80.14	-	-	-	-	-	-
EPNet [41]*	C+L	91.52	82.15	79.74	70.84	63.64	58.11	83.15	60.25	56.59
EPNet+Ours	C+L	92.18	82.82	80.33	72.22	63.96	57.53	84.57	61.44	57.33

we conduct quantitative and qualitative analysis of experimental results in Sections IV-C and IV-D.

A. Datasets and Evaluation Metric

KITTI Dataset is a standard benchmark dataset for automatic driving, containing 7481 training samples and 7518 test samples. Following the same dataset segmentation protocol as [9], [32], 7481 training samples are further divided into a training set of 3712 and a validation set of 3769 samples. In our experiments, we provide the results on the validation set of all three difficulty levels, namely, easy, moderate, and hard. Objects are categorized into different difficulty levels according to sizes, occlusion, and truncation.

Evaluation Metrics: According to the official evaluation protocol of the KITTI dataset, we adopt average accuracy (AP) as the metric. Furthermore, we apply a newly developed evaluation scheme [42] using 40 recall positions to calculate the average accuracy.

B. Implementation Details

Network Architecture: The multi-scale fusion architecture takes Lidar point clouds and camera images as input. For each 3D scene, we set the range of point clouds to be $[-40, 40]$, $[-1, 3]$, $[0, 70.4]$ meters along the X, Y, and Z axes of the camera coordinate system, and the orientation of θ is in the range of $[-\pi, \pi]$. We extract 16384 points from each scene as input to the Lidar feature extraction. For scenes with points less than 16384, we randomly repeat these points to obtain enough points, which is the same as PointRCNN [32]. And the 1280×384 resolution images are used as input for the image feature extraction. In the adaptive threshold generation network, we use 16384 points as centroids to calculate the point cloud density. We select the first 8000 boxes as proposals in the RPN stage according to the classification confidence. We randomly sample 512 points from each proposal as the input of the refinement stage.

The training scheme: Based on GeForce RTX 3090, the overall architecture is trained using Adam optimizer [43] with the batch size 6 from the initial learning rate 0.002 for 50 epochs. The weight decay and momentum factor are set to 0.001 and 0.9, respectively.

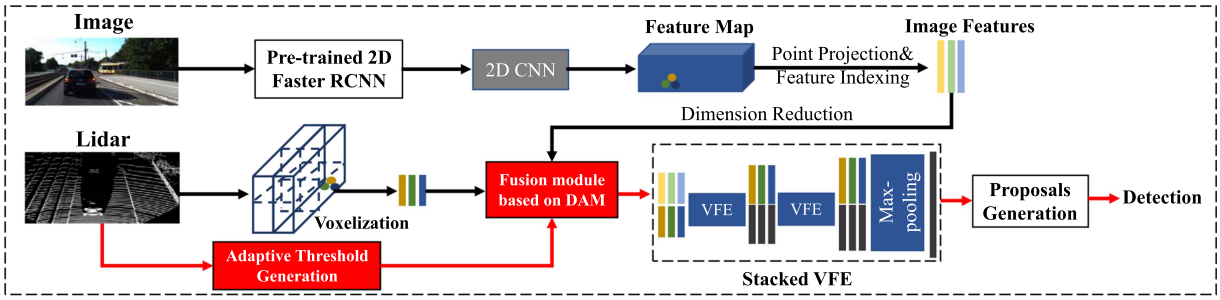
C. Quantitative Results and Analysis

The experimental results on the KITTI dataset: Table I presents the quantitative comparison with state-of-the-art 3D object detection methods on three categories of the KITTI validation set. Generally, our method achieves competitive 3D detection performance. For instance, it improves the LiDAR-based baseline PointRCNN [32] by 3.71%, 1.34%, and 3.81% on three levels of car class. And it improves the reproduced EPNet [41] by $0.7-0.9\%$ AP on car class and $1-2\%$ AP on cyclist class at three different levels, respectively.

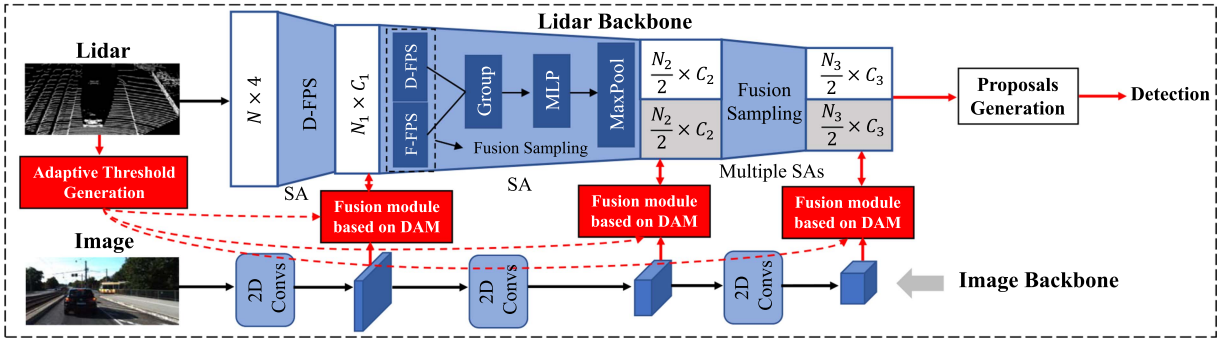
However, our method is lower than the reproduced EPNet [41] on pedestrian class at difficult level, which may be due to feature ambiguity and similarity between pedestrians and other instances (*e.g.*, traffic light poles). Furthermore, there is a large performance gap between our method and the state-of-the-art performance (bold) in Table I on three categories. A possible reason is that the camera projection matrix may produce inaccurate point-to-pixel projections due to calibration errors, leading to poor detection results. We will investigate these issues and provide solutions to achieve more robust and accurate performance in future work.

Performance comparison on three architecture backbones: In addition to EPNet [41], we also train and evaluate the fusion module based on the Depth Attention Mechanism (DAM) and adaptive threshold generation network on two other network architecture backbones: MVXNet [52] and 3DSSD [51]. These enhanced backbones do not require additional supervision.

The network architecture based on MVXNet [52] and 3DSSD [51] are shown in Fig. 5. We replace the Pointwise Concat part on MVXNet [52] PointFusion architecture with the fusion module based on DAM. In 3DSSD [51], image features at different resolutions are fused to Lidar features at three different scales, *i.e.*, 4096 points sampled by D-FPS, 1024 points sampled



(a) MVX-Net PointFusion method with fusion module based on DAM and adaptive threshold generation network



(b) 3DSSD with fusion module based on DAM and adaptive threshold generation network

Fig. 5. Proposed fusion module based on DAM and adaptive threshold generation network augment network architectures for different backbones.

TABLE II
PERFORMANCE COMPARISON OF 3D DETECTION ON KITTI VAL SPLIT WITH AP CALCULATED WITH 40 RECALL POSITIONS *(CARS)* AND WITH IOU THRESHOLD 0.7

Method	Modality	3D Detection				Bird's Eye View				FPS
		Easy	Moderate	Hard	3D mAP	Easy	Moderate	Hard	BEV mAP	
EPNet [41]	C+L	91.52	82.15	79.74	84.47	**95.73**	88.73	88.33	90.93	1.35
EPNet+Ours	C+L	**92.18**	**82.82**	**80.33**	**85.11**	95.62	**89.18**	**88.83**	**91.21**	1.30
3DSSD [51]	L	90.80	80.64	78.34	83.26	92.81	88.30	87.20	89.44	17.72
3DSSD+Ours	C+L	92.14	82.73	79.96	84.94	94.57	88.79	88.51	90.62	12.32
MVX-Net [52]	C+L	87.47	76.10	73.27	78.95	91.99	87.35	85.12	88.15	16.26
MVX-Net+Ours	C+L	88.31	76.83	74.06	79.73	92.43	88.17	87.10	89.23	16.22

TABLE III
ANALYSIS OF INFERENCE TIME AND MODEL COMPLEXITY FOR ENHANCED EPNET BACKBONE

Method	Modality	FPS	FLOPs
EPNet [41]	C+L	1.35	1531.17G
EPNet+Ours	C+L	1.30	1532.22G

by FS, and 512 points sampled by FS. Their codes are based on MMDetection3D [53], a typical 3D object detection platform.

The experimental results of the three enhanced backbones are shown in Table II. As is shown, the enhanced network is superior to the original network in 3D and BEV mAP. For instance, we enhance the three original networks by 0.76%, 2.02%, 0.99% on 3D mAP and 0.31%, 1.32%, 1.22% on BEV mAP, respectively. It is worth noting that we extend 3DSSD [51] to multi-sensor inputs, which inevitably increases inference time. These results demonstrate that our proposed modules achieve significant improvements. Moreover, Table III shows that the enhanced EPNet backbone increases by 0.069% in FLOPs compared to the original network, implying that our method does not drastically increase model complexity.

Ablation experiments: We use the enhanced EPNet [41] backbone to conduct ablation experiments on the KITTI Val dataset to evaluate the effectiveness of our fusion module based on the Depth Attention Mechanism (DAM) and the adaptive threshold generation network. We remove all fusion modules and adaptive threshold generation network from the enhanced EPNet [41] backbone as a baseline. As shown in Table IV, adding the fusion module based on the Depth Attention Mechanism (DAM) yields an improvement of 2.16% in terms of 3D mAP, which reveals that our fusion module can effectively fuse Lidar geometric and image semantic features and the generated fusion features are more conducive for the network to identify and locate objects. Based on adding fusion modules, applying the adaptive threshold yields an improvement of 2.79% over the baseline and 0.63% over only adding fusion modules. Compared with the artificial setting, the threshold generated in a learning manner can better

TABLE IV
ABLATION EXPERIMENTS ON THE EFFECTS OF DIFFERENT COMPONENTS OF THE ENHANCED EPNET BACKBONE

Fusion module	Adaptive threshold	Easy	Moderate	Hard	3D mAP	Gain	FPS	FLOPs
		89.49	79.80	77.67	82.32	-	-	-
√		91.83	81.95	79.67	84.48	↑2.16	1.30	1532.14G
√	√	**92.18**	**82.82**	**80.33**	**85.11**	↑2.79	1.30	1532.22G

TABLE V
ANALYSIS OF DIFFERENT FUSION MECHANISMS ON THE KITTI VAL DATASET

FC	SW	Ours	Easy	Moderate	Hard	3D mAP	Gain	Params	FPS	FLOPs
			89.49	79.80	77.67	82.32	-	-	-	-
√			89.20	78.87	76.62	81.56	↓0.76	15.15M	1.36	1531.00G
	√		91.58	80.71	78.74	83.68	↑1.36	17.26M	1.36	1531.69G
		√	**91.83**	**81.95**	**79.67**	**84.48**	↑2.16	18.84M	1.30	1532.14G

TABLE VI
QUANTITATIVE COMPARISON OF DIFFERENT THRESHOLDS ON THE KITTI VAL DATASET *(CARS)*

Threshold	Easy	Moderate	Hard	3D mAP
30m	91.73	82.07	79.82	84.54
35m	**92.18**	**82.82**	**80.33**	**85.11**
40m	92.02	82.50	80.19	84.90
45m	91.60	81.82	79.94	84.45
50m	91.61	82.03	80.06	84.56
55m	91.78	81.49	79.63	84.30

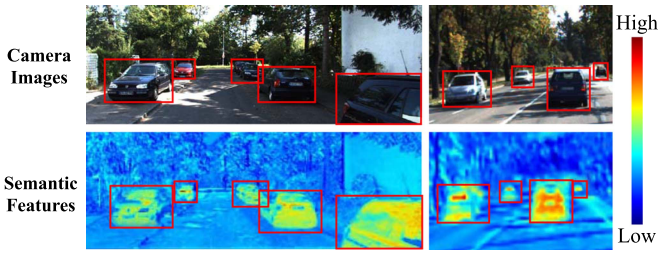


Fig. 6. Visualization of the learned semantic image features. Foreground objects are highlighted with red rectangle boxes.

divide point clouds and guide multi-modal fusion by extracting point cloud density information as input.

We also discuss the difference between our fusion module and the two fusion schemes. One way is feature concatenation (FC). We directly concatenate the obtained point-wise image semantic features and point cloud features in the channel dimension without dividing point clouds and generating weights. The other way is single-weight (SW) fusion. In this method, we only generate a weight map, multiply it with point-wise image semantic features, and then concatenate the obtained features with Lidar features in the channel dimension without dividing the point clouds. The results are shown in Table V. The 3D mAP generated by FC decreases by 0.76% compared with the baseline, suggesting that both semantic features and geometric features may introduce interference information, and the fusion method of simple concatenation cannot reduce the impact of interference information on network performance. In addition, our fusion module outperforms SW by 3D mAP 0.80%, demonstrating that sparse point clouds will also introduce fuzzy information, and it is necessary to generate weights for Lidar features. Meanwhile, it also indicates that our fusion module can effectively fuse multi-modal features and eliminate network performance degradation caused by feature defects.

Analysis of the threshold in the fusion module based on DAM: Concretely, we vary the size of the threshold for dividing point clouds in our fusion module and report the detection accuracy on the car class in Table VI. From the table, we can find that increasing the threshold from 30 m to 35 m improves the 3D mAP by 0.57. If continuously increasing the threshold to 55 m, the 3D mAP is reduced by 0.81 compared with 35 m. We infer that unreasonable threshold settings are insufficient for effective cross-modal fusion learning. Therefore, we construct an adaptive threshold generation network to generate a threshold in a learnable manner.

D. Qualitative Results and Analysis

Visualization of learned image features: We visualize the extracted image features to find areas where semantic features are more concerned during network training. As shown in Fig. 6, without additional explicit supervision information, the network still distinguishes foreground and background well at poor lighting and long distances, which proves that the fusion module based on the Depth Attention Mechanism (DAM) not only accurately establishes point-to-pixel projections, but also provides complementary image semantic information.

Visualization of detection results on the KITTI dataset: In Fig. 7, our network has an excellent correction effect on the orientation angle of object bounding boxes, especially for distant objects. As mentioned earlier, when distant point clouds are too sparse to clearly represent the geometric structure information of the objects, our fusion module uses the Depth Attention Mechanism (DAM) to improve the expression of Lidar features and realizes effective fusion of point clouds and images. Specifically, as shown in Fig. 7(a), false detection occurs in the original network due to the Lidar-dominated fusion in EPNet [41]. When the Lidar points are too sparse, the object structure information cannot be effectively characterized, leading to a lack of distinction

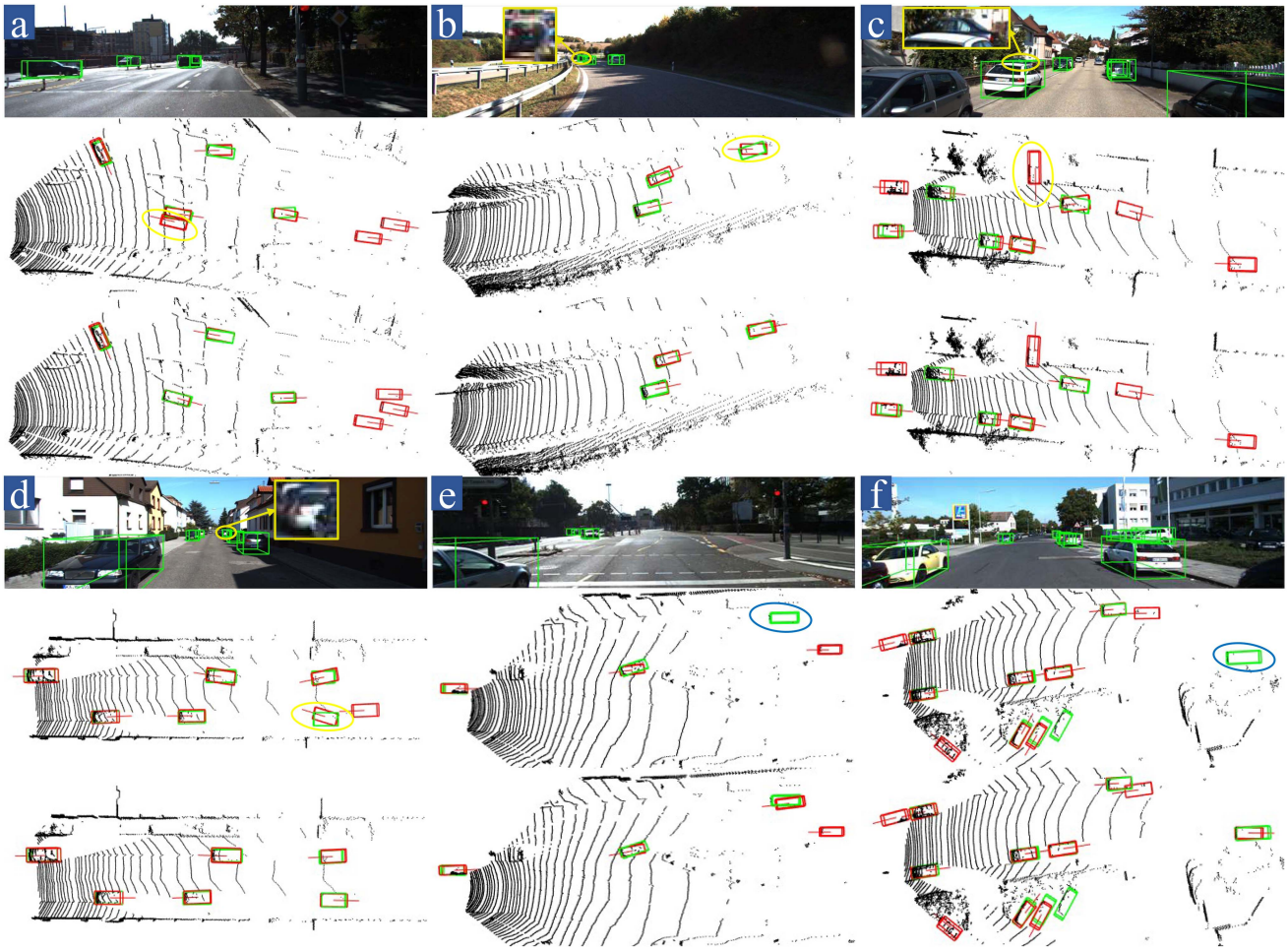


Fig. 7. Qualitative results of KITTI validation samples. For each group, the top is the 3D bounding boxes displayed on the image, the middle is the detection result of EPNet [41], and the bottom is the enhanced EPNet (Ours) detection result. The green boxes represent ground truth, and the red boxes represent the detection results of the network. The yellow circles denote false positives or wrongly oriented bounding boxes, and the blue circles indicate undetected objects.

between foreground and background, resulting in a false positive. By dividing point clouds, our network is able to determine where to rely on which features. Even if there are point cloud interference features, the network does not detect false positives at the same location due to the absence of color-texture information on the object. Moreover, although the original network locates the objects in Fig. 7(c) and (d), it incorrectly estimates the orientation of bounding boxes (given by the red line under the red bounding boxes). In contrast, our network achieves correct detection. Additionally, Fig. 7(e) and (f) demonstrate the excellent performance of our network in detecting distant objects, which is also due to the ideal balance between image semantic features and point cloud geometry features.

V. Conclusion

We propose a multi-scale fusion architecture that includes fusion modules based on the Depth Attention Mechanism (DAM) and an adaptive threshold generation network. The fusion module can adaptively learn image and point cloud feature weights, which encourages features to focus on more informative regions. And by using the depth threshold to divide point clouds, the fusion module can also independently judge where to rely on which features. Additionally, the adaptive threshold generation network extracts point cloud density information as the input, and generates the depth threshold through multi-layer perceptron encoding to guide the practical completion of the fusion process. The two proposed modules with high generalizations can be easily integrated into existing 3D object detection frameworks. Our modules have proven to be effective through quantitative and qualitative experiments on the KITTI dataset. However, there is a huge performance gap between our method and other SOTA methods. In the future, we will study the impact of calibration error and feature ambiguity on network performance to develop more efficient and robust 3D object detection methods.

References

[1] X. Chen et al., "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2147–2156.
[2] B. Xu and Z. Chen, "Multi-level fusion based 3D object detection from monocular images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2345–2353.

[3] X. Chen et al., "3D object proposals using stereo imagery for accurate object class detection," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 40, no. 5, pp. 1259–1272, May 2018.

[4] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[5] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3569–3577.

[6] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7652–7660.

[7] X. Wu et al., "Sparse fuse dense: Towards high quality 3D detection with depth completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5418–5427.

[8] Y. Zhang et al., "Not all points are equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18953–18962.

[9] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.

[10] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep sensor fusion for 3D bounding box estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 244–253.

[11] J. Lahoud and B. Ghanem, "2D-driven 3D object detection in RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4622–4630.

[12] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 808–816.

[13] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.

[14] C. Wang et al., "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3343–3352.

[15] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4421–4430.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc., IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[18] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D bounding box estimation using deep learning and geometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7074–7082.

[19] G. Brazil and X. Liu, "M3D-RPN: Monocular 3D region proposal network for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9287–9296.

[20] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," 2018, *arXiv:1811.08188*.

[21] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Proc. Conf. Robot Learn.*, 2022, pp. 1475–1485.

[22] Y. Liu, Y. Yixuan, and M. Liu, "Ground-aware monocular 3D object detection for autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 919–926, Apr. 2021.

[23] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-time monocular 3D detection from object keypoints for autonomous driving," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 644–660.

[24] X. Ma et al., "Delving into localization errors for monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4721–4730.

[25] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3289–3298.

[26] M. A. Haq et al., "One stage monocular 3D object detection utilizing discrete depth and orientation representation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21630–21640, Nov. 2022.

[27] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "LaserNet: An efficient probabilistic 3D object detector for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12677–12686.

[28] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018, Art. no. 3337.

[29] A. H. Lang et al., "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.

[30] G. Wang et al., "CenterNet3D: An anchor free object detector for autonomous driving," 2020, *arXiv:2007.07214*.

[31] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "RangeDet: In defense of range view for lidar-based 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2918–2927.

[32] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017.

[34] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1951–1960.

[35] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1711–1719.

[36] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.

[37] K. Shin, Y. P. Kwon, and M. Tomizuka, "RoarNet: A robust 3D object detection based on region approximation refinement," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 2510–2515.

[38] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7345–7353.

[39] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4604–4612.

[40] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "IPOD: Intensive point-based object detector for point cloud," 2018, *arXiv:1812.05276*.

[41] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 35–52.

[42] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1991–1999.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[44] J. Zarzar, S. Giancola, and B. Ghanem, "PointRGCN: Graph convolution networks for 3D vehicles detection refinement," 2019, *arXiv:1911.12236*.

[45] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10529–10538.

[46] J. Deng et al., "Voxel R-CNN: Towards high performance voxel-based 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1201–1209.

[47] H. Sheng et al., "Improving 3D object detection with channel-wise transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2743–2752.

[48] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 641–656.

[49] L. Xie et al., "PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12460–12467.

[50] S. Pang, D. Morris, and H. Radha, "CLOCs: Camera-LiDAR object candidates fusion for 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10386–10393.

[51] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11040–11048.

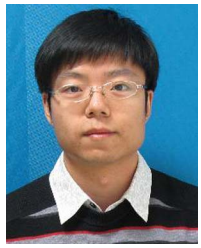
[52] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-Net: Multimodal voxelnet for 3D object detection," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 7276–7282.

[53] M. Contributors, "MMDetection3D: OpenMMlab next-generation platform for general 3D object detection," 2020.

Zhanwen Liu (Member, IEEE) received the B.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2006, and the M.S. and Ph.D. degrees in traffic information engineering and control from Chang'an University, Xi'an,, in 2009 and 2014, respectively. She is currently a Professor with the School of Information Engineering, Chang'an University. Her research interests include vision perception, autonomous vehicles, deep learning, and intelligent transportation systems.

Juanru Cheng received the B.S. degree in 2021 from Chang'an University, Xi'an, China, where she is currently working toward the M.S. degree with the Department of Computer Science and Technology. Her research interests include multi-modal fusion, point cloud and image feature extraction, and their applications in intelligent vehicle, and road infrastructure perception.

Jin Fan received the B.S. degree in 2021 from Chang'an University, Xi'an, China, where she is currently working toward the M.S. degree with the Department of Computer Science and Technology. Her research interests include multi-modal fusion, point cloud, and image matching mechanism and their applications in intelligent vehicle and road infrastructure perception.



Shan Lin received the B.S. degree from Xi'an Jiao Tong University, Xi'an, China, in 2006, and the Ph.D. degree from Chang'an University, Xi'an, in 2012. He is currently an Assistant Professor with the School of Electronic and Control Engineering, Chang'an University, His research interests include intelligent transportation systems and sustainable transportation.

Yang Wang (Member, IEEE) received the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2021. He is currently a Postdoctoral Researcher with the Department of Automation, University of Science and Technology of China. His research interests include machine learning and image processing.

Xiangmo Zhao (Member, IEEE) received the B.S. degree from Chongqing University, Chongqing, China, in 1987, and the M.S. and Ph.D. degrees from Chang'an University, Xi'an, China, in 2002 and 2005, respectively. He is currently a Distinguished Professor with the School of Information Engineering, Chang'an University. His research interests include intelligent transportation systems, internet of vehicles, connected and autonomous vehicles testing technology.