7-5-2021

P2V-RCNN: point to voxel feature learning for 3D object detection from point clouds

Jiale Li
College of Information Science and Electronic Engineering, Zhejiang University

Yu Sun
Zhejiang University

Shujie Luo
College of Information Science and Electronic Engineering, Zhejiang University

Ziqi Zhu
College of Information Science and Electronic Engineering, Zhejiang University

Hang Dai
Mohamed bin Zayed University of Artificial Intelligence

See next page for additional authors

Recommended Citation

J. Li et al., "P2V-RCNN: point to voxel feature learning for 3D object detection from point clouds," in *IEEE Access*, vol. 9, pp. 98249-98260, Jul. 5, 2021. doi: 10.1109/ACCESS.2021.3094562.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 27, 2021, accepted June 26, 2021, date of publication July 5, 2021, date of current version July 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094562

P2V-RCNN: Point to Voxel Feature Learning for 3D Object Detection From Point Clouds

JIALE LI[1], YU SUN[2], SHUJIE LUO[1], ZIQI ZHU[1], HANG DAI[3], ANDREY S. KRYLOV[4], (Member, IEEE), YONG DING[1], (Member, IEEE), AND LING SHAO[5], (Fellow, IEEE)

[1]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China
[2]School of Micro-Nano Electronics, Zhejiang University, Hangzhou 311200, China
[3]Computer Vision Department, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates
[4]Laboratory of Mathematical Methods of Image Processing, Lomonosov Moscow State University, 119991 Moscow, Russia
[5]Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates

Corresponding authors: Hang Dai (hang.dai@mbzuai.ac.ae) and Yong Ding (dingy@vlsi.zju.edu.cn)

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

ABSTRACT The most recent 3D object detectors for point clouds rely on the coarse voxel-based representation rather than the accurate point-based representation due to a higher box recall in the voxel-based Region Proposal Network (RPN). However, the detection accuracy is severely restricted by the information loss of pose details in the voxels. Different from considering the point cloud as voxel or point representation only, we propose a point-to-voxel feature learning approach to voxelize the point cloud with both the point-wise semantic and local spatial features, which maintains the voxel-wise features to build the high-recall voxel-based RPN and also provides the accurate point-wise features for refining the detection results. Another difficulty in object detection for point cloud is that the visible part varies a lot against the full view of object because of the perspective issues in data acquisition. To address this, we propose an attentive corner aggregation module to attentively aggregate the features of local point cloud surrounding a 3D proposal from the perspectives of eight corners in the proposal 3D bounding box. The experimental results on the competitive KITTI 3D object detection benchmark show that the proposed method achieves state-of-the-art performance.

INDEX TERMS 3D object detection, point clouds, attention mechanism, autonomous driving.

I. INTRODUCTION

Three Dimension (3D) point clouds captured with LiDAR sensors have been widely used for various applications, such as autonomous driving [1], [2], robotics [3] and augmented reality [4]. Different from 2D object detection that only locates the object on the 2D image, 3D object detection outputs the 3D position coordinates, 3D size, and orientation of the object in the form of a 3D bounding box, which is more practical and challenging.

According to the representations of point cloud data, the most popular methods can be divided into two major categories: voxel-based and point-based approaches. The voxel-based methods [5]–[9] divide the raw point cloud with local spatial features, such as 3D coordinates and reflection intensities, into the regularly arranged voxels for convolutional feature learning, then project the voxels into the Bird's Eye View (BEV) as the map-view features to construct the map-view Region Proposal Network (RPN). For computational efficiency, the initialized voxel size cannot be set too small, which leads to the rough quantization for objects. The point-based methods [10]–[13] conduct the point-wise feature learning directly from the raw point cloud without data conversion. Since the pose information of object is determined by the distribution of points, the point-based methods usually integrate a detection refinement stage that aims at extracting more accurate object pose features from the point-wise features [10], [12]. The detection refinement stage further decreases the residual error between the proposal bounding boxes and their ground truth with significant detection performance improvements. The map-view RPN [14], [15] has higher box recall than that in the point-based methods [10], [11], but it loses the object pose details in the voxelization process, which is not desirable for an accurate

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

detection refinement. There raises a question: can we maintain the point-wise features with object pose details for more accurate detection refinement while obtaining the map-view feature to construct a RPN with high box recall?

To achieve this, our method first learns the point-wise semantic features for the input point cloud in a segmentation network [16]–[18], then utilizes a novel point-to-voxel feature learning approach to obtain the voxel-wise features from both the point-wise semantic and local spatial features. Different from only the local spatial features in the existing methods [5], [8], the semantic features have larger receptive fields to perceive the semantic and 3D structure information of its surroundings. Moreover, the point-wise features learning can be jointly trained by the semantic supervision [10], which guides the network to understand the content of the point cloud and focus on the foreground regions. The predicted segmentation result can be used as an attention mask to penalize the voxel-wise features for further enhancement. In such a manner, the point-wise semantic features not only strengthen the voxel-wise features in the RPN stage, but also can be used in the detection refinement stage for improving the final detection performance.

Another difficulty in point cloud based object detection is that the visible part varies a lot against the full view of object because of the perspective issues. As shown in Fig. 1, when the objects of the same category are located in different positions or facing different directions, the visible part in the point cloud scene varies a lot. This is not desirable for learning the informative features and perceiving the distance between proposal 3D bounding boxes and their ground truth. To overcome this, we propose a perspective-invariant proposal feature learning approach to improve the proposal feature quality in the detection refinement stage. Given a local point cloud surrounding a 3D proposal, an Attentive Corner Aggregation (ACA) module divides the proposal feature into eight sub-features extracted from the eight perspectives of the proposal 3D bounding box's eight corners. Then the eight sub-features are adaptively penalized by the perspective-channel attention. The proposal feature obtained by the adaptive re-weighting in the eight perspectives is more robust to the variability in the relationship between the visible part and the full view of objects

The main contributions can be summarized as three-fold:

- We propose a novel 3D object detection method with point-to-voxel feature learning that achieves state-of-the-art performance on the highly competitive KITTI 3D object detection benchmark [19].
- We present a point-to-voxel feature learning approach, which maintains both the voxel-wise features for building the high-recall map-view RPN and the point-wise features for preserving the accurate pose information of object.
- We propose an attentive module to learn the perspective-invariant features that improves the detection accuracy in the refinement stage.

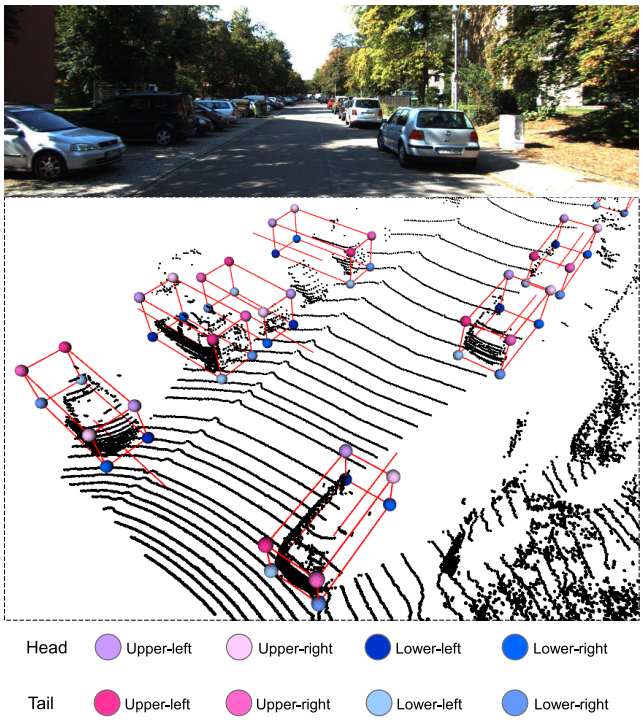


FIGURE 1. Illustration of the variability in the relationship between the visible part and full view of objects. The top row is the camera image for visualization, and the bottom row is the corresponding point cloud. The points in the point cloud and the ground truth bounding boxes are colored black and red, respectively. The lines extending from the bottom of bounding boxes point to objects' heading direction. For better observation, the point cloud is zoomed in and rotated appropriately.

The rest of the paper is organized as follows. Section II introduces the related works on 3D object detection for point clouds. Then we describe the proposed P2V-RCNN in Section III. In Section IV, we present the experimental results and the ablation study. A final section is used for conclusions.

II. RELATED WORKS

In the field of point cloud 3D object detection, some older methods are based on the multi-view image representation, and the recent methods are based on the voxel and point representations.

A. 3D OBJECT DETECTION BASED ON MULTI-VIEW IMAGES

To deal with the sparsity and disorder of a point cloud, some methods [20]–[26] project it into the images in the Bird's Eye View (BEV) and the Front View (FV), encoded with height, density, and other handcrafted statistics as the pseudo image channels. MV3D [20] firstly turns the point cloud into BEV images for region proposal generation and then refines the proposals with the camera image, the FV and BEV images of the point cloud. AVOD [21] merges the camera and BEV image features for region proposal generation with higher box recall on small objects. For higher computational efficiency, PIXOR [22] proposes a single-stage network only based on the BEV images of the point cloud. Later, some methods [24], [25] develop better and finer fusion strategies for

multiple views. Although the methods based on the multi-view image representation mentioned above can use a 2D Convolutional Neural Network (CNN) directly, they are always limited by the information loss introduced by being projected into a certain fixed-resolution 2D grid and the handcrafted input features.

B. 3D OBJECT DETECTION BASED ON VOXELS

The voxel-based methods [5], [9], [14], [27], [28] quantify the point cloud as regularly arranged voxels by a coarse step along the axes of X, Y, and Z, then collapse the Z axis to the channel axis forming a structured BEV map-view feature map for RPN. VoxelNet [5] proposes a Voxel Feature Encoding (VEF) layer to learn the voxel-wise local spatial features for 3D convolutional processing while SECOND [7] introduces the 3D sparse convolution [29], [30] to the VoxelNet for tackling the computational burden of the 3D CNN. Instead of dividing the point cloud along with three directions, PointPillars [8] and SCNet [6] voxelize it along the axes of X and Y to directly construct the map-view feature for 2D convolutional processing. The image features [31], [32] and the attention mechanism [33] are explored to strengthen the voxel feature learning. To improve the detection results from the coarse voxel-based representation, the methods [14], [15] integrate a detection refinement stage, which aggregate the voxel features around a proposal to the densely arranged grid points [14] or voxels [15] inside a proposal box. The reuse of the features on coarse voxel locations still loses the pose information in the accurate point coordinates.

C. 3D OBJECT DETECTION BASED ON POINTS

Although the point-based methods can perform the point-wise feature learning with accurate point coordinates, they generally require designing the specific region proposal strategies due to the unstructured representation format. F-PointNets [34] uses frustum proposals from 2D detection on the corresponding camera image to narrow the search scope in the point cloud, and directly regresses the 3D bounding boxes based on the interior points of a frustum proposal. In such a cascaded framework [34], [35], the 3D detection performance is severely limited by the result of 2D detection. Instead, some methods [10], [13], [36] perform the point-wise detection on the point-wise features provided by the PointNet++ [16] or graph neural networks [37]. The point-wise point cloud features can also be augmented with the camera image features by projecting the points onto the image plane [38], [39]. To improve the orientation coverage of the cubic anchor, a novel spherical anchor for point cloud space is proposed in STD [11], but the box recall is still lagging behind that of the map-view RPN in the voxel-based methods [14], [15].

Different from the above methods, we propose a two-stage network that employs the map-view RPN and the point-based detection refinement stage, which takes advantage of the structured voxel-based and the accurate point-based point cloud representations.

III. METHOD

As illustrated in Fig. 2, we propose a two-stage 3D object detection framework that consists of three parts: 1) a point-to-voxel feature learning from point-wise features to voxel-wise features; 2) a convolutional map-view RPN that generates 3D proposals; 3) a detection refinement stage that leverages the point-wise features to learn the perspective-invariant proposal features for refining the detection results from the RPN.

A. POINT-TO-VOXEL FEATURE LEARNING

1) POINT-WISE FEATURE LEARNING

To retain the information of the raw point cloud, we learn the high-level semantic features for each point via an encoder-decoder structure network PointNet++ [16]. Given an input point cloud $\mathcal{P}^{\text{in}}$ within the range of $P_{\min} = (X_{\min}, Y_{\min}, Z_{\min})$ and $P_{\max} = (X_{\max}, Y_{\max}, Z_{\max})$, the output point cloud $\mathcal{P}^{\text{sem}}$ with the point-wise semantic feature $f^{\text{sem}} \in \mathbb{R}^{1\times C^{\text{sem}}}$ can be denoted as

$$\mathcal{P}^{\text{in}} = \left\{(p_i, f_i^{\text{in}}) : i = 1, \ldots, N^{\text{p}}\right\}, \tag{1}$$

$$\mathcal{P}^{\text{sem}} = \left\{(p_i, f_i^{\text{sem}}) : i = 1, \ldots, N^{\text{p}}\right\}, \tag{2}$$

where $p \in \mathbb{R}^{1\times 3}$ denotes the floating-point 3D coordinates (x, y, z) of the raw point and the initial point-wise local spatial feature $f^{\text{in}} = (x, y, z, r)$ is adopted from the point-wise 3D coordinates and reflection intensity r.

With the point-wise semantic features in $\mathcal{P}^{\text{sem}}$, the features for 3D object detection task can be enhanced by the related semantic segmentation task in a co-training network [9], [10]. The additional point-wise semantic segmentation supervision can enhance the feature learning by guiding the network to understand the content of the point cloud and focus on the structure information of objects against the background. Thus, we integrate the semantic supervision into our method, which classifies the corresponding point as the foreground point or the background point by predicting a point-wise segmentation score s as

$$\mathcal{P}^{\text{seg}} = \left\{(p_i, s_i) : i = 1, \ldots, N^{\text{p}}\right\}. \tag{3}$$

Since the foreground points on the objects of each category are less than the background points, especially for the large-scale outdoor scenes, the points on objects of all the categories are regarded as the foreground points to ease such an imbalance. The category agnostic segmentation head follows the semantic feature f^{sem}, implemented by a 1-D convolutional block for feature embedding and another one followed by a sigmoid function for segmentation score $s \in [0, 1]$ output. The binary point-wise labels can be generated by determining whether the point is within the annotated box or not. Although all the points on objects of different categories are treated as the foreground points, it is still much less than the background points because most of the elements in a scene are backgrounds such as roads and plants. Hence, we adopt the focal loss [40] with the default settings as the segmentation loss $\mathcal{L}_{\text{seg}}$ to deal with the data imbalance.

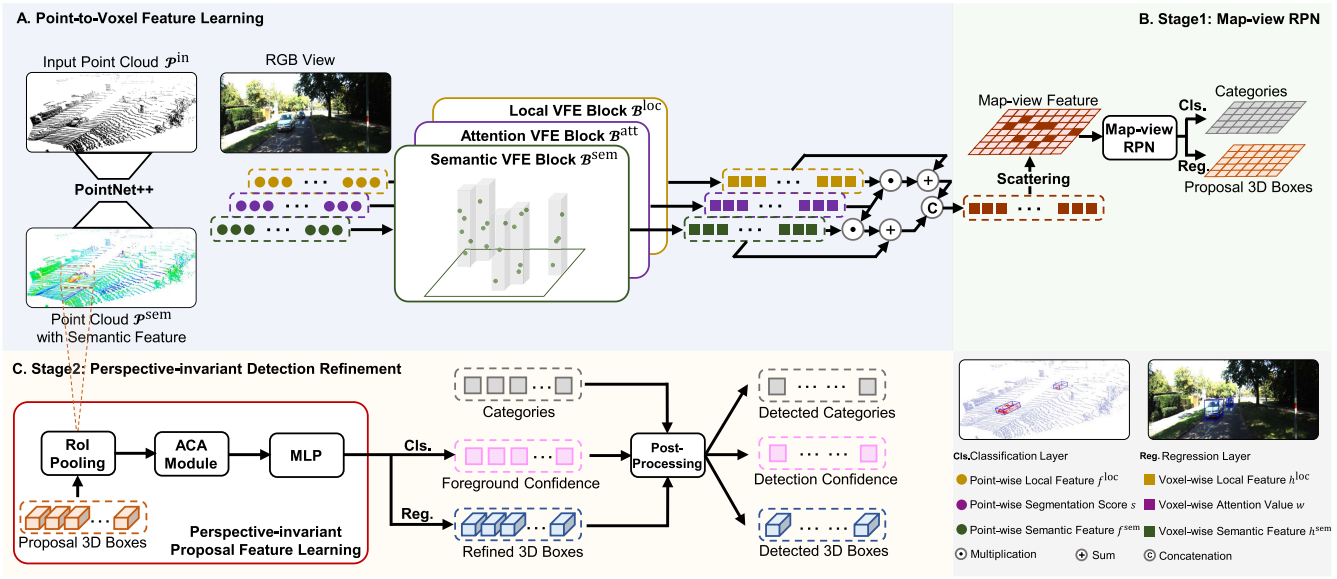


FIGURE 2. Framework of proposed 3D object detection network. Only the point cloud is used as the input, and the RGB image visualizes the scene. The figure is best viewed in color. The feature channels are not shown in this figure for clarity.

2) VOXEL-WISE FEATURE LEARNING

The voxel based methods [5], [7], [9], [14] directly voxelize the original point cloud into a sparse voxel representation by a quantization step $d = (d_x, d_y, d_z)$ as

$$\left\{\bar{p}_i = \lfloor \frac{p_i - P_{\min}}{d} \rfloor : i = 1, \ldots, N^{\mathrm{p}}\right\}, \tag{4}$$

where the unique integer values of $\bar{p}$ are set as the voxel indices v. A Voxel Feature Encoding (VFE) block [5], [8] is then defined on the point set $\mathcal{N}$ in the same voxel to obtain the voxel-wise feature h:

$$\left\{(v_i, h_i = \mathcal{G}(\mathcal{T}(\mathcal{N}_i))) : i = 1, \ldots, N^{\mathrm{v}}\right\}, \tag{5}$$

where the item (v, h) indicates the non-empty voxel located at v with the voxel-wise feature h. The feature transformation function $\mathcal{T}$ is responsible for projecting features to the high-dimensional space, and it is usually implemented by a shared Multi-Layer Perceptron (MLP) network. The symmetric function $\mathcal{G}$ with the permutation invariance [16] aggregates the point-wise features in $\mathcal{N}$ to the voxel-wise feature $h \in \mathbb{R}^{1\times C}$, which can be implemented by the max pooling Max(·) or average pooling Avg(·) along the point-axis. The reconstructed voxel tensor can be easily applied with the 3D sparse convolutional [5], [7] or 2D convlutional [8] processing.

The vanilla VFE block $\mathcal{B}^{loc}$ in [5] and [8] is performed on the input point cloud $\mathcal{P}^{in}$ with the initial point-wise local spatial features f^{in} and configured with the Max(·) for capturing the most discriminative features as $h^{loc} \in \mathbb{R}^{1\times C^{sem}}$. As the receptive fields gradually increase and the hidden space nonlinear transformations become complex, the high-level semantic features in the deeper layers of the neural network are more expressive than the raw data [10], [41]–[43]. The point-wise semantic features with 3D structure information are effectively supplement to the local spatial features. Thus, another semantic VFE block $\mathcal{B}^{sem}$ is applied on the $\mathcal{P}^{sem}$ for obaining the voxel-wise semantic feature $h^{sem} \in \mathbb{R}^{1\times C^{sem}}$. The feature transformation function $\mathcal{T}$ is not configured in the $\mathcal{B}^{sem}$ since the f^{sem} is already of the C^{sem} channels.

Besides, a point p with a larger segmentation score s has a greater probability of belonging to a foreground object. Such a point is supposed to contribute more to feature learning than that from the background regions. The point feature can be re-weighted by the segmentation score in the point feature learning [14]. Inspired by this, we employ another attention VFE block $\mathcal{B}^{att}$ on the $\mathcal{P}^{att}$ to encode the point-wise segmentation score s as the voxel-wise attention w for re-weighting both the voxel-wise h^{loc} and h^{sem} as

$$h^{\mathrm{loc'}} = (1 + w)h^{\mathrm{loc}}, \tag{6}$$

$$h^{\mathrm{sem'}} = (1 + w)h^{\mathrm{sem}}, \tag{7}$$

where 1 denotes the identity path in the residual feature learning [41] to avoid the background regions being completely suppressed to 0. The attention value is expected to be a scalar, so the feature transformation function $\mathcal{T}$ is not configured in the $\mathcal{B}^{att}$.

The enhanced voxel-wise local feature $h^{\mathrm{loc'}}$ and semantic feature $h^{\mathrm{sem'}}$ are concatenated along the channel-axis as the output voxel-wise feature h^{att}, which can be denoted as:

$$\mathcal{V} = \left\{(v_i, h_i^{\mathrm{att}} = h_i^{\mathrm{loc'}} \copyright h_i^{\mathrm{sem'}}) : i = 1, \ldots, N^{\mathrm{v}}\right\}, \tag{8}$$

where "©" indicates the concatenation. Another MLP is applied to transform the feature channels from $2C^{sem}$ to C^{sem} for saving computation. As our voxel-wise semantic feature h^{sem} is of 3D structural information, the expensive 3D sparse

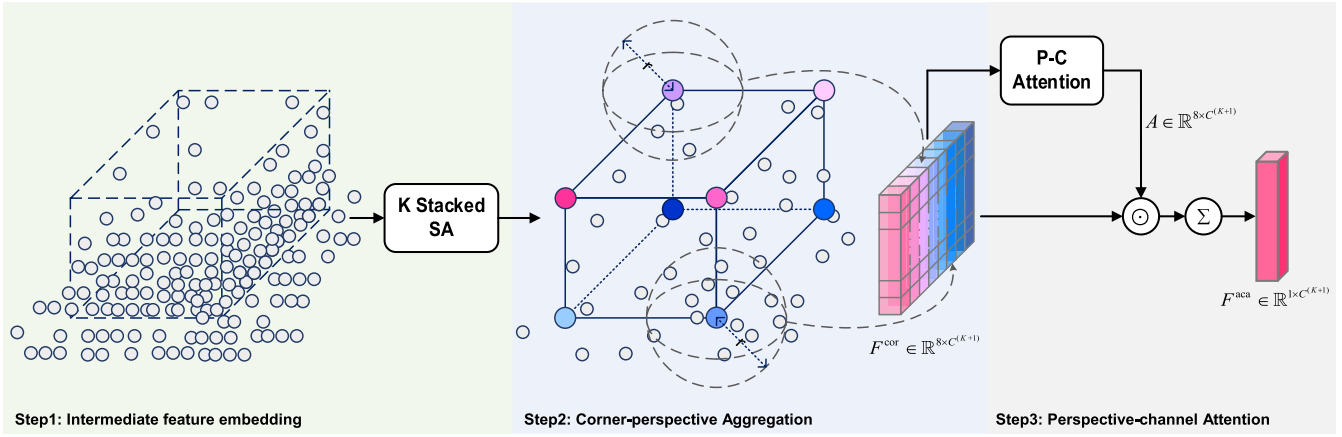


FIGURE 3. Illustration of the Attentive Corner Aggregation module. The eight corners are colored differently in the Step2. The $\odot$ and the $\sum$ in the Step3 denote the element-wise multiplication and the sum over eight corners, respectively.

convolutions are no longer necessary. In practice, the quantization step d_z is set as $Z_{\max} - Z_{\min}$ to ignore the voxelization along the Z-axis for subsequent 2D convolutional processing following PointPillars [8].

B. MAP-VIEW REGION PROPOSAL NETWORK

The voxel features in $\mathcal{V}$ are still in the form of a sequence. We initialize an all-zero tensor M with shape (C^{sem}, H, W), and then scatter the voxel-wise feature h^{att} to the corresponding position v in M to form the BEV map-view feature map following [8] as $M\left[:, v_y, v_x\right] = h^{\text{att}}$. The H and W can be decided as the $\lfloor \frac{Y_{\max}-Y_{\min}}{d_y} \rfloor$ and $\lfloor \frac{X_{\max}-X_{\min}}{d_x} \rfloor$, repectively.

As illustrated in the second part of Fig. 2, the RPN is built on the map-view feature M in the manner of the most popular voxel-based methods [7], [8], [14], which classifies and locates the objects as the 3D proposals. The feature map M is sequentially down-sampled $\times 2$, $\times 4$, and $\times 8$ times for expanding the receptive fields, passing through three convolutional blocks with stride of 2. To maintain sufficient feature representation and the resolution for small objects like cyclists and pedestrians, the down-sampled feature maps are further up-sampled to the desired resolution of $\times 2$ down-sampling time via deconvolutional blocks and concatenated together as M'. Two 1×1 conolutional layers take the M' as the input: one for classification, and the other one for regressing the residuals between the dense pre-defined anchors and the ground truth boxes.

C. PERSPECTIVE-INVARIANT DETECTION REFINEMENT NETWORK

The detection refinement stage can significantly improve the detection results in both 2D and 3D object detection [10], [11], [14], [43], [44]. As shown in the third part of Fig. 2, we first learn the perspective-invariant feature of each proposal, which is followed by two sets of fully-connected layers for further classifying whether it is a foreground object and narrowing the distance between the bounding boxes and the ground truth, respectively. The detection results with foreground confidence lower than the threshold θ_{cls} can be filtered as the negative results. The final output is obtained by suppressing overlapping bounding boxes via the standard Non-Maximum Suppression (NMS) post-processing in [45]. The following subsections describe the main perspective-invariant feature learning in detail.

1) ROI POOLING

We leverage the Region of Interest (RoI) pooling operation [10] to crop a local point cloud $\mathcal{P}^{\text{pro}}$ surrounding it as

$$\mathcal{P}^{\text{pro}} = \left\{(\tilde{p}_i, f_i^{\text{pro}}) : i = 1, \ldots, N^{\text{pro}}\right\}, \tag{9}$$

where N^{pro} is the number of the cropped points for a proposal. Each cropped point is normalized to the relative coordinate system (marked with " ˜ ") centered on the proposal center and aligned to the proposal rotation angle, which makes it more robust to the various rigid transformations. The point-wise feature $f^{\text{pro}} \in \mathbb{R}^{1\times C^{(0)}}$ can be formulated as that in [10]:

$$f^{\text{pro'}} = \text{MLP}_1(\tilde{x} \copyright \tilde{y} \copyright \tilde{z} \copyright r \copyright s \copyright d), \tag{10}$$

$$f^{\text{pro}} = \text{MLP}_2(f^{\text{pro'}} \copyright f^{\text{sem}}), \tag{11}$$

where d denotes the normalized depth. The f^{pro} combines both the high-dimensional semantic feature and low-dimensional spatial feature.

2) ATTENTIVE CORNER AGGREGATION

Given a cropped local point cloud $\mathcal{P}^{\text{pro}}$, the ACA module extracts the perspective-invariant proposal feature in three steps as illustrated in Fig. 3.

Step 1: **Intermediate feature embedding**. The $\mathcal{P}^{\text{pro}}$ can be regarded as a set of points with point-wise features. As a widely used operator on point sets [10], [11], [14], [46], [47], the Set Abstraction (SA) operation [16] is K levels stacked to learn the higher-dimensional intermediate features by

repeating the hierarchical down-sampling and the neighborhood aggregation.

In the k-th SA operation, the points in the intermediate point cloud $\mathcal{P}^{(\text{pro},k)}$ are set as the aggregation centroids, which are sampled from the previous points in $\mathcal{P}^{(\text{pro},k-1)}$ by the Furthest-Point-Sampling (FPS) algorithm [16]. For each aggregation centroids $\tilde{p}^{(k)}$, the features of m_k neighbors within radius r_k are grouped into the set $\mathcal{N}^{(\text{pro},k)}$ as

$$\mathcal{N}_i^{(\text{pro},k)} = \left\{ (\tilde{p}_j^{(k-1)} - \tilde{p}_i^{(k)}) \copyright f_j^{(\text{pro},k-1)} \right\}, \tag{12}$$

where $\left\| \tilde{p}_j^{(k-1)} - \tilde{p}_i^{(k)} \right\| < r^{(k)}$ and $(\tilde{p}_j^{(k-1)}, f_j^{(\text{pro},k-1)}) \in \mathcal{P}^{(\text{pro},k-1)}$. An $\text{MLP}_3^{(k)}$ is applied on each $\mathcal{N}^{(\text{pro},k)}$ to transform the features from $C^{(k-1)}$to $C^{(k)}$ channels as Eq. 13. A max pooling along the point-axis to aggregate the transformed features to point $\tilde{p}^{(k)}$ as

$$f_i^{(\text{pro},k)} = \text{Max}(\text{MLP}_3^{(k)}(\mathcal{N}_i^{(\text{pro},k)})) \in \mathbb{R}^{1\times C^{(k)}}, \tag{13}$$

$$\mathcal{P}^{(\text{pro},k)} = \left\{ (\tilde{p}_i^{(k)}, f_i^{(\text{pro},k)}) : i = 1, \ldots, N^{(\text{pro},k)} \right\}. \tag{14}$$

We set the multiple group radii in the SA operation for multi-scale information aggregation [16] in practice, which concatenates the aggregated features of each scale together.

Step 2: **Corner-perspective aggregation**. As illustrated in Fig. 3, we aggregate the intermediate local point cloud $\mathcal{P}^{(\text{pro},k)}$ with the eight corners of the proposal 3D bounding box, which performs like a $(K+1)$-th SA operation. For details, the proposal bounding box is decoded into eight corners like $(\pm l/2, \pm w/2, \pm h/2)$ in the relative coordinate system, where (l, w, h) represents the size of the proposal bounding box. We directly set these eight corners as the aggregation centroids $\tilde{p}^{(K+1)}$ instead of using eight sampled points by FPS, then also perform the Eq. 12 - 14 for aggregating the features in $\mathcal{P}^{(\text{pro},k)}$ to eight corner-perspectives as

$$F^{\text{cor}} = \left[f_1^{(\text{pro},K+1)}, \ldots, f_8^{(\text{pro},K+1)} \right] \in \mathbb{R}^{8\times C^{(K+1)}}. \tag{15}$$

Note that there are two main differences between the corner-perspective aggregation and the previous K stacked SA operations: (1) The eight aggregation centriods in the corner-perspective aggregation are not sampled from the local point cloud. So this process cannot be affected when the points at the eight centroids are missing in the local point cloud $\mathcal{P}^{\text{pro}}$. (2) The eight centroids are fixed patterns as defined across all the object instances, which is robust to the various point distributions.

Step 3: **Perspective-channel attention**. Based on the observation that the number of points near each corner is different, the eight corner-perspectives should contribute differently. Inspired by the attention on point cloud voxels [33], the proposed perspective-channel attention adaptively adjusts the contributions via re-weighting the sub-features in F^{cor} by the perspective-wise and channel-wise attentions. As illustrated in Fig. 4, the perspective-channel attention is dynamically generated from F^{cor} according to its internal responses.

For the perspective-wise attention, we first exploit a max pooling operation along the perspective-axis to obtain

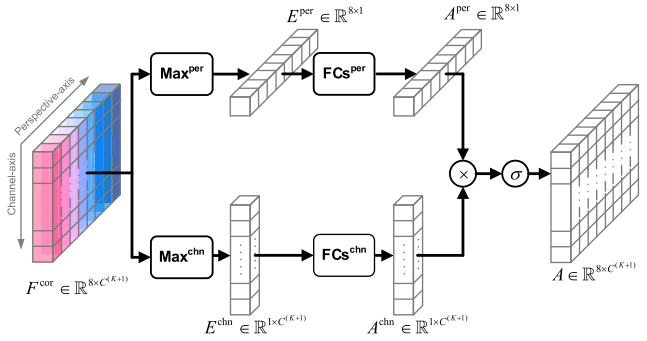


FIGURE 4. Illustration of the perspective-channel attention. Superscript $^{\text{per}}$ and superscript $^{\text{chn}}$ represent perspective-axis and channel-axis, respectively. The $\times$ and the σ denote the matrix multiplication and the sigmoid function, respectively.

the expressive perspective-wise responses $E^{\text{per}} \in \mathbb{R}^{8\times 1}$. To explore the different contributions of perspectives, two fully connected layers with weight parameters $W_1^{\text{per}} \in \mathbb{R}^{\frac{8}{r}\times 8}$ and $W_2^{\text{per}} \in \mathbb{R}^{8\times\frac{8}{r}}$ are utilized to learn the perspective-wise attention $A^{\text{per}} \in \mathbb{R}^{8\times 1}$ as

$$A^{\text{per}} = W_2^{\text{per}} \delta(W_1^{\text{per}} E^{\text{per}}), \tag{16}$$

where δ denotes the ReLU activation function. Similarly, the channel-wise attention $A^{\text{chn}} \in \mathbb{R}^{1\times C_{K+1}}$ can also be obtained by another two fully connected layers to strengthen the important channels as

$$A^{\text{chn}^T} = W_2^{\text{chn}} \delta(W_1^{\text{chn}} E^{\text{chn}^T}), \tag{17}$$

where the expressive channel-wise responses $E^{\text{chn}} \in \mathbb{R}^{1\times C_{K+1}}$ are aggregated by a max pooling operation along the channel-axis. In practice, the reduction ratio r is set as 1.

The perspective-channel attention matrix $A \in \mathbb{R}^{8\times C^{(K+1)}}$ combines the perspective-wise attention A^{per} and the channel-wise attention A^{chn} together through a matrix multiplication as

$$A = \sigma(A^{\text{per}} \times A^{\text{chn}}), \tag{18}$$

where the attention values are normalized to [0, 1] by a sigmoid function $\sigma(\cdot)$. Thus, the re-weighted feature $F^{\text{cor}'} \in \mathbb{R}^{8\times C^{(K+1)}}$ can be obtained by element-wise multiplication ($\odot$) with F^{cor} as

$$F^{\text{cor}'} = A \odot F^{\text{cor}}, \tag{19}$$

where A enhances the important information across the perspective-wise and channel-wise dimensions. We then take the re-weighted eight sub-features summation as the output of the ACA module as

$$F^{\text{aca}} = \sum_{i=1}^{8} f_i^{\text{cor}'}. \tag{20}$$

The final perspective-invariant feature F^{inv} for each proposal is further embedded from F^{aca} by MLP for channel transformation, which is set as the input of the foreground

confidence estimation and the bounding box refinement regression heads.

In [14], a small number of keypoints are sampled from the input point cloud by FPS to aggregate the coarse voxel features for RoI-grid pooling. The object becomes much more coarse because only a few sampled keypoints fall around the object. Our method directly learns the accurate point-wise feature for each point, which naturally provides more details on the object for detection. Moreover, our ACA module adaptively aggregates the local point cloud to eight corners instead of densely arranged grid points [14] or voxels [15] inside a proposal box. It reasonably follows the fact that the LiDAR points are only distributed on the surface rather than the inside of the object.

D. LOSS FUNCTION

The loss $\mathcal{L}_{\text{total}}$ is composed of the point cloud segmentation loss $\mathcal{L}_{\text{seg}}$, RPN loss $\mathcal{L}_{\text{rpn}}$ and detection refinement loss $\mathcal{L}_{\text{refine}}$ as

$$\mathcal{L}_{\text{total}} = \alpha_1 \mathcal{L}_{\text{seg}} + \alpha_2 \mathcal{L}_{\text{rpn}} + \alpha_3 \mathcal{L}_{\text{refine}}, \tag{21}$$

where α_1, α_2 and α_3 are empirically set to 4.0, 1.0 and 1.0 to balance each loss term of multi-task learning. The segmentation loss $\mathcal{L}_{\text{seg}}$ is based on the focal loss $\mathcal{L}_{\text{focal}}$ with default settings [40] as

$$\mathcal{L}_{\text{seg}} = \frac{1}{N^{(s^*>0)}} \sum_{i}^{N^{\text{p}}} \mathcal{L}_{\text{focal}} \left(s_i, s_i^* \right), \tag{22}$$

where s_i and s_i^* are the prediction and label of the point p_i in $\mathcal{P}^{\text{in}}$, $N^{(s^*>0)}$ denotes the number of foreground points with $s^* > 0$. The RPN loss $\mathcal{L}_{\text{rpn}}$ includes a classification loss for classifying the anchors into the required categories and a box regression loss for predicting the residuals between the anchor and ground truth boxes. The detection refinement loss $\mathcal{L}_{\text{refine}}$ is also composed of a classification loss for classifying the 3D proposals and a box regression loss for predicting the residuals between the proposal and ground truth boxes. Here we directly adopt the widely used RPN loss function in [7] and [8] and detection refinement loss function in [14] and [15].

IV. EXPERIMENT

In this section, we describe the experimental settings and compare our method with state-of-the-art methods on the widely used KITTI dataset [1]. The comprehensive ablation study is conducted to validate the effectiveness of each individual module.

A. EXPERIMENTAL SETUP

1) DATASET

The KITTI dataset [1] includes 7481 training images/point clouds and 7518 testing images/point clouds. It annotates the cars, cyclists and pedestrians in the camera Field of Vision (FOV) with the 3D bounding boxes. For each category, three difficulty levels are involved (Easy, Moderate and Hard), which depend on the size, occlusion level and truncation of 3D objects. Detection performance is generally compared following the official KITTI evaluation metric, the average precision with 40 recall positions on the 3D detection. The training samples are provided with labels, while the results on the *test* set must be submitted to the official test server [19] for evaluation. Since the ground truth for the test samples is not available, training samples are generally divided into the *train* split set (3712) and the *val* split set (3769) for training and validation [7], [10], [14], [20].

2) IMPLEMENTATION DETAILS

a: NETWORK SETTINGS

As the default settings of PointPillars [8] in [45], the point cloud range is set as $P_{\text{min}} = (0.00, -39.68, -3.00)$ meters and $P_{\text{max}} = (69.12, 39.68, 1.00)$ meters due to the FOV annotations. We follow [10] to devise the PointNet++ network to learn the point-wise semantic features of $C^{\text{sem}} = 128$ channels in $\mathcal{P}^{\text{sem}}$. The quantization step d is set as $(0.16, 0.16, 4.00)$ meters to voxelize the point cloud $\mathcal{P}^{\text{sem}}$, resulting in a map-view feature M of spatial resolution (H, W) as $(432, 496)$.

In the RPN stage, different categories employ different anchor sizes (l, w, h) of $(3.90, 1.60, 1.56)$ meters, $(1.76, 0.6, 1.73)$ meters, $(0.8, 0.6, 1.73)$ meters for the car, the cyclist, and the pedestrian, respectively. Each anchor has two directions in $\{0°, 90°\}$, which means that each location in the feature map M' has six anchors.

For the detection refinement stage, only 1 level SA operation is stacked for the intermediate feature embedding, which is mentioned in Sec. III-C2. The configuration details are listed in Tab. 1.

TABLE 1. Configuration details of the RoI pooling and the ACA module.

	$\mathcal{P}^{\text{sem}} \xrightarrow[\text{Pooling}]{\text{RoI}} \mathcal{P}^{\text{pro}}$	$\mathcal{P}^{\text{pro}} \xrightarrow{\text{SA}} \mathcal{P}^{(\text{pro},1)}$	$\mathcal{P}^{(\text{pro},1)} \xrightarrow{\text{SA}} \mathcal{P}^{(\text{pro},2)}$
# Points	512	256	8
# Input Channels	128	128	256
# Output Channels	128	128+128	256+256
radii	-	(0.4, 0.8) meters	(1.2, 1.6) meters
# Neighbors	-	(16, 32)	(16, 32)

b: TRAINING

Our model is trained from scratch in an end-to-end manner with the AdamW optimizer [48] and one-cycle policy [49] with LR 0.01, division factor 10, momentum ranges from 0.95 to 0.85, weight decay 0.01. A batch of 8 random point cloud samples is trained on 4 GeForce RTX 3090 GPUs with 80 epochs. To avoid overfitting, we employ four commonly used data augmentation strategies: ground truth sampling [7], random flipping along the X-axis, global scaling with a random scaling factor in $[0.95, 1.05]$, global rotation around the Z-axis with a random angle in $[-\frac{\pi}{4}, \frac{\pi}{4}]$. Please refer to the open source toolbox OpenPCDet [45] for more detailed training configurations since we conduct all experiments with it.

TABLE 2. Performance comparison with the state-of-the-art methods for car category on the KITTI *test* set. The top-2 results are in bold.

Method	Reference	Modality	3D AP		
			Easy	Mod.	Hard
MV3D [20]	CVPR 2017	RGB+LiDAR	74.97	63.63	54.00
AVOD-FPN [21]	IROS 2018	RGB+LiDAR	83.07	71.76	65.73
Conti-Fuse [24]	ECCV 2018	RGB+LiDAR	83.68	68.78	51.67
F-PointNet [34]	CVPR 2018	RGB+LiDAR	82.19	69.79	60.59
PointPainting [39]	CVPR 2020	RBG+LiDAR	82.11	71.70	67.08
PI-RCNN [38]	AAAI 2020	RGB+LiDAR	84.37	74.82	70.03
PFF3D [32]	Access 2021	RGB+LiDAR	81.11	72.93	67.24
SECOND [7]	Sensors 2018	LiDAR	83.34	72.55	65.82
PointPillars [8]	CVPR 2019	LiDAR	82.58	74.31	68.99
SCNet [6]	Access 2019	LiDAR	83.34	73.17	67.93
3D IoU Loss [50]	3DV 2019	LiDAR	86.16	76.50	71.39
PointRCNN [10]	CVPR 2019	LiDAR	86.96	75.64	70.70
Fast PointRCNN [51]	ICCV 2019	LiDAR	85.29	77.40	70.24
STD [11]	ICCV 2019	LiDAR	87.95	**79.71**	**75.09**
TANet [33]	AAAI 2020	LiDAR	84.39	75.94	68.82
Part-A2 [15]	TPAMI 2020	LiDAR	87.81	78.49	73.51
HotSpotNet [52]	ECCV 2020	LiDAR	87.60	78.31	73.34
Associate-3DDet [53]	CVPR 2020	LiDAR	85.99	77.40	70.24
Point-GNN [36]	CVPR 2020	LiDAR	88.33	79.47	72.29
3DSSD [13]	CVPR 2020	LiDAR	**88.36**	79.57	74.55
Ours	-	LiDAR	**88.34**	**81.45**	**77.20**

TABLE 3. Performance comparison for the categories of cyclist and pedestrian on the KITTI *test* set. The top-2 results are in bold.

Method	Cyclist			Pedestrian		
	Easy	Mod.	Hard	Easy	Mod.	Hard
AVOD-FPN [21]	63.76	50.55	44.93	50.46	42.27	39.04
PFF3D [32]	63.27	46.78	41.37	48.75	40.99	38.99
BirdNet+ [54]	67.38	47.72	42.89	37.99	31.46	29.46
SCNet [6]	67.98	50.79	45.15	47.83	38.66	35.70
F-PointNet [34]	72.27	56.12	49.01	50.53	42.15	38.08
PointRCNN [10]	74.96	58.82	52.53	47.98	39.37	36.01
PointPillars [8]	77.10	58.65	51.92	**51.45**	41.92	38.89
STD [11]	**78.69**	61.59	55.30	**53.29**	**42.47**	38.35
PointPainting [39]	77.63	**63.18**	**55.89**	50.32	40.97	37.87
SemanticVoxels [55]	N/A	N/A	N/A	50.90	42.19	**39.52**
Ours	**78.62**	**63.13**	**56.81**	50.91	**43.19**	**40.81**

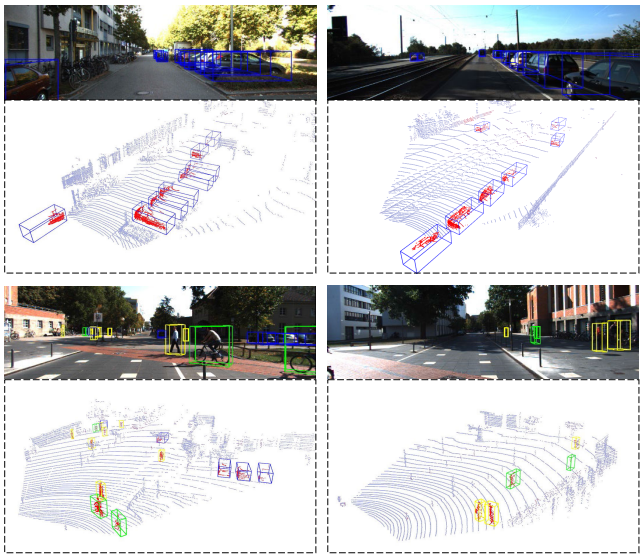


FIGURE 5. Qualitative results on KITTI *test* set. Detected cars, cyclists and pedestrians with blue, green and yellow boxes, respectively. The point cloud is segmented as foreground points in red and background points in grey.

c: INFERENCE

We first perform the NMS on the RPN proposals with IoU threshold 0.85 to take the top-100 proposals as the input of the detection refinement stage. After the refinement of the top-100 proposals, we ignore the negative detection results with their classification score lower than the threshold θ_{cls} of 0.3, and remove the redundant boxes by the NMS with IoU threshold 0.1.

B. MAIN RESULT

Tab. 2 and Tab. 3 compare our method with the most existing methods on the KITTI *test* set. For the mainly considered car category shown in Tab. 2, our method achieves state-of-the-art performance of across the multi-modality methods and LiDAR only methods. Compared with the point-based methods like PointRCNN [10], STD [11], Point-GNN [36] and 3DSSD [13], our method makes performance margins of +1.74% and +2.11% with the strong two-stage method STD [11] especially on the moderate and hard levels, which benefits from the improvements on the higher box recall of the map-view RPN and the more robust perspective-invariant feature learning in the detection refinement stage. Our method also has significant improvements of +0.53%, +2.96% and +3.69% when compared to the pure voxel-based two-stage method Part-A2 [15] on three levels, indicating that our point-wise feature learning exploits more pose details preserved in the points. Tab. 3 shows that our method also performs well on the categories of cyclist and pedestrian. Most methods do not compare the results for the cyclists and pedestrians due to the less object instances for stable training and the difficulty of detecting small objects. As shown in Fig. 5, our method detects objects of the three classes reasonably well in different scenarios. The results of our method can be retrieved on the KITTI benchmark [19] with the submission of "P2V-RCNN".

C. ABLATION STUDY

In the ablation study, all models are trained with the same training settings on the KITTI *train* split set and evaluated with 3D AP from 40 recall positions for the car category on the KITTI *val* split set following [9], [13], [14], [36]. The best results are in bold in tables.

1) ANALYSIS OF POINT-TO-VOXEL FEATURE LEARNING

Tab. 4 verifies the effects of three VFE blocks in our point-to-voxel feature learning. Three sets of experiments are conducted to analyze our design choices and show continuous performance improvements. To exclude the effects of detection refinement stage, we construct all the models in Tab. 4 with the single-stage architecture, and evaluate the 3D AP on the output of RPN as the performance metric. According to PointPillars [8], the local VFE with the symmetric function $\mathcal{G}$ of Max(·) works as the baseline, denoted as the model A in the 1^{st} row. The models B1 and B2 show that the semantic VFE block $\mathcal{B}^{\text{sem}}$ can significantly improve the 3D AP,

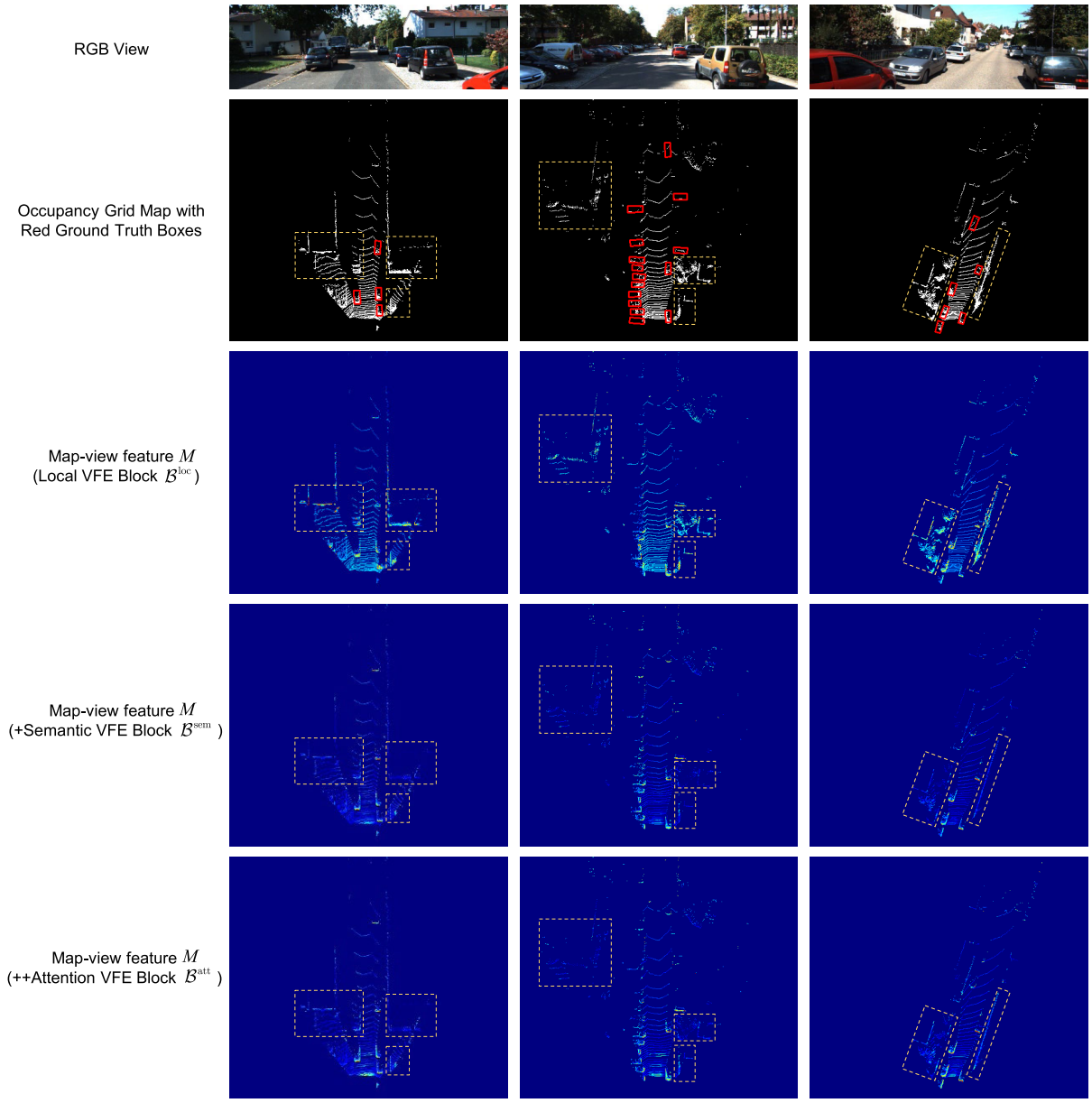


FIGURE 6. **Visualization of point-to-voxel feature learning. The black pixels in the occupancy grid map represent the empty voxels. The yellow dashed box emphasizes the main differences in the map-view feature map *M* with different VFE blocks.**

which reflects the importance of high-dimensional semantic features. The effect of the symmetric function $\mathcal{G}$ in $\mathcal{B}^{\text{sem}}$ is not obvious. As shown in the model C3, the performance still can be improved when the attention VFE block $\mathcal{B}^{\text{att}}$ with appropriate settings is added. Thus, we finally configure three VFE blocks with the settings of the model C3 in our network.

Fig. 6 shows the map-view feature M in the models A, B2, and C3 to illustrate how the $\mathcal{B}^{\text{sem}}$ and $\mathcal{B}^{\text{att}}$ affect the network further. The comparison between the 3^{rd} row and the last two rows in Fig. 6 clearly shows that the noisy background regions can be suppressed to highlight the regions of objects when the $\mathcal{B}^{\text{sem}}$ and $\mathcal{B}^{\text{att}}$ are added.

2) EFFECT OF DETECTION REFINEMENT STAGE

The effect of the detection refinement stage is further explored in Tab. 5. The 1^{st} row denotes the best single-stage model C3 in Tab. 4 while the 2^{nd} row represents our full model including the detection refinement stage. The significant performance gains of $+2.17\%$, $+3.76\%$, and $+4.57\%$ on three levels indicate that the detection refinement stage further improves the overall detection performance.

3) ANALYSIS OF COMPONENTS OF ACA MODULE

Tab. 6 shows the effectiveness of the ACA module that can be decomposed into the corner-perspective aggregation and perspective-channel attention. As shown in the comparison

TABLE 4. Performance comparison of single-stage models with different VFE blocks. "W./W.O." and "Sym. Func." denote "With/Without" and "Symmetric Function", respectively.

Model	Local VFE Block $\mathcal{B}^{loc}$		Semantic VFE Block $\mathcal{B}^{sem}$		Attention VFE Block $\mathcal{B}^{att}$		3D AP		
	W.($\surd$) /W.O.($\times$)	Sym. Func. $\mathcal{G}$	W.($\surd$) /W.O.($\times$)	Sym. Func. $\mathcal{G}$	W.($\surd$) /W.O.($\times$)	Sym. Func. $\mathcal{G}$	Easy	Mod.	Hard
A [8]	$\surd$	Max	$\times$	-	$\times$	-	87.51	78.54	75.75
B1	$\surd$	Max	$\surd$	Avg	$\times$	-	89.51	81.12	78.28
B2	$\surd$	Max	$\surd$	Max	$\times$	-	89.91	81.11	78.15
C1	$\surd$	Max	$\surd$	Avg	$\surd$	Avg	89.60	80.88	78.05
C2	$\surd$	Max	$\surd$	Avg	$\surd$	Max	89.02	80.18	77.62
C3	$\surd$	Max	$\surd$	Max	$\surd$	Avg	**90.41**	**81.34**	**78.38**
C4	$\surd$	Max	$\surd$	Max	$\surd$	Max	89.45	80.81	78.18

TABLE 5. Performance comparison of single-stage and two-stage models.

Stage1	Stage2	3D AP		
		Easy	Mod.	Hard
$\surd$	$\times$	90.41	81.34	78.38
$\surd$	$\surd$	**92.58**	**85.10**	**82.95**

TABLE 6. Effects of individual components of ACA module.

Aggregation	Merge	3D AP		
		Easy	Mod.	Hard
Random	Mean	91.18	82.44	82.21
Corners	Mean	92.42	85.05	82.94
Random	Attentive	91.47	82.42	82.30
Corners	Attentive	**92.58**	**85.10**	**82.95**

of the 1^{st} and 2^{nd} rows and the comparison of the 3^{rd} and 4^{th} rows, the performance drops a lot when replacing eight corners with eight random points sampled by FPS for aggregation, which validates that the proposed corner-perspective aggregation is more robust to capture the full view of an object in the point cloud scenes. Moreover, compared with treating each corner and channel equally in the 2^{nd} row, the performance also increases when the aggregated features are adaptively re-weighted by the perspective-channel attention in the 4^{th} row. The different distribution of points makes them contribute differently.

V. CONCLUSION

In this paper, we present a novel 3D object detection network for point clouds with the map-view RPN stage and point-based detection refinement stage. The proposed point-to-voxel feature learning combines both the point-wise and voxel-wise features rather than only one of them used in most of the existing methods. The proposed ACA module improves the quality of proposal-wise features in the detection refinement stage for perspective-invariant feature learning. Experimental results on the widely used KITTI dataset shows that our methods outperforms other state-of-the-art methods. The comprehensive ablation study indicates that each individual component in our method is effective with significant performance gains. In future work, we will exploit the visual clues from camera images to extend our method and achieve more robust performance on the various environmental conditions in the real world.

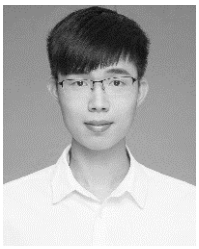
REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Int. Conf. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[2] M. Sualeh and G.-W. Kim, "Visual-LiDAR based 3D object detection and tracking for embedded systems," *IEEE Access*, vol. 8, pp. 156285–156298, 2020.
[3] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 3050–3057.
[4] B. Mahmood, S. Han, and D.-E. Lee, "BIM-based registration and localization of 3D point clouds of indoor scenes using geometric features for augmented reality," *Remote Sens.*, vol. 12, no. 14, p. 2302, Jul. 2020.
[5] 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. (CVPR)*, Jun. 2018, pp. 4490–4499.
[6] Z. Wang, H. Fu, L. Wang, L. Xiao, and B. Dai, "SCNet: Subdivision coding network for object detection based on 3D point cloud," *IEEE Access*, vol. 7, pp. 120449–120462, 2019.
[7] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, pp. 3337–3354, Oct. 2018.
[8] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.
[9] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11870–11879.
[10] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.
[11] 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. (ICCV)*, Oct. 2019, pp. 1951–1960.
[12] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep Hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9276–9285.
[13] 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. (CVPR)*, Jun. 2020, pp. 11037–11045.
[14] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10526–10535.
[15] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2020.
[16] 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. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2017, pp. 5099–5108.
[17] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5588–5597.

[18] X. Liang and Z. Fu, "MHNet: Multiscale hierarchical network for 3D point cloud semantic segmentation," *IEEE Access*, vol. 7, pp. 173999–174012, 2019.
[19] KITTI. (2021). *KITTI Leaderboard of 3D Object Detection Benchmark*. [Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d
[20] 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. (CVPR)*, Jul. 2017, pp. 6526–6534.
[21] 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. (IROS)*, Oct. 2018, pp. 1–8.
[22] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7652–7660.
[23] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3D object detection," in *Proc. 2nd Conf. Robot Learn. (CoRL)*, Dec. 2018, pp. 146–155.
[24] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 641–656.
[25] 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. (CVPR)*, Jun. 2019, pp. 7337–7345.
[26] J. Wang, M. Zhu, D. Sun, B. Wang, W. Gao, and H. Wei, "MCF3D: Multi-stage complementary fusion for multi-sensor 3D object detection," *IEEE Access*, vol. 7, pp. 90801–90814, 2019.
[27] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3D object detection in lidar point clouds," 2019, *arXiv:1910.06528*. [Online]. Available: https://arxiv.org/abs/1910.06528
[28] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-FPN: Multi-scale voxel feature aggregation in 3D object detection from point clouds," *Sensors*, vol. 20, pp. 704–721, Jan. 2020.
[29] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9224–9232.
[30] B. Graham and L. van der Maaten, "Submanifold sparse convolutional networks," 2017, *arXiv:1706.01307*. [Online]. Available: https://arxiv.org/abs/1706.01307
[31] V. A. Sindagi, Y. Zhou, and O. Tuzel, "MVX-Net: Multimodal VoxelNet for 3D object detection," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7276–7282.
[32] L.-H. Wen and K.-H. Jo, "Fast and accurate 3D object detection for Lidar-Camera-based autonomous vehicles using one shared voxel-based backbone," *IEEE Access*, vol. 9, pp. 22080–22089, 2021.
[33] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "TANET: Robust 3D object detection from point clouds with triple attention," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2020, pp. 11677–11684.
[34] 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. (CVPR)*, Jun. 2018, pp. 918–927.
[35] Z. Wang and K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1742–1749.
[36] 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. (CVPR)*, Jun. 2020, pp. 1708–1716.
[37] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[38] L. Xie, C. Xiang, Z. Yu, G. Xu, Z. Yang, D. Cai, and X. He, "PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2020, pp. 12460–12467.
[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. (CVPR)*, Jun. 2020, pp. 4603–4611.
[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[42] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[44] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6154–6162.
[45] (2021). *Openpcdet: An Open-Source Toolbox for 3D Object Detection From Point Clouds*. [Online]. Available: https://github.com/open-mmlab/OpenPCDet
[46] Z. Liu, H. Hu, Y. Cao, Z. Zhang, and X. Tong, "A closer look at local aggregation operators in point cloud analysis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 12368, 2020, pp. 326–342.
[47] M. Xu, Z. Zhou, and Y. Qiao, "Geometry sharing network for 3D point cloud classification and segmentation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2020, pp. 12500–12507.
[48] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," 2017, *arXiv:1711.05101*. [Online]. Available: https://arxiv.org/abs/1711.05101
[49] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Proc. SPIE*, vol. 11006, May 2019, Art. no. 1100612.
[50] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, "IoU loss for 2D/3D object detection," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 85–94.
[51] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9775–9784.
[52] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. L. Yuille, "Object as hotspots: An anchor-free 3D object detection approach via firing of hotspots," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 12366, 2020, pp. 68–84.
[53] L. Du, X. Ye, X. Tan, J. Feng, Z. Xu, E. Ding, and S. Wen, "Associate-3Ddet: Perceptual-to-conceptual association for 3D point cloud object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13326–13335.
[54] A. Barrera, C. Guindel, J. Beltrán, and F. García, "BirdNet+: End-to-end 3D object detection in LIDAR bird's eye view," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.
[55] J. Fei, W. Chen, P. Heidenreich, S. Wirges, and C. Stiller, "SemanticVoxels: Sequential fusion for 3D pedestrian detection using LiDAR point cloud and semantic segmentation," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Sep. 2020, pp. 185–190.

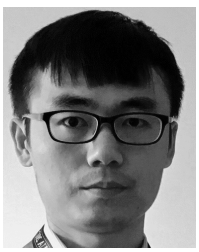
JIALE LI received the B.S. degree from Chongqing University, China, in 2018. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, China. His research interests include 3D object detection, point cloud processing, and computer vision.

YU SUN received the B.S. degree from the Huazhong University of Science and Technology, China, in 2021. He is currently pursuing the M.S. degree with the School of Micro-Nano Electronics, Zhejiang University, China. His research interests include 3D object detection and computer vision.



SHUJIE LUO received the B.S. degree from Zhejiang University, China, in 2018, where he is currently pursuing the M.S. degree with the College of Information Science and Electronic Engineering. His research interests include 3D object detection and computer vision.

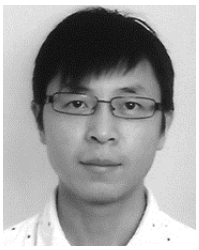
ZIQI ZHU received the B.S. degree from Zhejiang University, China, in 2019, where he is currently pursuing the M.S. degree with the College of Information Science and Electronic Engineering. His research interest includes computer vision.



HANG DAI received the Ph.D. degree from the Department of Computer Science, University of York, U.K., in 2019. He is currently an Assistant Professor with MBZUAI, Abu Dhabi, United Arab Emirates. He was an Oversea Research Scholar with the University of York, from 2015 to 2018. His research interests include medical image analysis, geometric deep learning, and 3D computer vision.

ANDREY S. KRYLOV (Member, IEEE) received the B.S. and Ph.D. degrees in applied mathematics from Lomonosov Moscow State University, Russia, in 1978 and 1983, respectively. Since then, he has been working with the Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. He is currently a Full Professor and the Head of the Laboratory of Mathematical Methods of Image Processing. His research interests include mathematical methods of image processing, computer vision, ill-posed and inverse problems, and mathematical methods in physical chemistry of liquid metals.

YONG DING (Member, IEEE) was born in 1974. He received the Ph.D. degree from the College of Electronic Science and Engineering, Southeast University, Nanjing, China. From 2000 to 2006, he was a Senior Engineer with the Research and Development Center, Hisense Group Company Ltd. From 2006 to 2008, he was a Senior Project Leader at Omnivision Company Ltd. Then he joined the Faculty of Zhejiang University, as a Full Professor, in 2009. He has published over 90 publications in journals and has made several plenary or invited talks on international conferences. Besides, he holds more than 30 Chinese patents. His research interests include image processing and SoC design and verification.



LING SHAO (Fellow, IEEE) is currently the CEO and the Chief Scientist with the Inception Institute of Artificial Intelligence and the EVP and Provost with MBZUAI, Abu Dhabi, United Arab Emirates. He is an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and other journals. His research interests include computer vision, machine learning, and medical imaging. He is a fellow of the IAPR, the IET, and the BCS.