Voxel-RCNN-Complex: An Effective 3-D Point Cloud Object Detector for Complex Traffic Conditions

Hai Wang, *Senior Member, IEEE*, Zhiyu Chen, Yingfeng Cai, *Senior Member, IEEE*, Long Chen, Yicheng Li, Miguel Angel Sotelo, *Fellow, IEEE*, and Zhixiong Li, *Senior Member, IEEE*

***Abstract*— The complex traffic conditions and high traffic flow are big challenges to the perception of autonomous vehicles. As the basis of environmental perception technology, object detection based on point cloud is of great significance for the normal operations of autonomous vehicles. Considering the complex traffic conditions, in this work, we use the One millioN sCenEs (ONCE) dataset to train an effective 3-D object detector, namely Voxel-region convolution neural network (RCNN)-Complex. This is accomplished by modifying the Voxel RCNN to make it suitable for complex traffic conditions. We add the residual structures in the 3-D backbone and design a heavy 3-D feature extractor, which is conducive to extracting high-dimensional information. We also design a 2-D backbone composed of residual structures, self-calibration convolution, and spatial attention and channel attention mechanism; this expands the receptive field and captures more context information. As compared with the Voxel RCNN, the proposed Voxel-RCNN-Complex significantly improves the detection performance for long-distance and small objects. In order to further increase the robustness of the proposed model and alleviate category imbalance, we use a class-balanced sampling strategy (CBSS). We evaluate the proposed model using the ONCE dataset. The results show that the proposed model achieves an mAP of 65.34% and an inference speed of 13.8 FPS. The experiments show that the proposed model performs better than other methods on the ONCE dataset. This demonstrates the effectiveness of the proposed Voxel-RCNN-Complex. Moreover, we also test the proposed model in an intelligent vehicle platform on real roads.**

***Index Terms*— 3-D object detection, complex traffic conditions, Lidar, point cloud.**

I. Introduction

WITH continuous advancements in the development of autonomous vehicles, the perceived requirements of these vehicles regarding the surrounding environment are constantly increasing. It is notable that object detection plays an important role in environmental perception algorithms [1]. Camera and Lidar are common on intelligent vehicles; the comparisons of camera 2-D and Lidar 3-D object detection are shown in Table I. The environmental perception algorithms provide essential environmental information for subsequent decision-making and controlling of autonomous vehicles [2]. The accuracies of traditional machine learning algorithms no longer meet the requirements of autonomous vehicles' operations. The deep learning algorithms have developed continuously and great progress has been made in 2-D detection [3] and 2-D segmentation [4]. However, it is well-known that the performance of the camera is often affected by night, rain, fog, strong light, and other conditions, which affects the detection performance [5]. With the reduction in the cost of Lidar technology and improvement in computing power, 3-D object detection has been widely adopted in autonomous vehicles [6]. As compared to a camera, Lidar is less affected by the environment and satisfies the perception requirements of autonomous vehicles. The 2-D detection algorithms only provide the positions of objects in the images. On the contrary, 3-D detection algorithms provide the position, shape, and heading angle of the object in the real environment. This information plays a very important role in subsequent decision-making and planning in autonomous vehicles [7].

Currently, most of the 3-D object detection models are trained on the KITTI dataset [8], nuScenes dataset [9], and Waymo Open dataset [10]. Recently, a new dataset One millioN sCenEs (ONCE) [11] for autonomous driving scenarios has been presented. In it, there are a lot of electric bicycles which do not exist in other datasets. Electric bicycles are fast and flexible in the urban cities, which are hard for drivers to deal with, as well as the perception of autonomous vehicles,

Manuscript received February 20, 2022; revised March 21, 2022; accepted March 27, 2022. Date of publication April 6, 2022; date of current version April 20, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant U20A20333, Grant 52072160, Grant 51875255, and Grant 61906076; in part by the Key Research and Development Program of Jiangsu Province under Grant BE2019010-2 and Grant BE2020083-3; in part by the Natural Science Foundation of Jiangsu Province under Grant BK20190853; in part by the China Postdoctoral Science Foundation under Grant 2020T130258; in part by the Zhenjiang Key Research and Development Program under Grant GY2020006; and in part by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) under Grant DE190100931. The Associate Editor coordinating the review process was Dr. Mohamad Forouzanfar. *(Corresponding author: Yingfeng Cai.)*

Hai Wang and Zhiyu Chen are with the School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China (e-mail: wanghai1019@163.com; 1445536148@qq.com).

Yingfeng Cai, Long Chen, and Yicheng Li are with the Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China (e-mail: caicaixiao0304@126.com; chenlong@ujs.edu.cn; liyucheng070@163.com).

Miguel Angel Sotelo is with the Department of Computer Engineering, University of Alcalá, 28801 Alcalá de Henares, Spain (e-mail: miguel.sotelo@uah.es).

Zhixiong Li is with the Yonsei Frontier Lab, Yonsei University, Seoul 03722, Republic of Korea, and also with the Faculty of Mechanical Engineering, Opole University of Technology, 45758 Opole, Poland (e-mail: zhixiong.li@yonsei.ac.kr).

Digital Object Identifier 10.1109/TIM.2022.3165251

TABLE I

COMPARISONS OF CAMERA 2-D AND LIDAR 3-D DETECTION

Sensors	Advantages	Disadvantages
Camera 2D detection	Low-cost, Rich semantic and texture information	Easily affected by environment
Lidar 3D detection	Provide the position, shape, and heading angle of the object in the real environment	Expensive, Hard to process sparse and irregular points

and make the traffic conditions more complex. Therefore, a 3-D object detector specialized for complex traffic conditions is a dire need. There are several 3-D object detectors trained using the ONCE dataset, including PointPainting [12], PointR-CNN [13], SECOND [14], PointPillars [15], PV-region convolution neural network (RCNN) [16], and CenterPoints [17] etc. However, the detection accuracies of these algorithms are not high enough to meet the perception requirements of autonomous vehicles, as is shown in the benchmark.[1] Especially, the detection performance for small objects, such as pedestrians and cyclists, is very poor. In addition, the detection performance of long-distance vehicles is also not quite suitable. In order to effectively relieve these problems, we design and train a detector for complex traffic conditions and our method can perceive the complex traffic conditions better than the above-mentioned methods.

Because the number of beams of Lidar is limited, the Lidar cannot receive a large number of points from long-distance objects. And because small objects occupy a little space, the Lidar cannot receive enough points. Due to this characteristic, it is hard to detect long-distance objects or small objects. In order to improve the detection performance of long-distance or small objects, it is a common way to use a camera to enhance the point cloud, such as EPnet [18] and PointPainting [12]. Besides, Faraway Frustum [19] trains two models for objects in short distances and objects in long distances, respectively. Because the direction of pedestrians is not always parallel or perpendicular to roads, it is hard to use an anchor to fit its directions. CenterPoints [17] predicts objects' center points and regresses their direction instead of using anchors, therefore, CenterPoints performs better for pedestrians. These methods must fuse the camera and Lidar together, or CenterPoints does not set anchors that perform worse for vehicles. But we want to modify the network to improve the performance of long-distance objects and small objects without cameras, and meanwhile, we must ensure the accuracy of vehicles. Therefore, we choose Voxel RCNN [20] which is a high-performance voxel-based 3-D object detector as a baseline and modify it to make it more suitable for complex traffic conditions.

In a voxel-based 3-D object detector, there is a 3-D backbone and a 2-D backbone. The 3-D backbone is used to process each voxel and extract its features. The features extracted by the 3-D backbone are projected on the bird's-eye view to generate 2-D pseudo images. Then, the 2-D backbone is used to process the pseudo images and extract the features to generate high-quality proposals or detection results. During the extraction of the voxel features by downsampling using the 3-D backbone, a lot of precise information is lost. Therefore, we add the residual module [21] to maintain the integrity of the information. Additionally, we design a heavy 3-D feature extractor, which is more conducive to the detection performance of small objects. The 2-D backbone of voxel-based methods is often similar to VoxelNet [22]. It is notable that most of these methods focus on the improvements of 3-D backbones to extract more features from the point cloud data, but ignore the impact of 2-D backbones in object detection. We modify the 2-D backbone and design a more sophisticated 2-D backbone comprising residual structures, self-calibrated convolutions [23], and channel attention and spatial attention [24]. The modified 2-D backbone expands the receptive field and enhances the presentation of features by using the attention mechanism. This 2-D backbone can capture more context information and improve detection performance, especially for long-distance and small objects.

Currently, it is difficult to label the 3-D datasets and the number of training samples is small. As a result, the models are not robust and the accuracy is not high enough. The long-tailed data distribution leads to category imbalance. Therefore, data augmentation is of great significance for improving the performance of detection and robustness of the model. The common data augmentation methods in 3-D object detection include random flipping along the X-axis, random global scaling, random rotation along the Z-axis, ground-truth sample augmentation [14], and class-balanced sampling strategy (CBSS) [25]. The CBSS alleviates the category imbalance and improves the detection performance for those classes which have a small number of training samples. Therefore, we use this data augmentation strategy during the training process, for improving the detection performance of pedestrians and cyclists.

The major contributions of this work are presented below:

1) We train a 3-D object detector Voxel-RCNN-Complex for complex traffic situations on the ONCE dataset.
2) We modify the 3-D and 2-D backbones of the original Voxel RCNN. As a result, the proposed method performs better than the original Voxel RCNN, especially on small objects and faraway objects, and other state-of-the-art (SOTA) algorithms, such as CT3D, CenterPoints, etc.
3) We use CBSS for data augmentation, which relieves class imbalance and enhances the robustness of the model.

II. RELATED WORKS

In the field of point cloud 3-D object detection, voxel-based methods account for the majority. Voxel-based methods first voxelize the input point cloud and then use the 3-D convolution to extract the features from the voxels of the complete scene. In VoxelNet [22], the point cloud is first divided into a large number of uniform voxels. Then, the voxel feature-encoding layer generates the voxel-wise feature, which is projected to the bird's-eye view map to generate the bounding boxes. It is the start of the voxel-based method, but

[1] https://once-for-auto-driving.github.io/index.html

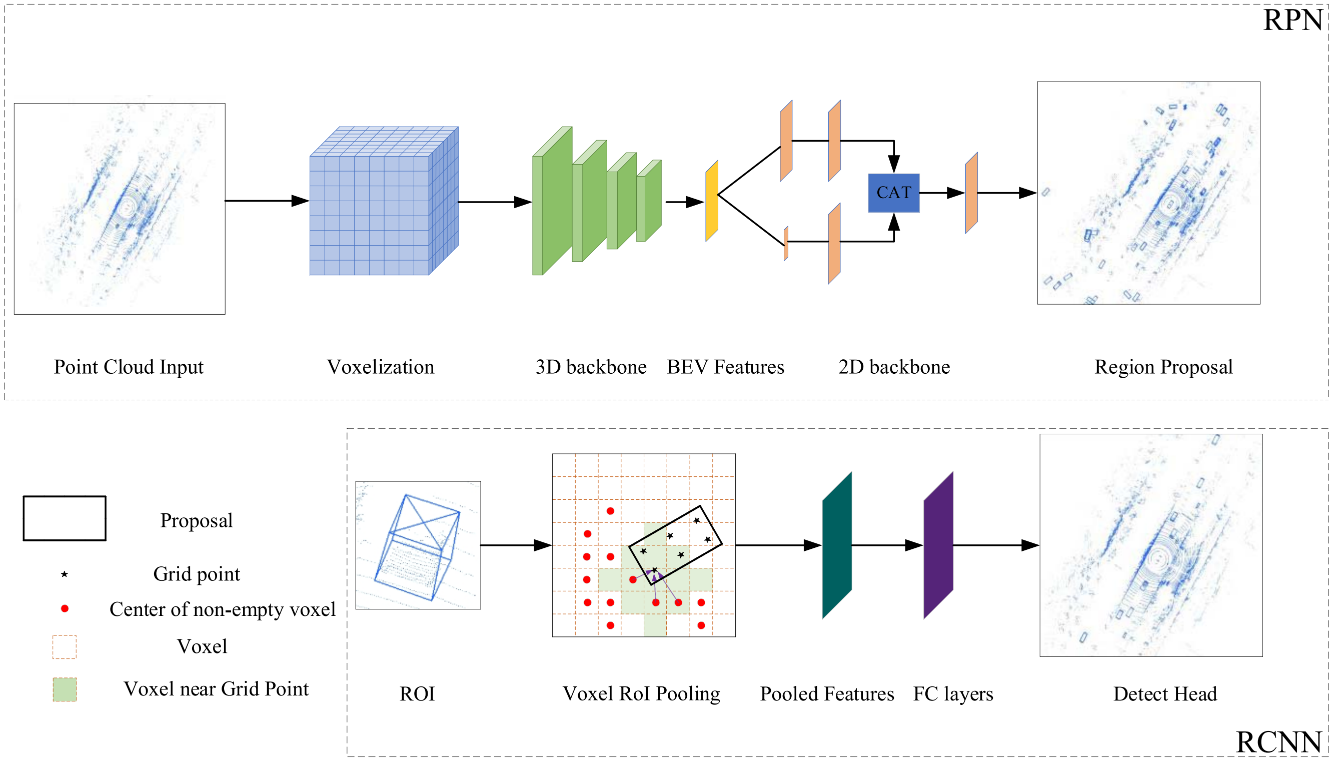

Fig. 1. Architecture of the original Voxel RCNN and the proposed Voxel-RCNN-Complex.

there are a lot of empty voxels in auto-driving scenes; dealing with empty voxels is a burden for the processor. SECOND [14] uses a sparse convolution algorithm to extract the features from the non-empty voxels, it uses sparse convolution to deal with non-empty voxels, which greatly improves the inference speed of voxel-based 3-D object detection algorithms. Since then, voxel-based methods develop fast in the academic world. PointPillars [15] divides the point cloud into pillars along the Z-axis and significantly improves the speed of point cloud feature encoding, this method has a simple network that does not use complex 3-D convolution, and it is popular in the industrial community. Part A^2 [26] considers the center of each voxel as a point and predicts the position of each point in the 3-D bounding box as an additional supervision information. PV-RCNN [16] uses sparse convolution to extract the features from the voxels and generate proposals. It encodes the multiscale voxel features to the key points and refines the boxes by aggregating the features of the key points around the grid points in the proposal. This method adds point information to voxels which improves the detection performance, but has a very low inferencing speed. Voxel RCNN [20] takes sparse convolution as the backbone to generate 3-D candidate boxes. It then distributes the uniform grid points in 3-D candidate boxes and encodes the voxel features around each grid point on the grid points. It is notable that the voxel RoI pooling module does not need the point information which makes the algorithm performs faster than PV-RCNN. CT3D [27] improves the voxel RoI pooling module with a channel-wise transformer, which can obtain the global information of proposals, which is quite useful for object refining in two-stage. M3DeTR [28] builds the relation between multirepresentation, multiscale with transformer; the relation information is of great significance for detection performance. CT3D and M3DETR all use transformer structures that can extract the feature better and improve the detection performance, but this will affect the inference speed greatly, while speed is very important for autonomous driving. All the above-discussed methods use anchors to fit the object, so these methods perform worse when confronting rotated objects because anchors cannot be enumerated in all directions. CenterPoints [17] predicts the center location of objects and regresses its direction, and it performs better for pedestrian detection.

In summary, through the analysis of the voxel-based method, we choose to use the Voxel RCNN algorithm as the baseline, which can keep a balance between precision and speed. In addition, in view of the poor performance of the algorithm on the ONCE dataset, especially for the performance of pedestrians and long-distance vehicles, we modify Voxel RCNN and improve the detection effect of long-distance vehicles and pedestrians while ensuring the overall accuracy.

III. Methodology

In this work, we use Voxel RCNN [20] as the baseline and modify its architecture to design the proposed Voxel-RCNN-Complex. The proposed network is an effective 3-D object detector for complex traffic conditions. Both the original Voxel RCNN and the proposed Voxel-RCNN-Complex are voxel-based two-stage 3-D object detectors. They share an overall similar framework as presented in Fig. 1. In the first stage, the input point

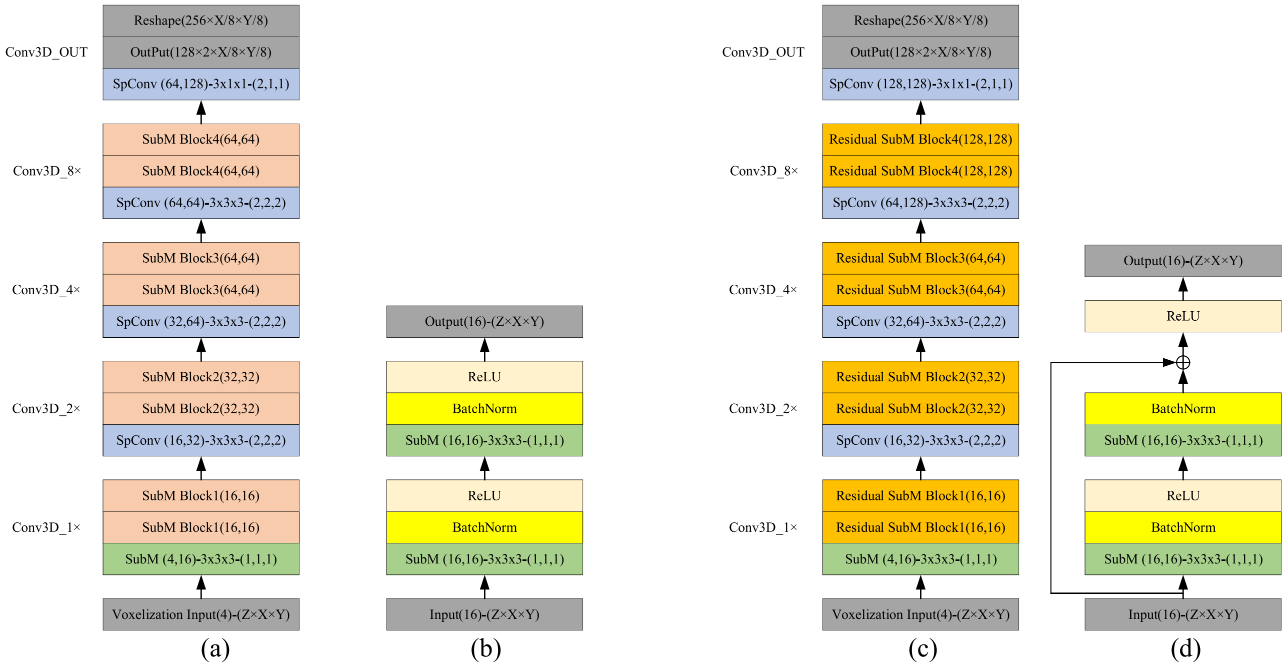
Reshape(256×X/8×Y/8)
Conv3D_OUT
OutPut(128×2×X/8×Y/8)
SpConv (64,128)-3x1x1-(2,1,1)
Conv3D_8×
SubM Block4(64,64)
SubM Block4(64,64)
SpConv (64,64)-3x3x3-(2,2,2)
Conv3D_4×
SubM Block3(64,64)
SubM Block3(64,64)
SpConv (32,64)-3x3x3-(2,2,2)
Conv3D_2×
SubM Block2(32,32)
SubM Block2(32,32)
SpConv (16,32)-3x3x3-(2,2,2)
Conv3D_1×
SubM Block1(16,16)
SubM Block1(16,16)
SubM (4,16)-3x3x3-(1,1,1)
Voxelization Input(4)-(Z×X×Y)
(a)
Output(16)-(Z×X×Y)
ReLU
BatchNorm
SubM (16,16)-3x3x3-(1,1,1)
ReLU
BatchNorm
SubM (16,16)-3x3x3-(1,1,1)
Input(16)-(Z×X×Y)
(b)
Reshape(256×X/8×Y/8)
Conv3D_OUT
OutPut(128×2×X/8×Y/8)
SpConv (128,128)-3x1x1-(2,1,1)
Conv3D_8×
Residual SubM Block4(128,128)
Residual SubM Block4(128,128)
SpConv (64,128)-3x3x3-(2,2,2)
Conv3D_4×
Residual SubM Block3(64,64)
Residual SubM Block3(64,64)
SpConv (32,64)-3x3x3-(2,2,2)
Conv3D_2×
Residual SubM Block2(32,32)
Residual SubM Block2(32,32)
SpConv (16,32)-3x3x3-(2,2,2)
Conv3D_1×
Residual SubM Block1(16,16)
Residual SubM Block1(16,16)
SubM (4,16)-3x3x3-(1,1,1)
Voxelization Input(4)-(Z×X×Y)
(c)
Output(16)-(Z×X×Y)
ReLU
BatchNorm
SubM (16,16)-3x3x3-(1,1,1)
ReLU
BatchNorm
SubM (16,16)-3x3x3-(1,1,1)
Input(16)-(Z×X×Y)
(d)

Fig. 2. (a) Architecture of the original 3-D backbone. (b) Architecture of the SubM Block 1. (c) Architecture of the modified 3-D backbone. (d) Architecture of the residual SubM Block 1.

cloud is voxelized. The 3-D backbone comprises sparse convolution [29] and submanifold convolution [30], which are used for voxel feature extraction. The voxel features are projected on the bird's-eye view to generate the pseudo image features. A 2-D backbone is used for feature extraction from the pseudo images to generate the proposals. In the second stage, the uniform grid points are seeded in the proposals. The voxel features around the grid points are gathered, and the pooled features are generated based on the voxel RoI pooling module. Finally, the detection results are predicted by the fully connected layers. It is noteworthy that we add residual structure [21] to the 3-D backbone of the original Voxel RCNN and design a heavy 3-D feature extractor. Moreover, we also design a 2-D backbone comprising residual structure, self-calibration convolution [23], and channel attention and spatial attention [24].

In the following sections, we present the detailed architecture of the proposed Voxel-RCNN-Complex. The complete pseudocode is shown in Algorithm 1.

A. Voxelization

The input point cloud is voxelized along the X, Y, and Z-axes, and the whole point cloud is divided into uniform voxels. This enables us to efficiently extract the features from the point cloud based on 3-D convolution. We represent the original features of each point by X, Y, and Z coordinates and the reflection intensity of each point. Each voxel is represented by the mean of the original features of the point cloud in each non-empty voxel.

B. 3-D Backbone

In voxelized 3-D object detectors, the 3-D backbone is composed of submanifold convolution and sparse convolution

Algorithm 1 Training Procedure of Proposed Method

```
Inputs: Training dataset $D$, maximal iteration $T$, batch size $N$, model $\phi$ parameterized by $\theta$, learning rate $\gamma$.
Outputs: updated model $\phi$ parameterized with $\theta^*$.
for $t = 1$ to max iteration $T$ do:
    Select a batch of Lidar files $X$ from dataset $D$
    Voxelize the Lidar points as $X^*$
    Extract feature $F$ on $X^*$ by 3-D and 2-D backbone
    Predict proposals and scores by $F$
        Sort proposals by scores -> Top-9000 proposals
        NMS (threshold = 0.8) −>512 high-quality proposals $P$
    Extract Voxel RoI Pooling features $R$ by $P$ and $F$
    Predict Refined boxes and scores by $R$
        Filter out boxes (scores <0.1) -> boxes(scores >=0.1)
        NMS (threshold = 0.5) -> final results
    Compute loss with Eq.2
    Update the parameters $\theta$
end for
Return updated model $\phi$ parameterized with $\theta^*$.
```

to extract the features from the non-empty voxels. This greatly improves the speed of feature extraction. The architecture of the 3-D backbone of the original Voxel RCNN is presented in Fig. 2(a). The input is $Z \times X \times Y$ voxel representation, where each voxel has 4-D original features, i.e., X, Y, Z coordinates of the points and reflection intensity. First, the original features of voxels are extracted and 1× downsampling by submanifold convolution followed by two continuous submanifold blocks is used for feature extraction. Then, three sparse convolutions are used to perform 2×, 4×, 8× downsampling along X, Y, and Z-axes to obtain multiscale 3-D features, and each sparse convolution followed by two continuous submanifold blocks is used to extract features. The architecture of submanifold block 1 is shown in Fig. 2(b). It consists of two submanifold convolutions. Each submanifold convolution is followed by BatchNorm and ReLU layers. The architecture of other submanifold convolution blocks is similar to submanifold convolution block 1. Finally, sparse convolution is used to perform 2× downsampling along the Z-axis, and then the feature map is reshaped to $256 \times X/8 \times Y/8$ pseudo image representation.

We modify the original 3-D backbone and replace the submanifold block with a residual submanifold block to obtain more rich features. The residual submanifold block adds a shortcut between the input and output of each block, thus making it easier to maintain the complete features and optimize the network. Besides, when using sparse convolution 8× downsampling, the dimension of voxel feature vector increases from 64 to 128. As a result, the network can extract rich information for improving the detection performance. The architecture of the modified 3-D backbone is presented in Fig. 2(c). Please note that this architecture is similar to the original 3-D backbone; however, the original submanifold block is replaced by the residual submanifold block. The architecture of the residual submanifold block 1 is shown in Fig. 2(d). The input feature of the residual submanifold block is extracted based on two submanifold convolutions. Then, the output of the second submanifold convolution is added to the input of the first submanifold convolution. The architecture of the other residual submanifold block is similar to residual submanifold block 1.

C. 2-D Backbone

The 3-D feature map obtained by the 3-D backbone is projected to the bird's-eye view along the Z-axis for generating a $256 \times X/8 \times Y/8$ pseudo image representation. Afterward, the 2-D backbone is used to extract the features from the pseudo image representation and high-quality 3-D proposals generation. The 2-D backbone of the original Voxel RCNN extracts the features through a series of 2-D convolutions. The network architecture is shown in Fig. 3(a). Specifically, in 2-D pseudo image representations, six standard 3×3 convolutions are used for feature extraction, one standard 3×3 convolution is used for performing 2× downsampling, and five standard 3×3 convolutions are used for feature extraction. The features of two scales are deconvoluted to the same size and concatenated to obtain multiscale features. Finally, another 3×3 convolution is used to compress the dimension.

However, this feature extraction method has a low reception field and cannot capture large context information which is important for locating the long distances and small objects. Therefore, we modify the 2-D backbone to improve the performance for long-distances and small objects. We design a 2-D backbone composed of residual structure, self-calibration convolution [23], and channel attention and spatial attention [24], as shown in Fig. 3(b). We use three 3×3 convolutions for performing 1×, 2×, and 4× downsampling to obtain multi-scale features. Then, after each 3×3 convolution, two continuous self-calibration convolutions are used to extract more representative features. In addition, a shortcut is used between the input of the first self-calibration convolution and the output of the second self-calibration convolution. The self-calibration convolution [23] extracts the features in self-calibrated scale space and original scale space. This expands the receptive field and captures rich context information. In addition, the spatial and channel attention mechanism [24] is used after self-calibration convolution to enhance useful feature expression. Based on these modifications, more representative features are extracted, which is conducive to the generation of high-quality proposals, and more long-distance and small objects can be recalled. The architecture of self-calibration convolution 1 and channel attention and spatial attention 1 is shown in Fig. 3(c) and (d).

D. Region Proposal

The size of the 2-D feature map obtained by the 2-D backbone is $128 \times X/8 \times Y/8$. Now, 2 × num_classes anchors are placed on each pixel of the 2-D feature map. The 2-D feature map is used for category prediction and size regression of each anchor. As a result, we obtain $2 \times \text{num\_classes} \times X/8 \times Y/8$ proposals. During the training process, we select 9000 proposals with the highest classification score and perform non-maximum suppression (NMS) to obtain 512 high-quality proposals.

E. Voxel ROI Pooling

For the proposals generated in the region proposal network (RPN) stage, we use the voxel RoI pooling module [20] to aggregate the multiscale voxel features. First, uniform $6 \times 6 \times 6$ grid points are sampled for each proposal. Second, we query the non-empty voxels within a certain distance around these grid points. Third, we extract the features from non-empty voxels by using the pointnet, fully connected layer, and max-pooling layer to obtain the pooled features. When querying voxels around the grid points, we query 2×, 4×, and 8× downsampled voxels to obtain the multiscale voxel features.

F. Detect Head

The pooled features of each proposal are used as input. Two fully connected layers are used to perform confidence prediction and refinement of the regression box. In the confidence prediction branch, based on [16], [20], [31], the IoU_related score is predicted as its confidence c_i and the target c_i^* is computed using the following equation. This alleviates the mismatch between the positioning accuracy and the classification confidence

$$c_i^* = \begin{cases} 0, & \text{IoU}_i < \theta_L \\ \dfrac{\text{IoU}_i - \theta_L}{\theta_H - \theta_L}, & \theta_L < \text{IoU}_i < \theta_H \\ 1, & \text{IoU}_i > \theta_H \end{cases} \tag{1}$$

where, c_i^* denotes the target confidence for the ith proposal, IoU_i denotes the intersection over union for the

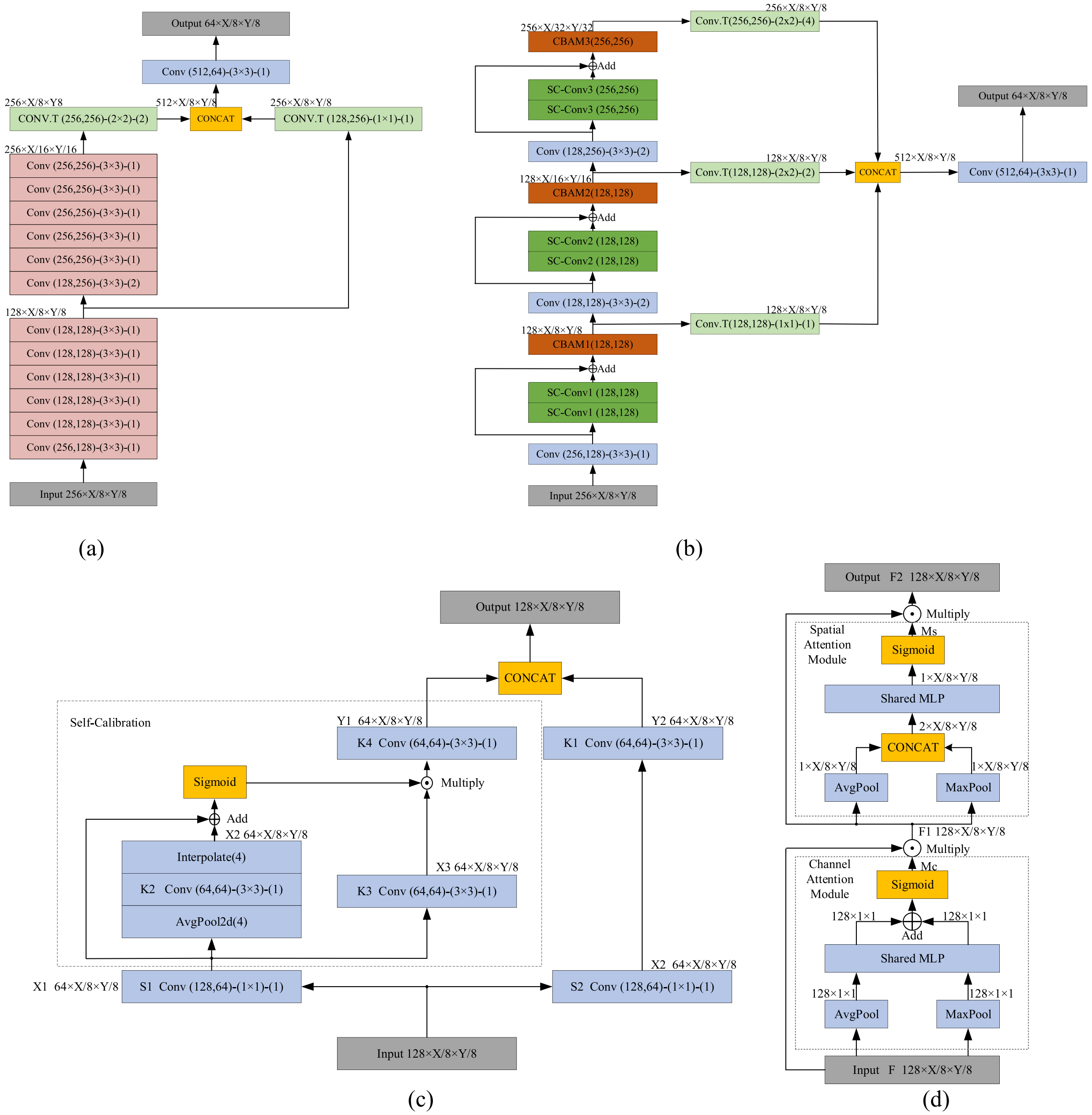

Fig. 3. (a) Architecture of original 2-D backbone. (b) Architecture of modified 2-D backbone. (c) Architecture of self-calibration convolution 1 (SC-Conv1). (d) Architecture of channel attention and spatial attention 1 (CBAM1).

ith proposal, and its ground truth box, θ_H and θ_L denote the threshold for the foreground and background, respectively.

G. Loss Function

We train the proposed network in an end-to-end way. The total loss L_{TOTAL} is composed of RPN loss L_{RPN} and RCNN loss L_{RCNN}. In the RPN stage, the focal loss [32] is used to compute the classification loss L_{cls}, and smooth L1 loss is used to compute the regression loss L_{reg1}. In the RCNN stage, binary cross-entropy loss is used to calculate the confidence loss L_{iou}, and smooth L1 loss is used to calculate the regression loss L_{reg2}

$$L_{\text{TOTAL}} = L_{\text{RPN}} + L_{\text{RCNN}} \tag{2}$$

$$L_{\text{RPN}} = \frac{1}{N_{fg}}\left[\sum_i L_{\text{cls}}\left(s_i, s_i^*\right) + \alpha L_{\text{reg1}}\left(\delta_i^1, t_i^*\right)\right] \tag{3}$$

where, N_{fg} denotes the number of foreground proposals, s_i denotes the prediction of category, s_i^* denotes the ground truth of category, α means that the regression loss is calculated only for the foreground proposals, δ_i^1 denotes the prediction of box regression parameters, and t_i^* represents the ground truth of box regression parameters

$$L_{\mathrm{RCNN}} = \frac{1}{N_s}\left[\sum_i L_{\mathrm{iou}}\left(c_i, c_i^*\right) + \beta L_{\mathrm{reg2}}\left(\delta_i^2, t_i^*\right)\right] \tag{4}$$

where, N_s denotes the number of sampled proposals, c_i denotes the prediction of confidence, s_i^* denotes the ground truth of confidence, β means regression loss is calculated only for the sampled foreground proposals, δ_i^2 denotes the prediction of box regression parameters, and t_i^* denotes the ground truth of box regression parameters.

IV. Experiments and Results

A. Experiment Details

1) Dataset: ONCE dataset [11] is the latest 3-D object detection dataset presented by the Huawei Corporation. This dataset is collected in China by using seven cameras and a 40-beam Lidar. It contains five categories, including cars, trucks, buses, pedestrians, and cyclists. The data are acquired in different weather conditions (sunny, cloudy, rainy, etc.), at different times of a day (morning, noon, afternoon, and night), and for different road conditions (downtowns, suburbs, highways, tunnels, bridges, etc.). Thus, it appropriately represents the complex traffic conditions. We evaluate the proposed Voxel-RCNN-Complex and perform ablation studies using this dataset. Moreover, we also compare the proposed Voxel-RCNN-Complex with other methods using the ONCE dataset.

2) Data Augmentation: During the training stage, we use common data augmentation methods that are widely used in 3-D object detection, including random flipping along the X-axis, random global scaling with a scaling factor ranging from 0.95 to 1.05, and random rotation along Z-axis ranging from $-\pi/4$ to $\pi/4$. We also use the ground truth sample method, randomly copying a few ground-truth objects from other scenes to the current training scene. In addition, we use the CBSS [25], randomly copying scenes with a small number of categories to relieve the category imbalance. About CBSS, we firstly duplicate samples of each category according to its fraction of all samples. The fewer a category's samples are, the more samples of this category are duplicated to form the larger training dataset. Then we randomly select the same number of samples for each category from the larger training dataset to form the final training dataset.

3) Training: We train the cars, trucks, buses, pedestrians, and cyclists simultaneously in an end-to-end way. We train the network for 90 epochs using two RTX 2080Ti GPUs using the Adam optimizer. During the training process, the maximum learning rate is 0.003, the division factor is 10, the momentum ranges from 0.95 to 0.85, the weight decay is 0.01, and the batch size is 6. The complete training process requires around 23 h. With CBSS, it takes around 77 h to train the network. Although CBSS consumes more time in the training stage, it will not affect inference speed. In addition, we use the cosine annealing learning rate strategy [33] during the training stage to obtain a robust model. The code of this work is based on an open-source 3-D object detection framework OpenPCDet.

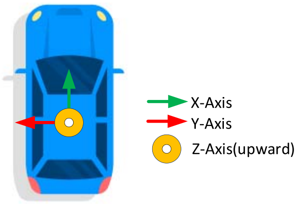

Fig. 4. Coordinate of Lidar.

TABLE II
Anchor Size of Different Categories

Category	Anchor Size/meters
car	4.38×1.87×1.59
bus	11.11×2.88×3.41
truck	7.52×2.50×2.62
pedestrian	0.75×0.76×1.6
cyclist	2.18×0.79×1.43

4) Parameter Settings: As is shown in Fig. 4, we set the Lidar center as the origin. Along the car, the forward direction denotes the X-axis, the left denotes the Y-axis, and the upward denotes the Z-axis. The complete 3-D scene, i.e., X, Y, Z, is in the range $[(-75.2,75.2), (-75.2,75.2), (-5,3)]$ m, respectively. The size of each voxel is $0.1 \times 0.1 \times 0.2$ m. Therefore, the whole scene is divided into $1504 \times 1504 \times 40$ voxels. For anchor settings, the size of different categories is shown in Table II. Five categories of anchors are placed at each pixel of the feature map. The directions of each category are 0° and 90°, respectively.

B. Results on ONCE Dataset

1) Evaluation Criterion: We use the official evaluation script provided by the ONCE dataset to evaluate the detection performance of the proposed model. We first sort the prediction results based on the confidence score and filter out the prediction results with low confidence. Second, the results for cars, buses, trucks, pedestrians, and cyclists, with 3-D IoU less than 0.7, 0.7, 0.7, 0.3, and 0.5, respectively, are filtered out. Third, we filter out the prediction results with mismatched heading angles. The remaining prediction results are true positives. Finally, we draw the precision-recall curve for 50 recalls and obtain the mean average precision (mAP) through integration.

2) Comparison With the Original Voxel RCNN: As presented in Table III, the mAP of the original Voxel RCNN using the ONCE validation set is 56.35%. On the other hand, the mAP of the modified Voxel-RCNN-Complex (without CBSS) is 63.66%, an increase of 7.31%. In addition, after applying the CBSS to the proposed Voxel-RCNN-Complex, the mAP reaches 65.34%, which is 8.99% higher than the original Voxel RCNN. We analyze the detection results of each category at different distances. In terms of vehicle detection, the detection performance of the proposed Voxel-RCNN-Complex is 2.41%, 9.94%, and 12.46% higher than the original Voxel RCNN at short-distance, medium-distance, and long-distance respectively, and the average accuracy is improved by 7.21%.

TABLE III

RESULTS OF ORIGINAL VOXEL RCNN AND THE PROPOSED VOXEL-RCNN-COMPLEX ON ONCE DATASET

Method	Vehicle				Pedestrian				Cyclist				mAP
	overall	0-30m	30-50m	50m-inf	overall	0-30m	30-50m	50m-inf	overall	0-30m	30-50m	50m-inf	
Original-Voxel RCNN	73.53	87.07	65.59	49.97	35.66	42.38	29.8	18.15	59.85	73.57	51.59	33.65	56.35
Ours (without CBSS)	80.40	89.55	74.77	61.73	43.04	51.15	36.92	22.82	67.54	79.08	61.39	44.14	63.66
Ours (With CBSS)	80.74	89.48	75.53	62.43	45.63	55.83	37.78	21.50	69.65	80.93	63.79	45.83	65.34
Improvements	**7.21**	**2.41**	**9.94**	**12.46**	**9.97**	**13.45**	**7.98**	**3.35**	**9.8**	**7.36**	**12.2**	**12.18**	**8.99**

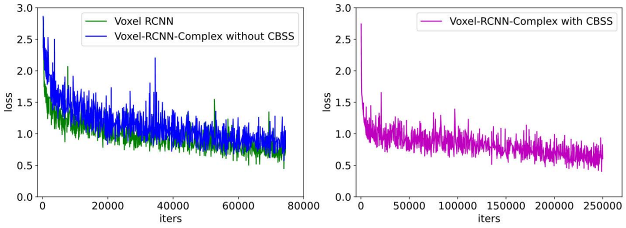

Fig. 5. Loss curve of Voxel RCNN and Voxel-RCNN -Complex (with or without CBSS).

In terms of pedestrian detection, the detection performance of our Voxel-RCNN-Complex is 13.45%, 7.98%, and 3.35% higher than the original Voxel RCNN at short-distance, medium-distance, and long-distance, respectively, and the average accuracy is improved by 9.97%. In terms of cyclist detection, the detection performance of the proposed Voxel-RCNN-Complex is 7.36%, 12.2%, and 12.18% higher than the original Voxel RCNN at short-distance, medium-distance, and long-distance, respectively, and the average accuracy is improved by 9.8%. The comparison between the original Voxel RCNN and the proposed Voxel-RCNN-Complex shows that the proposed method performs better for long-distance and small objects. For example, for cars situated 50 m away, the detection accuracy is improved by 12.46%, and for pedestrians, the overall accuracy is improved by 9.97%. Fig. 5 presents the loss curve of Voxel-RCNN-Complex and original Voxel RCNN. It can be seen that our Voxel-RCNN-Complex converges better than the original Voxel RCNN, and after using CBSS, the final loss become lower.

3) Comparison With Other Methods: Table IV shows the performance of other algorithms and the proposed Voxel-RCNN-Complex on the ONCE dataset. It is evident from Table IV that the mAP of the proposed Voxel-RCNN-Complex is 7.56%, 36.60%, 21.00%, 13.45%, 11.79%, 7.58%, 5.29%, and 9.39% greater than PointPanting [12], PointRCNN [13], PointPillars [15], SECOND [14], PV-RCNN [16], CT3D [27], CenterPoints [17], and M3DeTR [28], respectively. The mAP of the proposed method is 7.56% higher as compared to the multimodal PointPainting and 5.29% higher as compared to the CenterPoints, which performs best among the single-modal methods. Although the pedestrian performance of our method is still not as well as that of CenterPoint, which results from the disadvantages of the anchor-based method, our method performs better on vehicles and cyclists. The comparisons between the other methods and the proposed model show the effectiveness of the proposed Voxel-RCNN-Complex.

4) Ablation Studies: In this section, we present the impact of different components in the proposed Voxel-RCNN-Complex. The modification of the network architecture includes adding residual structures to the 3-D backbone, increasing the dimension of 3-D voxel features, adding residual structures, performing three times downsampling, self-calibration convolution, channel attention, and spatial attention mechanism in the 2-D backbone, and using CBSS.

5) Effectiveness of Modified 3-D Backbone: We take the original Voxel RCNN as the baseline and replace the original submanifold module with the residual submanifold module in the 3-D backbone and increase the dimension of voxel features. As presented in Table V, after adding the residual module, the accuracy is improved by 1.89% as compared with baseline, especially for pedestrians. This shows that adding the residual structure is conducive to maintaining the integrity of information and alleviating the loss of information for small objects in the downsampling process. For 8× downsampling, the dimension of each voxel increases from 64 to 128, and the accuracy improves by 1.01%; this shows that the heavy backbone appropriately extracts the high-dimension features which is important for the detection performance.

6) Effectiveness of Modified 2-D Backbone: We take the modified 3-D backbone model as the baseline. We add the residual structure, downsample the 2-D pseudo times for feature extraction, add the self-calibration convolution, and channel attention and spatial attention mechanisms. As presented in Table VI, after adding the residual structure, the accuracy improves by 0.57% as compared to the baseline. The BEV feature map is extracted three times, and the accuracy is improved by 1.25%. The extraction of features at three different scales is conducive to the detection of different-sized objects. The self-calibration convolution is used to extract the features in different scale spaces, and the accuracy is improved by 1.51%. Based on the self-calibration convolution, the receptive field is expanded and more context information is obtained. By adding the channel attention and spatial attention mechanism after self-calibration convolution, the accuracy is improved by 1.08%. The attention mechanism highlights the useful information in the spatial and channel directions, weakens the useless information, and enhances the expression of features.

7) Effectiveness of CBSS: We take the model with a modified 3-D backbone and 2-D backbone as the baseline. As presented in Table VII, because of CBSS, the mAP is improved by 1.68%. On the one hand, adding CBSS alleviates the category imbalance, as is seen in Fig. 6; the distribution

TABLE IV
RESULTS OF VOXEL-RCNN-COMPLEX AND OTHER METHODS

Method	Source	Sensors	Vehicle	Pedestrian	Cyclist	mAP	FPS/Hz
PointPainting	CVPR2020	Lidar+Camera	66.17	44.84	62.34	57.78	-
PointRCNN	CVPR2019	Lidar	52.09	4.28	29.84	28.74	1.3
PointPillars	CVPR2019	Lidar	68.57	17.63	46.81	44.34	**27.1**
SECOND	Sensors2018	Lidar	71.19	26.44	58.04	51.89	26.6
PV-RCNN	CVPR2020	Lidar	77.77	23.50	59.37	53.55	5.9
VoxelRCNN	AAAI2021	Lidar	75.53	35.66	59.85	56.35	21.3
CT3D	ICCV2021	Lidar	79.14	32.05	62.07	57.76	11.6
CenterPoints	CVPR2021	Lidar	66.79	**49.90**	63.45	60.05	14.2
M3DeTR	WACV2022	Lidar	77.78	29.93	60.14	55.95	3.3
Ours	**-**	**Lidar**	**80.74**	45.63	**69.65**	**65.34**	13.8

The mAP of PointPainting,PointRCNN,PointPillars,SECOND,PV-RCNN,CenterPoints is released by ONCE Dataset, the mAP of M3DeTR,VoxelRCNN,CT3D are trained and tested by ourselves using 2 RTX 2080Ti GPUs with their default settings, the FPS is all tested on 2080Ti GPU.

TABLE V
ABLATION STUDY RESULTS FOR 3-D BACKBONE

baseline	Residual module	Increase dimension	vehicle	pedestrian	cyclist	mAP
✓			73.53	35.66	59.85	56.35
✓	✓		74.51	38.45	61.76	58.24
✓	✓	✓	76.21	38.54	62.99	59.25

TABLE VI
ABLATION STUDY RESULTS FOR 2-D BACKBONE

baseline	Residual structure	Down sample 3 times	SC-Conv	CBAM	vehicle	pedestrian	cyclist	mAP
✓					76.21	38.54	62.99	59.25
✓	✓				77.16	39.19	63.10	59.82
✓	✓	✓			78.55	39.54	65.10	61.07
✓	✓	✓	✓		80.18	41.23	66.34	62.58
✓	✓	✓	✓	✓	80.40	43.04	67.54	63.66

TABLE VII
ABLATION STUDY RESULTS FOR CLASS-BALANCED SAMPLING STRATEGY

baseline	Class-Balanced Sampling Strategy	vehicle	pedestrian	cyclist	mAP
✓		80.40	43.04	67.54	63.66
✓	✓	80.74	45.63	69.65	65.34

of each class becomes more balanced after using CBSS and this improves the detection performance of these classes with fewer training samples. On the other hand, the training set is expanded by about 3.3 times and the training data are expanded from 5000 frames to 16 600 frames, which makes the model more robust.

C. Visualization on ONCE Dataset

In order to clearly compare the original Voxel RCNN, our proposed Voxel-RCNN-Complex, and CenterPoints, Fig. 7 presents the visualization results of these three methods on the ONCE dataset. The first, second, and third columns represent the detection results of the original Voxel RCNN, Voxel-RCNN-Complex, and CenterPoints, respectively. In addition,

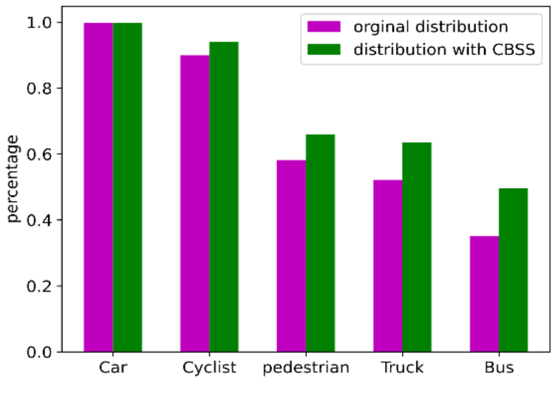

Fig. 6. Distribution of each class with or without CBSS.

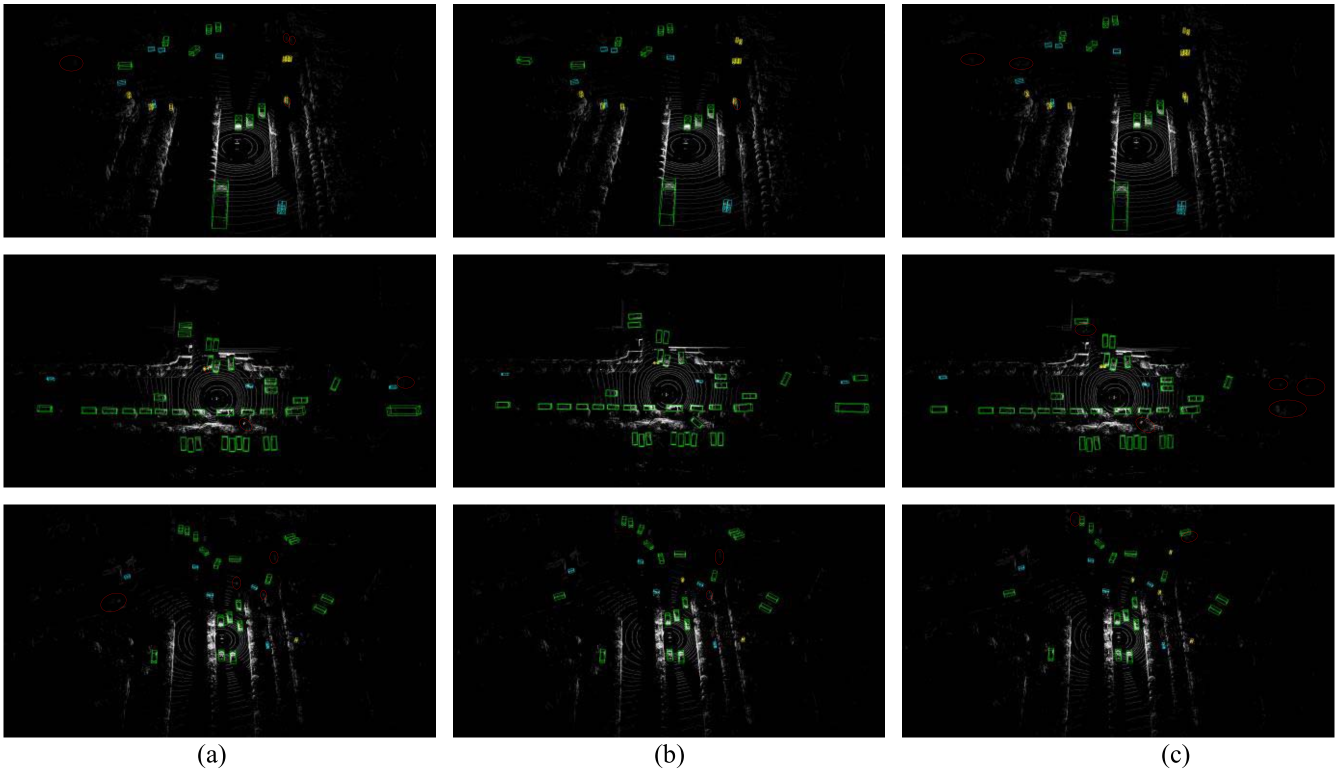

Fig. 7. Visualization results of original Voxel RCNN, Voxel-RCNN-Complex, and CenterPoints. (a) Original Voxel RCNN. (b) Voxel-RCNN-Complex. (c) CenterPoints.

the green box represents the vehicle, the blue box represents the cyclist, the yellow box represents the pedestrian, and the red circle represents the missed detection. Based on the comparison between the three methods, it is evident that the original Voxel RCNN has many missed pedestrians and long-distance vehicles, and CenterPoints has many missed vehicles. The proposed Voxel-RCNN-Complex has significantly improved compared with the original Voxel RCNN, especially for pedestrians and long-distance vehicles. Although our Voxel-RCNN-Complex also has several missed pedestrians compared with CenterPoints, our Voxel-RCNN-Complex performs better than CenterPoints. Through the comparisons between the three methods, it can be concluded that our method performs well on the ONCE dataset, except for the pedestrians.

D. Real Experiment on Intelligent Vehicles on Real Roads

In order to verify the effectiveness of the proposed Voxel-RCNN-Complex on the real roads, we deploy the proposed Voxel-RCNN-Complex in our intelligent car Chery Arrizo-5e as is shown in Fig. 8, which is equipped with an 80-beam RoboSense Lidar and an RTX 2080Ti industrial computer. We test the Voxel-RCNN-Complex on the real roads in China. The detection performance is shown in Fig. 9. The real road vehicle experiment shows that our proposed Voxel-RCNN-Complex adapts to complex traffic environments and has a good detection performance for vehicles, pedestrians, and cyclists.

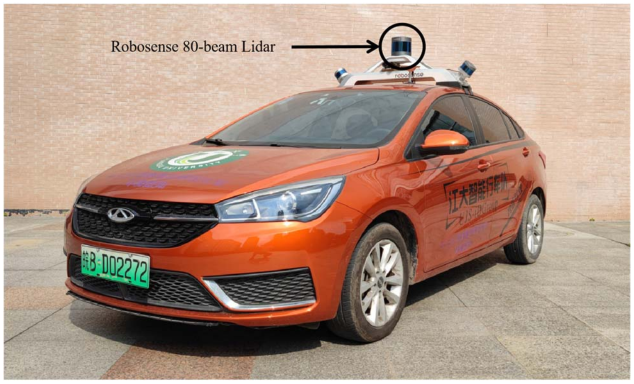

Fig. 8. Image of our intelligent vehicle and Lidar.

V. Conclusion

In this work, we design a 3-D object detector called Voxel-RCNN-Complex for a complex traffic environment. We add the residual structure to the original 3-D backbone and design a heavy 3-D feature extractor. We also design a 2-D backbone composed of residual structure, self-calibration convolution, and spatial attention and channel attention mechanism. In addition, during the training process, we use CBSS to alleviate the problem of class imbalance. The proposed Voxel-RCNN-Complex performs well on the ONCE dataset. Especially, it significantly improves the detection performance for long-distance objects and small objects and performs well in vehicle

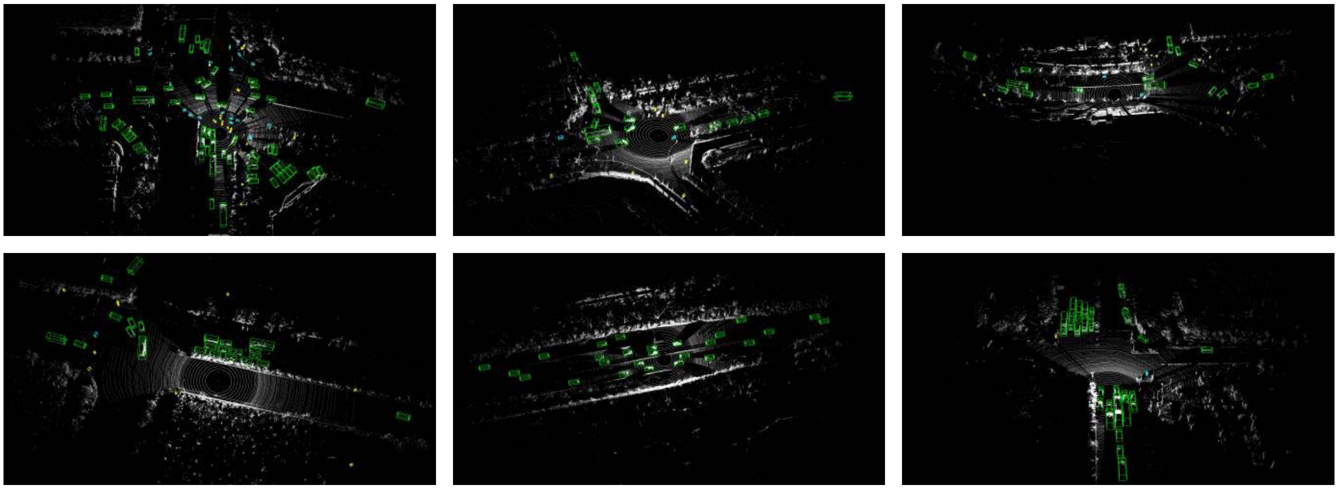

Fig. 9. Visualization results of Voxel-RCNN-Complex on real road experiment.

experiments on real roads. Therefore, our proposed method can perceive complex traffic environments better than other methods. In the future, we will continue to improve this algorithm. In terms of shortcomings, the performance of our method for pedestrians is worse than CenterPoints, which results from that the anchor cannot enumerate all directions of pedestrians, the direction of pedestrians can be 360°, and the direction of vehicles is parallel to the road at most time. But CenterPoints does not perform well on vehicles detection. In our future work, we will combine our method with CenterPoints and improve the detection accuracy of pedestrians under the premise of maintaining the detection accuracy of vehicles.

References

[1] Y. Cai *et al.*, "YOLOv4-5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[2] Y. Cai *et al.*, "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 28, 2021, doi: 10.1109/TITS.2021.3052908.

[3] 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, doi: 10.1109/TPAMI.2016.2577031.

[4] Y. Cai, L. Dai, H. Wang, L. Chen, and Y. Li, "DLnet with training task conversion stream for precise semantic segmentation in actual traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 25, 2021, doi: 10.1109/TNNLS.2021.3080261.

[5] Z. Liu *et al.*, "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 24, 2021, doi: 10.1109/TITS.2021.3059674.

[6] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019, doi: 10.1109/TITS.2019.2892405.

[7] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3D object detection networks using LiDAR data: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1152–1171, Jan. 2021.

[8] 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.*, Jun. 2012, pp. 3354–3361.

[9] H. Caesar *et al.*, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[10] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2443–2451.

[11] J. Mao *et al.*, "One million scenes for autonomous driving: ONCE dataset," 2021, *arXiv:2106.11037*.

[12] 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.

[13] 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. (CVPR)*, Jun. 2019, pp. 770–779.

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

[15] 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.

[16] S. Shi *et al.*, "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.

[17] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11784–11793.

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

[19] H. Zhang, D. Yang, E. Yurtsever, K. A. Redmill, and O. Ozguner, "Faraway-frustum: Dealing with lidar sparsity for 3D object detection using fusion," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2646–2652.

[20] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3d object detection," in *Proc. AAAI*, 2021, pp. 1–9.

[21] 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.

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

[23] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10093–10102.

[24] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[25] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," 2019, *arXiv:1908.09492*.

[26] 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, doi: 10.1109/TPAMI.2020.2977026.

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

[28] T. Guan *et al.*, "M3DETR: Multi-representation, multi-scale, mutual-relation 3D object detection with transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 772–782.

[29] B. Graham, "Sparse 3D convolutional neural networks," 2015, *arXiv:1505.02890*.

[30] B. Graham and L. van der Maaten, "Submanifold sparse convolutional networks," 2017, *arXiv:1706.01307*.

[31] J. Deng, W. Zhou, Y. Zhang, and H. Li, "From multi-view to hollow-3D: Hallucinated hollow-3D R-CNN for 3D object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4722–4734, Dec. 2021, doi: 10.1109/TCSVT.2021.3100848.

[32] 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, doi: 10.1109/TPAMI.2018.2858826.

[33] 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.

Hai Wang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, in 2006, 2008, and 2012, respectively.

In 2012, he joined the School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang, China, where he is currently working as a Professor. He has published more than 50 papers in the field of machine vision-based environment sensing for intelligent vehicles. His research interests include computer vision, intelligent transportation systems, and intelligent vehicles.

Zhiyu Chen received the B.S. degree from the Nanjing Institute of Technology, Nanjing, China, in 2020. He is currently pursuing the Ph.D. degree with Jiangsu University, Zhenjiang, China.

His research interests include computer vision, deep learning, and intelligent vehicles.

Yingfeng Cai (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Instrument Science and Engineering, Southeast University, Nanjing, China, in 2006, 2009, and 2013, respectively.

In 2013, she joined the Automotive Engineering Research Institute, Jiangsu University, Zhenjiang, China, where she is currently working as a Professor. Her research interests include computer vision, intelligent transportation systems, and intelligent automobiles.

Long Chen received the Ph.D. degree in vehicle engineering from Jiangsu University, Zhenjiang, China, in 2002.

His research interests include intelligent automobiles and vehicle control systems.

Yicheng Li received the Ph.D. degree in vehicle engineering from the Wuhan University of Technology, Wuhan, China, in 2018.

He is currently an Assistant Professor with the Automotive Engineering Research Institute, Jiangsu University, Zhenjiang, China. His research interests include intelligent vehicle localization, intelligent transportation systems, computer vision, and 3-D data processing.

Miguel Angel Sotelo (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Alcalá (UAH), Alcalá de Henares, Madrid, Spain, in 2001.

He is currently a Full Professor with the Department of Computer Engineering, University of Alcala (UAH). His research interests include autonomous vehicles and the prediction of intentions.

Dr. Sotelo is a member of the IEEE ITSS Board of Governors and Executive Committee. He has served as a Project Evaluator, a Rapporteur, and a Reviewer for the European Commission in the field of ICT for intelligent vehicles and cooperative systems in FP6 and FP7. He served as the Editor-in-Chief of the *IEEE Intelligent Transportation Systems Magazine* and the *ITSS Newsletter* and an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. At present, he is the President of the IEEE Intelligent Transportation Systems Society.

Zhixiong Li (Senior Member, IEEE) received the Ph.D. degree in transportation engineering from the Wuhan University of Technology, Wuhan, China, in 2013.

He is with the Yonsei Frontier Lab, Yonsei University, Seoul, Republic of Korea, and also with the Faculty of Mechanical Engineering, Opole University of Technology, Poland. He is the Director of the International Joint Research Center on Renewable Energy and Sustainable Marine Vehicles. He is the author/co-author of two books and over 100 articles. His research interests include dynamic system modeling, renewable energy, and machine learning applications.

Dr. Li is an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and a Column Editor of *IEEE Intelligent Transportation Systems Magazine*.