Intersection Is Also Needed: A Novel LiDAR-Based Road Intersection Dataset and Detection Method

Zhiheng Li, Yubo Cui, and Zheng Fang, *Member, IEEE*

***Abstract*— 3D object detection is crucial for autonomous driving. However, most existing methods focus on the foreground objects, such as vehicles and pedestrians, while ignoring some important background objects for traffic scene understanding, especially road intersections. Moreover, existing datasets (e.g., KITTI, Waymo) do not provide the labels for intersections, and the evaluation metric is also unsuitable for intersection detection. To address the above issues, we first present a LiDAR-based intersection dataset on the basis of KITTI dataset, called *KITTI-Intersection Dataset*. The new dataset includes 4718 frames with 5178 instances belonging to Forkroad and Crossroad, respectively. To weaken the impact of uncertain intersection size on the performance evaluation, we introduce CEIOU instead of IOU as a new evaluation metric. Then, we propose *MInsectDet* and *MMInsectDet*, two LiDAR-based detection methods, to solve the intersection detection problem. We start with a lightweight BEV backbone to alleviate the influence of numerous dynamic foreground objects at the intersection and obtain discriminative features. After that, to obtain more abundant and complete intersection features, we propose a Multi-Representation Backbone that integrates the BEV and voxel features to achieve better detection performance. Furthermore, in order to better adapt to various appearances and sizes of intersection, we propose a Class-Aware MultiHead, which classifies and regresses different categories with specific head. Finally, we evaluate our MInsectDet and MMInsectDet methods on the proposed KITTI-Intersection Dataset with the state-of-the-art foreground 3D detection methods. The results show that MMInsectDet achieves the best performance, and MInsectDet ranks second but could run at 65.0 FPS.**

***Index Terms*— 3D road intersection detection, road intersection dataset, LiDAR point cloud.**

I. Introduction

NOWADAYS, 3D object detection has been widely used in autonomous driving. Generally, the detectors directly take point cloud data as input and adopt single-stage [1], [2], [3], [4] or two-stage [5], [6], [7], [8], [9] methods to perceive foreground objects, such as vehicles and pedestrians. According to the different data representations, these works can be divided into point-based [5], [8], [9], [10], [11] and voxel-based methods [3], [4], [12]. The former can obtain fine-grain geometric information from the raw point cloud. However, due to direct operation on point cloud, those methods usually suffer from huge computation burden and time consumption. Differently, voxel-based methods convert the unordered point cloud into ordered regular voxels and directly use mature 2D detectors, which guarantees high real-time performance but loses some data information.

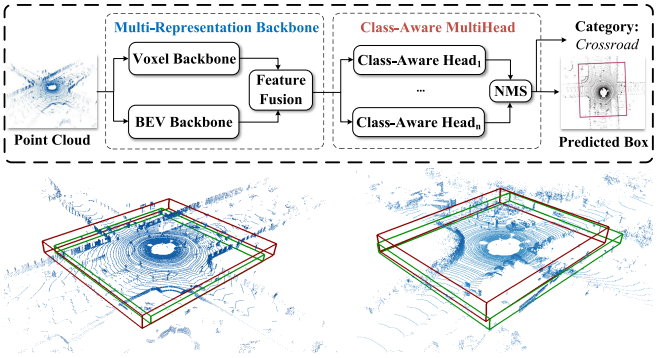


Fig. 1. The pipeline and dataset for LiDAR-based road intersection detection. The top one shows our pipeline, which first fuses BEV and voxel features by the proposed MRB and then uses CAM Head to implement the classification and regression of road intersections. The bottom one shows the detection results and the ground truths, in red and green, respectively.

Although most previous works [13], [14], [15] obtain promising detection results for foreground objects that are main participants in the traffic system, they overlook background objects, such as road intersections, which are essential traffic indicators for autonomous vehicles. In real environments, the road intersection conditions are complex, unpredictable, and even unknowable on electronic map. Meanwhile, there are no traffic lights or traffic signs at many intersections, especially in urban alleys, countryside and field environments. In these cases, if the vehicle has the ability to perceive road intersections by its own sensors, it could prepare to slow down or change lanes in advance to avoid potential collision risks without relying on electronic map. Moreover, it could be further combined with foreground detection to estimate traffic flow at the intersection and provide supplemental information for subsequent vehicle planning. Thus, intersection detection in autonomous driving could not only improve the understanding of the surrounding environment but

Manuscript received 6 July 2022; revised 25 March 2023 and 11 August 2023; accepted 8 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62073066 and Grant U20A20197, in part by the Fundamental Research Funds for Central Universities under Grant N2226001, and in part by 111 Project under Grant B16009. The Associate Editor for this article was J. Li. *(Corresponding author: Zheng Fang.)*

Zhiheng Li and Zheng Fang are with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China, also with the National Frontiers Science Center for Industrial Intelligence and Systems Optimization, Northeastern University, Shenyang 110819, China, and also with the Key Laboratory of Data Analytics and Optimization for Smart Industry, Ministry of Education, Northeastern University, Shenyang 110819, China (e-mail: fangzheng@mail.neu.edu.cn).

Yubo Cui is with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China.

Digital Object Identifier 10.1109/TITS.2024.3354939

also provide the important traffic information for downstream tasks. Besides, correcting localization drift is another potential application. For example, in some scenarios without GNSS signals, autonomous vehicles mainly rely on IMU and LiDAR for long-term localization, which will cause accumulative error. One possible way to solve this issue is to construct a prior map containing intersection locations and view intersections as stable markers. When vehicles detect an intersection during driving, the detection result can be matched with the prior map, where the accurate intersection location is obtained and used to correct localization drift.

Nevertheless, different from foreground object detection, the road intersection detection has the following challenges, as shown in Fig. 2.

- Intersections usually contain numerous foreground objects, which leads to intersection information tending to be more complex and containing much more noise.
- Affected by different road standards, the size and appearance of road intersections exist certain intra-class and inter-class gaps.
- Different from most foreground objects, intersections have no strict and clear objective boundaries.

Considering the sensor data for road intersection detection, image is difficult to reflect the complete appearance of large intersections since limited viewing angle. Besides, visual degradation problem causes the inferior performance of image-based methods in harsh environments. More importantly, image inherently lacks the depth information of objects. In contrast, LiDAR data has wider horizontal perception FOV and could obtain accurate spatial location of intersection. Thus, LiDAR point cloud is very suitable for intersection detection. But, unfortunately, the lack of dataset limits the development of related research. Till now, there are no excellent road intersection detection methods based on point cloud.

In this paper, we try to solve the above problems from two aspects, namely dataset and algorithms. **For dataset**, we propose a pure point cloud intersection dataset called *KITTI-Intersection Dataset* which should be the first LiDAR dataset for road intersection detection. We select the frames with intersections to label and divide the categories into Forkroad and Crossroad, as shown at the bottom of Fig. 1. Since there is no explicit boundary for intersection, we label their rough sizes based on our subjective judgment. We will show the details of labeling in Section III, and demonstrate the necessity of labeling subjective intersection sizes and the unbiasedness of this labeling manner in Section V. In addition, due to the unclear intersection boundary, using IOU to evaluate the quality of the predicted results is no longer suitable for intersection detection. Therefore, we propose CEIOU, a joint quality evaluation metric that combines location error and IOU of the predicted box, to distinguish true positives (TP) and false positives (FP). We believe that the proposal of KITTI-Intersection Dataset and corresponding evaluation metric could provide a fair and unified benchmark, which is also beneficial to make up for the lack of work in the intersection detection.

For algorithms, we propose *MInsectDet* and *MMInsectDet* methods to predict the category, spatial location, dimension, and orientation of intersections. Generally, in traditional 3D object detection, it is widely believed that BEV representation of point clouds usually leads to poor detection performance because of losing much geometric information. However, in road intersection detection, we argue that BEV is beneficial for weakening the interference of foreground objects and preserving the obvious appearance characteristics of the intersection. Inspired by this, we first use a lightweight BEV backbone in our proposed MInsectDet, which could obtain more discriminative intersection features and take the balance in detection precision and real-time performance. To further compensate for lacking geometric information in BEV features, we propose a Multi-Representation Backbone (MRB) in MMInsectDet to fuse BEV features and voxel features that have rough geometric information. In this way, we could obtain a more complete feature representation to improve classification and regression ability. Furthermore, we design a Class-Aware MultiHead (CAM Head), which adopts a specific head for each intersection category. By separating the predictions of different categories, it can effectively adapt the appearance gap and mitigate the scale-sensitive problem for all intersections. Finally, we evaluate our methods with the state-of-the-art foreground 3D object detection methods on the KITTI-Intersection Dataset. The experimental results show that our proposed methods are better than others. More specifically, MMInsectDet with multi-representation feature achieves the best detection performance and localization accuracy. Meanwhile, MInsectDet ranks second but could run at 65.0 FPS inference speed.

In summary, the main contributions of this paper are as follows:

- This is the first work that proposes a LiDAR-based road intersection dataset and introduces CEIOU as a new metric to evaluate the quality of predicted results. We think this work will open a new and interesting research topic for autonomous driving and robotics related areas.
- We present MRB combining features of different representations to obtain complete intersection feature. Meanwhile, we propose CAM Head to further improve the classification and regression ability for intersections with significant differences regardless of appearance or scale. Both of them can be used as plug-and-play modules to improve detectors' performance in intersection detection.
- Our proposed two practical algorithms, named MInsectDet and MMInsectDet, achieve the best detection precision and maintain real-time performance on the KITTI-Intersection Dataset. The dataset and methods will be available soon at https://github.com/LeoZhiheng/KITTI-Intersection.

The rest of this paper is organized as follows. In Section II, we discuss the related work on datasets and algorithms. Section III describes the annotation details of our dataset and relevant evaluation metrics. Section IV presents the framework of MInsectDet and MMInsectDet. In Section V, we evaluate

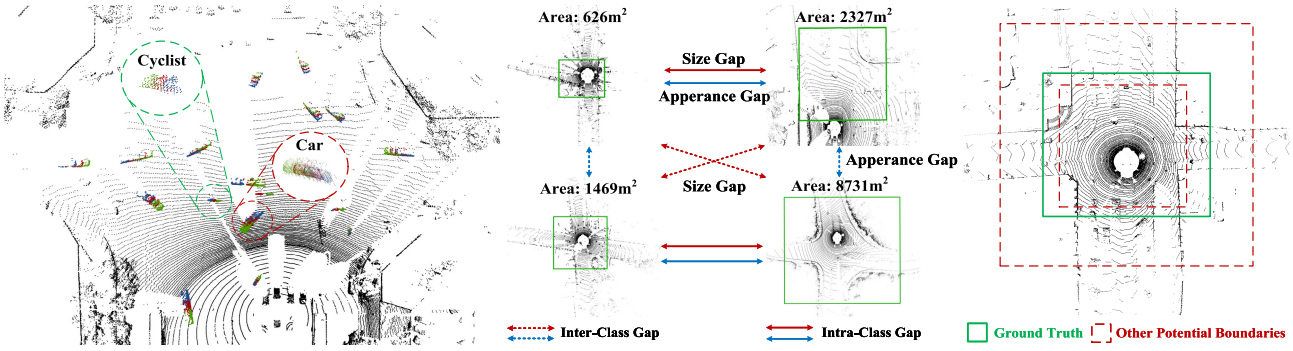


Fig. 2. (1) The left figure shows the visualization of dynamic foreground objects at the road intersection. Different colored point clouds represent moving objects in different frames. (2) The middle figure demonstrates the size and appearance gap among the different intersections. (3) The right figure displays that the intersection lacks clear boundaries.

the performance of proposed methods and conduct ablation studies. Finally, we conclude in Section VII.

II. Related Work

A. Datasets

1) Traffic Information Datasets: As deep learning methods [16], [17], [18] begin to be applied to the perception of traffic information, researchers construct many image-based datasets containing rich traffic information, such as traffic signs [19], [20], road surface markings [21], traffic lights [22], lane lines [23], etc., for training robust neural networks. Although the above datasets have covered most traffic information, they usually ignore vital road intersections, where autonomous vehicles are prone to traffic accidents. Meanwhile, in terms of sensor data, image misses depth information of intersection and suffers from visual degradation problem in some challenging scenes. Therefore, it is necessary to collect sensor data that can reflect the spatial characteristic of intersections to complement current traffic information datasets.

2) 3D Object Detection Datasets: In recent years, 3D LiDAR sensors have been widely used in autonomous driving and provide point cloud with long-range perception ability. Meanwhile, the accurate point cloud can also reflect the detailed 3D structure of objects. For these reasons, many LiDAR-based datasets emerge one after the other, such as KITTI [24], Nuscenes [25], and Waymo [26], which has greatly promoted related research in 3D object detection. However, these datasets mainly focus on foreground objects and ignore background objects that are significant for scene understanding. To this end, we propose a point cloud intersection dataset, which supplements the categories of existing datasets and establishes a fair evaluation metric to provide a benchmark for road intersection detection.

B. Algorithms

1) 3D Detection on Point Cloud: According to different representations of point clouds, LiDAR-based detection methods could be divided into voxel-based and point-based methods. The voxel-based methods convert the unordered point clouds into regular voxels by voxelization. However, numerous empty voxels lead to large computation burden. To solve this problem, SECOND [3] applied sparse convolution to obtain compact 3D voxel feature representation. PointPillars [2] generated pseudo-image by extracting features from points and utilized 2D detector to predict. Furthermore, Li et al. [4] adopted an anchor-free regression head to avoid generating numerous anchors and achieved better detection performance. Differently, point-based methods directly encode the raw point clouds. For example, PointRCNN [5] applied PointNet++ [27] to extract point-wise features, generated proposals by RPN, and finally refined the proposals by the following ROI. To sufficiently exploit object information, Zhou et al. [8] proposed a unified framework for joint 3D object detection and instance segmentation at the point level.

Generally, the voxel-based methods achieve fast computation but lose some geometric information, while the point-based methods could reserve fine-grained 3D structure but would be time-consuming. Therefore, some works attempt to combine the advantages of both sides. For example, Part-A^2 [6] extracted voxel features by sparse convolution and proposed RoI-aware to obtain the point information within the proposals. PV-RCNN [7] combined the voxel features with the set abstraction in PointNet [28] to obtain more discriminative point features. Besides, some semi-supervised [29], weakly supervised [9] and unsupervised [30] methods obtained promising results without numerous expensive and time-consuming annotations. And graph neural network [31] was also used in 3D detection to aggregate temporal information from multiple frames. However, the above methods only focus on foreground object detection, such as vehicles and pedestrians. Thus, they may not be suitable for the intersection detection since the characteristic gap between foreground and background objects. To solve this problem, we propose MInsectDet and MMInsectDet to detect road intersections better.

2) Road Intersection Perception Algorithms: Previous works on the perception of road intersection could be divided into image-based and point-based methods, respectively. Jochem et al. [32] first tried to use image and ANN to recognize Y shape intersection. Koji and Kanji [33] and Bhatt et al. [34] further adopted the combination of CNN and LSTM to classify the intersection. Ballardini et al. [35] used teacher/student training paradigm to improve intersection classification results. Moreover, Sugimoto et al. [36] presented a PDoT-based method that identified an intersection from

TABLE I
STATISTIC OF ROAD INTERSECTION NUMBERS IN THE TRAINING AND TESTING SET OF KITTI-INTERSECTION DATASET

	Forkroads	Crossroads	Total
Train	3328	1084	4412
Test	573	202	775
Total	3901	1286	5187

individual frames in 360° videos. Although image-based works have achieved well-classified results, they mainly focus on classification and lack the capacity to obtain the spatial information of intersection. Compared to prior works, we propose a point-based method that implements classification and localization for intersection at the same time.

Although we can see that utilizing point cloud to detect road intersections has many advantages from the aforementioned analysis (in Section I), only a few works utilize point cloud to detect intersections due to the shortage of available point cloud intersection datasets. For example, Habermann et al. [37] designed ANN to process point clouds to identify road geometry categories. By exploiting the LiDAR data, Hata et al. [38] employed machine learning to classify road structures. To promote the research of LiDAR-based intersection detection, in this paper, we first present a LiDAR-based intersection dataset and the corresponding evaluation metrics. Then, we propose Multi-Representation Backbone and Class-Aware MultiHead to improve the advanced 3D object detection method so that it can provide intersection categories and spatial information while maintaining real-time performance. To the best of our knowledge, this is the first work that provides point cloud dataset and new evaluation metric for intersection detection.

III. BENCHMARK DATASET

A. Data Collection and Statistics

Geiger et al. introduced KITTI [24] Dataset in 2012, which collected point cloud data in diverse scenes through Velodyne LiDAR sensor (10 HZ spin-rate, 64 laser beams, range: 100m). However, the dataset only provides the label of foreground objects, such as cars, cyclists, and pedestrians. With the help of its rich point cloud, we annotate the road intersections in the 3D object detection benchmark of KITTI to construct our KITTI-Intersection Dataset, which aims to complement the background categories for KITTI. Specifically, we select 4718 frame point clouds containing intersections from KITTI training set and annotate 5187 intersection instances for Forkroads and Crossroads. The detailed data statistics of the KITTI-Intersection are shown in Table I. Furthermore, in Fig. 4, we display the distribution of intersection areas in our dataset.

B. Ground Truth Labels

In the KITTI dataset, only objects within the camera view have annotation information. However, this setting makes the number of intersections very few and further affects the performance of networks. Thus, we label all intersections within the LiDAR scan range to obtain sufficient samples. Meanwhile, our label format and coordinate system are both consistent with KITTI. That is to say, the label format is [*type, truncated, occluded, alpha, bbox, dimensions, location, y-rotation*] in the camera coordinate system.

The difference is that we ignore the degree of truncation and occlusion of intersections. Thus, the trained network can detect potential intersections as much as possible. Additionally, when labeling the dimension of intersections, we empirically give the relatively inaccurate length and width of intersections and set the height to $3m$ since the intersections do not have objectively accurate boundaries. In Section V-C, we will show the necessity of labeling subjective dimension. Meanwhile, to eliminate the impact of the empirically labeling and ensure the fairness of the evaluation, we further propose a reasonable evaluation metric in Section III-C. Finally, we align the orientation of ground truth with the main road. The visualization of some annotation results is shown in Fig. 3. In summary, our KITTI-Intersection Dataset can provide category, spatial location, dimension, and orientation of intersections to train and test the detectors.

C. Evaluation Metrics

The key to intersection detection is distinguishing the categories correctly and regressing the location accurately. Therefore, to obtain a fair comparison in evaluating different methods, we first introduce Average Center Error as the metric of localization accuracy. Meanwhile, we propose CEIOU to avoid the impact of unclear intersection boundary on distinguishing predicted boxes with different qualities. Furthermore, we employ CEIOU-based average precision and recall to evaluate detection performances comprehensively.

1) Average Center Error (ACE): To compute the value of ACE, we first need to match the predicted box with the ground-truth box by computing the IOU between them, and match the boxes if the IoU is higher than the threshold. Then, in frame k, we use Euclidean distance to define the center error $d_{k,i}$ between the matched boxes i as follows:

$$d_{k,i} = \sqrt{(x_{p_i} - x_{t_i})^2 + (y_{p_i} - y_{t_i})^2} \tag{1}$$

where (x_{p_i}, y_{p_i}) and (x_{t_i}, y_{t_i}) are the center coordinates of the predicted box p_i and the ground-truth box t_i on the x-y plane, respectively. Finally, we calculate the average center error (ACE) of the matched boxes in all frames, where c_k is the number of pairs in the frame k.

$$ACE = \frac{\sum_{k,i} d_{k,i}}{\sum_k c_k} \tag{2}$$

2) CEIOU: In terms of evaluating detection performance, IOU is usually the criterion for determining whether the predicted box is a true positive (TP) or false positive (FP). The predicted box will be identified as FP when the IOU is lower than the certain threshold, otherwise as TP. Then, the average precision (AP) is further calculated by counting the number of TP and FP to measure the detection precision.

However, different from previous foreground object detection, intersections lack objective boundary, leading to ambiguity in TP and FP. As shown on the left of Fig. 5, we found that

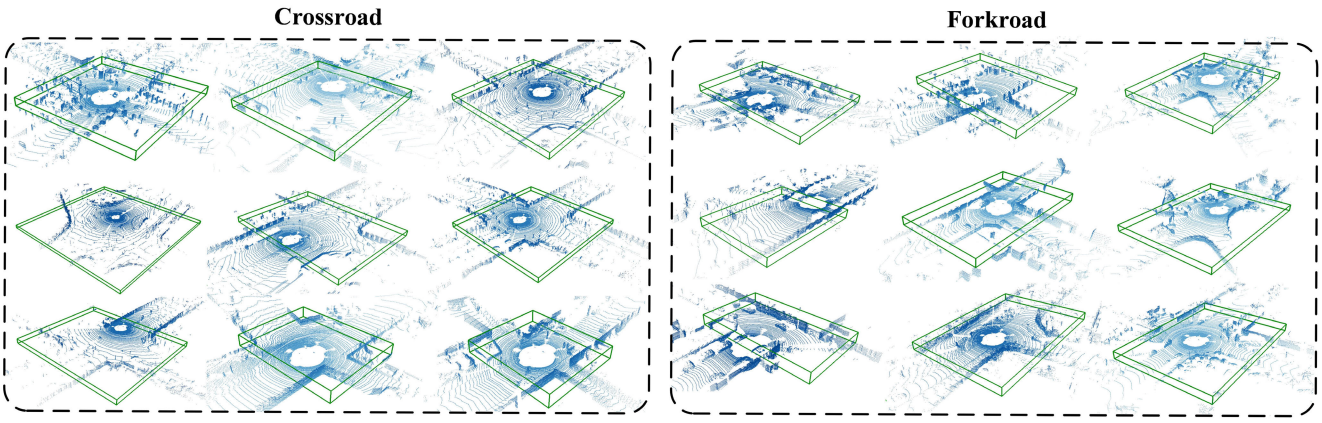


Fig. 3. Visualization results of the ground-truth boxes in the KITTI-Intersection Dataset. The road intersection includes two categories, Forkroad and Crossroad. The center of the ground-truth box corresponds to that of the intersection. The height of ground truth is a fixed value, and the width and length are empirically labeled. The orientation aligns with the main road.

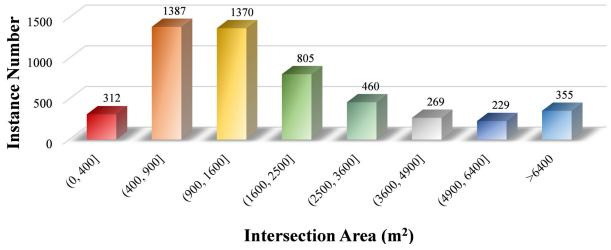
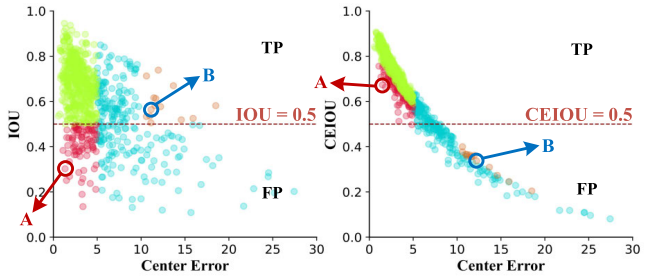
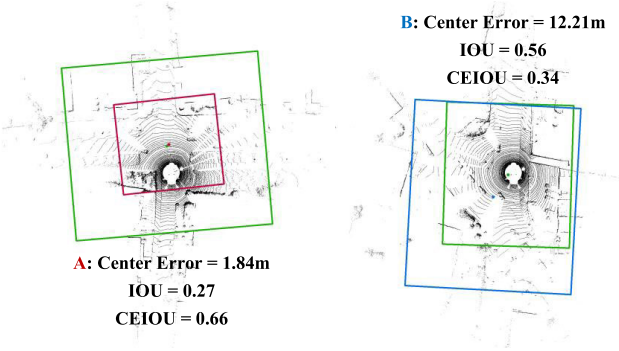


Fig. 4. Distribution of road intersection areas in the KITTI-Intersection Dataset. The scale change of different intersections may be more than 10 times.



(a) Comparison of the correspondence between CEIOU/IOU and Center Error. The definition of three qualities of predicted box: Optimal boxes (green points, IOU>0.5, Center Error<5.0m), Sub-optimal boxes (red points, IOU<0.5, Center Error<5.0m), Poor boxes (orange points, IOU>0.5, Center Error>10.0m).



(b) The visualization of sub-optimal predicted box A and poor predicted box B. Both red and blue bounding boxes are predicted results. The green bounding box is the ground-truth box.

Fig. 5. The explanation of the reason for CEIOU as the new evaluation metric to distinguish TP and FP.

simply adopting IOU to evaluate the quality of predicted boxes will introduce many low-quality boxes in TP. On the other hand, some sub-optimal results are incorrectly classified as FP, which leads to unfairness in the subsequent calculation of AP. Specially, when the IOU threshold equals 0.5, the optimal predicted boxes (green points) can be correctly classified as TP. However, the ambiguous intersection boundary inevitably leads to some sub-optimal predicted boxes (red points) with low IOU but high localization accuracy being incorrectly classified as FP. As shown on the left side of Fig. 5, although the center error of sub-optimal predicted box A is only 1.84m, A is wrongly identified as FP due to its IOU being too low. On the contrary, some poor predicted boxes (orange points) with high IOU but inferior localization accuracy will be classified as TP, as shown on the right side of Fig. 5.

To avoid this unreasonable evaluation, we propose a joint quality metric, named CEIOU, which introduces center error into the quality evaluation. Instead of only considering area overlap, CEIOU simultaneously takes localization accuracy into account, compensating for the adverse effect of ambiguous intersection boundary. The CEIOU could be computed as follows:

$$CEIOU = (IOU)^{1-\beta}(e^{\alpha \times d})^{\beta} \tag{3}$$

where d and IOU are the center error and intersection over union, respectively. By default, $\alpha = -0.1$, $\beta = 0.8$.

The correspondence between CEIOU and center error is shown on the right of Fig. 5. Compared with IOU, the sub-optimal predicted box (red points including predicted box A) will be corrected to TP. Besides, the poor predicted box (orange points including predicted box B) is corrected to FP. Meanwhile, the original evaluation result of the optimal predicted box (green points) does not change. Based on that, we replace IOU with CEIOU as our evaluation criterion of predicted box quality in the intersection detection.

For calculating the recall, we also adopt CEIOU as the criterion and further count the number of false negatives (FN):

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

In Section V, we will calculate AP and recall of our methods and state-of-the-art 3D detection algorithms in intersection detection task based on CEIOU. The specific performance comparison is shown in Table II and Table III.

IV. Methdology

A. Overview

For traditional 3D detection, the foreground objects typically have relatively fixed sizes and are impossible to overlap with others. However, different from the foreground objects, road intersection often has dynamic foreground objects within it (Fig. 2). Even for the same intersection, the numerous dynamic foreground objects would lead to inconsistent intersection representation information in different frames. Therefore, it is difficult for network to learn unified and consistent feature for intersections. Moreover, due to different road design standards, intersections have significantly different sizes and appearances. Hence, perfectly classifying and regressing intersection is also a challenging task.

Nowadays, point-based methods [5], [11] usually use PointNet++ [27] to aggregate local structure information. However, PointNet++ is difficult to encode global features of large intersections and also time-consuming. Voxel-based methods [2], [3], [39] divide the point cloud into ordered voxels to extract rough geometric features. However, mixed foreground and background points in voxel hinder the extraction of discriminative intersection features. Usually, BEV-based methods [40], [41] may lose rich geometric information when projecting the point cloud to the BEV-plane, which leads to poor performance in 3D object detection. However, different from previous conclusions, we found that the BEV perspective avoids spatial occlusion and retains the complete intersection appearance. Meanwhile, it can weaken the influence of foreground points within the intersection, which benefits encoding discriminative intersection features.

Inspired by this, we propose a simple yet efficient method called MInsectDet, which exploits a lightweight BEV backbone to encode the intensity, occupancy, and height of intersections. Moreover, we propose Multi-Representation Backbone in MMInsectDet (Fig. 6), which fuses BEV features with voxel features that have rough geometric information. Additionally, to deal with various sizes and appearances in intersections, we introduce Class-Aware MultiHead with stronger classification and regression capability. Note that both of our methods adopt CAM Head to generate prediction results. Although our network looks simple, our lightweight and efficient method is very suitable for autonomous driving. In the following, we present the details of MInsectDet and MMInsectDet methods.

B. BEV Backbone and Multi-Representation Backbone (MRB)

In order to get a trade-off between detection precision and efficiency, we use BEV backbone as part of MInsectDet to extract intersection features. Similar to [41], we define the point cloud range as $x \in [X_{min}, X_{max}]$, $y \in [Y_{min}, Y_{max}]$, $z \in [Z_{min}, Z_{max}]$, which consists of a series of points $p \in \{p_0, p_1, p_2, \ldots, p_m\}$. Each point with its own 3D location and intensity information is expressed as $p_e = (x_e, y_e, z_e, I_e), e \in \{0, \ldots, m\}$.

$$\begin{cases} \triangle X = X_{max} - X_{min} \\ \triangle Y = Y_{max} - Y_{min} \\ \triangle Z = Z_{max} - Z_{min} \end{cases} \tag{5}$$

Then, along the x-y plane, we divide points p into a series of pillars $P[i, j]$ with size $(x_p, y_p, \triangle Z)$, where $i \in [0, [\triangle X/x_p]]$, $j \in [0, [\triangle Y/y_p]]$. After that, we respectively encode the height, occupancy, and intensity information in $P[i, j]$. As shown in Equation 6, $H_{channel}[i, j]$ is the z-axis value of the highest point in each pillar. $O_{channel}[i, j]$ indicates whether the pillar contains point. If the pillar contains point, the channel value is set to 1, otherwise 0. $I_{channel}[i, j]$ represents the average intensity value of points in the pillar, where M is the point number in each pillar. Finally, we combine all channels to obtain a BEV image of size $([\triangle X/x_p], [\triangle Y/y_p], 3)$.

$$\begin{cases} O_{channel}[i, j] \in \{0, 1\} \\ I_{channel}[i, j] = I_{sum}[i, j]/M \\ H_{channel}[i, j] = z_{max}[i, j] \end{cases} \tag{6}$$

Furthermore, we adopt ResNet18 [42] to extract features of BEV image encoded by point cloud. Since ResNet18 uses 32x downsampling factor, which makes the output feature map too small, we remove the last ResNet Block in ResNet18 and get 16x downsampling feature map $F_b \in R^{W \times H \times C}$.

Although BEV efficiently represents point clouds, it cannot reflect the geometric characteristic of 3D objects. To compensate for the missing spatial information, we propose MRB in MMInsectDet, which contains two mutually independent voxel backbone and BEV backbone for extracting features of different representations. Similar to SECOND [3], voxel backbone divides points p into ordered voxels of size (x_v, y_v, z_v). Moreover, 3D sparse convolution is utilized to extract rough geometric information of intersections, such as roadside and wall features, to help identify intersections. Then, by merging the z-axis and feature channels, the 3D sparse voxel feature is converted into a 2D dense feature map $F_v \in R^{W \times H \times C}$. Meanwhile, we ensure that the BEV feature F_b has the same size as the voxel feature F_v through setting appropriate parameters, such as $(x_v, y_v, z_v, x_p, y_p, \triangle Z)$ and channel values in the backbone. Finally, simple but efficient element-wise addition is utilized to fuse the discriminative BEV features with the geometric features encoded by voxel backbone to obtain multi-representation feature map $F \in R^{W \times H \times C}$.

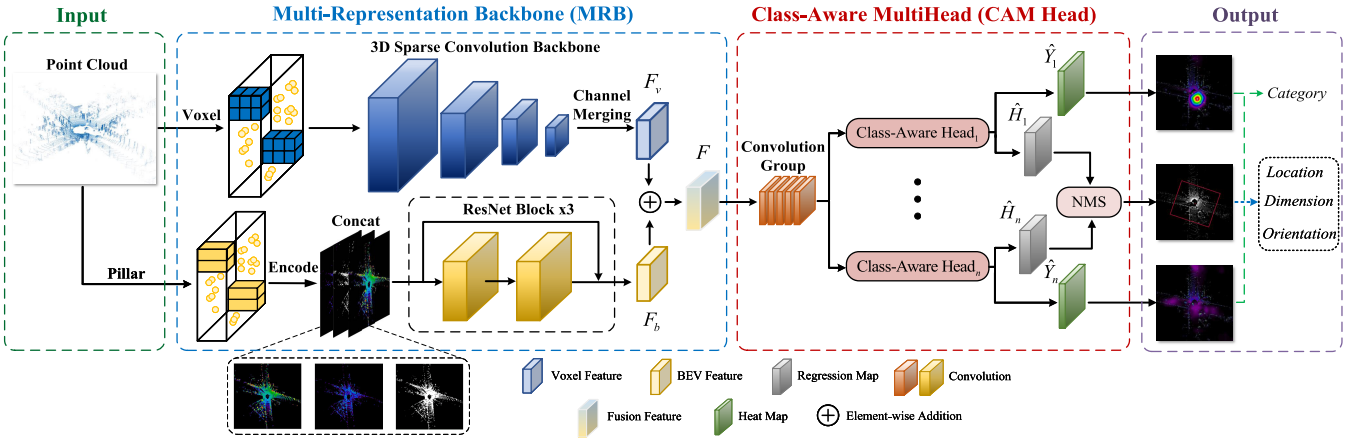


Fig. 6. The MMInsectDet architecture for road intersection detection based point cloud. The network mainly consists of two parts: (a) Multi-Representation Backbone (MRB) contains voxel backbone and BEV backbone, using 3D sparse convolution and simplified ResNet18 to extract features of voxel and BEV, respectively. (b) Class-Aware MultiHead (CAM Head) employs class-special heads to accomplish classification and regression for different categories of intersections. Finally, the candidate boxes generated by different heads are processed by NMS to obtain the final predicted boxes.

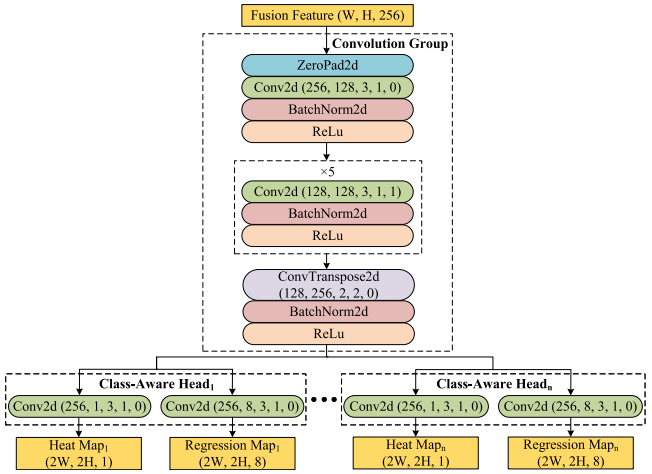


Fig. 7. The details of our proposed CAM Head. Conv2d and ConvTranspose2d mean convolution and transposed convolution. The parameters in them denote input channel, output channel, kernel size, stride, and padding, respectively.

C. Class-Aware MultiHead (CAM Head)

In this part, we improve the head of CenterNet [12] and CenterPoint [12] to make it more suitable for intersection classification and regression. Specially, in CenterNet, classification branch predicts a keypoint heatmap $\hat{Y} \in [0,1]^{W\times H\times K}$, where K is the category number. The area with the highest value in the heatmap is where the target is most likely to exist. Then, in $Y \in [0,1]^{W\times H\times K}$, all ground truth keypoints are generated by Gaussian kernel $Y_{xyk} = \exp\left(-\frac{(x-\tilde{p}_x)^2+(y-\tilde{p}_y)^2}{2\delta_p^2}\right)$, where $(\tilde{p}_x, \tilde{p}_y)$ is the location of keypoints in the feature F and δ_p is the object size-adaptive standard deviation. Additionally, the output $\hat{H} \in R^{W\times H\times 8}$ of regression branch contains location offset $\hat{L} \in R^{W\times H\times 3}$, 3D dimensions $\hat{\Gamma} \in R^{W\times H\times 3}$ and orientation encoding $\hat{A} \in R^{W\times H\times 2}$.

Nevertheless, CenterNet employs a single head to predict multiple categories and the regression is category-independent. Thus, the head has to fit multiple distributions of different class features with large gap, which may exceed the prediction capability of a single head. Meanwhile, various intersection scale further increases the difficulty of predicting multi-classes geometric properties within a single regression map.

To address this problem, we propose CAM Head which consists of multiple heads. Each head is used for class-specific classification and regression. In this way, we transform the multi-classification into multiple simple binary classifications so that each head could focus on a specific category. Meanwhile, since the regression of CAM Head is category-dependent, each regression map can better adapt to the scale changes of a specific category. In CAM Head (Fig. 7), fusion feature F is encoded by the convolution group. After that, the feature is further fed into class-aware head for classification and regression. In the n-th head, we use convolution kernels K_{hn} and K_{rn} to generate heatmap $\hat{Y}_n \in [0,1]^{W\times H\times 1}$ and regression value $\hat{H}_n \in R^{W\times H\times 8}$ respectively. Here, the number of heads n equals two due to the KITTI-Intersection dataset containing two classes.

$$\hat{Y}_n = K_{hn}\text{ConvGroup(F)} \tag{7}$$

$$\hat{H}_n = K_{rn}\text{ConvGroup(F)} \tag{8}$$

Finally, we perform NMS processing on all candidate output boxes from each head to obtain the final predicted boxes. The experiment shows that CAM Head effectively improves the detection performance by separating the classification and regression of different categories but will not introduce too much computational burden.

D. Loss

For the classification loss, we employ the focal loss [43] shown as Equation 9, where $\hat{Y}_{xyn}$ is the class score for location (x, y) in $\hat{Y}_n$ and S is the number of keypoints. Y_{xyn} denotes the ground truth keypoints generated by Gaussian kernel. By default, the hyper-parameter $\alpha = 2.0$, $\beta = 4.0$.

$$L_{heat} = \frac{-1}{S}\sum_{xyn}\begin{cases}(1-\hat{Y}_{xyn})^{\alpha}\log\hat{Y}_{xyn} & \text{if } Y_{xyn}=1\\ (1-Y_{xyn})^{\beta}(\hat{Y}_{xyn})^{\alpha} & \\ \log(1-\hat{Y}_{xyn}) & \text{otherwise}\end{cases} \tag{9}$$

Meanwhile, $\hat{H}_n$ which is the output from regression branch of the n-th class-aware head consists of $\hat{L}_n \in R^{W\times H\times 3}$, $\hat{\Gamma}_n \in R^{W\times H\times 3}$ and $\hat{A}_n \in R^{W\times H\times 2}$. From above three features, we respectively extract $\hat{l}_{ns} \in R^3$, $\hat{\gamma}_{ns} \in R^3$ and $\hat{\alpha}_{ns} \in R^2$ vectors corresponding to the center position of each ground truth instance s. And then we employ L1 loss to compute location and dimension loss in Equation 10.

$$L_{loc} = \frac{1}{S}\sum_{n=1}^{N}\sum_{s=1}^{S_n}|\hat{l}_{ns} - l_{ns}|, L_{dim} = \frac{1}{S}\sum_{n=1}^{N}\sum_{s=1}^{S_n}|\hat{\gamma}_{ns} - \gamma_{ns}| \tag{10}$$

where l_{ns} and γ_{ns} are the ground truth of location offset and dimension, respectively. And the orientation encoding α_{ns} is represented by $(sin(\theta), cos(\theta))$, where θ is the yaw rotation angle. Thus, the orientation loss is calculated as follows:

$$L_{ori} = \frac{1}{S}\sum_{n=1}^{N}\sum_{s=1}^{S_n}|\hat{\alpha}_{ns} - \alpha_{ns}| \tag{11}$$

The total loss is obtained by weighted sum of classification loss and regression losses in Equation 12, where λ represents the weight of different losses.

$$L = \lambda_{heat}L_{heat} + \lambda_{loc}L_{loc} + \lambda_{dim}L_{dim} + \lambda_{ori}L_{ori} \tag{12}$$

V. Experiment

In this section, we first introduce training details and evaluation settings, and then compare our proposed methods with state-of-the-art 3D detection algorithms. Specially, we demonstrate the necessity of labeling subjective intersection size and the role of our modules in improving model performance. Besides, we prove that our method can better overcome point cloud sparsity under harsh conditions. Finally, we illustrate that MRB and CAM Head can be used as plug-and-play modules to improve other methods.

A. Training and Inference Details

Different from previous works, which only train objects within the camera FOV, we view most intersections in the LiDAR scan range as training samples. Thus, we set the detection range within $[-80.0, 80.0]m$ for the X and Y axes, and $[-6.0, 2.0]m$ for the Z axis. Moreover, considering the size of the intersection is 5∼20 times bigger than the foreground object, too small voxel will lead to limited receptive field and huge time consumption. Therefore, for the voxel-based method [3], [12], we set the voxel size to $[1.0, 1.0, 0.2]m$, and each voxel contains at most 100 points. Similarly, the pillar size is set to $[1.0, 1.0, 8.0]m$ for the pillar-based method [2]. In addition, we set the anchor size to $[30.0, 30.0, 3.0]m$ and the rotation angle to $(0, \frac{4}{\pi})$ for anchor-based method [2], [3], [6], [7]. Other settings are the same as previous works. For data augmentation, we employ random flips along the x-axis, random rotations in the angle range $[-\frac{4}{\pi}, \frac{4}{\pi}]$, and random scaling in the scale range $[0.95, 1.05]$. Meanwhile, all algorithms are trained for 80 epochs with a batch size of 16. All experiments are conducted using NVIDIA GTX-3090 GPU.

B. Road Intersection Detection on KITTI-Intersection Dataset

As shown in Table I, our KITTI-Intersection dataset contains 4412 training samples and 775 testing samples. We compare our methods with other state-of-the-art methods in the 3D object detection field. All the methods are trained on training set and evaluated on testing set. Note that we evaluate detection performance of methods in the bird's eye view. The results are summarized in Table II, Table III, and Fig. 8.

1) Quantitative Results: Most of the current state-of-the-art detectors are voxel-based and point-based methods. However, as shown in Table II, in most evaluation metrics, MInsectDet with lightweight BEV Backbone and MMInsectDet based on MRB achieve better results than others. Although MInsectDet ranks second in detection precision, its running speed is twice as fast as CenterPoint [12], reaching 65.0 FPS. Meanwhile, MInsectDet outperforms CenterPoint by 3.64% of mAP. The result demonstrates that the BEV representation has the ability to balance the precision and efficiency in intersection detection. In particular, benefiting from the MRB and CAM Head, the AP of MMInsectDet on Forkroad and Crossroad reaches 87.26% and 91.21%, respectively. Besides, MMInsectDet is only 5.85 FPS slower than CenterPoint, which shows that our proposed modules can improve performance without consuming too many computation resources. Finally, compared with the BEV-based method [44], our proposed methods are also better in most aspects. Specifically, MInsectDet surpasses BirdNet+ on mAP by 3.04%, and its inference speed is significantly faster than BirdNet+. And MMInsectDet also outperforms BirdNet+ in precision and localization accuracy.

As illustrated in Table III, due to re-scoring the predicted boxes, the two-stage detectors Part-A^2 [6] and PV-RCNN [7] achieve the best performance in recall but speed is below 10 FPS. On the contrary, for the single-stage detectors, MMInsectDet and MInsectDet achieve high recall while maintaining good real-time performance.

2) Qualitative Results: Fig. 8 demonstrates some qualitative results of our proposed MMInsectDet on the KITTI-Intersection Dataset. Compared with CenterPoint, MMInsectDet better adapts to the scale change of intersection and achieves more accurate location. Therefore, we believe that our methods have capacity to detect intersection in challenging scenes, no matter narrow alley or large-scale arterial road.

C. Ablation Studies

1) Necessity of Subjective Intersection Boundary: Based on CenterPoint and MMInsectDet, we study the effect of regressing different bounding box attributes on intersection detection. In Table IV, after adding the constraint of subjective boundary, the performance of both CenterPoint and our MMInsectDet has been significantly improved. Specifically, when adding subjective length and width constraints, the ACE of CenterPoint and MMInsectDet decline by $3.14m$ and $0.64m$, respectively. And the overall performance of detectors will be further improved by height constraints. The difference is that after adding orientation constraints, the performance of CenterPoint becomes worse, while MMInsectDet becomes

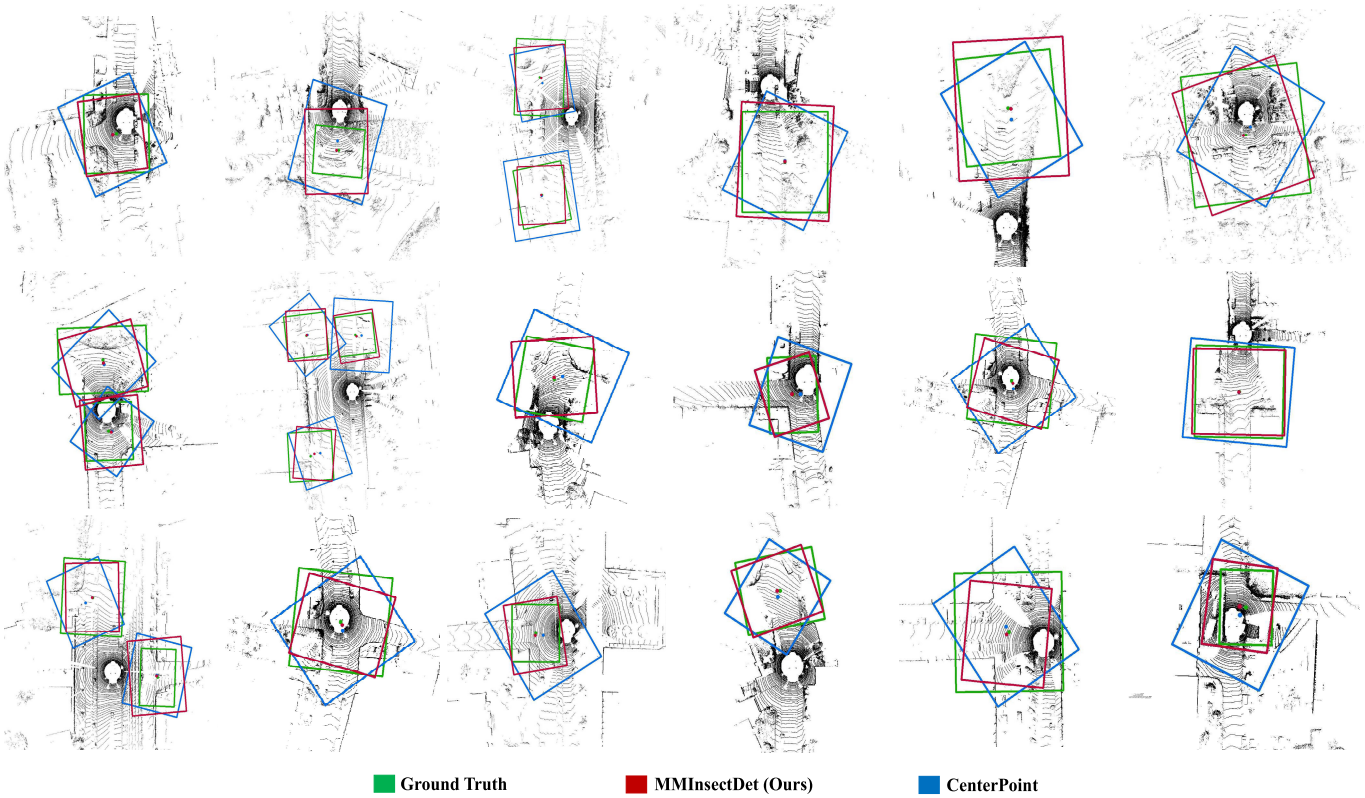


Fig. 8. Qualitative results on the testing set of KITTI-Intersection Dataset. The ground-truth boxes are in green. The predicted boxes of MMInsectDet and CenterPoint are in red and blue, respectively. The center of boxes are displayed by the corresponding color.

TABLE II

PERFORMANCE COMPARISON OF ROAD INTERSECTION DETECTION WITH PREVIOUS 3D DETECTION METHODS. THE AP@0.3 MEANS THE AVERAGE PRECISION WITH CEIOU THRESHOLD 0.3. THE MAP REPRESENTS THE MEAN PRECISION OF ALL CATEGORIES. AND THE ACE IS AVERAGE CENTER ERROR. ↑ IS HIGHER BETTER AND ↓ IS LOWER BETTER

Method	Reference	Method Type	Forkroad AP @0.3 ↑	Crossroad AP @0.3 ↑	mAP @0.3 ↑	ACE ↓	FPS ↑
Part-A^2 [6]	TPAMI 2020	Point + Voxel	82.41	81.64	82.03	4.62	7.98
PV-RCNN [7]	CVPR 2020	Point + Voxel	77.51	74.49	76.00	4.36	6.19
SECOND [3]	Sensors 2018	Voxel	83.52	71.60	77.56	4.59	21.09
PointPillars [2]	CVPR 2019	Pillar	71.69	51.69	61.69	4.90	54.42
BirdNet+ [44]	IEEE Access 2021	BEV	82.43	86.33	84.38	4.36	26.28
CenterPoint [12]	CVPR 2021	Voxel	83.12	84.44	83.78	4.51	33.04
MInsectDet (Ours)	-	BEV	85.89	88.95	87.42	4.44	**65.00**
MMInsectDet (Ours)	-	BEV + Voxel	**87.26**	**91.21**	**89.23**	**4.25**	27.19

TABLE III

RECALL OF OUR METHODS AND PREVIOUS 3D DETECTION METHODS WITH CEIOU THRESHOLD 0.5. ↑ IS HIGHER BETTER

Method	Method Type	Stage	Recall @0.5 ↑
Part-A^2 [6]	Point + Voxel	Two-stage	88.67
PV-RCNN [7]	Point + Voxel	Two-stage	**88.77**
SECOND [3]	Voxel	One-stage	79.61
PointPillars [2]	Pillar	One-stage	78.58
BirdNet+ [44]	BEV	Two-stage	80.52
CenterPoint [12]	Voxel	One-stage	80.77
MInsectDet (Ours)	BEV	One-stage	83.35
MMInsectDet (Ours)	BEV + Voxel	One-stage	83.10

TABLE IV

EFFECT OF REGRESSING DIFFERENT BOUNDING BOX ATTRIBUTES ON THE PERFORMANCE. THE BLANK MEANS THAT IT DOES NOT PARTICIPATE IN THE REGRESSION LOSS

Method	x	y	z	w	l	h	ry	ACE	Recall @0.5	mAP @0.3
CenterPoint [12]	✓	✓	✓					7.78	-	-
	✓	✓	✓	✓	✓			4.64	55.61	83.42
	✓	✓	✓	✓	✓	✓		**4.38**	**81.03**	**85.03**
	✓	✓	✓	✓	✓	✓	✓	4.51	80.77	83.78
MMInsectDet	✓	✓	✓					5.02	-	-
	✓	✓	✓	✓	✓			4.38	70.45	88.10
	✓	✓	✓	✓	✓	✓		4.26	81.29	88.75
	✓	✓	✓	✓	✓	✓	✓	**4.25**	**83.10**	**89.23**

better. This result shows that regressing intersection orientation is a challenging problem for CenterPoint. In contrast, our method can better solve this issue, which is also demonstrated in Fig. 8. In summary, complete intersection size constraints are beneficial to improve detection performance. And our method can solve the problem of orientation prediction better.

Additionally, as shown in Table V, the size-changing of ground-truth box would not obviously influence the detection performance. For CenterPoint and MMInsectDet, the largest deviations in ACE, mAP, and recall are 0.18m, 1.03%, and 1.17%, respectively. Therefore, we believe that labeling subjective boundary of the intersection is necessary and would not cause too many negative effects.

2) Effect of MRB and CAM Head on Detection Performance: In this section, we analyze the impact of different

TABLE V

EFFECT OF DIFFERENT SIZES OF THE GROUND-TRUTH BOX ON THE PERFORMANCE. OFFSET REPRESENTS SIMULTANEOUSLY INCREASING OR DECREASING THE WIDTH AND LENGTH OF GROUND-TRUTH BOX

Method	Offset (m)	ACE	mAP @0.3	Recall @0.5
CenterPoint [12]	-4	$4.51_{\uparrow 0.00}$	$83.04_{\downarrow 0.74}$	$79.60_{\downarrow 1.17}$
	-2	$4.57_{\uparrow 0.06}$	$83.39_{\downarrow 0.39}$	$79.86_{\downarrow 0.91}$
	0	4.51	83.78	80.77
	+2	$4.63_{\uparrow 0.12}$	$84.81_{\uparrow 1.03}$	$81.41_{\uparrow 0.64}$
	+4	$4.60_{\uparrow 0.09}$	$83.82_{\uparrow 0.04}$	$80.89_{\uparrow 0.12}$
MMInsectDet	-4	$4.21_{\downarrow 0.04}$	$88.26_{\downarrow 0.97}$	$82.07_{\downarrow 1.03}$
	-2	$4.07_{\downarrow 0.18}$	$88.78_{\downarrow 0.45}$	$82.84_{\downarrow 0.26}$
	0	4.25	89.23	83.10
	+2	$4.25_{\downarrow 0.00}$	$90.08_{\uparrow 0.85}$	$83.88_{\uparrow 0.78}$
	+4	$4.13_{\downarrow 0.12}$	$89.22_{\downarrow 0.01}$	$82.86_{\downarrow 0.24}$

TABLE VI

EFFECT OF MRB AND CAM HEAD ON DETECTION PERFORMANCE

Backbone	Head	ACE	mAP @0.3	Recall @0.5
Voxel	Single	4.51	83.78	80.77
BEV	Single	4.34	83.51	82.32
MRB	Single	4.54	83.53	79.74
Voxel	CAM	4.33	87.28	82.84
BEV	CAM	4.44	87.42	**83.35**
MRB	CAM	**4.25**	**89.23**	83.10

backbones and heads on the performance with CenterPoint as our baseline. Note that CenterPoint extracts features through voxel backbone and uses a single head to predict. The results are shown in Table VI.

Firstly, we replace the voxel backbone of baseline with BEV (2^{nd} row). In this way, ACE is reduced by $0.17m$, and recall is increased by 1.55%, while mAP is only decreased by 0.27%. The result shows that BEV backbone is very suitable for extracting discriminative intersection features.

Moreover, we combine the BEV backbone with CAM Head (5^{th} row), compared with BEV backbone alone (2^{nd} row), the mAP and recall are increased by 3.91% and 1.03%, respectively. This reflects that separating classification and regression of different categories in the CAM Head could enhance detection performance. In addition, as shown in the 4^{th} row, replacing the single head in the baseline with the CAM Head can also achieve performance improvement, which further confirms the effectiveness of the CAM Head.

Finally, we try to directly replace the backbone of baseline with MRB. As shown in the 3^{rd} row, the limited prediction ability of single head results in that richer features cannot help the network obtain better results. However, when MRB is combined with CAM Head (6^{th} row), it can play an interactive effect and achieve the best performance. That is, the ACE is $4.25m$, the mAP and recall achieve 89.23% and 83.10%, respectively.

3) Effect of MRB and CAM Head on Robustness: In previous work, foreground objects are usually sensitive to point cloud sparsity. Therefore, for intersection detection, we downsample the point cloud by 50%, 75%, 87.5%, and 93.75% to check the impact of different backbones and heads on the robustness. As shown in Table VII, since the number of background points belonging to intersection far exceeds that of foreground, when the downsampling ratio achieves 87.5%, the detectors begin to show significant performance degradation.

a) Analysis of mAP: When the downsampling ratio is 50%, MInsectDet achieves mAP of 89.86% with 2.44% improvement. The result shows that random sampling of points would filter out some foreground points while retaining most background points, thus improving the precision. Meanwhile, MInsectDet maintains a minimal precision drop for all downsampling ratios. The average precision drop of CenterPoint* (BEV Backbone + Single Head) is 12.19%, while MInsectDet is only 6.69%, which demonstrates that CAM Head could improve robustness. Additionally, as shown in Fig. 9, we compare our methods with others in different downsampling ratios. In the case of different point cloud sparsity, MInsectDet and MMInsectDet are better than other methods in terms of detection precision.

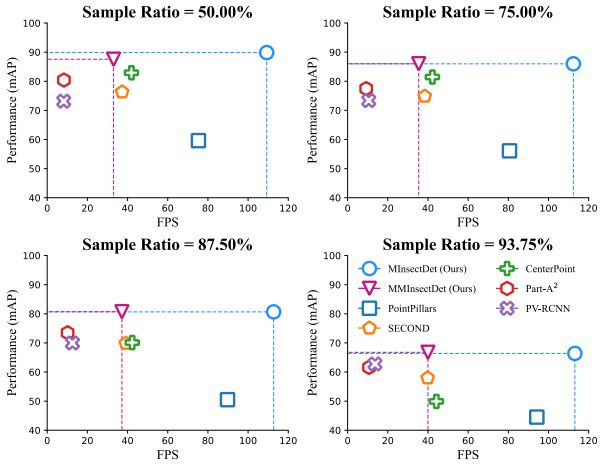


Fig. 9. Comparison of our methods and others under different point downsampling ratios. Note that when the downsampling ratio equals 50%, the mAP of MInsection is 89.86% and the inference speed reaches 109 FPS.

b) Analysis of Recall: Compared with CenterPoint, CenterPoint** (MRB + Single Head) can maintain relatively high recall in different downsampling ratios, which shows that the rich features brought by MRB can improve the robustness. Meanwhile, MMInsectDet has good robustness in the harshest environment. Specially, when the downsampling ratio reaches 93.75%, it can obtain a relatively small decrease in recall.

4) Effect of Object Distance on Detection Performance: As shown in Table. VIII, we count the number of intersections in a set of distance intervals and explore the impact of object distance on performance. Compared with CenterPoint [12], our MMInsectDet achieves better Recall in most distance intervals. Specially, when the distance from the intersection surpasses $60m$, our method outperforms CenterPoint by 5.41%, 10.81% and 18.92% in different CEIOU thresholds respectively. Additionally, we test how far an intersection can be detected in our KITTI-Intersection dataset. The result shows that CenterPoint and our MMInsectDet can detect intersections up to distances of $62.31m$ and $65.12m$ with CEIOU threshold of 0.7.

D. Generalization Analysis of MRB and CAM Head

To validate the generalization ability of our modules, we replace the corresponding parts in SECOND [3] and PointPillars [2] with MRB and CAM Head. Since both SECOND and PointPillars are anchor-based methods, we adopt anchor-based CAM Head to replace their heads. As shown in Table IX, although CAM Head causes a slight drop in localization

TABLE VII

EFFECT OF MRB AND CAM HEAD ON ROBUSTNESS. THE RESULTS ARE THE DETECTION PERFORMANCE UNDER DIFFERENT POINT DOWNSAMPLING RATIOS. * MEANS ADOPTING BEV BACKBONE, WHILE ** MEANS ADOPTING MRB AS THE BACKBONE

Metrics	mAP @0.3					Recall @0.5				
Sample Ratio	0%	50.00%	75.00%	87.50%	93.75%	0%	50.00%	75.00%	87.50%	93.75%
CenterPoint [12]	83.78	$82.92_{\downarrow 0.86}$	$81.46_{\downarrow 2.32}$	$70.12_{\downarrow 13.66}$	$49.91_{\downarrow 33.87}$	80.77	$79.10_{\downarrow 1.67}$	$77.29_{\downarrow 3.48}$	$72.13_{\downarrow 8.64}$	$60.13_{\downarrow 20.64}$
CenterPoint*	83.51	$81.67_{\downarrow 1.84}$	$78.40_{\downarrow 5.11}$	$73.07_{\downarrow 10.44}$	$53.51_{\downarrow 30.00}$	82.32	$81.16_{\downarrow 1.16}$	$77.68_{\downarrow 4.46}$	$74.97_{\downarrow 7.35}$	$62.45_{\downarrow 19.87}$
CenterPoint**	83.53	$81.18_{\downarrow 2.25}$	$78.56_{\downarrow 4.87}$	$72.76_{\downarrow 10.77}$	$52.67_{\downarrow 30.86}$	79.74	$79.61_{\downarrow 0.13}$	$76.90_{\downarrow 2.84}$	$72.26_{\downarrow 7.48}$	$61.03_{\downarrow 18.71}$
MInsectDet (Ours)	87.42	$89.86_{\uparrow 2.44}$	$86.00_{\downarrow 1.42}$	$80.66_{\downarrow 6.76}$	$66.40_{\downarrow 21.02}$	83.35	$82.58_{\downarrow 0.77}$	$80.90_{\downarrow 2.45}$	$74.58_{\downarrow 8.77}$	$63.74_{\downarrow 19.61}$
MMInsectDet (Ours)	89.23	$87.57_{\downarrow 1.66}$	$86.02_{\downarrow 3.21}$	$80.71_{\downarrow 8.52}$	$66.77_{\downarrow 22.46}$	83.10	$80.90_{\downarrow 2.20}$	$80.13_{\downarrow 2.97}$	$75.35_{\downarrow 7.75}$	$66.71_{\downarrow 16.39}$

TABLE VIII

EFFECT OF OBJECT DISTANCE ON DETECTION PERFORMANCE

Method	Distance	$0\sim20m$	$20\sim40m$	$40\sim60m$	$>60m$
	Instance Num.	452	223	63	37
CenterPoint [12]	Recall @0.3	96.46	93.27	93.65	91.89
	Recall @0.5	83.18	81.17	69.84	81.08
	Recall @0.7	48.45	43.50	34.92	32.43
MMInsectDet	Recall @0.3	97.12	96.41	92.06	97.30
	Recall @0.5	86.06	84.30	73.02	91.89
	Recall @0.7	52.88	48.88	41.27	51.35

TABLE IX

GENERALIZATION ABILITY OF OUR MODULES IN SECOND AND POINTPILLARS. CAM HEAD† MEANS ANCHOR-BASED CLASS-AWARE MULTIHEAD

Method	Module	ACE	mAP @0.3	Recall @0.5
SECOND [3]	-	4.59	77.56	79.61
	+ CAM Head†	$4.60_{\uparrow 0.01}$	$82.80_{\uparrow 5.24}$	$83.67_{\uparrow 4.06}$
	+ CAM Head†+ MRB	$4.38_{\downarrow 0.21}$	$84.21_{\uparrow 6.65}$	$83.87_{\uparrow 4.26}$
PointPillars [2]	-	4.90	61.69	78.58
	+ CAM Head†	$5.01_{\uparrow 0.11}$	$70.95_{\uparrow 9.26}$	$81.29_{\uparrow 2.71}$
	+ CAM Head†+ MRB	$4.79_{\downarrow 0.11}$	$78.35_{\uparrow 16.66}$	$81.94_{\uparrow 3.36}$

accuracy, it significantly improves the detection performance. Subsequently, we further replace backbone in [3] and [2] with MRB. Compared with original model, the mAP and recall are improved by 6.65% and 4.26%, 16.66% and 3.36% for SECOND and PointPillars, respectively. Meanwhile, the ACE is also reduced by $0.21m$ and $0.11m$. Therefore, it is verified that our proposed methods have good generalization ability and can be used as plug-and-play modules.

VI. FUTURE WORK DISCUSSION

Due to the limited computational resources in autonomous vehicles, the lightweight network is the prerequisite for real-time environmental perception. Although our MInsectDet and MMInsectDet have achieved 65.00 and 27.19 FPS on NVIDIA GTX-3090 GPU, their running speed still has great potential for improvement. For example, we can distil our frameworks to obtain a smaller, faster and more accurate detector through the teacher-student distillation model [45]. Moreover, replacing the vanilla convolution with dilated convolution (eg. [46]) is another way to save memory costs. Apart from optimizing algorithm, in the future, we will continue to increase intersection types and expand the scale of road intersection dataset.

VII. CONCLUSION

In this work, we propose a KITTI-Intersection Dataset, the first LiDAR-based road intersection dataset, which provides a unified benchmark for subsequent related research. To fairly evaluate the performance of different methods in the intersection detection task, we utilize Average Center Error to measure localization accuracy. Besides, we present CEIOU as a novel metric to distinguish predicted boxes with different quality.

Furthermore, to address intersection detection problem, we introduce MInsectDet and MMInsectDet methods to obtain intersection category, location, dimension, and orientation. In MInsectDet, we extract discriminative features through an efficient BEV Backbone. In MMInsectDet, we propose Multi-Representation Backbone to obtain the complete intersection features. Moreover, we propose Class-Aware MultiHead to improve the classification and regression capabilities of our methods for intersection detection. Experimental results on the KITTI-Intersection Dataset show that our MInsectDet and MMInsectDet can achieve competitive performance.

REFERENCES

[1] 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.

[2] 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.

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

[4] H. Li, S. Zhao, W. Zhao, L. Zhang, and J. Shen, "One-stage anchor-free 3D vehicle detection from LiDAR sensors," *Sensors*, vol. 21, no. 8, p. 2651, Apr. 2021.

[5] 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.

[6] 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. 2021.

[7] 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.

[8] D. Zhou et al., "Joint 3D instance segmentation and object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1836–1846.

[9] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. Van Gool, "Towards a weakly supervised framework for 3D point cloud object detection and annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4454–4468, Aug. 2022.

[10] 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.

[11] 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.

[12] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11779–11788.

[13] S. Yang, H. Lu, and J. Li, "Multifeature fusion-based object detection for intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1126–1133, Jan. 2023.

[14] K. Zhao et al., "3D vehicle detection using multi-level fusion from point clouds and images," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15146–15154, Sep. 2022.

[15] D. Zhou, X. Song, J. Fang, Y. Dai, H. Li, and L. Zhang, "Context-aware 3D object detection from a single image in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18568–18580, Oct. 2022.

[16] P. S. Zaki, M. M. William, B. K. Soliman, K. G. Alexsan, K. Khalil, and M. El-Moursy, "Traffic signs detection and recognition system using deep learning," 2020, *arXiv:2003.03256*.

[17] T. Wu and A. Ranganathan, "A practical system for road marking detection and recognition," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 25–30.

[18] M. Haris and A. Glowacz, "Lane line detection based on object feature distillation," *Electronics*, vol. 10, no. 9, p. 1102, May 2021.

[19] C. G. Serna and Y. Ruichek, "Classification of traffic signs: The European dataset," *IEEE Access*, vol. 6, pp. 78136–78148, 2018.

[20] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. Computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.

[21] O. Jayasinghe, S. Hemachandra, D. Anhettigama, S. Kariyawasam, R. Rodrigo, and P. Jayasekara, "CeyMo: See more on roads–A novel benchmark dataset for road marking detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3381–3390.

[22] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 2341–2345.

[23] S. Shirke and R. Udayakumar, "Lane datasets for lane detection," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2019, pp. 792–796.

[24] 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.

[25] 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.

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

[27] 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.*, vol. 30, 2017, pp. 5105–5114.

[28] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[29] J. Yin et al., "Semi-supervised 3D object detection with proficient teachers," in *Proc. ECCV*, 2022, pp. 727–743.

[30] J. Yin et al., "ProposalContrast: Unsupervised pre-training for LiDAR-based 3D object detection," in *Proc. ECCV*, 2022, pp. 17–33.

[31] J. Yin, J. Shen, X. Gao, D. J. Crandall, and R. Yang, "Graph neural network and spatiotemporal transformer attention for 3D video object detection from point clouds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, pp. 9822–9835, 2023, doi: 10.1109/TPAMI.2021.3125981.

[32] T. M. Jochem, D. A. Pomerleau, and C. E. Thorpe, "Vision-based neural network road and intersection detection and traversal," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Systems. Hum. Robot Interact. Cooperat. Robots*, Aug. 1995, pp. 344–349.

[33] T. Koji and T. Kanji, "Deep intersection classification using first and third person views," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 454–459.

[34] D. Bhatt, D. Sodhi, A. Pal, V. Balasubramanian, and M. Krishna, "Have I reached the intersection: A deep learning-based approach for intersection detection from monocular cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4495–4500.

[35] A. L. Ballardini, Á. H. Saz, and M. Á. Sotelo, "Model guided road intersection classification," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 703–709.

[36] N. Sugimoto, S. Ikehata, and K. Aizawa, "Intersection prediction from single 360° image via deep detection of possible direction of travel," 2022, *arXiv:2204.04634*.

[37] D. Habermann, C. E. O. Vido, F. S. Osório, and F. Ramos, "Road junction detection from 3D point clouds," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4934–4940.

[38] A. Y. Hata, D. Habermann, F. S. Osorio, and D. F. Wolf, "Road geometry classification using ANN," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 1319–1324.

[39] D. Zeng Wang and I. Posner, "Voting for voting in online point cloud object detection," in *Proc. Robot., Sci. Syst. XI*, Rome, Italy, Jul. 2015, pp. 10–15.

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

[41] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3517–3523.

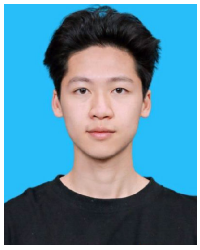
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[44] A. Barrera, J. Beltrán, C. Guindel, J. A. Iglesias, and F. García, "BirdNet+: Two-stage 3D object detection in LiDAR through a sparsity-invariant Bird's eye view," *IEEE Access*, vol. 9, pp. 160299–160316, 2021.

[45] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled Siamese networks for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8896–8909, Dec. 2022.

[46] Z. Zhao, S. Zhao, and J. Shen, "Real-time and light-weighted unsupervised video object segmentation network," *Pattern Recognit.*, vol. 120, Dec. 2021, Art. no. 108120.



Zhiheng Li received the B.S. degree from the Civil Aviation University of China. He is currently pursuing the M.S. degree with Northeastern University. His research interests include deep learning, visual perception, and 3D object tracking.

Yubo Cui received the B.S. degree in automation and the M.S. degree in robot science and engineering from Northeastern University, China, in 2017 and 2020, respectively, where he is currently pursuing the Ph.D. degree. His research interests include deep learning, 3D object tracking, and 3D detection.

Zheng Fang (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, China, in 2002 and 2006, respectively. He was a Post-Doctoral Research Fellow with Carnegie Mellon University from 2013 to 2015. He is currently a Full Professor with the Faculty of Robot Science and Engineering, Northeastern University. His research interests include visual/laser SLAM, perception, and autonomous navigation of various mobile robots.