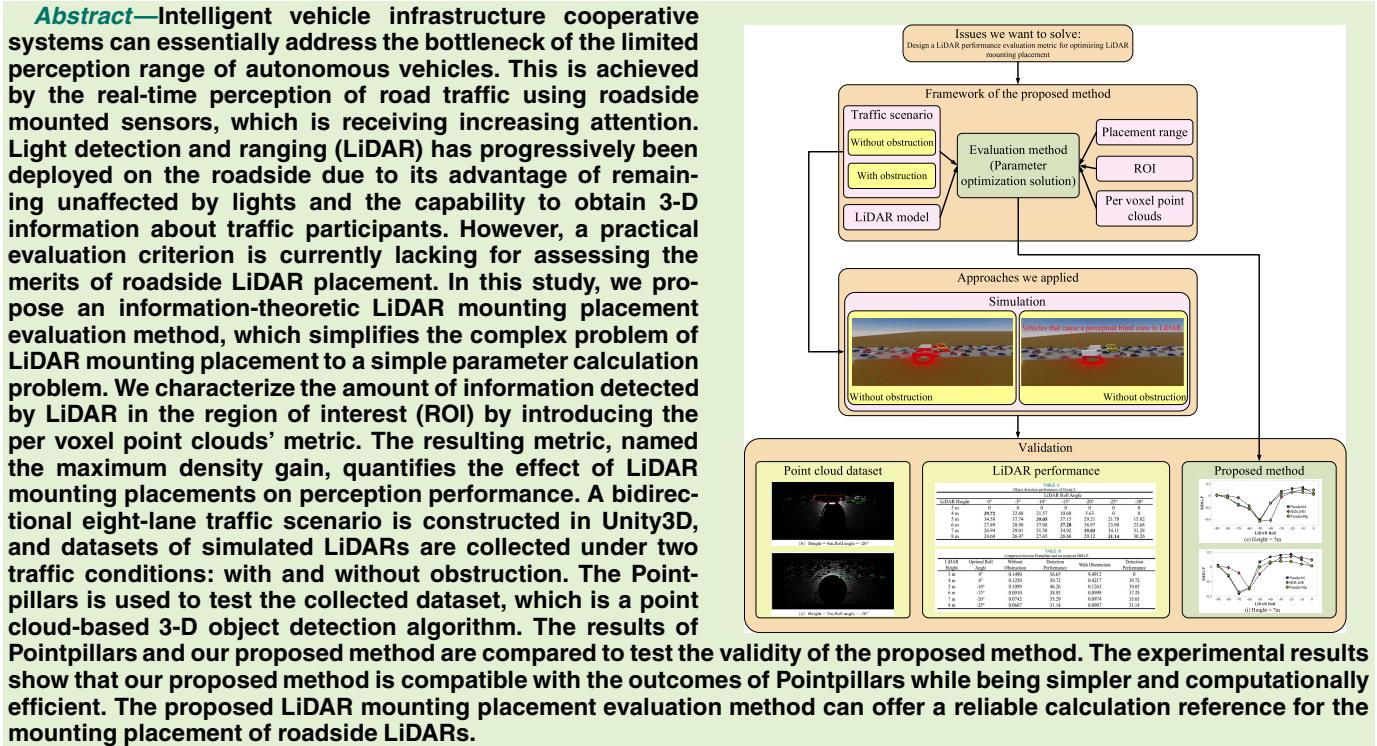


# A Novel Information Theory-Based Metric for Evaluating Roadside LiDAR Placement

Shaojie Jin<sup>ID</sup>, Ying Gao, Fei Hui, Xiangmo Zhao<sup>ID</sup>, Cheng Wei, Tao Ma, and Weihao Gan



**Index Terms**—Entropy, information theory, light detection and ranging (LiDAR), mounting placement, performance analysis.

## I. INTRODUCTION

AUTONOMOUS vehicles (AVs) are designed to free drivers from driving tasks and are expected to improve

Manuscript received 14 June 2022; revised 7 August 2022 and 25 August 2022; accepted 4 September 2022. Date of publication 13 September 2022; date of current version 31 October 2022. This work was supported in part by the National Key Research and Development Program under Grant 2021YFB2501204, in part by the National Natural Science Foundation of China under Grant 52172380, and in part by the Fundamental Research Funds for the Central Universities, CHD, under Grant 300102242711. The associate editor coordinating the review of this article and approving it for publication was Prof. Dongsoo Har. (*Corresponding author: Fei Hui*.)

Shaojie Jin, Ying Gao, Fei Hui, Xiangmo Zhao, and Cheng Wei are with the School of Information and Engineering, Chang'an University, Xi'an, Shaanxi 710064, China (e-mail: jinshaojie@chd.edu.cn; gaoying888@chd.edu.cn; feihui@chd.edu.cn; xmzhao@chd.edu.cn; chengwei@chd.edu.cn).

Tao Ma and Weihao Gan are with the Autonomous Driving Group, SenseTime, Shanghai 200000, China (e-mail: matao@pjlab.org.cn; ganweihao@sensetime.com).

Digital Object Identifier 10.1109/JSEN.2022.3204836

traffic safety and efficiency [1]. However, the AVs have a limited perceptive range and precision. The intelligent vehicle infrastructure cooperative systems (IVICSs) based on the AVs complete the collection of all-time dynamic traffic information through roadside sensors and realize dynamic real-time information interaction among traffic participants through wireless communication and other technologies. This assists the AVs in environmental perception, calculation and decision-making, and control and execution, and accelerates the maturity of AV applications [2].

Cameras have been frequently employed as roadside sensors since they can provide frames closest to human vision. However, their visual effect is poor in bad weather, backlight or complex light, and shadow, and it is difficult to ensure high reliability. Therefore, light detection and ranging (LiDAR), thanks to its high resolution and advantage of real-time 3-D modeling of the traffic environment, has become a critical sensor in high-grade AVs [3], [4], [5]. In recent years, most LiDAR research in the realm of AVs has focused on improv-

ing deep learning models to enhance sensing performance. Research on roadside LiDAR placements is still in its infancy, and there are no widely agreed standards that can evaluate the benefits of LiDAR placements and determine the optimal installation locations.

However, evaluating the effectiveness of different LiDAR placements based on real-world sensing performance is extremely challenging. The reason is that it necessitates a full cycle of LiDAR deployment, data collection, model training, and evaluation, which can be extremely costly and time-consuming [6]. When using multiple LiDARs, the cost of evaluation and optimization increases exponentially. In addition, as it is not possible to collect two identical point-cloud frames in the real world, it is difficult to guarantee the fairness of LiDAR placement evaluation. Therefore, quantitative evaluation of roadside LiDAR placement is a vital yet unsolved problem. An in-depth investigation of the interaction between LiDAR placement and sensing performance is crucial for roadside sensing systems in IVICS, as it can save both deployment costs and maximize LiDAR sensing performance. Therefore, a practical indicator has to be developed for evaluating the LiDARs' placement. In this article, we propose a practical indicator that can address this problem. Our main contributions to this article are given as follows.

- 1) We propose a surrogate metric to evaluate the effect of LiDAR placement on perception performance by obtaining the per voxel point clouds (PVPCs). We show the correlation between the proposed metric and the detection performance based on Pointpillars, and the results validate the proposed method.
- 2) We collect a dataset corresponding to two typical traffic scenarios in the Unity3D simulator: without obstruction and with obstruction. Pointpillars is applied to the collected dataset to evaluate the LiDARs' placement. The results reveal that LiDAR placement plays an important role in its perceptual performance.
- 3) We investigate the relationship between LiDAR placement and perceptual performance using the proposed method, which can serve as a reference for choosing the deployment parameters of LiDAR in the real world.

The rest of this article is organized as follows. A brief literature review is given in Section II. In Section III, we elaborate on the framework of the proposed LiDAR performance quantification method. Experimental results are provided in Section IV. Conclusions and future work are finally presented in Section V.

## II. LITERATURE REVIEW

### A. LiDAR-Based 3-D Object Detection

LiDAR sensors are commonly used in AVs and robotics to capture 3-D scene information as sparse and irregular point clouds, which offer crucial cues for 3-D scene perception and comprehension [4]. Most existing 3-D detection methods may be divided into two categories with respect to point cloud representations: grid-based approaches and point-based approaches.

To deal with point clouds' irregular data format, grid-based approaches often convert them to regular representations, such as 3-D voxels [7], [8], [9], [10] or 2-D bird-view

maps [11], [12], [13], [14], [15], [16], [17], [18]. These representations may then be efficiently processed using 2-D or 3-D convolutional neural networks (CNNs) to train point features for 3-D detection. The pioneering work MV3D [11] projects point clouds onto 2-D bird-view grids and uses a large number of predefined 3-D anchors to generate 3-D bounding boxes. Subsequent works [13], [17] improve multisensor fusion strategies, while Yang *et al.* [14], [15] and Lang *et al.* [16] propose more efficient frameworks with a bird-view representation. Some other works [3], [9] partition the point clouds into 3-D voxels that are processed by a 3-D CNN. The 3-D sparse convolution [19] is introduced in [8] for efficient 3-D voxel processing. Wang *et al.* [20] and Zhu *et al.* [21] employ several detecting heads, whereas Shi *et al.* [5] investigate the positions of object parts in order to improve the performance. The receptive fields are constrained by the kernel size of 2-D/3-D convolutions although these grid-based approaches are often efficient for accurate 3-D proposal creation.

Although grid-based approaches are computationally more efficient compared to point-based approaches, information loss inevitably affects their fine-grained localization accuracy. On the other hand, point-based methods have a higher computational cost but can easily achieve a broader receptive field due to point set abstraction. The F-PointNet [22] first proposes to apply PointNet [23], [16] for discerning objects in 3-D from cropped point clouds based on 2-D picture bounding boxes. For 3-D detection using only point clouds, the PointRCNN [24] generates 3-D proposals directly from the whole point clouds instead of 2-D images, while STD [25] proposes the sparse-to-dense technique for better proposal refining. Qi *et al.* [26] propose the Hough voting approach for better object feature grouping. These point-based approaches are mostly based on the PointNet series, particularly the set abstraction operation [16], which allows for configurable receptive fields for learning point cloud features. Liu *et al.* [27] combine both voxel-based CNN and point-based SharedMLP for efficient point cloud feature learning. In conclusion, representation learning on point clouds is still an important topic worthy of study for improving the performance of point cloud classification and segmentation [9], [16], [23], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37].

### B. Sensor Placement

In the literature, there are generally two kinds of formulations for optimal sensor mounting placement problems: 1) optimal control [38], [39] and 2) parameter optimization [40], [41]. The optimal control formulation is typically used for cooperative path planning [38], [39], with the goal of estimating the target location, on the one hand, and planning the path of sensor platforms in order to reduce estimation uncertainty, on the other hand. This problem is also referred to as simultaneous localization and planning (SLAP). In the SLAP problem, target motion and sensor measurement models are considered process and measurement models, respectively. The Kalman filter is commonly used to describe the estimation covariance and estimate the target position. An optimal control problem is designed in order to minimize the estimation covariance. The disadvantage of this type of formulation is that

numerical methods are usually required to solve the optimal control with various restrictions. It is frequently impossible to obtain analytical properties [42].

Many research approaches, including the one reported in this study, consider the optimal placement problem as a parameter optimization problem in order to avoid the complexity of the optimal control problem. This type of formulation has a long history and has been thoroughly studied in a number of publications, including [40], [41], [43], and [44]. This formulation assumes that a rough estimate of the target position has already been acquired using other approaches. This rough estimate can be used to optimize the sensor placements and characteristics, resulting in a more accurate target localization based on the optimized placement [45]. In most parameter optimization formulations, the Fisher information matrix (FIM) is used as the objective function. The FIM is the inverse of the Cramer–Rao lower bound (CRLB), which is the lowest estimation covariance that may be achieved. The term efficient refers to an unbiased estimator that meets the CRLB. An optimal placement that maximizes an FIM function, such as the determinant, can be considered equivalent to maximizing the target information acquired by the sensors while simultaneously minimizing the estimation covariance of any effective estimators [46].

### C. Objectives

Compared to the optimal control formulation, the parameter optimization formulation can be solved analytically. The analytical solutions are important as they can provide insight into the effect of LiDAR mounting placement on LiDAR performance. Therefore, in this article, an information theory-based LiDAR performance evaluation method is proposed based on the parameter optimization perspective to find the optimal LiDAR mounting placement on the roadside.

## III. METHODOLOGY

This section describes the method to efficiently evaluate LiDAR placement for perception performance based on the information theory. The proposed method is designed to avoid the extensive effort required for data collection and complicated analysis of learning-based detection for evaluating each LiDAR placement. The problem of LiDAR placement evaluation is formalized, and the terms region of interest (ROI) and PVPC are introduced. We show how to calculate the surrogate metric for quick evaluation of LiDAR configurations. Fig. 1 shows a general overview of the proposed framework, and more details will be described in the following context.

First, we should determine the configuration information of the LiDAR. The configuration includes the placement and LiDAR model. The installation range includes the range of translation [ $x, y, z$ ] and rotation angle [roll, pitch, yaw] of the LiDAR with respect to the origin of the world coordinate system. Different LiDAR models correspond to different fields of views (FOVs), the number of beams, and related parameters. Second, we should define the ROI. After determining these inputs, we need to calculate the quantified results of the LiDAR sensing capability. In these calculations, we compute

the LiDAR sensing capability with and without obstruction, respectively. In this case, occlusion refers to the situation when high-roof vehicles, such as trucks and buses, appear in the vicinity of the LiDAR and cause obstruction to the LiDAR perception FOV. After calculating the perception capability with and without obstruction separately, we merge the two cases to obtain the combined sensing capability of LiDAR. The merged calculation here refers to assigning corresponding weights to the obstruction cases according to the frequency of obstruction cases in the real-world scenes and then weighting them with the nonobstruction cases to increase the usability of the final calculation results. Finally, we validate the proposed method by simulation experiments.

### A. Definition of ROI

This section begins with the definition of ROI, which is the foundation of the LiDAR configuration evaluation problem. As in previous work [6], [47], we define the ROI as the space where we keep track of the objects to be detected. To accommodate for the LiDAR's limited detection range, we designate the ROI's cuboid dimensions as  $[l, w, h]$  in the XYZ coordinate system. The dimensions are given as length  $l$  along the  $x$ -axis, width  $w$  along the  $y$ -axis, and height  $h$  along the  $z$ -axis, as illustrated in Fig. 2. The length  $l$ , the width  $w$ , and the height  $h$  are mainly determined by the maximum LiDAR sensing range, the total road width, and the highest height of a moving vehicle, respectively. Subsequently, we split the ROI into voxels with a fixed resolution  $\gamma$  to represent ROI as the set of voxels as follows:

$$V = \{v_1, v_2, \dots, v_N\}, \quad N = \frac{l}{\gamma} \times \frac{w}{\gamma} \times \frac{h}{\gamma} \quad (1)$$

where  $N$  represents the total number of voxels in the ROI. The ROI provides us with a fixed perception field. Each voxel can be expressed as  $v_i = [v_x, v_y, v_z]$  with the size of  $0.1 \text{ m} \times 0.1 \text{ m} \times 0.1 \text{ m}$  in the world coordinate system. Consequently, when computing the perception metrics for the LiDAR-based 3-D object detection task, we only consider the objects within the ROI for each point cloud frame.

### B. Quantitative Calculation of LiDAR Sensing Performance

It is incorrect and inefficient to evaluate LiDAR placement in the real world using 3-D object detection metrics because it is difficult to collect point cloud data that are the same for all scenes and objects under different LiDAR placements. Furthermore, each evaluation procedure for LiDAR configuration may require several days for data collection and training the detection models in order to obtain final detection metrics. In order to simplify the calculation process and enable a fair comparison of results, a new surrogate metric based on the PVPC is proposed in this section.

**1) Per Voxel Point Clouds:** As we consider the 3-D object detection in ROI among all the frames, intuitively, the LiDAR configurations that acquire a higher number of point clouds will perform better in the object detection task. To this end, we propose to count the number of point clouds within each

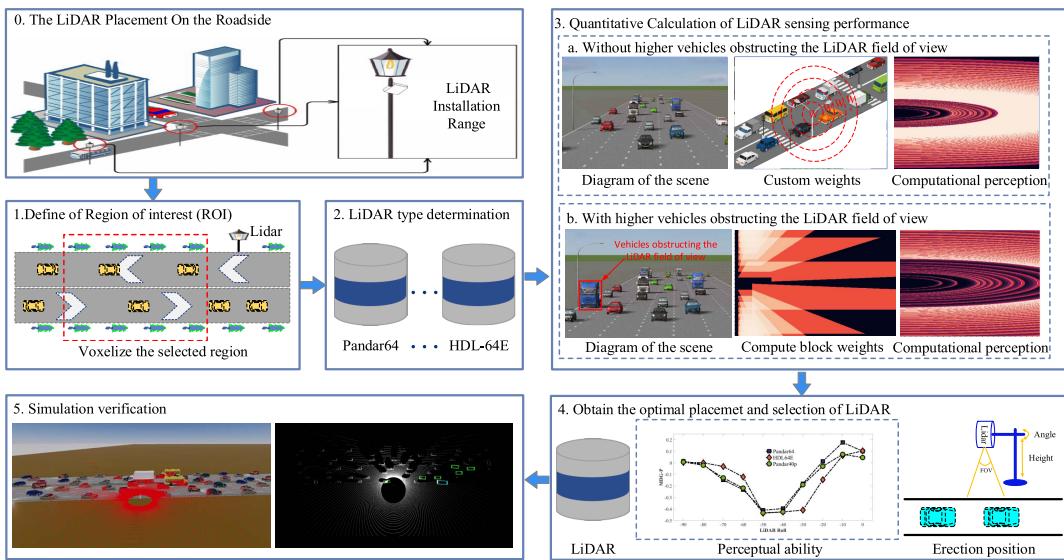


Fig. 1. Proposed LiDAR perception capability evaluation framework.

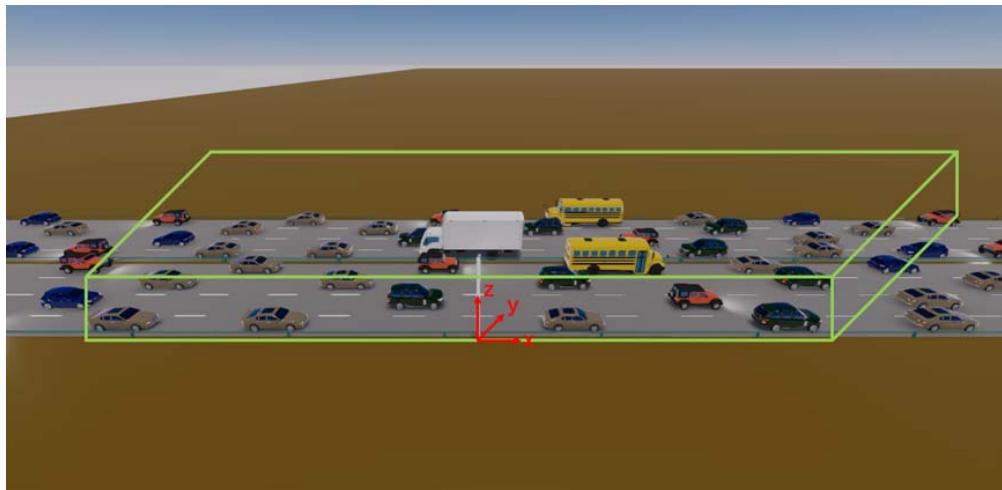


Fig. 2. Division of the ROI and definition of the coordinate system. LiDAR sensors installed along the roadside calculate LiDAR sensing capabilities based on the ROI regions.

voxel in the ROI as PVPC. Given the LiDAR configuration  $L = L_0$ , which includes the LiDAR model, and translation and rotation angle information relative to the origin of the world coordinate system, a conditional PVPC can be developed to represent the conditional point cloud of each voxel corresponding to the specific LiDAR configuration with the assumption of conditional independence. This conditional PVPC is given by

$$p_{\text{PVPC}|L_0} = p(v_1^{L_0}, v_2^{L_0}, \dots, v_N^{L_0} | L = L_0) \quad (2)$$

where  $p_{\text{PVPC}|L_0}$  represents the LiDAR with configuration  $L_0$  and  $v_i^{L_0}$  denotes the occupied voxel random variable  $v_i | L = L_0$ .

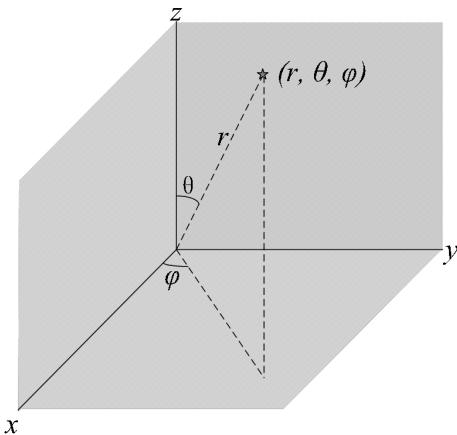
To calculate the PVPC from each LiDAR configuration, the LiDAR perception model should be defined first. As shown in Fig. 3, the LiDAR sensor placement is defined in a spherical coordinate system [48]. Since LiDAR emits a detection signal (laser beam) to the target, the received signal reflected back from the target is then compared and processed with the emitted signal to obtain information about the target. We refer

to the point where the beam hits the target as the return point. The return point in a spherical coordinate system is defined by a distance and two angles, where the former refers to the straight-line distance between the center of the LiDAR and the return point. Two angles adopt the azimuth  $\theta$  and polar angle  $\varphi$  conventions. Consequently, the return point in the LiDAR coordinate system is defined as  $(r, \theta, \varphi)$ . After the coordinate transformation, we can calculate the return point's Cartesian coordinates, which are given as follows:

$$\begin{cases} x_L = r \times \sin \theta \times \cos \varphi \\ y_L = r \times \sin \theta \times \sin \varphi \\ z_L = r \times \cos \theta. \end{cases} \quad (3)$$

After obtaining the point's position in the LiDAR coordinate system, we need to convert the coordinates to the world coordinate system. The calculation formula is given as

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} \quad (4)$$



**Fig. 3.** LiDAR spherical coordinates schematic. The coordinates of the star-shaped point in the figure can be determined by the points  $(r, \theta, \varphi)$ .

where  $R$  and  $T$  represent the LiDAR's rotation and translation matrices, respectively, which are included in the LiDAR placement. The coordinates in the LiDAR coordinate system are given by  $(x_L, y_L, z_L)$ , and  $(x_W, y_W, z_W)$  denotes the coordinates in the world coordinate system.

The positions of all points scanned by the LiDAR under a single line beam can be determined by varying the value of  $(r, \theta, \varphi)$ . The return points are denoted as the set  $p = \{p_1, p_2, \dots, p_T\}$ , where  $T$  represents the total number of return points from the current LiDAR configuration at different distances and beam angles. Each return point can be represented as  $p_j = (p_j^x, p_j^y, p_j^z)$ , where  $(x, y, z)$  denotes the coordinate values in the world coordinate system. A point within a voxel is denoted by  $v_i = [v_{x_i}, v_{y_i}, v_{z_i}]$ ,  $p_j \in v_i$ , where the size of each voxel is  $0.1 \text{ m} \times 0.1 \text{ m} \times 0.1 \text{ m}$

$$p(v_i) = \sum_{j=1}^T \mathbb{I}(p_j \in v_i), \quad i = 1, 2, \dots, N \quad (5)$$

where  $\mathbb{I}(\cdot)$  is an indicator function,  $N$  is the total number of voxels in the ROI, and  $p(v_i)$  represents the number of point clouds contained within each voxel.

### 2) Without High-Roof Vehicles Obstructing the LiDAR FOV:

In this section, we derive our surrogate metric based on the information theory using the conditional PVPC for a given LiDAR configuration  $L$ . This metric is used to evaluate the effectiveness of the LiDAR configuration for sensing the objects in the ROI. For this purpose, we introduce the theory of conditional entropy, described as follows:

$$\mathbf{H}(Y | X) = \sum_x p(x) \mathbf{H}(Y | X = x) \quad (6)$$

In this article, we name the proposed surrogate metrics as maximum density gain based on the PVPC (MDG-P). Therefore, the MDG-P is calculated as

$$\begin{cases} \mathbf{MDG}_P = \mathbf{H}(\mathbf{PVPC} | L) = \mathbb{E}_{v_i^{L_0} \sim p_{v|L_0}} \sum_{i=1}^N \mathbf{H}(v_i^{L_0}) \\ \mathbf{H}(v_i^{L_0}) = w(v_i) \log p(v_i) \end{cases} \quad (7)$$

where  $H(v_i^{L_0})$  is calculated as  $w(v_i) \log p(v_i)$  instead of  $p(v_i) \log p(v_i)$  in the original conditional entropy formula

because, although each voxel has the same probability of appearing in the ROI, the degree of influence of different voxels in different positions on the final result will vary. For example, ramps need to pay more attention to the merging (diverging) zone. Therefore,  $w(v_i)$  represents the weight of each voxel and is used to indicate the degree of attention of different voxels. It can be calculated as follows:

$$w(v_i) = \frac{w_{v_i}}{\sum_{i=1}^N w_{v_i}} \quad (8)$$

where  $w_{v_i}$  represents the weight of  $v_i$  that can be specified according to different requirements and  $\sum_{i=1}^N w_{v_i}$  represents the sum of the weights of all voxels. Consequently, the entropy  $H_n$  for each voxel in the absence of any obstacle vehicle can be calculated by the following expression:

$$H_n(v_i^{L_0}) = \frac{w_{v_i}}{\sum_{i=1}^N w_{v_i}} \log p(v_i). \quad (9)$$

### 3) With High-Roof Vehicles Obstructing the LiDAR FOV:

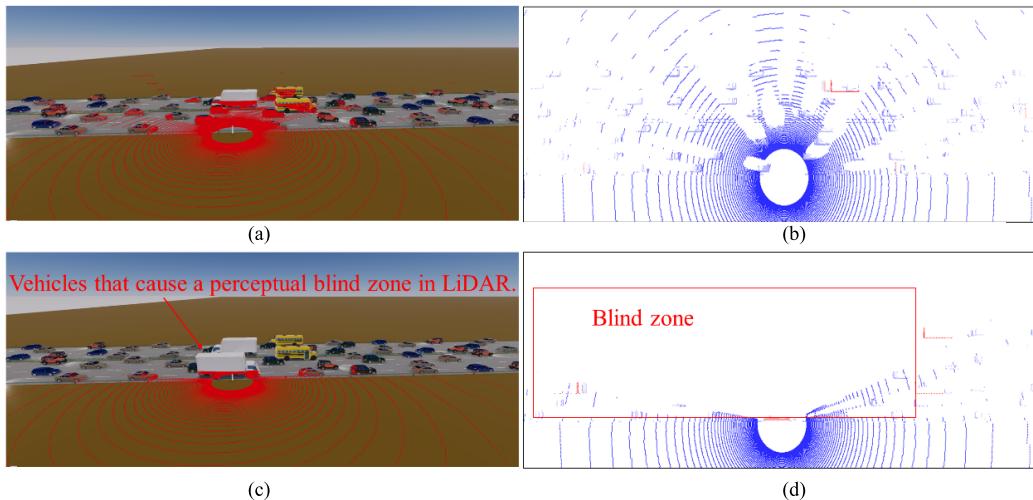
In this section, we quantify the performance loss caused by vehicle obstruction of the LiDAR FOV. In contrast to a vehicle-mounted LiDAR, a roadside LiDAR is designed to perceive the entire road environment. However, when a high-roof vehicle approaches the LiDAR in close proximity, it blocks the LiDAR's multiple line beams. Consequently, a large perceptual blind zone appears to result in the vehicle cannot be detected efficiently by LiDAR, as shown in Fig. 4. Note that the high-roof vehicle here refers to nonhousehold vehicles, such as buses and trucks.

The performance loss for each voxel is defined as  $H_b(v_i^{L_0})$ , which is calculated as follows:

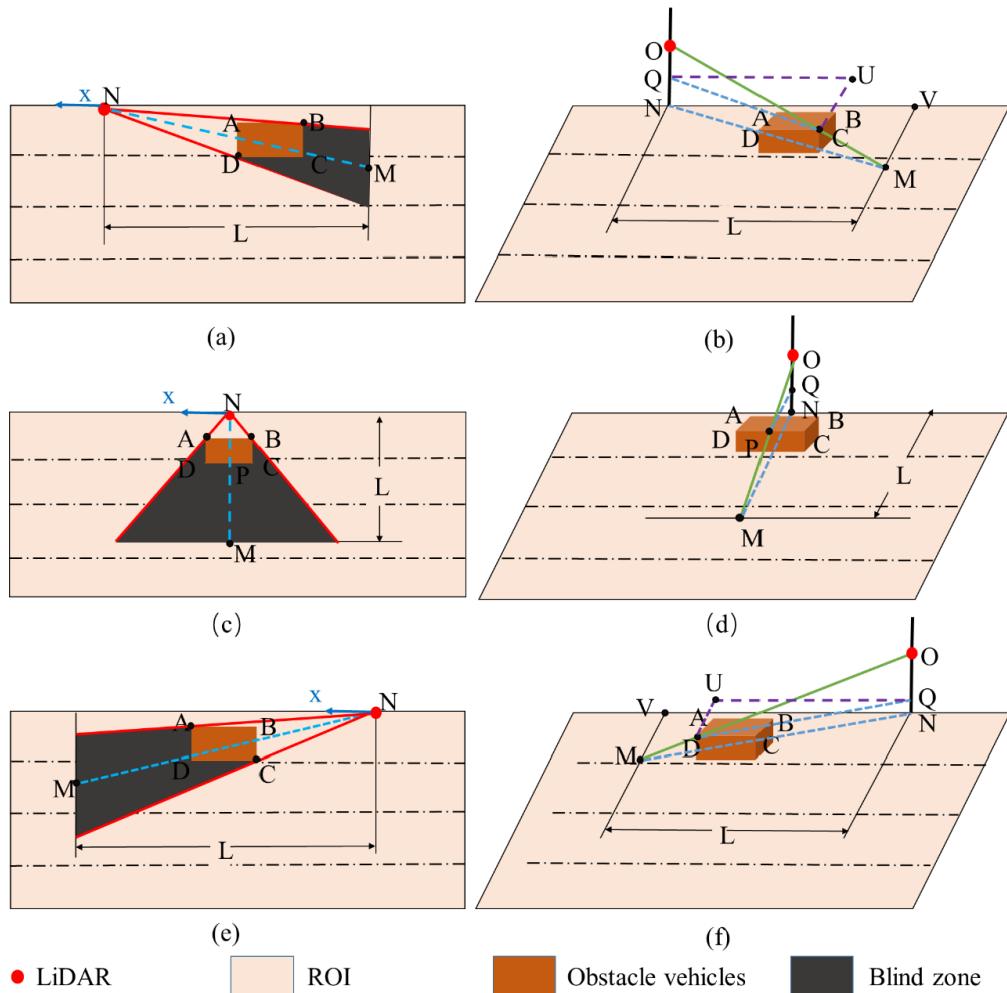
$$H_b(v_i^{L_0}) = w_b(v_i) \log p_b(v_i) \quad (10)$$

In order to quantify the performance loss caused by occlusion, the voxel weights  $w_b(v_i)$  in the ROI should be calculated first. The following assumptions were made during the calculation process: 1) the LiDAR is placed at a position higher than the passing vehicle's highest point because, otherwise, the LiDAR can completely lose its effectiveness and 2) we only examine the presence of obstacles in the lane closest to the LiDAR because such impediments have the greatest impact on sensing capabilities. The calculation of the blind zone is shown in the following.

To facilitate the calculations, we mark the position of the LiDAR as  $O$  and approximate the obstacle as a rectangle.  $A$ ,  $B$ ,  $C$ , and  $D$  are the projections of the obstructed vehicles on the ground, respectively. We need to calculate the longitudinal (transverse) range to complete the calculation of the 2-D area of the blind zone. Subsequently, we can obtain the 3-D area of the blind zone by filling. However, the calculation of the longitudinal (transverse) range changes when the relative position of the obstacle vehicle and the LiDAR varies, as shown in Fig. 5. Consequently, we differentiated the relative position of the LiDAR to the obstacle object based on the  $x$ -axis. The  $x$ -axis of the coordinate system is shown as the blue arrow in arrow in Fig. 5. The calculation is divided into three categories based on the relative position of the LiDAR and the obstructing vehicle: the obstacle vehicle is in



**Fig. 4.** Schematic of scene comparison with and without an obstructed vehicle. (c) Loss of LiDAR perceptual FOV due to the presence of high-roof vehicles. (d) Accompanying point cloud. As shown by the red rectangular box in (d), the LiDAR perception of the entire road is nearly completely lost due to the presence of the high-roof vehicles, causing a large perceptual blind area. (a) Schematic with the unobstructed vehicle. (b) Point cloud with the unobstructed vehicle. (c) Schematic with the obstructed vehicle. (d) Point cloud with the obstructed vehicle.



**Fig. 5.** Blind zone division is depicted in this diagram. LiDAR is represented by the red origin in the diagram. The blue arrow depicts the x-axis direction in the world coordinate system, while the black region shows the blind zone caused by the obstacle vehicle. (a) Negative direction top view. (b) Negative direction 3-D view. (c) Positive center top view. (d) Positive center top 3-D view. (e) Positive direction top view. (f) Positive direction 3-D view.

the LiDAR's negative direction, positive center, and positive direction, where the center point of the vehicle determines its position.

When the obstacle is in the negative direction of LiDAR, the range of the blind zone in the y-direction can be determined by calculating the slope of  $OB$  and  $OD$ , as shown in Fig. 5.

Subsequently, the transverse range  $L$  of the blind zone can be obtained by calculating the intersection of the lower right boundary point  $C$  of the obstacle and the straight line where the LiDAR  $O$  is located with the ground. This refers to the contact point  $M$  between the extension line where  $OC$  is located and the ground, as shown in Fig. 5(b), and the length of  $L$  can be obtained by calculating the length of  $CQ$  and  $MN$ . Based on the above information, the 2-D range of the occluded area can be determined.

Considering the point  $N$  as the origin of the coordinate system, the slope of  $OB$  and  $OD$  can be determined by the following equations:

$$\begin{cases} OB = \frac{y_{\text{block}} - \text{object.width}/2}{x_{\text{block}} - \text{object.long}/2} \\ OD = \frac{y_{\text{block}} + \text{object.width}/2}{x_{\text{block}} + \text{object.long}/2} \end{cases} \quad (11)$$

where `object.width` and `object.long` represent the width and length of the obstacle vehicle, respectively, and  $x_{\text{block}}$  and  $y_{\text{block}}$  denote the obstacle coordinates. Subsequently, using simple mathematical principles,  $L$  may be determined using the equation below:

$$L = NV = \frac{ON \cdot QU}{OQ} = \frac{\text{lidar.h} * (x_{\text{block}} - \text{object.long}/2)}{\text{lidar.h} - \text{object.height}} \quad (12)$$

where `lidar.h` represents the height of the LiDAR, and `object.height` denotes the height of the obstacle vehicle.

After completing the blind zone in 2-D, we need to expand it into 3-D. The formula for this conversion is obtained as follows by calculating the angle  $\theta$  formed by  $OM$  and  $MN$ :

$$\tan \theta = \frac{\text{lidar.h} - \text{object.height}}{\sqrt{(x_{\text{block}} - \text{object.long}/2)^2 + (y_{\text{block}} + \text{object.width}/2)^2}}. \quad (13)$$

The filling height  $h$  of the point  $Z$  in the 2-D blind area is determined as follows, where the horizontal coordinate of point  $Z$  is  $m$ , the vertical coordinate of point  $Z$  is  $n$ :

$$h = \text{distance} * \tan \theta \quad (14)$$

where `distance` specifies the lateral distance from point  $Z$  to the line where  $M$  is placed, i.e., the distance between point  $Z$  and the maximum absolute  $x$  value of the blind zone, when  $n$  is within the range of  $AB$  and  $DC$ .

When the point  $Z$  is in the 2-D blind area but not in the range of  $AB$  and  $DC$ , the  $y$ -value of the lower limit of the 2-D blind area must be calculated, and `distance` refers

to the distance between the point  $Z$  and the blind area's maximum  $y$ -value. In this case,  $h$  can be calculated as (15), shown at the bottom of the page.

The filling height  $h$  can be calculated according to (15). At this point, we have completed the blind zone calculation when the obstacle is located in the negative direction of the LiDAR (with the  $x$ -axis direction orientation as the reference). The blind zone distribution for the remaining two cases can be calculated based on the same principle. To mimic the distance transformation between the obstacle and the LiDAR, we also need to keep changing the  $x$ -coordinates of the obstacles so that they complete all traversals within the  $x$ -axis range in the ROI. This is followed by altering the obstacle length, breadth, and height to calculate the distribution of blind zones under different types of obstacles. The number of return points within each voxel is considered to be the weight of that voxel, which is computed publicly as follows:

$$w_b(v_i) = \frac{\sum_{j=1}^B \mathbb{I}(b_j \in v_i), i = 1, 2, \dots, N}{\sum_{i=1}^N w_b(v_j)} \quad (16)$$

where  $\mathbb{I}(\cdot)$  is an indicator function,  $N$  is the total number of voxels in the ROI, and  $B$  is the total number of LiDAR points.

The factor loss is introduced for calculating the number of point clouds of voxels in the blind zone, which is represented as  $p_b(v_i) = \text{loss} \times p(v_i)$ . It can be calculated as follows:

$$\text{loss} = 1 - \frac{p(v_i)_{\text{before}} - p(v_i)_{\text{after}}}{p(v_i)_{\text{before}}} \quad (17)$$

where  $p(v_i)_{\text{before}}$  and  $p(v_i)_{\text{after}}$  refer to the PVPC in the unobscured and obscured cases, respectively. In our experiments, we approximate loss by calculating the beam loss, which can be calculated as follows:

$$\text{loss} = 1 - \frac{\text{beam}_{\text{obstacled}}}{\text{beam}} \quad (18)$$

where `beam_blocked` refers to the number of beams blocked by the obscured vehicle and the `beam` refers to the total number of beams of LiDAR. Therefore, the LiDAR performance loss caused by the obstacle vehicle can be calculated using the following equation:

$$H_b(v_i) = \frac{\sum_{j=1}^B \mathbb{I}(b_j \in v_i), i = 1, 2, \dots, N}{\sum_{i=1}^N w_b(v_j)} \times \log \left( \left( 1 - \frac{\text{beam}_{\text{obstacled}}}{\text{beam}} \right) * p(v_i) \right). \quad (19)$$

**4) Merging Calculation:** As described in Section II-B3, there is a great probability of higher vehicle obstruction of the LiDAR FOV, so we need to merge the LiDAR sensing capability with and without obstruction to make the calculation

$$\begin{cases} h = \frac{\text{lidar.h} - \text{object.height}}{\sqrt{(x_{\text{block}} - \text{object.long}/2)^2 + (y_{\text{block}} + \text{object.width}/2)^2}} \times |L - m| \\ n \in (y_{\text{block}} - \text{object.width}/2, y_{\text{block}} + \text{object.width}/2) \\ h = \frac{\text{lidar.h} - \text{object.height}}{\sqrt{(x_{\text{block}} - \text{object.long}/2)^2 + (y_{\text{block}} + \text{object.width}/2)^2}} \times |L \times OD - n|, \text{ else} \end{cases} \quad (15)$$

**TABLE I**  
OBJECT DETECTION PERFORMANCE OF GROUP 1

LiDAR Height	LiDAR Roll Angle						
	0°	-5°	-10°	-15°	-20°	-25°	-30°
3 m	<b>56.65</b>	51.61	46.65	39.65	32.12	20.51	10.62
4 m	49.07	<b>50.72</b>	48.83	42.68	37.85	25.16	14.92
5 m	41.53	43.17	<b>46.26</b>	42.15	38.09	32.58	20.26
6 m	28.98	31.89	33.17	<b>38.93</b>	37.01	36.58	34.49
7 m	27.68	29.97	32.58	35.13	<b>35.29</b>	34.48	32.06
8 m	24.68	26.47	27.65	28.68	29.12	<b>31.14</b>	30.26

results more realistic for reference. However, the frequency of occlusions can vary from scene to scene. Therefore, we use a merging factor  $\sigma$  to calculate the merge result. It is used for approximate simulation of the frequency of occlusion situations in reality. The value of  $\sigma$  can be obtained by calculating the ratio of the number of obstructing vehicles to the total number of vehicles in the lane adjacent to the LiDAR in the ROI. It is calculated as follows:

$$\sigma = \frac{\sum \text{vehicle}_{\text{obstacle}}}{\sum \text{vehicle}} \quad (20)$$

where  $\text{vehicle}_{\text{obstacle}}$  refers to the obstacle vehicle that can cause a loss in the LiDAR sensing performance and the vehicle represents all vehicles. This numerical count is limited to vehicles located in lanes adjacent to the LiDAR. The final sensing capability of LiDAR can be calculated using the following equation:

$$\begin{aligned} H_{\text{LiDAR}} &= H_n(v_i^{L_0}) + \sigma H_b(v_i^{L_0}) \\ &= w(v_i) \log p(v_i) - \sigma w_b(v_i) \log(\text{loss} * p(v_i)). \end{aligned} \quad (21)$$

Consequently, the MDG-P can be calculated for different LiDAR configurations as follows the larger the  $\text{MDG}_P^{L_0}$ , the better the performance of the LiDAR in that configuration, i.e., the more superior that configuration is:

$$\begin{aligned} \text{MDG}_P^{L_0} &= \mathbb{E}_{v_i^{L_0} \sim p_{v|L_0}} \sum_{i=1}^N (w(v_i) \log p(v_i) - \sigma w_b(v_i) \\ &\quad \times \log(\text{loss} * p(v_i))). \end{aligned} \quad (22)$$

#### IV. EXPERIMENTS

This section aims to investigate the impact of different LiDAR placements on the performance of 3-D object detection and the validity of the proposed method. We carry out extensive experiments in Unity 3-D to address these questions. Due to the requirement of fairly evaluating different LiDAR placements while keeping all other environmental factors fixed, the simulation scenarios in Unity 3-D are chosen instead of a public real-world dataset.

##### A. Simulation Setup

Unity3D integrates a built-in physical engine, which is used to power the simulation so that it resembles the real world. A bidirectional eight-lane traffic scenario is developed

in Unity3D. Five vehicle models are introduced, including sport utility vehicle (SUV), sedan, school bus, truck, and jeep. Each of them is equipped with rigid body components so that they can be powered by the physical engine. The traffic flow is generated using the Poisson distribution with a traffic intensity  $\tau$ . The ROI is defined with length  $l \in [-80 \text{ m}, 80 \text{ m}]$ , i.e., 160 m in total, width  $w \in [0, 28 \text{ m}]$ , i.e., eight lanes each with a width of 3.5 m, and height  $h \in [0, 3 \text{ m}]$ .

Unity3D offers the advantage of being able to functionalize the Lidar's main components, namely, the ray technology. A LiDAR simulation may be constructed by setting up several beams with the capacity to spin. The collision process between the rays and rigid bodies can be used to obtain the object distance information. The data are collected at different LiDAR mounting heights ranging from 3 to 8 m with a 1-m increment and roll angles ranging from  $-90^\circ$  to  $0^\circ$  with a  $5^\circ$  step size. A total of ten frames of point cloud data are collected in each scenario, and a total of 1140 frames of point clouds are collected.

All the simulations are carried out on a Windows 10 Educational Version System with Unity3D 2019.4.18f1. A workstation with a 2.9-GHz Intel Core i7-10700 CPU, 64-GB RAM, an NVIDIA GeForce RTX 3070 GPU with 8-GB memory, and using the Ubuntu16.04 Operating System is used to run the simulations.

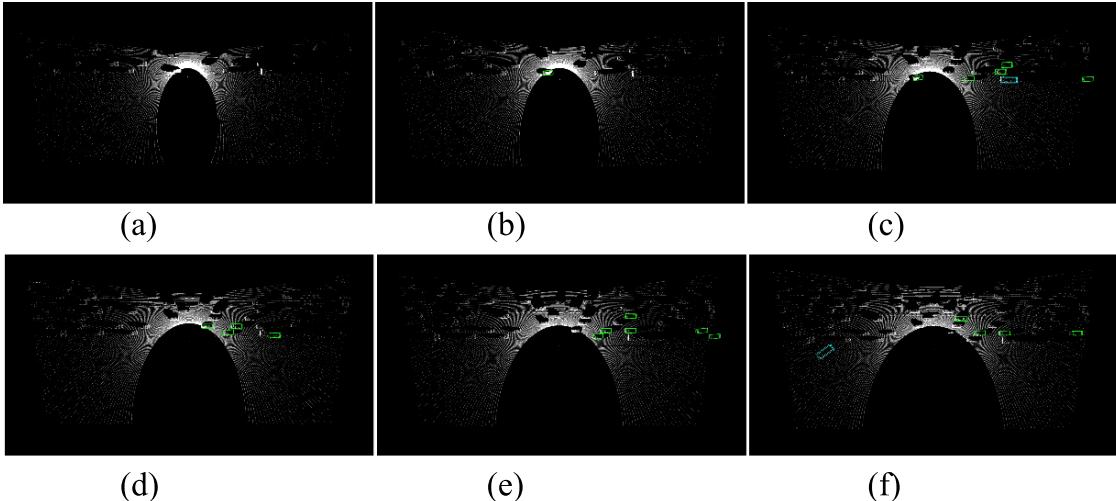
##### B. Impact of LiDAR Mounting Placement on the Performance of 3-D Object Detection

The experiment is divided into two groups using LiDAR HDL-64E: Group 1 with obstruction and Group 2 without obstruction. The simulation is carried out under different LiDAR mounting placements, with heights ranging from 3 to 8 m with a step size of 1 m, and roll angles varying from  $0^\circ$  to  $-30^\circ$  with a  $-5^\circ$  increment. A total of 430 frames of point cloud data are collected at each LiDAR mounting placement, where the Pointpillars [16] is applied to measure the LiDAR performance with respect to object detection. Pointpillars is a deep learning detection algorithm, which is a point cloud-based model for 3-D target detection. Tables I and II present the results for Groups 1 and 2, respectively.

It can be observed that different LiDAR placements inevitably affect the object detection performance in Group 1. When the height is more than 4 m, the LiDAR performance increases at first and then decreases with the increase in the roll angle. A similar trend of LiDAR performance with respect

**TABLE II**  
OBJECT DETECTION PERFORMANCE OF GROUP 2

LiDAR Height	LiDAR Roll Angle						
	0°	-5°	-10°	-15°	-20°	-25°	-30°
3 m	0	0	0	0	0	0	0
4 m	<b>39.72</b>	32.68	21.57	10.68	5.63	0	0
5 m	34.58	37.74	<b>39.05</b>	37.15	29.21	21.78	15.82
6 m	27.89	28.98	37.08	<b>37.28</b>	36.97	25.98	23.68
7 m	26.94	29.01	31.58	34.92	<b>35.03</b>	34.11	31.28
8 m	24.68	26.47	27.65	28.68	29.12	<b>31.14</b>	30.26



**Fig. 6.** Example of object detection results in Group 1 with the roll angle of  $-20^\circ$ . The green rectangular boxes in the figure represent the cars identified by the algorithm, and the blue boxes represent the trucks identified by the algorithm. (a) Height = 3 m and roll angle =  $-20^\circ$ . (b) Height = 4 m and roll angle =  $-20^\circ$ . (c) Height = 5 m and roll angle =  $-20^\circ$ . (d) Height = 6 m and roll angle =  $-20^\circ$ . (e) Height = 7 m and roll angle =  $-20^\circ$ . (f) Height = 8 m and roll angle =  $-20^\circ$ .

to the mounting height can be observed when the roll angle is more than  $-10^\circ$ . The optimal LiDAR roll angle increases as the LiDAR mounting height increases. This may be caused by the requirement of a larger roll angle to ensure that all the beams hit the ROI as the mounting height increases. However, if the roll angle increases continuously, it will cause a perception loss at the outer road, and the lower the erection height, the higher the loss, as shown in Fig. 6.

In Group 2, findings similar to those in Group 1 are obtained. The difference is that there is a high-roof vehicle blocking the LiDAR, which causes the LiDAR to almost lose the sensing ability at the 3- and 4-m mounting heights. This phenomenon improves as the LiDAR mounting height increases and the roll angle changes, as shown in Fig. 7.

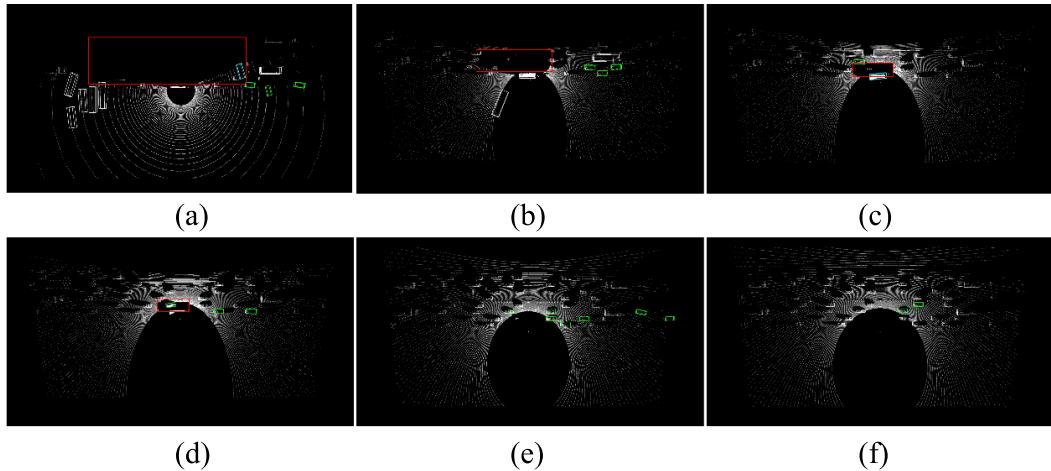
In conclusion, in the absence of any high-roof vehicles, e.g., truck and bus, which will obstruct the LiDAR FOV, the lower the LiDAR mounting height within the allowable range, the better the perception performance. When the mounting height increases, the roll angle should be appropriately increased to make the beams as concentrated as possible in the ROI, reducing the blind zone of perception centered on the LiDAR. When there are vehicles that will obstruct the FOV of LiDAR, the installation height of the LiDAR should be increased appropriately; generally speaking, about 1 m higher than the highest height of the obstacle vehicle can achieve a satisfactory perception effect.

### C. Relationship Between the 3-D Object Detection Performance and the Proposed Method

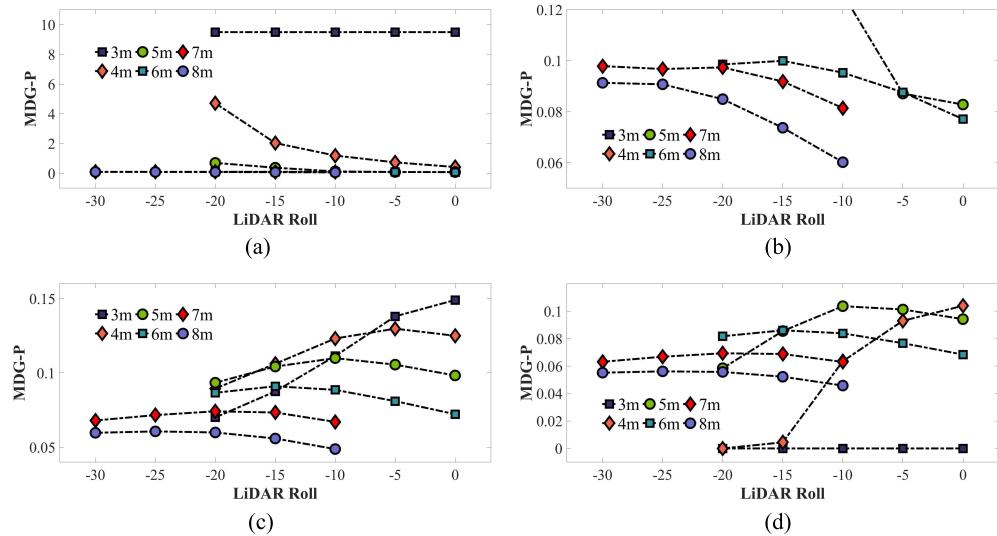
To verify the feasibility of our proposed MDG-P method to represent the LiDAR perception performance, we consider LiDAR HDL-64E as an example and compare the results with the deep learning algorithm results. We choose Pointpillars as the deep learning detection algorithm as mentioned above. The comparison results are provided in Table III. For both deep learning and our proposed method, the larger the result, the better the LiDAR sensing performance is represented, i.e., the more superior that LiDAR configuration is. Similar changing trends in the two groups of data can be observed. Therefore, it is proven that the results of our proposed MDG-P are consistent with those of the Pointpillars. Our proposed MDG-P method can be used for evaluating radar performance.

### D. Results of MDG-P

Figs. 8–10 show the proposed MDG-P results of LiDAR HDL-64E, LiDAR Pandar64, and LiDAR Pandar40P, respectively. In each figure, subfigure (a) represents the results with obstruction, (b) is the partially enlarged figure of (a), (c) represents the results without obstruction, and (d) is the merge results. In this experiment, the roll angle rotation along the right-hand rule is considered positive, while the opposite is considered negative. All the ranges of the roll angle mentioned in this section are shown in the corresponding results plots.



**Fig. 7.** Example of object detection results in Group 2 with the roll angle of  $-20^\circ$ . The red rectangle indicates the blind zone. (a) Height = 3 m and roll angle =  $-20^\circ$ . (b) Height = 4 m and roll angle =  $-20^\circ$ . (c) Height = 5 m and roll angle =  $-20^\circ$ . (d) Height = 6 m and roll angle =  $-20^\circ$ . (e) Height = 7 m and roll angle =  $-20^\circ$ . (f) Height = 8 m and roll angle =  $-20^\circ$ .



**Fig. 8.** MDG-P results of LiDAR HDL-64E: (a) with obstruction, (b) expanded view of (a) (details), (c) without obstruction, and (d) merge result.

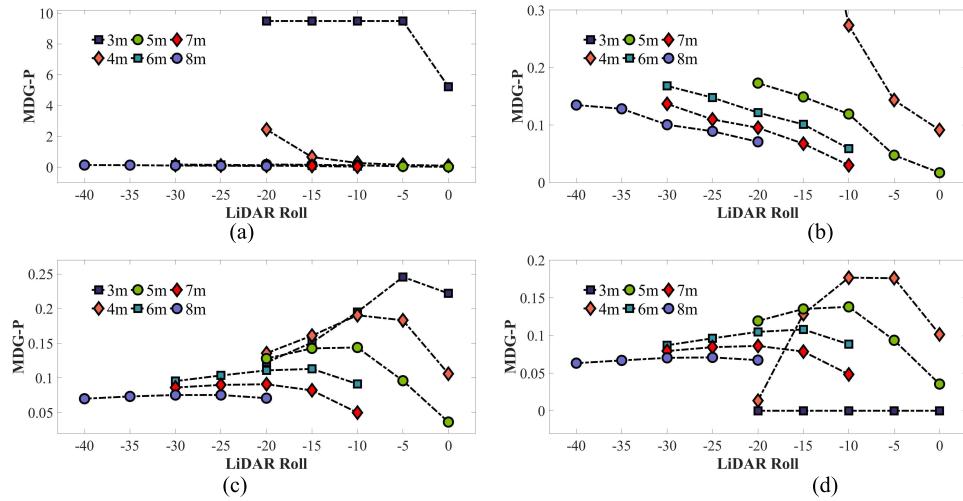
**TABLE III**  
COMPARISON BETWEEN POINTPILLARS AND OUR PROPOSED MDG-P

LiDAR Height	Optimal Roll Angle	Without Obstruction	Detection Performance	With Obstruction	Detection Performance
3 m	$0^\circ$	0.1490	56.65	9.4912	0
4 m	$0^\circ$	0.1250	50.72	0.4217	39.72
5 m	$-10^\circ$	0.1099	46.26	0.1263	39.05
6 m	$-15^\circ$	0.0910	38.93	0.0999	37.28
7 m	$-20^\circ$	0.0742	35.29	0.0974	35.03
8 m	$-25^\circ$	0.0607	31.14	0.0907	31.14

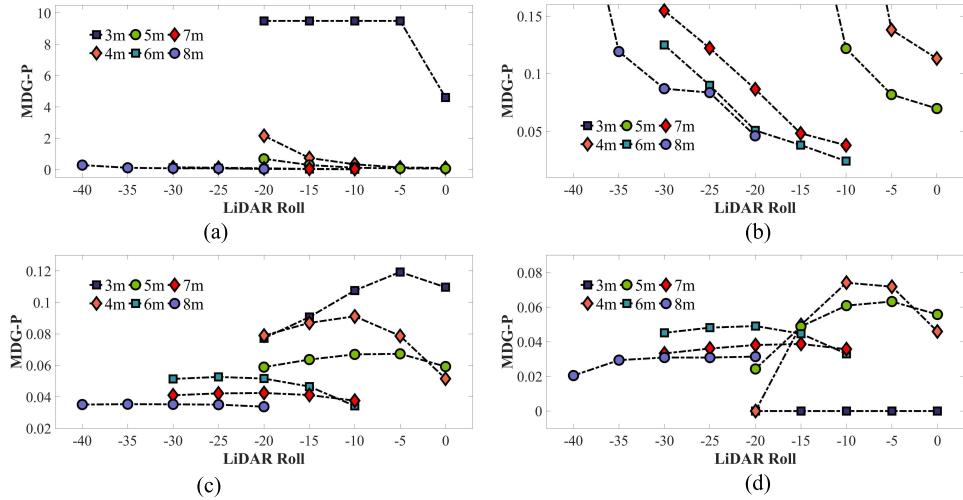
The subgraphs (a) and (b) of Figs. 8–10 show that the curves corresponding to the same color, which corresponds to a fixed height, basically decrease as the absolute value of the roll decreases. That is, for a constant height in the presence of obstruction, the perception loss of the LiDAR increases as the absolute value of the roll angle of the LiDAR increases. The reason for this behavior may be that, as the roll angle of the LiDAR increases, it causes a higher number of beams to

be obscured by the obstacle, which causes an increase in the perceptual loss, as shown in Fig. 11.

Figs. 10(c), 11(c), and 12(c) show that each curve first increases and then decreases with the increase in the absolute value of the roll angle for a height of more than 4 m. That is to say, in the absence of obstruction, for a constant height of more than 4 m, the perceptual performance of LiDAR first increases and then decreases with the increase in the absolute



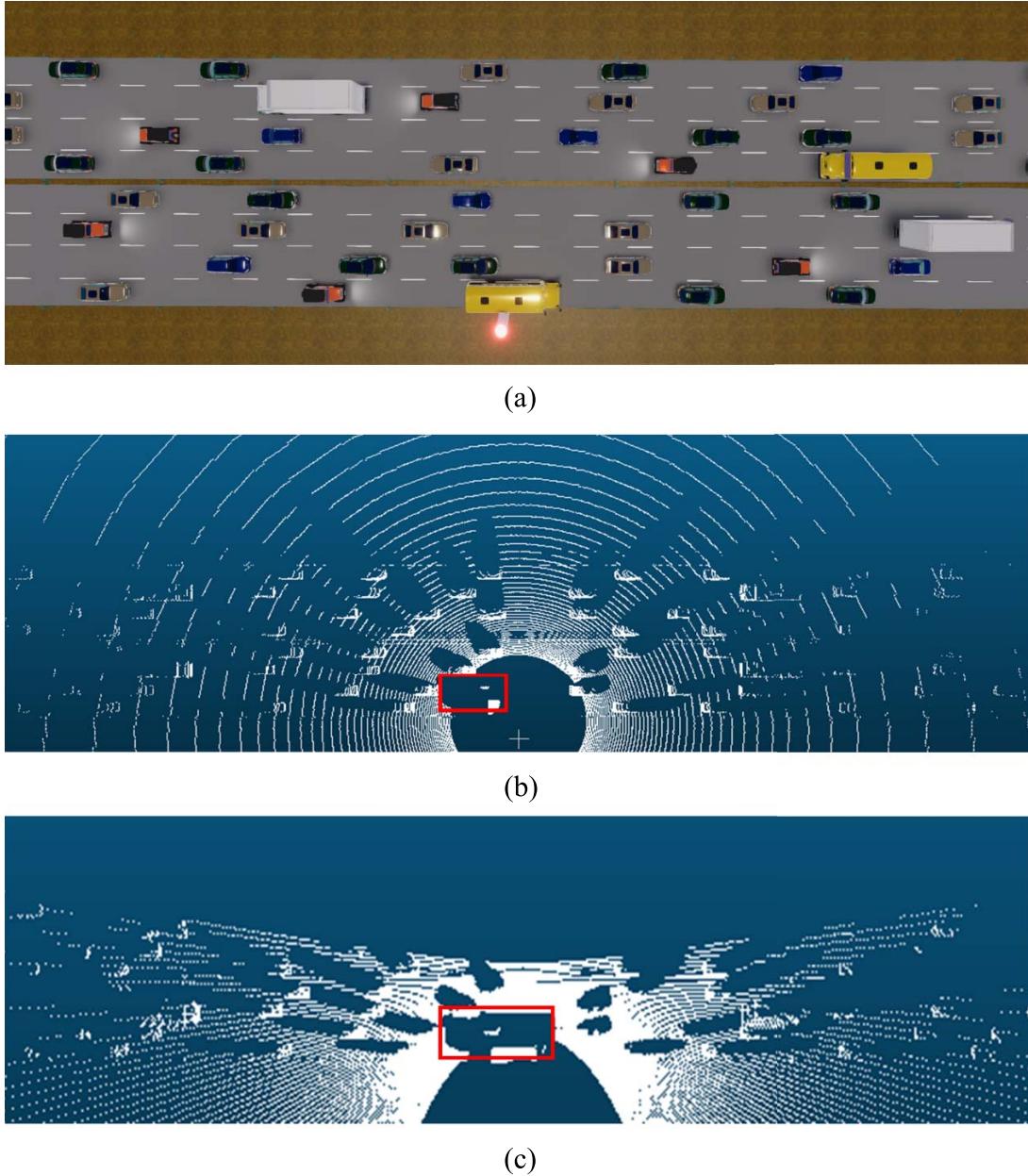
**Fig. 9.** MDG-P results of LiDAR Pandar64: (a) with obstruction, (b) expanded view of figure (a) (details), (c) without obstruction, and (d) merge result.



**Fig. 10.** MDG-P results of LiDAR Pandar40P: (a) with obstruction, (b) expanded view of (a) (details), (c) without obstruction, and (d) merge result.

TABLE IV  
EXPERIMENTAL RESULTS

LiDAR Height	LiDAR	Optimal Roll Angle	Without Obstruction	With Obstruction	Optimal MDG-P
3 m	HDL-64E	0°	0.1490	9.4912	/
	Pandar64	-5°	0.2458	9.4912	/
	Pandar40p	-5°	0.1193	9.4912	/
4 m	HDL-64E	0°	0.1250	0.4217	0.1039
	Pandar64	-10°	0.1905	0.2733	0.1768
	Pandar40p	-10°	0.0912	0.3418	0.0707
5 m	HDL-64E	-10°	0.1099	0.1263	0.1036
	Pandar64	-10°	0.1440	0.1192	0.1380
	Pandar40p	-15°	0.0673	0.0820	0.0632
6 m	HDL-64E	-15°	0.0910	0.0999	0.0861
	Pandar64	-15°	0.1130	0.1011	0.1080
	Pandar40p	-20°	0.0516	0.0509	0.0491
7 m	HDL-64E	-20°	0.0742	0.0974	0.0693
	Pandar64	-20°	0.0909	0.0952	0.0861
	Pandar40p	-25°	0.0412	0.0483	0.0387
8 m	HDL-64E	-25°	0.0607	0.0907	0.0562
	Pandar64	-25°	0.0753	0.0891	0.0709
	Pandar40p	-30°	0.0338	0.0461	0.0314



**Fig. 11.** Blind zone demonstration of LiDAR HDL-64E at 5 m height. **(b)** With roll angle of  $0^\circ$ . **(c)** With a roll angle of  $20^\circ$ . The red rectangle indicates the blind zone. **(a)** Simulation scene schematic. **(b)** LiDAR = 5 m and roll angle =  $0^\circ$ . **(c)** LiDAR = 5 m and roll angle =  $-20^\circ$ .

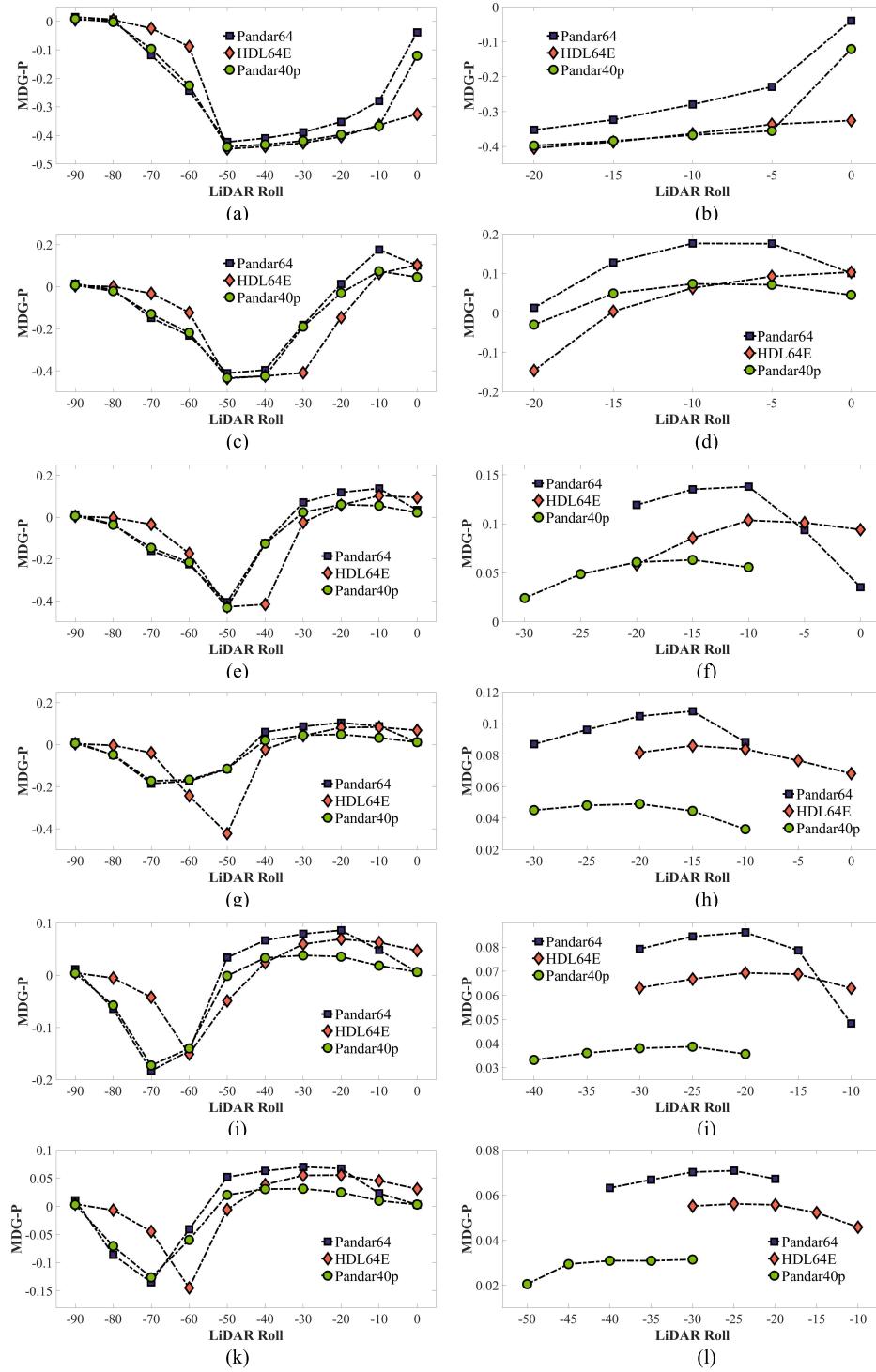
value of roll angle. This behavior may be caused by a larger roll angle needed to concentrate the LiDAR beam in the ROI as the height increases.

Figs. 8(d), 9(d), and 10(d) show the combined results obtained by integrating the results of (a) and (c) according to our proposed method. It can be observed that, even though the model of LiDAR varies, a broadly similar performance is exhibited, i.e., a LiDAR height of 3 m causes the maximum perception loss, and in addition, a lower height results in a better perception performance. This may be caused by the fact that the lower the mounting height, the smaller the perceptual blind zone centered on the LiDAR. As the height gradually increases, although increasing the roll angle improves the perception performance, it causes a higher number of line beams to be concentrated in the center of the LiDAR,

and the perception performance on both sides of the road degrades.

Fig. 12 compares the performance of the three LiDARs at the same height and different roll angles. It can be observed that all LiDARs show a similar performance trend. However, the performance is limited due to technical differences in LiDARs produced by different manufacturers. For example, LiDAR HDL-64E and LiDAR Pandar64 are both 64-line LiDARs; however, their resolution is different that causes a difference in performance. Overall, the 64-line LiDAR (HDL-64E and Pandar64) performs better than the 40-line LiDAR (Pandar40P), and Pandar64 performs better than HDL-64E.

Table IV shows the optimal mounting placements (height and roll angle) of the three LiDARs under the comprehensive



**Fig. 12.** Variation of sensing performance caused by the variation of LiDAR mounting height and roll angle. (a) Height = 3 m. (b), (d), (f), (h), (j), and (l) Details. (c) Height = 4 m. (e) Height = 5 m. (g) Height = 6 m. (i) Height = 7 m. (k) Height = 8 m.

conditions and their MDG-P, which can be used as a reference for the mounting placement of roadside LiDARs.

## V. CONCLUSION

In this article, we proposed an information-theory-based LiDAR mounting placement evaluation method. The goal of the proposed method was to convert the complicated problem of LiDAR mounting placement into a parameter calculation

problem in order to achieve the optimal mounting placement for LiDARs. We proposed a metric of PVPC to measure the amount of information sensed by the LiDAR in the ROI. This metric was named the MDG-P. A two-way eight-lane traffic scenario was constructed in Unity3D, and data were collected corresponding to different LiDAR mounting placements for two cases: with obstruction and without obstruction. Subsequently, the Pointpillars was applied to the collected dataset,

and the results were compared and analyzed with respect to the results of the proposed MDG-P. In conclusion, our proposed MDG-P was simpler and could effectively evaluate the LiDAR placement. We hope that the MDG-P proposed in this study can serve as a useful and powerful testbed for roadside LiDAR mounting placement optimization. In the future, in addition to continuing to study the issues related to the optimal layout of roadside equipment in detail, we shall focus on the multisensor placement optimization under the cooperative vehicle-infrastructure collaborative system.

## DATA AVAILABILITY

The data used to support the findings of this study are available from the corresponding author upon request.

## CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

## REFERENCES

- [1] Y. Gao, X. Zhao, Z. Xu, J. Cheng, and W. Wang, "An indoor vehicle-in-the-loop simulation platform testing method for autonomous emergency braking," *J. Adv. Transp.*, vol. 2021, pp. 1–12, Apr. 2021, doi: [10.1155/2021/8872889](https://doi.org/10.1155/2021/8872889).
- [2] *Key Technologies and Prospects of Vehicle-Road Collaboration for Autonomous Driving*, TUBA Institute for AI Industry Research, Beijing, China, 2021.
- [3] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepVCP: An end-to-end deep neural network for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 12–21.
- [4] G. Zamanakos, L. Tsochatzidis, A. Amanatiadis, and I. Pratikakis, "A comprehensive survey of LiDAR-based 3D object detection methods with deep learning for autonomous driving," *Comput. Graph.*, vol. 99, pp. 153–181, Oct. 2021.
- [5] 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.
- [6] H. Hu, Z. Liu, S. Chitlangia, A. Agnihotri, and D. Zhao, "Investigating the impact of multi-LiDAR placement on object detection for autonomous driving," 2021, [arXiv:2105.00373](https://arxiv.org/abs/2105.00373).
- [7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.
- [8] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.
- [9] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.
- [10] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9775–9784.
- [11] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
- [12] 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.*, Oct. 2017, pp. 2980–2988.
- [13] J. M. Ku, M. Lee, J. Harakeh, A. Waslander, and L. Steven, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.
- [14] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3D object detection," in *Proc. Conf. Robot Learn.*, Oct. 2018, pp. 146–155.
- [15] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7652–7660.
- [16] 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.*, Jun. 2019, pp. 12697–12705.
- [17] A. Simonelli, S. R. Bulo, L. Porzi, M. Lopez-Antequera, and P. Kortschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1991–1999.
- [18] H. Yi *et al.*, "SegVoxelNet: Exploring semantic context and depth-aware features for 3D vehicle detection from point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2274–2280.
- [19] B. Graham, M. Engelcke, and L. V. D. Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.
- [20] B. Wang, J. An, and J. Cao, "Voxel-FPN: Multi-scale voxel feature aggregation in 3D object detection from point clouds," 2019, [arXiv:1907.05286](https://arxiv.org/abs/1907.05286).
- [21] 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](https://arxiv.org/abs/1908.09492).
- [22] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.
- [23] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.
- [24] 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.*, Jun. 2019, pp. 770–779.
- [25] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 1711–1719.
- [26] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep Hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9277–9286.
- [27] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel CNN for efficient 3D deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
- [28] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2626–2635.
- [29] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.
- [30] H. Su *et al.*, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539.
- [31] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3075–3084.
- [32] M. Jaritz, J. Gu, and H. Su, "Multi-view pointnet for 3D scene understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, Oct. 2019, pp. 1–9.
- [33] J. Schult, F. Engelmann, T. Kontogianni, and B. Leibe, "DualConvMeshNet: Joint geodesic and Euclidean convolutions on 3D meshes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 8612–8622.
- [34] J. Mao, X. Wang, and H. Li, "Interpolated convolutional networks for 3D point cloud understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1578–1587.
- [35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.
- [36] M. Finzi, S. Stanton, P. Izmailov, and A. G. Wilson, "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 3165–3176.
- [37] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Deep learning on 3D point clouds," *Remote Sens.*, vol. 12, no. 11, p. 1729, 2020.
- [38] J. Ousingsawat and M. E. Campbell, "Optimal cooperative reconnaissance using multiple vehicles," *J. Guid., Control, Dyn.*, vol. 30, no. 1, pp. 122–132, 2007.

- [39] A. J. Sinclair, R. J. Prazenica, and D. E. Jeffcoat, "Optimal and feedback path planning for cooperative attack," *J. Guid., Control, Dyn.*, vol. 31, no. 6, pp. 1708–1715, 2008.
- [40] A. N. Bishop and P. Jensfelt, "An optimality analysis of sensor-target geometries for signal strength based localization," in *Proc. Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Dec. 2009, pp. 127–132.
- [41] A. N. Bishop, B. Fidan, B. D. O. Anderson, K. Doğançay, and P. N. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, Mar. 2010.
- [42] S. Xu, "Optimal sensor placement for target localization using hybrid RSS, AOA and TOA measurements," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 1966–1970, Sep. 2020.
- [43] S. Xu and K. Doğançay, "Optimal sensor placement for 3-D angle-of-arrival target localization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 3, pp. 1196–1211, Jun. 2017.
- [44] M. Sadeghi, F. Behnia, and R. Amiri, "Optimal sensor placement for 2-D range-only target localization in constrained sensor geometry," *IEEE Trans. Signal Process.*, vol. 68, pp. 2316–2327, 2020.
- [45] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3D point cloud classifiers," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2279–2283.
- [46] N. H. Nguyen and K. Dogancay, "Optimal geometry analysis for elliptic target localization by multistatic radar with independent bistatic channels," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 2764–2768.
- [47] T. Ma, Z. Liu, and Y. Li, "Perception entropy: A metric for multiple sensors configuration evaluation and design," 2021, *arXiv:2104.06615*.
- [48] *LiDAR Basics: The Coordinate System*. Accessed: Sep. 29, 2018. [Online]. Available: <https://hackernoon.com/lidar-basics-the-coordinate-system-a26529615df9>



**Shaojie Jin** received the B.S. degree in computer science from Chang'an University, Xi'an, China, in 2019, where she is currently pursuing the Ph.D. degree in traffic information engineering and control with the School of Information Engineering.

Her research interests include 3-D object detection and multisensor fusion perception.



**Ying Gao** received the B.S. degree from Chang'an University, Xi'an, China, in 2018, where he is currently pursuing the Ph.D. degree in traffic information engineering and control with the School of Information Engineering.

His research interests are related to autonomous driving environment perception.



**Fei Hui** received the Ph.D. degree from the Department of Computer System Architecture, Xi'an Institute of Microelectronics Technology, Xi'an, China, in 2009. He is currently a Professor with the School of Information Engineering, Chang'an University, Xi'an. He has been involved in the China "863" Project, the Information Technology Major Project of the Transportation Ministry, the National Natural Science Foundation, the National K&R Programmer of China, and so on. His current research interests include connected vehicles, intelligent transportation systems, and autonomous driving virtual simulation testing.



**Xiangmo Zhao** received the Ph.D. degree from Chang'an University, Xi'an, China, in 2006.

He is currently a Professor with the School of Information Engineering, Chang'an University. He is the Vice-President of the Joint Laboratory for Connected Vehicles, the Ministry of Education and China Mobile, and the Shaanxi Road Traffic Intelligent Detection and Equipment Engineering Technology Research Centre and a Leader of the National Key Subjects-Traffic Information Engineering and Control, Chang'an University.

His current research interests include connected vehicles, automated vehicles, intelligent transportation systems, and computer science.

Dr. Zhao is the Director of the Information Professional Committee; a member of the Advisory Expert Group of the China Transportation Association, the National Motor Vehicle Operation Safety Testing Equipment Standardization Committee, and the Leading Group of the National Traffic Computer Application Network; the Vice-Chairman of the Institute of Highway Association on Computer Professional Committee; and the Deputy Director of the Institute of Computer in Shaanxi Province.



**Cheng Wei** received the M.S. degree from Chang'an University, Xi'an, China, in 2020, where he is currently pursuing the Ph.D. degree.

His research interests include autonomous driving virtual simulation testing and cooperative vehicle infrastructure systems.



**Tao Ma** received the B.E. and master's degrees in information engineering from Northwestern Polytechnical University, Xi'an, China, in 2020.

He is currently a full-time Researcher with the Autonomous Driving Group, SenseTime Research, Shanghai, China. His research interests include 3-D object perception and multimodal fusion.



**Weihao Gan** received the B.E. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2012, and the M.S. and Ph.D. degrees in electrical engineering from the Media Communication Laboratory, University of Southern California, Los Angeles, CA, USA, in 2014 and 2017, respectively.

He published more than 20 papers in top-tier conferences and journals. His research interests include computer vision, artificial intelligence, and multimedia data understanding.