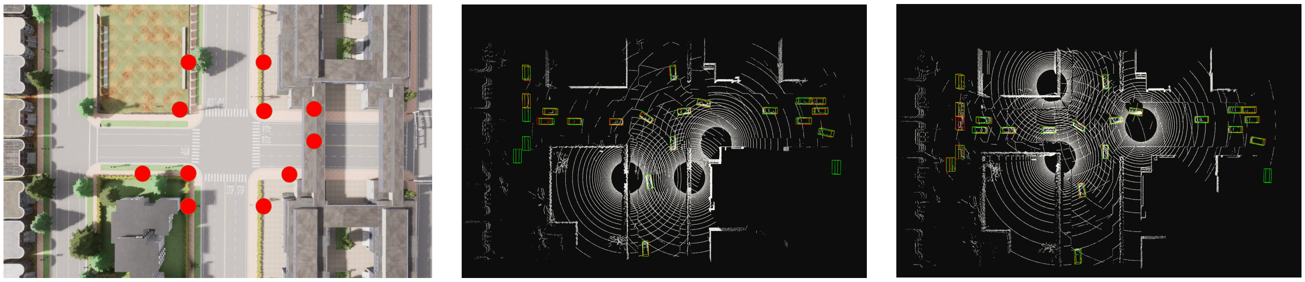
# Optimizing the Placement of Roadside LiDARs for Autonomous Driving

Wentao Jiang[1,2,4] Hao Xiang[3] Xinyu Cai[2] Runsheng Xu[3]
Jiaqi Ma[3] Yikang Li[2*] Gim Hee Lee[4] Si Liu[1*]
[1] Beihang University [2] Shanghai AI Laboratory [3] UCLA [4] NUS



(a) Bird's eye view of a simulated scene (b) Random roadside LiDAR placement (c) Optimized roadside LiDAR placement

Figure 1: An example of roadside LiDAR placement optimization. (a) is a simulated scene in CARLA, where red circles indicate potential LiDAR deployment locations. We perform a random selection as well as our proposed method to select 3 locations for LiDAR setup, as shown in (b) and (c). The green and red bounding boxes denote the ground truth vehicles and the detected vehicles using the detection model, respectively. It can be observed that by optimizing the placement, the red box overlaps more with the green box, indicating an improvement in the detection performance.

## Abstract

*Multi-agent cooperative perception is an increasingly popular topic in the field of autonomous driving, where roadside LiDARs play an essential role. However, how to optimize the placement of roadside LiDARs is a crucial but often overlooked problem. This paper proposes an approach to optimize the placement of roadside LiDARs by selecting optimized positions within the scene for better perception performance. To efficiently obtain the best combination of locations, a greedy algorithm based on perceptual gain is proposed, which selects the location that can maximize the perceptual gain sequentially. We define perceptual gain as the increased perceptual capability when a new LiDAR is placed. To obtain the perception capability, we propose a perception predictor that learns to evaluate LiDAR placement using only a single point cloud frame. A dataset named Roadside-Opt is created using the CARLA simulator to facilitate research on the roadside LiDAR placement problem. Extensive experiments are conducted to demonstrate the effectiveness of our proposed method.*

## 1. Introduction

Accurate perception of the driving environment is critical for autonomous driving, and recent advancements in deep learning have significantly improved the robustness of single-vehicle perception algorithms in tasks such as object detection [19, 29, 10, 1, 26, 34, 39]. Compared to single-vehicle perception, utilizing both vehicle and infrastructure sensors for collaborative perception brings significant advantages, such as a global perspective beyond the current horizon and coverage of blind spots. Current progress in multi-agent communication has enabled such collaboration by sharing the visual data with other agents [36, 2, 25, 33]. Among the collaboration, the LiDAR sensors play a critical role as they can provide reliable geometry information and be directly transferred to real-world coordinate systems from any agent's perspective[6].

To optimize the LiDAR perception performance, the placement of LiDAR sensors is critical [14]. However, current research mainly focuses on the placement of LiDAR sensors on vehicles and neglects the placement problem for roadside LiDAR [14, 23]. As Vehicle-to-Everything (V2X) applications continue to develop, it becomes increasingly important to determine the optimal placement for roadside

[*] Equal corresponding author

LiDARs to maximize their benefits. Unlike vehicle LiDAR, the placement of roadside LiDAR has greater flexibility. As depicted in Figure 1, the roadside LiDAR placement optimization problem aims to maximize perception ability for a scene, usually an intersection with dense traffic flow, by selecting $M$ optimal locations from $N$ potential locations for placing roadside LiDARs.

Existing literature on LiDAR placement such as [14, 23] propose surrogate metrics such as S-MIG and Perception Entropy to evaluate LiDAR placement, while [4] analyzes the correlation between perception performance and LiDAR point cloud density and uniformity. However, these methods cannot be directly applied to roadside LiDAR optimization since they only evaluate LiDAR placement rather than directly optimizing sensor position. Additionally, these surrogate metrics are based on probabilistic occupancy and may not be directly related to Average Precision (AP) obtained by multi-agent detection models, which may result in sub-optimal performance when tested with multi-agent 3D detection models.

To address these challenges, we propose a novel method to directly optimize the placement of roadside LiDAR through a *perceptual gain* based greedy algorithm and a *perception predictor* that learns to evaluate placements quickly. To optimize the placement of roadside LiDAR, it is necessary to tackle the efficiency problem. A straightforward method is to use the brute-force algorithm, which searches all possible combinations of selected locations (*i.e.*, $C_N^M$ choices in total). However, with the number of selected LiDAR $M$ and potential setup locations $N$ increasing, the runtime of the brute-force algorithm becomes unacceptable. Fortunately, we observe that if we select the LiDAR placement sequentially, the obtained perceptual performance exhibits an intuitive diminishing returns property, also known as submodularity [18, 28].

Our proposed *perceptual gain* based greedy algorithm uses this property to obtain approximate optimal solutions that dramatically reduce computation time. Specifically, we place the LiDAR sequentially and choose the potential location to maximize the perceptual gain. The perceptual gain is defined as the increase in perception ability when a newly added LiDAR is placed. To obtain perceptual gain, testing the LiDAR configuration on datasets and calculating Average Precision (AP) is a straightforward way. However, it is still inefficient because calculating the AP on a large dataset is also time-consuming. To overcome this challenge, we develop a novel *perception predictor* that can predict the perception ability of a LiDAR placement using only a single frame of point cloud data. Additionally, we create a dataset called "Roadside-3D" using the realistic self-driving simulator, CARLA [8] since no suitable dataset for roadside LiDAR optimization research exists. We conduct extensive experiments to demonstrate the efficiency and effectiveness of our proposed method.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to study the optimization problem of roadside LiDAR placement. We build the "Roadside-3D" dataset using the CARLA simulator to facilitate the related research.
- We propose a *perceptual gain* based greedy algorithm that obtains approximate optimal solutions for roadside LiDAR placement optimization. This approach dramatically reduces the computational time compared to the simple brute-force algorithm.
- We introduce a novel *perception predictor* that can quickly obtain the perceptual gain by predicting the perception ability of a LiDAR placement.

## 2. Related Work

**Multi-agent/V2X perception.** Multi-agent/V2X perception aims at detecting objects by leveraging the sensing information from multiple connected agents. Current research on V2X perception mainly focuses on V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure). Multi-agent perception can be broadly categorized into early, intermediate, and late fusion. Early fusion [12] directly transforms the raw data and fuses them to form a holistic view for cooperative perception. Intermediate fusion [21, 7, 32, 37, 36, 20, 30] extract the intermediate neural features from each agent's observation and then broadcast these features for feature aggregation while late fusion [24, 11, 5, 27] circulates the detection outputs and uses non-maximum suppression to aggregate the received predictions to produce final results. Although early fusion can achieve outstanding [32] performance as it can preserve complete raw information, it requires large transmission bandwidth to transmit raw LiDAR point clouds, which makes it infeasible to deploy in the real world. On the other hand, late fusion only requires small bandwidth, but its performance is constrained due to the loss of environmental context. Therefore, intermediate fusion has attracted increasing attention due to its good balance between accuracy and transmission bandwidth. V2VNet [32] proposes a spatially aware message-passing mechanism to jointly reason detection and prediction. [31] regresses vehicle localization errors with consistent pose constraints to mitigate outlier messages. DiscoNet [21] uses knowledge distillation to improve the intermediate process. V2X-ViT [36] presented a unified transformer architecture for V2X perception that can capture the heterogeneous nature of V2X systems with strong robustness against various noises. This paper uses early, intermediate, and late fusion to evaluate the average precision of selected roadside LiDAR placement.

**LiDAR placement.** Previous research on LiDAR placement has mainly focused on analyzing and evaluating the placement of vehicle LiDARs for autonomous driving. Ma *et al*. [23] proposed a new method based on Bayesian theory conditional entropy to evaluate the vehicle sensor configuration. Liu *et al*. [22] proposed a bionic volume to surface ratio (VSR) index to solve the optimal LiDAR configuration problem of maximizing utility. Dybedal *et al*. [9] used the mixed integer linear programming method to solve the problem of finding the best placement position of 3D sensors. Hu *et al*. [14] proposed a method for optimizing the placement of vehicle-mounted LiDAR and compared the perceptual performance effects of many traditional schemes. Kim *et al*. [16] investigated how to increase the number of point clouds in a given area and reduce the dead zone as much as possible. Cai *et al*. [4] propose a LiDAR simulation library for the simulation of infrastructure LiDAR sensors. They also analyze the correlation between perception performance and the density/uniformity of the LiDAR point cloud. However, it is not feasible to adapt the existing works to our roadside LiDAR optimization problem since they only propose methods to evaluate different LiDAR placements instead of directly optimizing and searching sensor positions. To this end, we present a novel efficient greedy algorithm to search the optimal positions for roadside LiDAR.

**Algorithm 1:** perceptual gain based greedy method

```
Input: The number of LiDAR $M$, a set of feasible
       LiDAR positions $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N\}$.
Output: A set of selected LiDAR positions
        $\mathcal{S} = \{\boldsymbol{p}_1^*, \boldsymbol{p}_2^*, ..., \boldsymbol{p}_M^*\}$.
Initialize LiDAR position set $\mathcal{S} \leftarrow \varnothing$
for $i \leftarrow 1$ to $M$ do
    Initialize selected LiDAR position $\boldsymbol{p}_i^* \leftarrow \varnothing$
    Initialize maximum perceptual gain $g^* \leftarrow -\infty$
    Calculate the perception score $k_{\mathcal{S}}$ given the
      selected LiDAR positions in $\mathcal{S}$
    for $\boldsymbol{p}$ in $\mathcal{P}$ do
        $\mathcal{S}' \leftarrow \mathcal{S} \cup \{\boldsymbol{p}\}$
        Calculate the perception score $k_{\mathcal{S}'}$ for $\mathcal{S}'$
        Obtain the perceptual gain $g \leftarrow k_{\mathcal{S}'} - k_{\mathcal{S}}$
        if $g > g^*$ then
            $\boldsymbol{p}_i^* \leftarrow \boldsymbol{p}$
            $g^* \leftarrow g$
        end
        $\mathcal{S} \leftarrow \mathcal{S}' \setminus \{\boldsymbol{p}\}$
    end
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{p}_i^*\}$
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \{\boldsymbol{p}_i^*\}$
end
return $\mathcal{S}$
```

## 3. Method

To optimize roadside LiDAR placement, we aim to place $M$ LiDAR sensors at $N$ potential locations where roadside LiDAR can be deployed (*e.g*., street lights, traffic signals, and traffic signs in the scene) such that the optimized placement can maximize the perceptual performance of downstream multi-agent 3D detection models. To avoid using the brute-force algorithm to evaluate every combination of possible locations, we design a novel perceptual gain based greedy algorithm that dramatically reduces computation time. Specifically, we greedily select the location that can maximize the perceptual gain sequentially. The perceptual gain is defined as the increase in perception ability when a new LiDAR is placed, which we will describe in Sec 3.1. To obtain the perceptual gain efficiently, we propose a perception predictor that learns to evaluate LiDAR placements using only a single frame point cloud data, which is detailed in Sec 3.2.

### 3.1. Perceptual gain based greedy method

Given a feasible set of LiDAR positions $\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N\}$, a straightforward way to optimize roadside LiDAR placement is to use a brute-force algorithm. After evaluating all possible combinations of selected locations (*i.e*., $C_N^M$ decisions in total), the location with the highest average precision is selected. However, the running time of the brute-force algorithm becomes unacceptable as $N$ and $M$ become larger. We observe that when we sequentially select the placement of the LiDAR, the obtained perceptual performance exhibits an intuitive diminishing returns property, also known as submodularity [18, 28, 15]. Inspired by this property, we propose a *perceptual gain* based greedy algorithm to obtain an approximate optimal placement, as shown in the Algorithm 1.

Our algorithm sequentially selects $M$ position from a set of feasible LiDAR positions $\mathcal{P}$. In each loop, we tentatively place a LiDAR at each feasible position and compute the perceptual gain $g = k_{\mathcal{S}'} - k_{\mathcal{S}}$ of the trial placement $\boldsymbol{p}$, where $k_{\mathcal{S}}$ and $k_{\mathcal{S}'}$ are the perception scores before and after placing the LiDAR at $\boldsymbol{p}$ respectively. The concept of perceptual gain denotes the increase in perceptual ability when an additional LiDAR is placed at $\boldsymbol{p}$. The feasible position that yields the maximum perceptual gain $\boldsymbol{p}^*$ will be added to the selected LiDAR position set $\mathcal{S}$ and removed from $\mathcal{P}$. After selecting $M$ LiDARs sequentially, we can obtain a set of optimized LiDAR positions $\mathcal{S}$.

Noted that our method does not need practical trial-and-error for selection. Instead, we only need to collect sequential point cloud data at each location once before using our method. In the implementation, we select an ego LiDAR

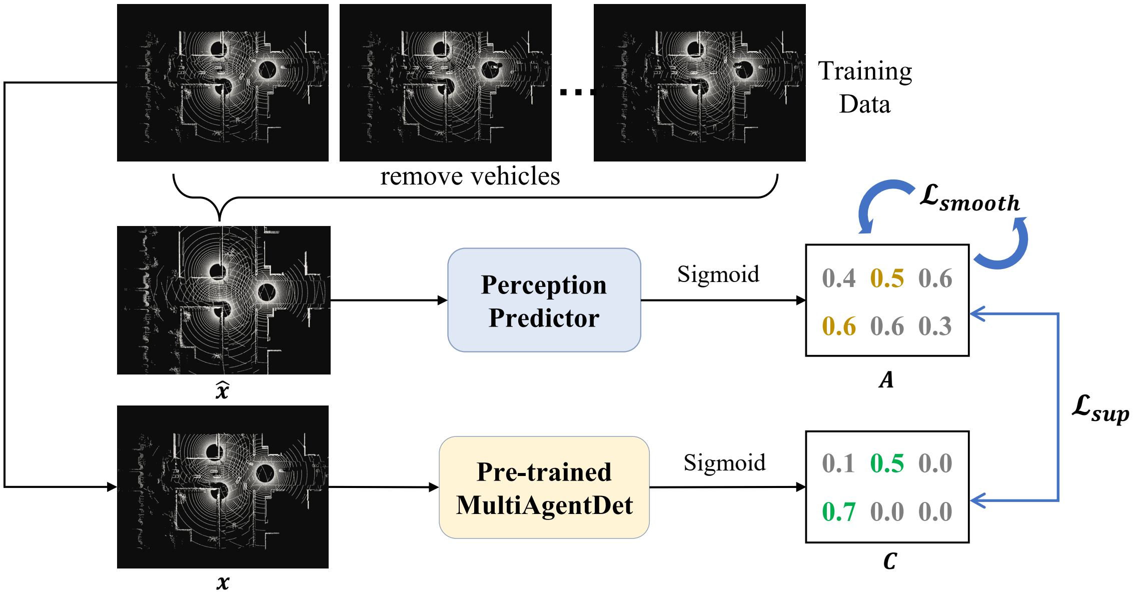


Figure 2: We obtain a frame of point cloud data without vehicle $\hat{\boldsymbol{x}}$ by selectively fusing different frames using labeled bounding boxes. Our perception predictor takes $\hat{\boldsymbol{x}}$ as input and predicts a perception ability map $\boldsymbol{A}$. We supervise $\boldsymbol{A}$ with $\mathcal{L}_{sup}$ using the confidence map $\boldsymbol{C}$ generated from a pre-trained multi-agent detection model. Note that we supervise $\boldsymbol{A}_{i,j}$ only if $\boldsymbol{C}_{i,j} >$ threshold (*i.e.*, elements with color in the image above). A smoothing loss $\mathcal{L}_{smooth}$ is also adopted for robustness.

and then project point cloud data obtained by other LiDARs to the ego coordinate system, eliminating the need to place and remove LiDARs. This makes our method very efficient.

## 3.2. Perception predictor for perceptual gain

**Perception predictor.** In our proposed greedy method for optimizing sensor placement, the calculation of the perceptual gain $g = k_{\mathcal{S}'} - k_{\mathcal{S}}$ is the most important part. Nevertheless, it is time-consuming and computationally inefficient to evaluate the true detection performance for every possible position based on a whole dataset. To this end, we propose a *perception predictor* that learns to predict the perception score for arbitrary sensor placement, as Figure 2 shows.

In the training phase, we first randomly select multiple roadside LiDARs from $\mathcal{S}$ to form a subset $\hat{\mathcal{S}}$. We then project the sequential training point cloud data collected from $\hat{\mathcal{S}}$ into the coordinate system of a randomly selected ego LiDAR. Since we have sequential point cloud data labeled with vehicle bounding boxes, we can obtain a frame of point cloud data *without* vehicle by selectively fusing different frames. The selective fusion removes vehicles through bounding box annotations and obtains a frame of point cloud data termed as $\hat{\boldsymbol{x}}$. Our proposed perception predictor aims to predict a perception ability map $\boldsymbol{A} \in \mathbb{R}^{H\times W}$ for LiDAR placement $\hat{\mathcal{S}}$ given $\hat{\boldsymbol{x}}$, where $\boldsymbol{A}_{i,j}$ denote model's perception ability for the placement $\hat{\mathcal{S}}$ if there is a vehicle at location $(i, j)$.

The motivation of the perception predictor is to learn the correlation between the perception ability and the pattern/distribution of the point cloud, enabling a quick evaluation of a roadside LiDAR placement. We remove the vehicles in the point cloud data to retain the original distribution of the point cloud data. The proposed perception predictor uses PointPillar [19] as the backbone to extract point cloud features from $\hat{\boldsymbol{x}}$ and then generate a perception ability map $\boldsymbol{A}$. We supervise the perception predictor using the confidence map $\boldsymbol{C}$ produced by a pre-trained multi-agent detection model [37]. Mathematically, the process of training is as follows:

$$\begin{aligned}
\boldsymbol{A} &= \mathrm{Sigmoid}\left(\mathrm{predictor}(\hat{\boldsymbol{x}})\right) \\
\boldsymbol{C} &= \mathrm{Sigmoid}(\mathrm{MultiAgentDet}(\boldsymbol{x})) \\
\boldsymbol{K}_{i,j} &= \begin{cases} 1, & \text{if } \boldsymbol{C}_{i,j} > \text{threshold} \\ 0, & \text{otherwise} \end{cases} \\
\mathcal{L}_{sup} &= \|\boldsymbol{K}\cdot(\boldsymbol{C}-\boldsymbol{A})\|,
\end{aligned} \tag{1}$$

where $\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{K} \in \mathbb{R}^{H\times W}$ denote the predicted confidence map, supervision confidence map, and supervision mask. $H$ and $W$ denote the output height and width. $\mathcal{L}_{sup}$ denote the supervision loss. Note that for different training point cloud data $x$, the vehicles appear in different locations. We should only supervise $\boldsymbol{A}_{i,j}$ when vehicles appear to exist at the location $(i, j)$ of $x$ since $\boldsymbol{A}_{i,j}$ denote the perception ability if there is a vehicle at location $(i, j)$. Thus, we set a threshold for the produced confidence map $\boldsymbol{C}$ and obtain the supervision mask $\boldsymbol{K}$, which enables the supervision on location $(i, j)$ only if $\boldsymbol{C}_{i,j} >$ threshold.

To increase the robustness and generalization, we also add a smoothing loss $\mathcal{L}_{smooth}$ to the overall loss $\mathcal{L}$:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{sup} + \gamma \mathcal{L}_{smooth}\\ \mathcal{L}_{smooth} &= \frac{1}{HW} \sum_{i,j} |\boldsymbol{A}_{i,j} - \boldsymbol{A}_{i-1,j}| + |\boldsymbol{A}_{i,j} - \boldsymbol{A}_{i,j-1}|\\ &+ |\boldsymbol{A}_{i,j} - \boldsymbol{A}_{i+1,j}| + |\boldsymbol{A}_{i,j} - \boldsymbol{A}_{i,j+1}|.\end{aligned} \tag{2}$$

Intuitively, the perceptual abilities of neighboring regions do not vary much. So for each element in $\boldsymbol{A}$ we compute the difference between it and its neighbors. We then include $\mathcal{L}_{smooth}$ in the overall supervision $\mathcal{L}$ with a relatively small factor $\gamma$.

**Perceptual gain.** The trained perception predictor learns to find the correlation between the perception ability and the pattern/distribution of the point cloud, which allows a quick evaluation of a roadside LiDAR placement. Thus, we accumulate the perception ability map $\boldsymbol{A}$, which we refer to as the perception score $k$:

$$k = \sum_{i,j} \boldsymbol{A}_{i,j}. \tag{3}$$

Through tentatively placing a LiDAR at every feasible position, we compute the perceptual gains $g = k_{\mathcal{S}'} - k_{\mathcal{S}}$ of the trial placement $\mathcal{S}'$, where $k_{\mathcal{S}}$ and $k_{\mathcal{S}'}$ are the perception scores before and after placing the LiDAR respectively. The feasible position that maximizes perceptual gain will be added to the selected LiDAR position set $\mathcal{S}$ and removed from $\mathcal{P}$.

# 4. Experiment

## 4.1. Dataset and settings

**Dataset.** In recent years, an increasing number of autonomous driving datasets have been released. In addition to single-agent datasets such as KITTI [13] and nuScenes [3], there are several multi-agent/V2X datasets that contain roadside LiDAR data, including the synthetic dataset OPV2V [37], V2XSet [36], V2XSim [21], and realistic dataset DAIR-V2X [40], Rope3D [38]. However, all these multi-agent datasets contain only 1 or 2 roadside LiDARs in each scene, which is far from what we need for the experiment. Since there is no suitable dataset for roadside LiDAR optimization research, we build our own dataset named Roadside-Opt using the realistic CARLA simulator [8] and simulation tool OpenCDA [35]. With the simulator, we build our dataset so that we can fairly evaluate different LiDAR placements with all other environmental factors fixed, such as the trajectories of the ego vehicle and surrounding objects.

To collect the Roadside-Opt dataset, we build 10 different scenes in CARLA Town $3, 4, 5, 6$ with recorded dense traffic flow. For each scene, we set up at least 15 feasible roadside LiDAR positions and spawn at least 50 vehicles in the scene. Feasible roadside LiDAR positions are collected by finding streetlights and traffic lights on the scene. If there are only a few streetlights or traffic lights, we randomly generate multiple feasible roadside LiDAR positions in the scene. The collected Roadside-Opt dataset contains $37,641$ point cloud frames from 126 roadside LiDARs in total. Each roadside LiDAR has a rotation frequency of 20Hz and approximately 20000 points per frame. We take $40\%$ of the scenes as the test set and the remaining data as the training set for all the experiments.

**Implementation details.** For every experiment, we customize the range of each LiDAR to be $l = 140$m, $w = 40$m, $h = 4$m following the settings of [37, 36]. We pre-train the multi-agent 3D object detection method [37] in the training set of Roadside-Opt for the supervision of perception predictor. The factor $\gamma$ for the smoothing loss $\mathcal{L}_{smooth}$ is set to $0.1$. The threshold set for calculating $\mathcal{L}_{sup}$ for training the perception predictor is $0.2$.

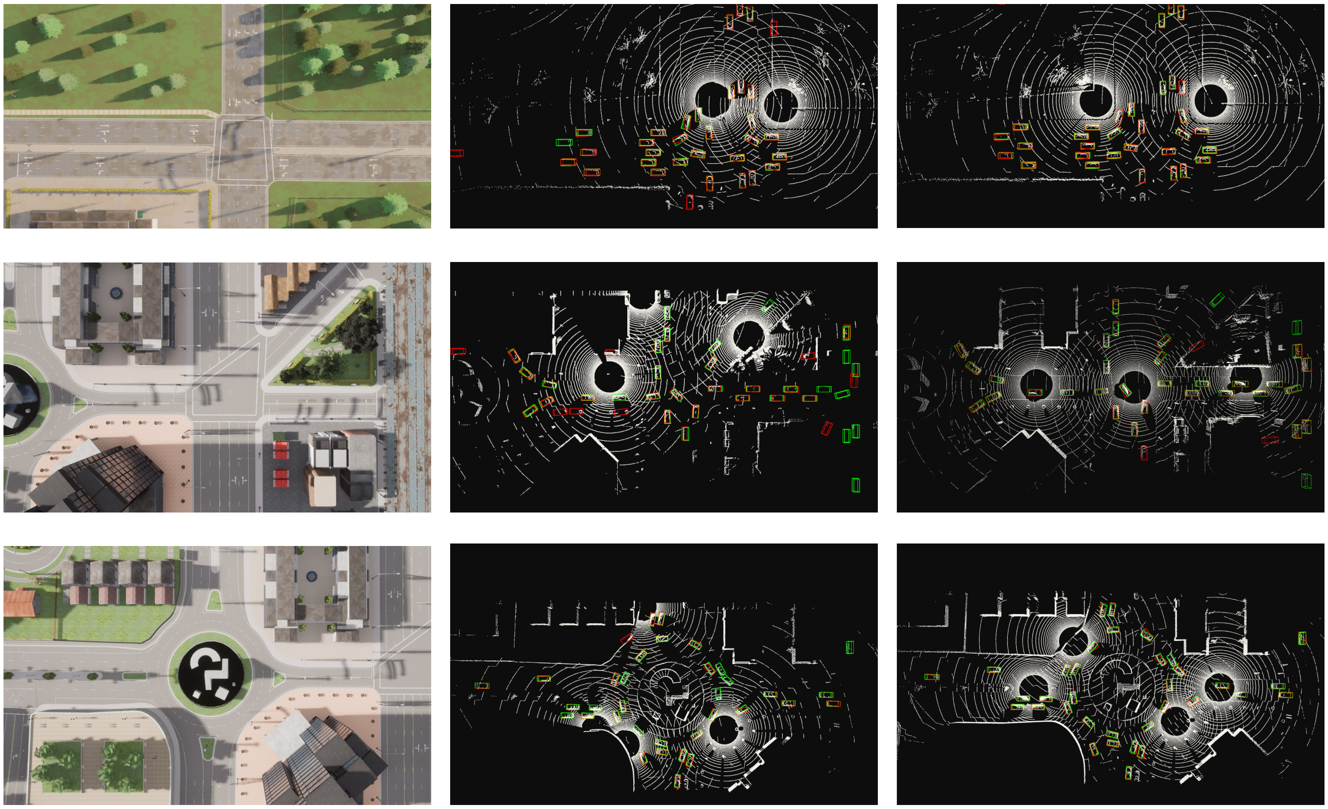
We follow the existing multi-agent perception papers [37, 36] to use Average Precision (AP) as the metric to evaluate the perception performance after selecting a roadside LiDAR placement. We compute APs through three types of fusion methods including early fusion, intermediate fusion, and late fusion. Early fusion directly projects and aggregates raw LiDAR point clouds from other agents. Late fusion gathers all detected outputs from all agents and applies Non-Maximum Suppression (NMS) to obtain the final results. For intermediate fusion, we evaluate APs using a well-known multi-agent 3D detection method [37]. For a fair comparison, all three types of fusion methods use PointPillar [19] as the backbone and are trained on the training set of the Roadside-Opt dataset for 50 epochs. We use the Adam optimizer [17] with an initial learning rate of 0.001 for all the models. Experiments compared with [14] and a coverage/density baseline that first maximizes the coverage of LiDARs and then the density of the point cloud can be found in the supplemental material.

## 4.2. Experimental Results

**Quantitative experiments.** We conduct quantitative experiments on the test set of our proposed Roadside-Opt dataset and set the number of LiDARs $M = 2, 3, 4$, as shown in Table 1. We randomly select a combination of LiDAR positions for 3 times and calculate the mean and variance of the obtained APs as a baseline. We employ three

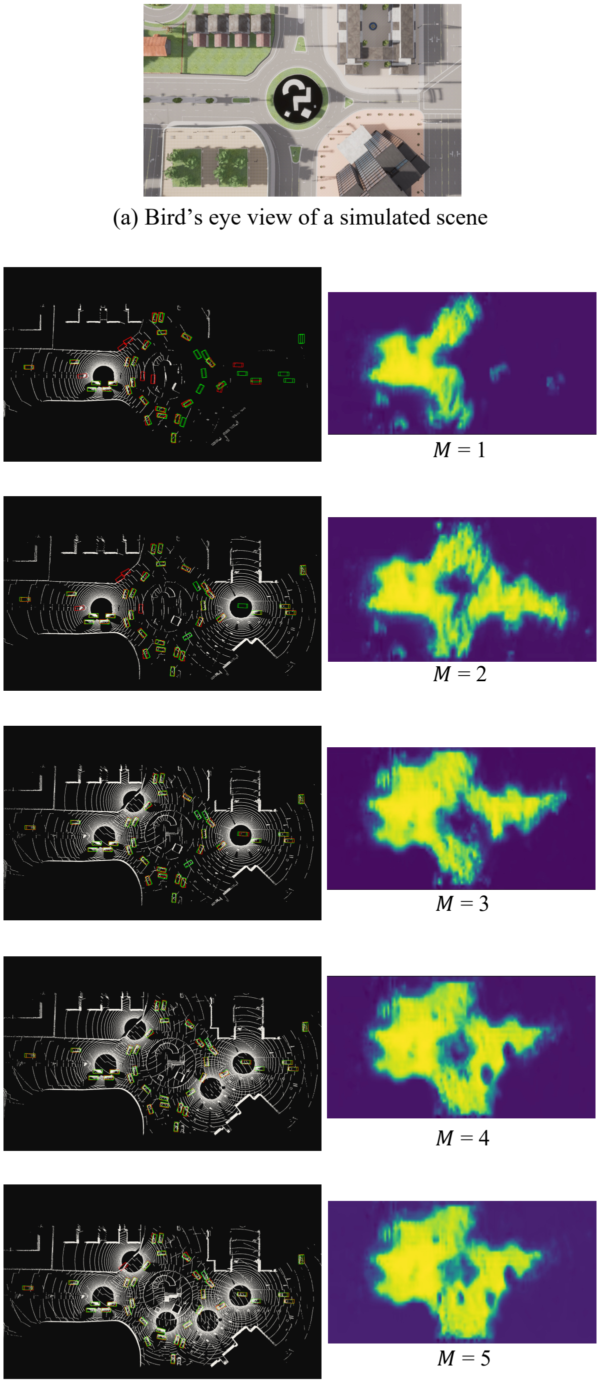
Table 1: Quantitative experiments on the test set of the Roadside-Opt dataset. Three types of fusion methods are adopted to evaluate the average precision after selecting the LiDAR placement. The row "Random" indicates randomly selecting $M$ LiDAR positions from all the potential positions. The results of the Upper bound are obtained using the brute-force algorithm that tests every possible placement (*i.e.*, $C_M^N$ in total).

| | Early Fusion | | | Intermediate Fusion | | | Late Fusion | | |
|---|---|---|---|---|---|---|---|---|---|
| | AP@0.3 | AP@0.5 | AP@0.7 | AP@0.3 | AP@0.5 | AP@0.7 | AP@0.3 | AP@0.5 | AP@0.7 |
| The number of LiDARs $M = 2$ | | | | | | | | | |
| Random | $0.68 \pm 0.11$ | $0.64 \pm 0.10$ | $0.54 \pm 0.08$ | $0.48 \pm 0.08$ | $0.44 \pm 0.08$ | $0.31 \pm 0.04$ | $0.40 \pm 0.18$ | $0.31 \pm 0.15$ | $0.22 \pm 0.09$ |
| Ours | **0.82** | **0.78** | **0.56** | **0.74** | **0.63** | **0.40** | **0.65** | **0.56** | **0.31** |
| Upper bound | 0.85 | 0.82 | 0.72 | 0.78 | 0.71 | 0.52 | 0.70 | 0.61 | 0.43 |
| The number of LiDARs $M = 3$ | | | | | | | | | |
| Random | $0.75 \pm 0.20$ | $0.73 \pm 0.19$ | $0.62 \pm 0.18$ | $0.57 \pm 0.10$ | $0.54 \pm 0.08$ | $0.43 \pm 0.07$ | $0.49 \pm 0.13$ | $0.40 \pm 0.12$ | $0.33 \pm 0.10$ |
| Ours | **0.89** | **0.88** | **0.81** | **0.80** | **0.75** | **0.56** | **0.71** | **0.65** | **0.45** |
| The number of LiDARs $M = 4$ | | | | | | | | | |
| Random | $0.79 \pm 0.17$ | $0.77 \pm 0.16$ | $0.70 \pm 0.13$ | $0.70 \pm 0.14$ | $0.69 \pm 0.14$ | $0.61 \pm 0.12$ | $0.65 \pm 0.16$ | $0.61 \pm 0.15$ | $0.51 \pm 0.12$ |
| Ours | **0.92** | **0.91** | **0.89** | **0.83** | **0.81** | **0.65** | **0.78** | **0.73** | **0.56** |



(a) Bird's eye view of a simulated scene    (b) Random roadside LiDAR placement    (c) Optimized roadside LiDAR placement

Figure 3: We visualize one frame from each of the three test scenes, where $2, 3, 4$ LiDARs are placed in each scene. (a) are the simulated scenes of the test set. (b) are the visualization of the baseline, which generate random placements from all the potential locations. (c) are the visualization of optimizing the placement using our proposed method. The green and red bounding boxes denote the ground truth vehicles and the detected vehicles using the pre-trained detection model, respectively.


(a) Bird's eye view of a simulated scene

$M = 1$

$M = 2$

$M = 3$

$M = 4$

$M = 5$

(b) Optimized placement (c) Perception ability map $\boldsymbol{A}$

Figure 4: Visualization of the process of selecting optimal locations that maximize the perceptual gain sequentially (from 1 LiDAR to 5 LiDARs).

Table 2: Quantitative experiments on the process of selecting optimal locations that sequentially maximize perceptual gain (from 1 LiDAR to 5 LiDARs).

| | Intermediate Fusion | | |
|---|---|---|---|
| | AP@0.3 | AP@0.5 | AP@0.7 |
| The number of LiDARs $M = 1$ | | | |
| Ours | 0.50 | 0.38 | 0.17 |
| The number of LiDARs $M = 2$ | | | |
| Ours | 0.81 | 0.69 | 0.38 |
| The number of LiDARs $M = 3$ | | | |
| Ours | 0.87 | 0.81 | 0.54 |
| The number of LiDARs $M = 4$ | | | |
| Ours | 0.90 | 0.88 | 0.67 |
| The number of LiDARs $M = 5$ | | | |
| Ours | 0.91 | 0.90 | 0.76 |

types of multi-agent fusion strategies, including early fusion, intermediate fusion, and late fusion. The perceptual gain based greedy method sequentially computes the feasible position that can maximize the perceptual gain. And our proposed perception predictor can efficiently evaluate perceptual gain for a trial location. By incorporating the perceptual gain based greedy method and the perception predictor, our method significantly outperforms the random baseline method. We also compute the upper bound of perceptual performance, which is to use the brute-force algorithm that tests every possible placement (*i.e.*, $C_M^N$ in total). Since the brute-force algorithm is very time-consuming, we only experiment on the cases where $M = 2$. We can observe that the result obtained by our greedy method is close to the upper bound, which shows that our method is able to obtain approximately optimal solutions and also dramatically reduce the computation time. The calculation time of the brute-force method (upper bound) exceeds 11 hours and 46 hours when the number of LiDAR is 2 and 3, as it requires evaluating the Average Precision (AP) on the test set for every possible combination. While our proposed greedy method calculates the optimal placement within 10 seconds.

**Visualization of optimized example.** We perform the experiment on the test set of the Roadside-Opt dataset and visualize some of the examples, as shown in Figure 3. We visualize one frame from each of the three test scenes, where $2, 3, 4$ LiDARs are placed in each scene. Column (a) shows the bird's eye view of three simulated scenes in the CARLA simulator. Column (b) shows the baseline of randomly select $M$ LiDARs from all the feasible setup locations. Column (c) indicates the visualization of optimizing the placement with our proposed method. We can observe that by optimizing the placement, the selected LiDAR positions are more reasonable, which covers more areas in the scene. The red box obtained by optimal LiDAR placement overlaps more with the green box, indicating an improvement in perceptual performance.

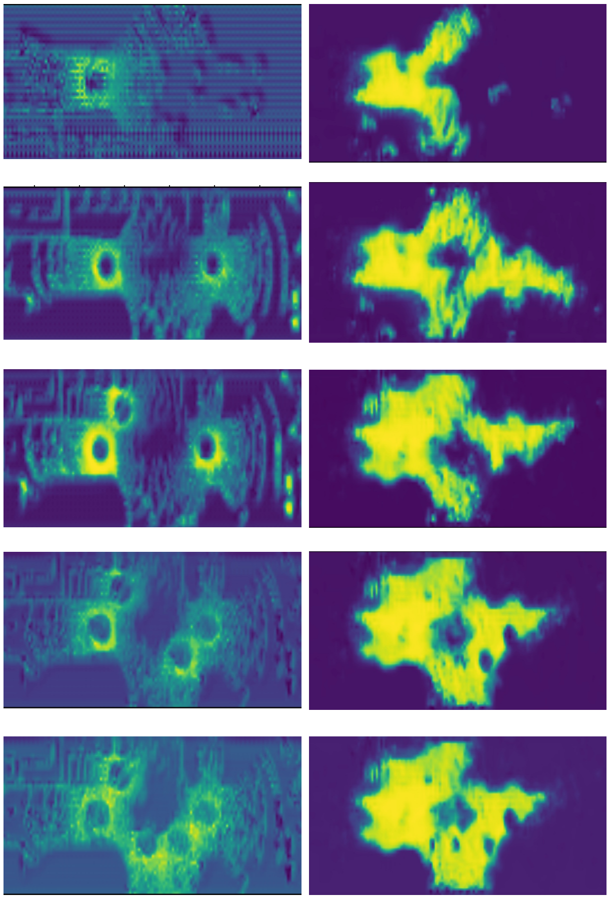
## 4.3. Analysis

**Visualization of the sequential selection.** In the method section, we propose to use a perceptual gain based greedy method to sequentially select optimal LiDAR based on the calculated perceptual gain. In Figure 4, we visualize the process of sequentially selecting optimal locations, where the number of LiDAR increase from 1 to 5. All experiments use the same pre-trained multi-agent 3D detection model for a fair comparison. The left column visualizes different roadside LiDAR placements under the same point cloud data from the test set. In the greedy method, we sequentially select the locations to deploy the LiDAR, which covers more and more areas of the intersection and produces increasingly accurate red bounding boxes, as shown in the left column. We also visualize the predicted perceptual map for each step, as shown in the right column. As the number of LiDAR deployed increases, the perception ability map expands to cover most of the intersection.

**Quantitative analysis of selecting optimal LiDAR sequentially.** To quantitatively analyze the process of the greedy algorithm, we evaluate the average precision for the 5 placement shown in Table 2 using the intermediate fusion method [37]. As the number of placed LiDARs increases, the obtained perceptual performance rapidly increases as we select the location that can maximize the perceptual gain. The perceptual performance also exhibits an intuitive diminishing returns property as we place more LiDARs.

**Visualization of perception ability map.** In the method section, we propose to use the perception predictor to quickly evaluate the LiDAR placement. We train the perception predictor using $\mathcal{L}_{sup}$, which uses the confidence map produced by the off-the-shelf detection model for supervision. A smoothing loss $\mathcal{L}_{smooth}$ is also adopted with a factor $\gamma$ to increase the robustness and generalization. We visualize the perception ability map $\boldsymbol{A}$ obtained from the perception predictor trained with and without smoothing loss, as shown in Figure 5. We can observe that the perception ability maps in the left column are sparse and discontinuous, while the perception ability maps in the right column are more continuous and smoother. This is because when training the perception predictor, the vehicles do not appear in every location of the training frames. Thus the confidence map $\boldsymbol{C}$ used for supervision is sparse since we set a threshold to obtain the supervision mask $\boldsymbol{K}$. Intuitively, the perception abilities of neighboring regions do not vary much. We then add a smoothing loss for the training of the perception predictor with an appropriate factor $\gamma$, which makes the obtained perception ability map more continuous.

**Limitations.** The main limitation of our method and also the current methods is the application to the real world. Both ours and existing papers [4, 14, 23] that study the LiDAR configuration problem are all tested in the simulation environment, as evaluating placements with all other environmental factors fixed in the real world is infeasible. To adapt our method to the real world, users need to build digital towns in CARLA simulator that mimic real-world road topology and spawn realistic traffic flow in the simulator. For instance, OPV2V [37] has created a digital town of Culver City, Los Angeles, following the aforementioned steps, which accurately replicates the real-world environment.



(a) Perception ability map $\boldsymbol{A}$ w.o. $\mathcal{L}_{\text{smooth}}$

(b) Perception ability map $\boldsymbol{A}$ with $\mathcal{L}_{\text{smooth}}$ , $\gamma = 0.1$

Figure 5: Visualization of the perception ability map $\boldsymbol{A}$ in the process of selecting optimal locations sequentially using our method. The detail of the scene is shown in Figure 4. $\boldsymbol{A}$ obtained by perception predictor trained with and without the smoothing loss $\mathcal{L}_{smooth}$ are shown in column (b) and (a) respectively.

## 5. Conclusion

In this paper, we aim to optimize the placement of roadside LiDARs for multi-agent cooperative perception in autonomous driving, which is a crucial but rarely studied problem. We propose a novel perceptual gain based greedy method that selects optimized positions sequentially. For each step in the sequence, we greedily choose the position that can maximize the perceptual gain. Perceptual gain indicates the increase in perceptual ability when a new LiDAR is placed. To obtain the perceptual gain, we design a perception predictor that learns to evaluate a LiDAR placement using a single frame of point cloud data without vehicles. Since there is no suitable dataset for the experiment, we build our own Roadside-Opt dataset containing scenes with multiple roadside LiDARs. Extensive experiments are performed on the newly proposed dataset, demonstrating the effectiveness of our method. Our proposed greedy method can obtain approximately optimal solutions and also greatly reduce the computation time. We believe our method can also be used in other tasks that require optimization on LiDAR placement.

**Acknowledgement.** This work was supported in part by Science and Technology Commission of Shanghai Municipality (Grant No. 22DZ1100102), in part by the National Key R&D Program of China under Grant 2022ZD0115502, in part by the National Natural Science Foundation of China under Grant 62122010, in part by the Fundamental Research Funds for the Central Universities, in part by the Tier 2 grant MOE-T2EP20120-0011 from the Singapore Ministry of Education, and in part by Shanghai Artificial Intelligence Laboratory.

## References

[1] W. Ali, S. Abdelkarim, M. Zahran, M. Zidan, and A. E. Sallab. Yolo3d: End-to-end real-time 3d oriented object bounding box detection from lidar point cloud. In *ECCV 2018: ”3D Reconstruction meets Semantics” workshop*, 2018. 1

[2] Sushma U Bhover, Anusha Tugashetti, and Pratiksha Rashinkar. V2x communication protocol in vanet for co-operative intelligent transportation system. In *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 602–607. IEEE, 2017. 1

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5

[4] Xinyu Cai, Wentao Jiang, Runsheng Xu, Wenquan Zhao, Jiaqi Ma, Si Liu, and Yikang Li. Analyzing infrastructure lidar placement with realistic lidar. *arXiv preprint arXiv:2211.15975*, 2022. 2, 3, 8

[5] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019. 2

[6] Guang Chen, Fan Lu, Zhijun Li, Yinlong Liu, Jinhu Dong, Junqiao Zhao, Junwei Yu, and Alois Knoll. Pole-curb fusion based robust and efficient autonomous vehicle localization system with branch-and-bound global optimization and local grid map method. *IEEE Transactions on Vehicular Technology*, 70(11):11283–11294, 2021. 1

[7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 2, 5

[9] Joacim Dybedal and Geir Hovland. Optimal placement of 3d sensors considering range and field of view. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1588–1593. IEEE, 2017. 3

[10] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 1

[11] Chen Fu, Chiyu Dong, Christoph Mertz, and John M Dolan. Depth completion via inductive fusion of planar lidar and monocular camera. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10843–10848. IEEE, 2020. 2

[12] Hongbo Gao, Bo Cheng, Jianqiang Wang, Keqiang Li, Jianhui Zhao, and Deyi Li. Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics*, 14(9):4224–4231, 2018. 2

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5

[14] Hanjiang Hu, Zuxin Liu, Sharad Chitlangia, Akhil Agnihotri, and Ding Zhao. Investigating the impact of multi-lidar placement on object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2550–2559, 2022. 1, 2, 3, 5, 8

[15] Qiangqiang Huang, Joseph DeGol, Victor Fragoso, Sudipta N Sinha, and John J Leonard. Optimizing fiducial marker placement for improved visual localization. *arXiv preprint arXiv:2211.01513*, 2022. 3

[16] Tae-Hyeong Kim and Tae-Hyoung Park. Placement optimization of multiple lidar sensors for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):2139–2145, 2019. 3

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3:71–104, 2014. 2, 3

[19] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of*

*the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 1, 4, 5

[20] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. *arXiv preprint arXiv:2207.08560*, 2022. 2

[21] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552, 2021. 2, 5

[22] Zuxin Liu, Mansur Arief, and Ding Zhao. Where should we place lidars on the autonomous vehicle?-an optimal design approach. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2793–2799. IEEE, 2019. 3

[23] Tao Ma, Zhizheng Liu, and Yikang Li. Perception entropy: A metric for multiple sensors configuration evaluation and design. *arXiv preprint arXiv:2104.06615*, 2021. 1, 2, 3, 8

[24] Gledson Melotti, Cristiano Premebida, and Nuno Gonçalves. Multimodal deep-learning for object recognition combining camera and lidar data. In *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 177–182. IEEE, 2020. 2

[25] Cristina Olaverri-Monreal, Javier Errea-Moreno, Alberto Díaz-Álvarez, Carlos Biurrun-Quel, Luis Serrano-Arriezu, and Markus Kuba. Connection of the sumo microscopic traffic simulator and the unity 3d game engine to evaluate v2x communication-based systems. *Sensors*, 18(12):4399, 2018. 1

[26] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[27] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer. Car2x-based perception in a high-level fusion architecture for cooperative perception systems. In *Intelligent Vehicles Symposium*, 2012. 2

[28] Mike Roberts, Debadeepta Dey, Anh Truong, Sudipta Sinha, Shital Shah, Ashish Kapoor, Pat Hanrahan, and Neel Joshi. Submodular trajectory optimization for aerial 3d scanning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5324–5333, 2017. 2, 3

[29] M. Simon, S. Milz, K. Amende, and H. M. Gross. Complex-yolo: Real-time 3d object detection on point clouds. *arXiv preprint arXiv.1803.06199*, 2018. 1

[30] Sanbao Su, Yiming Li, Sihong He, Songyang Han, Chen Feng, Caiwen Ding, and Fei Miao. Uncertainty quantification of collaborative detection for self-driving. 2023. 2

[31] N. Vadivelu, M. Ren, J. Tu, J. Wang, and R. Urtasun. Learning to communicate and correct pose errors. 2020. 2

[32] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision*, pages 605–621. Springer, 2020. 2

[33] Hao Xiang, Runsheng Xu, Xin Xia, Zhaoliang Zheng, Bolei Zhou, and Jiaqi Ma. V2xp-asg: Generating adversarial scenes for vehicle-to-everything perception. *arXiv preprint arXiv:2209.13679*, 2022. 1

[34] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, 2014. 1

[35] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162. IEEE, 2021. 5

[36] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. *arXiv preprint arXiv:2203.10638*, 2022. 1, 2, 5

[37] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE, 2022. 2, 4, 5, 8

[38] Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. Rope3d: the roadside perception dataset for autonomous driving and monocular 3d object detection task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21341–21350, 2022. 5

[39] Z. Yin and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[40] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022. 5