# Vehicle-Infrastructure Cooperative 3D Detection via Point Cloud Filtering

1st Wenchao Yan
*College of Software*
*Huazhong University of Science and Technology*
WuHan, China
wenchaoyan@foxmail.com

2nd Hua Cao*
*College of Software*
*Huazhong University of Science and Technology*
WuHan, China
caohua226@hust.edu.cn

3rd Jiazhong Chen
*College of Computer Science and Technology*
*Huazhong University of Science and Technology*
WuHan, China
jzchen@hust.edu.cn

2023 8th International Conference on Image, Vision and Computing (ICIVC) | 979-8-3503-3523-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIVC58118.2023.10269906

***Abstract*—Restricted by the vehicle's field of view and location, there are problems such as a lack of global perspective and limited long-distance sensing capability in the process of autonomous driving, so the assistance of road-side information is needed. At present, vehicle-infrastructure cooperation methods mainly suffer from the problems of vehicle-infrastructure sensor heterogeneity, spatial and temporal matching of vehicle-infrastructure sensors, high transmission cost, and significant computation on the vehicle-side. This paper proposes an early fusion with the point cloud filtering method for 3D object detection to reduce the transmission cost and alleviate the computational effort at the vehicle-side, which divides the whole 3D object detection process into three stages: First, a 3D object detection performed at the road-side, and the output prediction boxes are used as the road-side point cloud filter boxes, then the road-side point cloud is filtered according to the filter boxes, and only the point clouds within a specific range of the filter boxes are retained and transmitted to the vehicle-side. Finally, the vehicle-side fuses the point clouds from the road-side for vehicle 3D object detection and outputs the final detection results. The experimental results show that compared with the benchmark of the DAIR-V2X dataset, the transmission cost of this method is reduced by 92.07%. The detection accuracy is improved by 1.37% and 12.88% on average compared with the early fusion and late fusion methods in the benchmark, maintaining high accuracy while significantly reducing the communication load and saving the consumption of computational resources on the vehicle-side.**

***Index Terms*—automatic driving, vehicle-infrastructure coordination, 3D object detection, point cloud fusion, data filtering**

## I. Introduction

Autonomous driving requires a high level of accuracy and real-time performance. At the same time, the vehicle has limited perceivable external information and computing power, so it needs the support of external information and arithmetic power. With the development of the automotive industry driven by electrification, intelligence, networking, and sharing, all major global powers have made intelligent networked vehicles a national strategic development direction. The technological development of cellular telematics, edge computing networks, and high-precision positioning systems has provided adequate support for the comprehensive integration of vehicle-vehicle, vehicle-road, vehicle-human, and vehicle-cloud systems, forming a complete set of vehicle-infrastructure collaboration solutions [1].

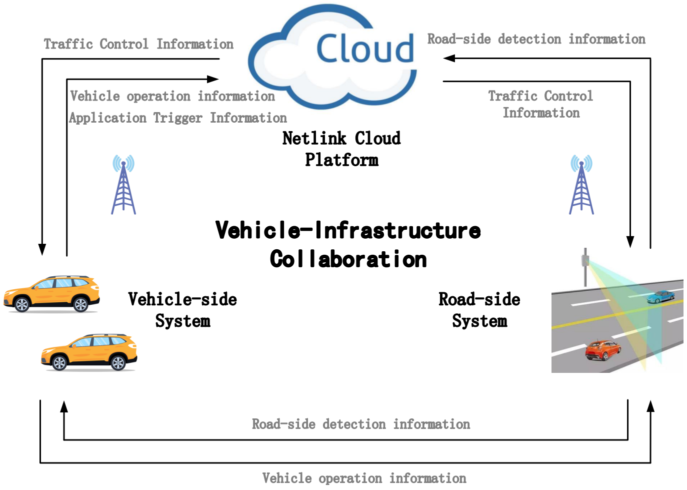

Cloud
Traffic Control Information
Road-side detection information
Vehicle operation information
Application Trigger Information
Traffic Control Information
Netlink Cloud Platform
Vehicle-Infrastructure Collaboration
Vehicle-side System
Road-side System
Road-side detection information
Vehicle operation information

Fig. 1. Vehicle-infrastructure collaboration framework.

As shown in Fig. 1, vehicle-infrastructure collaboration can divide into three core components: the vehicle-side system, the road-side system, and the network-linked cloud platform, which transmit and interact through communication facilities such as base stations [2]. The vehicle system is equipped with cameras, LiDAR, and positioning modules, which can sense and process information on the road and transmit it to the road-side system and the cloud platform. The road-side system is also equipped with surveillance cameras, LiDAR, and other detection equipment, which can detect traic information on the road and transmit information to the cloud platform and the vehicle-side system to assist detection. The netlink cloud platform will receive and calculate the data sent from the

This work was supported by the National Key R&D Program of the Ministry of Science and Technology, 2022YFB2601802, "Digital Construction Demonstration Application of Traffic Facilities in Xiongan New Area", Project 2, "Research and Application of Universal and Reliable Traffic Condition Sensing Technology for Urban Roads".

* Corresponding author: Hua Cao (caohua226@hust.edu.cn)

979-8-3503-3523-1/23/$31.00 ©2023 IEEE

vehicle-side and road-side systems, and send the computed results to the vehicle-side system and road-side system according to the corresponding rules to assist detection, and also send traffic control and other information to play a coordinating role in planning. Vehicle-infrastructure collaboration is considered the path to achieving L5-level autonomous driving. With the help of road-side data, it can, on the one hand, improve the perception accuracy and assist in the precise positioning of vehicles, and on the other hand, expand the perception range of vehicles to cover blind areas. However, the following problems still exist in the large-scale practical application of vehicle-infrastructure collaboration. 1) Heterogeneity of vehicle-road sensors. 2) Spatial and temporal matching of vehicle-road sensors. 3) Cost and time delay of data transmission from road-side to vehicle-side. 4) Computing power and real-time performance of road-side and vehicle-side [3].

To address these issues, numerous scholars have conducted research: V2vnet [4] created a V2V simulation dataset that uses vehicle-to-vehicle (V2V) communication to improve autonomous vehicles' perception and prediction performance. By intelligently aggregating information received from multiple nearby vehicles to view the same scene from different angles, it not only solves the single-vehicle point-of-view occlusion problem but also improves the ability to detect long-range objects. Cui et al. [5] introduced a new generation of enhanced road-side LiDAR, where road-side LiDAR sensors actively sense the high-resolution status of surrounding traffic participants and broadcast vehicle information via DSRC road-side units. Zhao et al. [6] proposed a system to detect and track pedestrians and vehicles at intersections through road-side LiDAR sensors. They analyzed real-time information about their position, speed, and direction. DiscoNet [23] applies distillation in the feature fusion training. V2X-ViT [24] introduces the vision transformer to fuse information across on-road agents. V2X-Sim [25] and OPV2V [26] are two simulated datasets for multi-vehicle cooperative perception research. Valiente et al. [27] integrate infrastructure data for end-to-end autonomous driving. Some works use infrastructure data to improve 3D perception ability. These studies were validated in experiments and initially addressed the problems in large-scale applications of vehicle-infrastructure collaboration. However, these works do not consider the mitigation of the vehicle-infrastructure communication and onboard computing loads in vehicle-infrastructure collaboration to further improve the performance and reliability of the overall system.

This paper proposed an early fusion with filter (EFWF) method, which divides the whole process of 3D object detection under vehicle-infrastructure cooperation into three stages: firstly, a 3D object detection performed at the road-side before transmitting the point cloud, and the output prediction frame use the road-side point cloud filter frame, then the point cloud is filtered according to the filter frame, and only retain a small number of point clouds around the filter box, and finally the retained point clouds are transferred to the vehicle-side for fusion, and then the 3D object detection at the vehicle-side is completed, and the final detection results are output. As there is a large amount of information in the point cloud at the road-side that is not relevant to the vehicle-side detection, the point cloud filtering can significantly reduce the transmission cost and bandwidth consumption from the road-side to the vehicle-side and also reduce the demand for computing resources at the vehicle-side.

## II. Early Fusion with Filter

This section first describes the overall process of the proposed EFWF algorithm and the specific tasks of the three phases and then will focus on the particular method of road-side point cloud filtering.

### A. Overall Flow of the EFWF Algorithm

As shown in Fig. 2, the EFWF algorithm divides the 3D object detection task into three stages: the first stage is road-side 3D object detection, the second stage is the filtering, coordinate conversion, and data transmission of the point cloud, and the third stage is the fusion of the point cloud and vehicle-side 3D object detection. The first and second stages are completed on the road-side, and the third is completed on the vehicle-side, which can effectively reduce the amount of computation and save computational resources consumption.

The main goal of the first stage is to perform 3D object detection on the point cloud data collected by LiDAR on the road-side. Fig. 3(a) and 3(b) show that the model takes the road-side point cloud data as input and outputs filter boxes, categories, and confidence levels after detection.

In the second stage, since a large amount of point cloud data in the figure is road surface, trees, walls, and other information that is not relevant to vehicle detection, the number of objects point clouds of interest to the vehicle-side only accounts for a tiny portion of the total number of point clouds, so the irrelevant information needs filtered with the help of the filter box in the first stage. As shown in Fig. 3(c), only the point clouds within or around the filter frame are retained, and then the retained point clouds are converted from the road-side LiDAR coordinate system to the vehicle-side LiDAR coordinate system. Since the point cloud data volume is large and the coordinate conversion requires many computing resources, the road-side has more computing resources and lower power requirements than the vehicle-side, so completing the coordinate conversion of the point cloud on the road-side can improve the conversion efficiency, reduce the computational load on the vehicle-side and further improve the real-time performance.

The world coordinate system is used as a transition to obtain the rotation matrix and translation vector from the road-side LiDAR coordinate system to the vehicle-side LiDAR coordinate system. The rotation matrix $R_{i2w}$ and translation vector $t_{i2w}$ are obtained from the calibration file, and the rotation matrix and translation vector from the vehicle-side LiDAR coordinate system to the world coordinate system are inverted to get the rotation matrix $R_{w2v}$ and translation vector $t_{w2v}$ from the world LiDAR coordinate system to the vehicle-side LiDAR coordinate system. The rotation matrix $R_{i2v}$ and

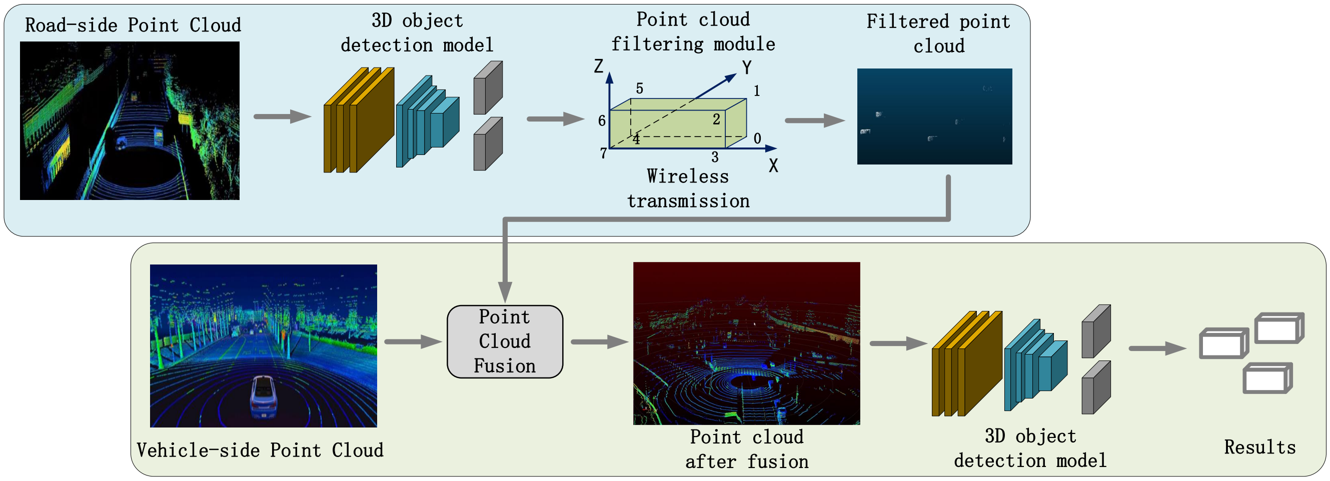


Fig. 2. EFWF algorithm flowchart.

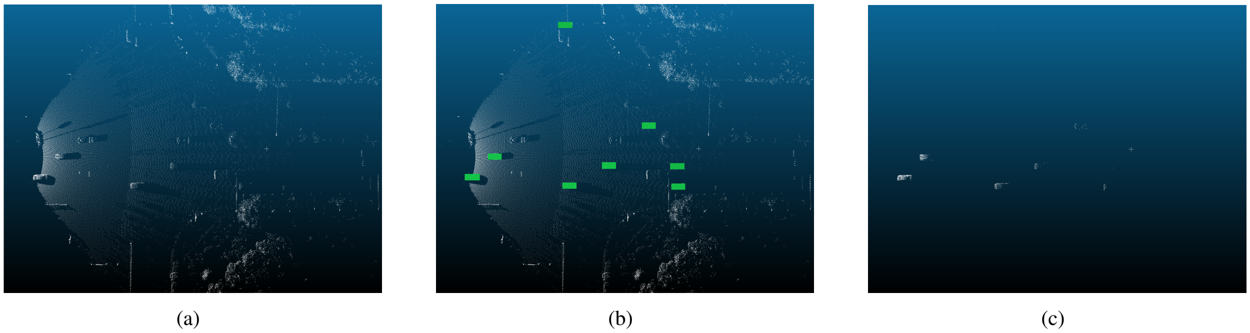


Fig. 3. Road-side point cloud, road-side detection result and filtered road-side point cloud from left to right.

translation vector $t_{i2v}$ from the road-side LiDAR coordinate system to the vehicle-side LiDAR coordinate system are obtained according to Eq.1.

$$\begin{bmatrix} R_{i2v} & t_{i2v} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{w2v} & t_{w2v} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{i2w} & t_{i2w} \\ 0 & 1 \end{bmatrix} \quad (1)$$

Finally, the point cloud is converted from the road-side LiDAR coordinate system to the vehicle-side LiDAR coordinate system according to the coordinate conversion Eq.2.

$$X_v = R_{i2v} X_i + t_{i2v} \quad (2)$$

$X_i$ and $X_v$ are the coordinates of the point cloud in the road-side LiDAR coordinate system and the vehicle-side LiDAR coordinate system, respectively.

In the third stage, the point cloud received from the road-side is fused with the collected point cloud on the vehicle-side, and then the fused data is fed into the vehicle-side model for 3D object detection to get the final detection result.

As shown in Fig. 4(a), there are two problems with the point cloud collected from the single vehicle side. 1) Weak perception capability and sparse object point cloud (shown in the red box). 2) Limited sensing range and missed objects (shown in the green box). Fig. 4(b) shows that the fusion of the point cloud of the road-side can effectively increase the density of the object point cloud while complementing the missed object at the vehicle-side, making up for the two shortcomings of single vehicle-side detection.

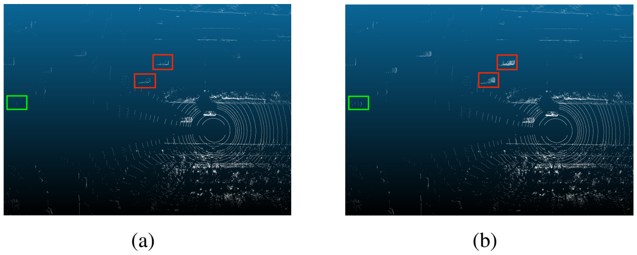


Fig. 4. Point cloud of the vehicle-side before fusion and after fusion from left to right.

### B. Road-side Point Cloud Filtering Method

The filter box output from the road-side model is represented as eight vertices, the shape of which is a non-regular rectangular shape, with the possibility of some vertices being significantly off-center of the object. Their center coordinates and width, length, and height are converted to a regular

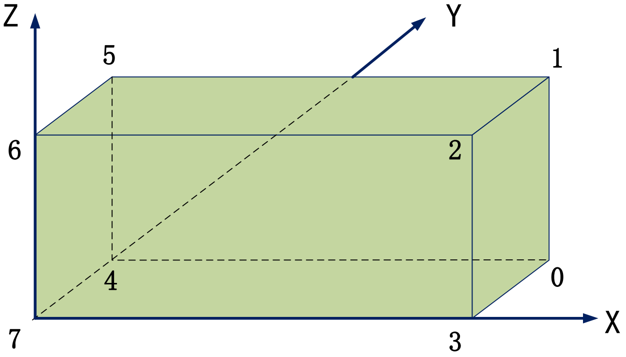


Fig. 5. Position of the eight vertices of the filter box.

rectangular shape and are calculated as follows:

$$\begin{bmatrix} center_x \\ center_y \\ center_z \end{bmatrix} = \begin{bmatrix} \frac{1}{8}\sum_{i=0}^{7} x_i \\ \frac{1}{8}\sum_{i=0}^{7} y_i \\ \frac{1}{8}\sum_{i=0}^{7} z_i \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} w \\ l \\ h \end{bmatrix} = \begin{bmatrix} \frac{x_3-x_7+x_1-x_5}{2} \\ \frac{y_4-y_7+y_1-y_2}{2} \\ \frac{z_6-z_7+z_1-z_0}{2} \end{bmatrix} \tag{4}$$

where $(center_x, center_y, center_z)$ denotes the coordinates of the center point under the x, y, z axes, $(x_i, y_i, z_i)$ denotes the coordinates of the $i_{th}$ vertex under the x, y, z axes, and w, l, h denotes the width, length, and height of the rectangle. The filter box and the location information of the eight vertices are shown in Fig. 5.

For a point cloud i, note its coordinates as $(point_x, point_y, point_z)$ for a filter box j, note its centroid coordinates and width, length, and height as $(center_x, center_y, center_z, w, h, l)$. The distances between point cloud i and the centroid of filter box j in x, y, and z dimensions are calculated respectively and then compared with the width, length, and height and decided to keep or not according to the following rules:

- If the distance between the point cloud and the center point in x, y, and z dimensions is less than or equal to one-half of the filtered width, length, and height, respectively, i.e., the point cloud is located inside the filter box, then the point cloud is kept.
- If the distance between the point cloud and the center point in some dimension is more significant than one-half of the corresponding edge length, i.e., the point cloud is located outside the filter box, then the point cloud is discarded. The calculation formula is as follows:

$$\begin{cases} |point_x^{(i)} - center_x^{(j)}| < \dfrac{w^{(j)}}{2} \\ |point_y^{(i)} - center_y^{(j)}| < \dfrac{l^{(j)}}{2} \\ |point_z^{(i)} - center_z^{(j)}| < \dfrac{h^{(j)}}{2} \end{cases} \tag{5}$$

To further investigate the effect of point cloud transmission on the detection results, the variable K was introduced to adjust the road-side point cloud filtering range. Eq. 5 was rewritten in the following form:

$$\begin{cases} |point_x^{(i)} - center_x^{(j)}| < \dfrac{w^{(j)}}{2} * K \\ |point_y^{(i)} - center_y^{(j)}| < \dfrac{l^{(j)}}{2} * K \\ |point_z^{(i)} - center_z^{(j)}| < \dfrac{h^{(j)}}{2} * K \end{cases} \tag{6}$$

Different filter ranges can be obtained around the filter box of the road-side detection output by adjusting K. Since the road-side model cannot fit the proper position of all objects exactly, there is a different degree of offset between the filter box and the appropriate position of particular objects. By increasing the filtering range, the number of retained objects will subsequently increase, which will help to improve the vehicle-side detection accuracy. However, at the same time, the transmission cost will also increase. As shown in Fig. 9, as K increases, the filtering range gradually increases, the transmission cost slowly rises, and the detection accuracy increases.

## III. Experiments and Analysis of Results

This section first describes the experimental configuration, including the hardware and software configurations. The DAIR-V2X dataset [7] and PointPillars model [8] adopted during the experiments are then described, and details of the model training and the testing process are shown. Finally, the experimental results are analyzed, including the trend of the vehicle-side detection accuracy with increasing transmission cost, the selection of the optimal filtering range, and a comparison with the benchmark algorithm in the dataset.

### A. Experimental Configuration

The main hardware configuration used for this experiment is GeForce RTX3090 (24 GB) and 15 vCPU Intel(R) Xeon(R) Platinum 8338C CPU @ 2.60GHz. The main software configuration is Ubuntu 16.04, CUDA 12.0, Python 3.7.1, and mmdetetion3d 0.17.1.

### B. Experimental Procedure

*1) DAIR-V2X dataset:* The DAIR-V2X dataset released by the Institute of Artificial Intelligence Industry (AIR) of Tsinghua University is the world's first large-scale, multi-modal, multi-view dataset for vehicle-infrastructure collaboration research. All the data it contains are from natural scenes in Beijing's high-grade autonomous driving demonstration area and include 2D and 3D annotations. The dataset consists of 71,254 frames of image data and 71,254 frames of point cloud data and is divided into three parts: DAIR-V2X vehicle- infrastructure collaboration dataset (DAIR-V2X-C), DAIR-V2X infrastructure dataset (DAIRV2X-I) and DAIR-V2X vehicle dataset (DAIR-V2X-V). In DAIR-V2X-C, a frame of image and point cloud data from the vehicle-side and road-side is

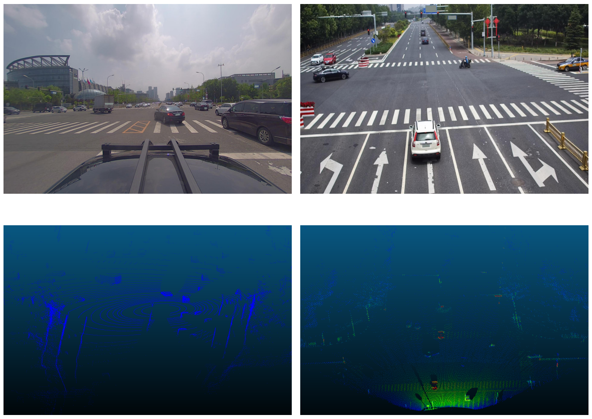


Fig. 6. Image and point cloud data of DAIR-V2X dataset.

shown in Fig. 6. DAIR-V2X-C contains 18,330 frames of road-side multi-modal data and 20,515 frames of vehicle-side multi-modal data with the corresponding 2D and 3D annotations. The joint annotation is also obtained by merging the vehicle-side and road-side annotations, as:

$$GT = GT_i \cup GT_v \tag{7}$$

where GT is the joint vehicle-road annotation, $GT_i$ is the annotation of road-side data, and $GT_v$ is the annotation of vehicle-side data. The specific annotation method is as follows [7]:

- Form a data pair with similar data $(|t_i - t_v| < 10ms)$ between the road-side acquisition time $t_i$ and the vehicle-side acquisition time $t_v$.
- The road-side labeling be converted from the road-side LiDAR coordinate system to the vehicle-side LiDAR coordinate system according to Eq.1 and Eq.2.
- Road-Side annotations matching and fused with vehicle-side annotations. Road-side annotations that cannot be matched on the vehicle-side are added directly to the vehicle-side annotations.
- Manually observe and adjust annotation information for more accurate joint annotation.

The next part of this paper will use this data to conduct experiments.

*2) 3D object detection models:* 3D object detection aims to predict an object's location, category, and other information in a realistic scene by feeding in the data collected by different sensors. Cameras and LiDAR are the two most commonly used sensors, corresponding to image and point cloud data. Image data is dense and regular, and convolutional neural networks can effectively exploit its local spatial correlation. However, cameras only capture appearance information and cannot directly acquire 3D structural information, and detection from images is susceptible to extreme weather and time conditions. LiDAR can obtain 3D structural information about objects by measuring the reflection of laser beams. Point cloud data is sparsely distributed, irregularly structured, and geometrically disordered, so the direct application of convolution to point clouds can lead to severe distortions [9]. To address this problem, the literature [8], [10], [11] voxelated their inputs to accommodate convolution operators, while the literature [12], [13] proposes diverse architectures to detect 3D objects directly from the original points [14]. Depending on the representation of the data, existing methods can be classified into three categories [15]: 1) voxel-based methods, 2) point-based methods, and 3) Voxel-point-based methods. The PointPillars model is a voxel-based approach proposed by Lang et al. [8], which weighs the speed and accuracy of inference, making it widely used in practical scenarios. As shown in Fig. 7, the PointPillars model consists of three main components: a feature encoding network that converts the point cloud into a pseudo-image, a backbone network that extracts features from the pseudo-image, and a detection head for detection and regression.

First, the feature encoding network converts the data from point cloud format to pseudo-image format to facilitate subsequent feature extraction using the 2D convolution of the backbone network. It proposes the pillars that convert point clouds into pillar clusters and transforms them into pseudo-images after feature extraction by a simplified PointNet [16]. Based on this, PointPillars uses a backbone network similar to Voxelnet [10] for feature extraction of pseudo-images. The backbone network consists of network blocks (S, L, F) with different step sizes, each with a step size of S and L 3 * 3 2D convolutional layers and F output channels, each followed by a BatchNorm [17] layer and a ReLU [18] layer. Network blocks of different step sizes can go to different-size feature maps, which are up-sampled and concatenated, and then the combined feature map tensor formed is output. Finally, the detection head used for detection and regression is the single shot detector (SSD) proposed by Liu et al. [19]. SSD is faster as the whole process takes only one step by evenly extracting anchor frames of different sizes over the image, then using CNN to extract features for classification and regression.

The PointPillars model has several advantages: firstly, it does not rely on fixed encodings and makes eicient use of the complete information of the point cloud. In addition, it proposes pillars instead of voxel operations, eliminating the need to adjust the vertical grading manually. Finally, all the critical functions of pillars can be represented using convolution, which is very eicient for computing on the GPU.

Therefore, this paper uses it as an experimental model for the road-side point cloud filtering and vehicle-side fusion detection phases.

*3) Model training and testing:* First, to train the road-side model for point cloud filtering, the road-side data in DAIRV2X-C was converted to KITTI [20] format, and the dataset was divided into a training set and a test set according to the benchmark. The dataset was split into a training set and a test set according to 5:2 (DAIR-V2X currently only releases a training set and a validation set; its benchmark is based on the validation set, so the validation set is used here as the test set). The point clouds and annotation files from the road-side training set are then fed into the model, as in Fig. 8(a), and after 22,300 rounds of iterations, the loss values tend to

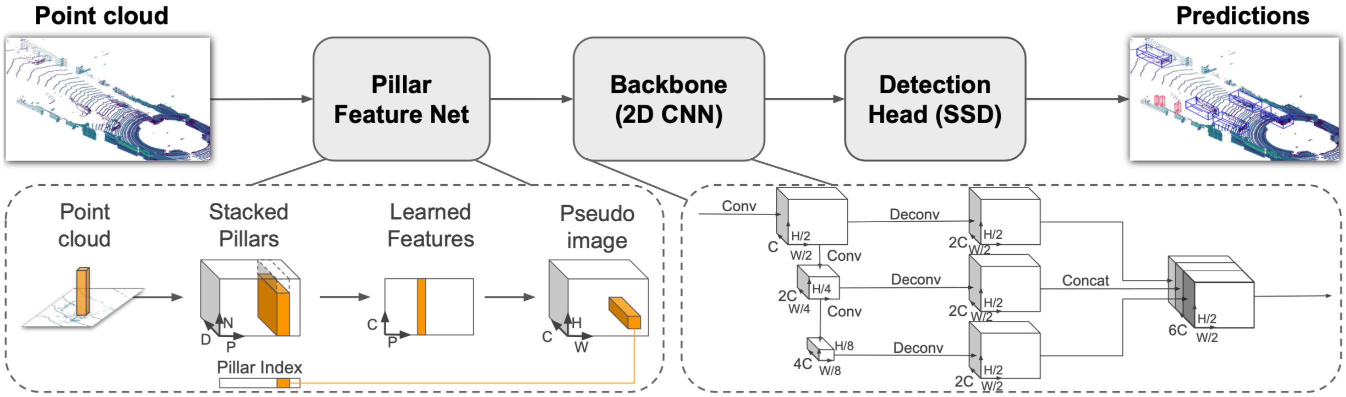
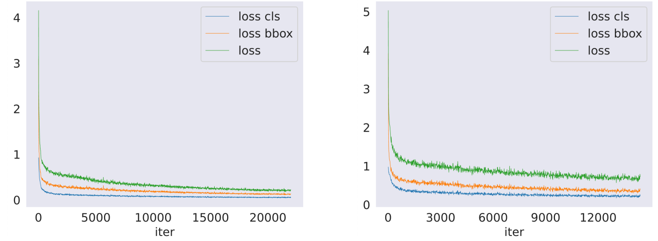


Fig. 7. PointPillars model architecture.



(a)

(b)

Fig. 8. Loss curve of road-side and vehicle-side from left to right.

stabilize, and the road-side model is obtained. Next, to get the vehicle-side object detection model after point cloud fusion, the road-side point cloud data of DAIRV2X-C is converted to vehicle-side coordinates and then fused with the vehicle-side point cloud according to Eq.1 and Eq.2, i.e., the vehicle-side point cloud is merged with the converted road-side point cloud data under the vehicle-side coordinate system, and the vehicle-side annotation is replaced with the collaborative annotation to obtain the fused vehicle-side data. Then the exact format conversion and data set division is done with the road-side, and the fused point cloud and annotations are input into the vehicle-side model. After 14,400 iterations, the loss values tend to stabilize, and the vehicle-side model is obtained, as in Fig. 8(b).

Finally, the test set is tested using these two models for inference, and the object detection accuracy is calculated based on the test set. The inference process is referred to in Section 2.1. Different filtering ranges were taken at the road-side to test the transmission cost, and the object detection accuracy at the vehicle-side, respectively, and a table of the relationship between transmission cost and detection accuracy was obtained, as shown in Tab. I. The trend of detection accuracy with increasing transmission point cloud is shown in Fig. 9.

### C. Analysis of Results

*1) Evaluation indicators:* To verify the accuracy of the point cloud filtering algorithm in this paper, the Average Precision (AP) [20] from Birds eye view (BEV) is used as the evaluation metric of the accuracy of the algorithm. Meanwhile, to measure the transmission cost, the Average Byte (AB) [7] is used as the evaluation metric of transmission cost.

(1) AP [21] is an essential metric for assessing the accuracy of 2D target detection, extending it to 3D space:

$$AP = 100 \int_{0}^{1} max\{P(r'|r' \geq r)\}dr \tag{8}$$

where p(r) is the same precision-recall curve as Lin et al. [21], the main difference between AP in 3D and 2D space is the matching criterion between the ground truth and the predictions when calculating accuracy and recall. In this paper, we will use the $AP_{3D}$ and $AP_{BEV}$ criteria proposed by KITTI [20], which are based on the intersection over union (IoU) of two cuboids from the birds-eye view (BEV IoU) and 3D view (3D IoU). The IoU is set to 0.5 in this paper.

(2) AB is a transmission cost evaluation metric proposed by Yu et al. [7], where a Byte consists of eight bits. To simplify the problem, the data encoding and decoding time consumption during transmission is ignored, so the smaller the data transmission cost, the smaller the time delay. In this paper, only point cloud data is transmitted. The point cloud format used is (x, y, z, r), where x, y, z are the location coordinates of the point cloud in space, and r is the intensity value of this point, where each value is stored as a float32 data type, and each point cloud is measured in Byte. The size of each point cloud is 16 bytes. When evaluating, the number of point clouds for each transmission is summed up, multiplied by the size of each point cloud, and divided by the number of transmissions to obtain AB, calculated as follows:

$$AB = \frac{16}{n} \sum_{i=0}^{n} N_i \tag{9}$$

where n denotes the total number of transmissions, and $N_i$ indicates the number of point clouds transmitted for the $i_{th}$ time.

*2) Comparison of accuracy at different transmission costs:* According to Eq.6, different filtering ranges can be obtained

TABLE I
AP and AB at Different K Values.

| K | $AP_{3D}$ | $AP_{BEV}$ | AB |
|---|---|---|---|
| 0.5 | 56.23 | 62.13 | 2693.78 |
| 1 | 62.06 | 68.18 | 30262.64 |
| **3** | **63.72** | **69.57** | **109660.76** |
| 4 | 63.94 | 69.76 | 164009.62 |
| 8 | 64.07 | 70.07 | 431448.78 |
| 12 | 64.43 | 70.47 | 674657.36 |

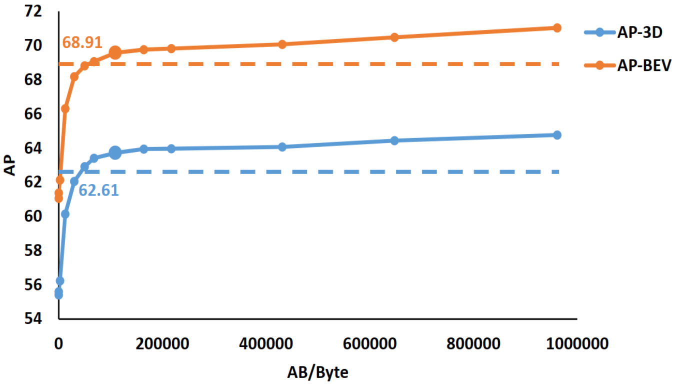


Fig. 9. Trend of AP with the size of the transmitted point cloud.

with different K values. When K is taken as 0, no road-side point cloud data is transmitted, only the point cloud collected by the vehicle-side is used for detection, and AB is 0. When K is taken as infinite, all point cloud data from the road-side is transmitted to the vehicle-side for fusion detection, and AB reaches its maximum value at this time. As shown in Tab. I, a series of different K values (only some of the K values are shown) were used to obtain different relationships between AB and AP, and this was used to make a graph of the change in AP with the increase in transmitted point clouds, as shown Fig. 9.

Fig. 9 shows that the AP increases sharply with AB < 100000 and increases gently with AB > 100000. To make AB and AP relatively balanced, according to the principle of the elbow method, K=3 is selected as the filtering range (the coarse point in Fig. 9), at which AB is 109660.76, $AP_{3D}$ and $AP_{BEV}$ is 63.72 and 69.57, respectively. Both are higher than the early fusion benchmark of 62.61 and 68.91.

This result shows that only a tiny portion of the point cloud data on the road-side is valid, and most of the point clouds are irrelevant for vehicle-side detection, so transmitting only a tiny portion of useful point cloud data can significantly reduce transmission costs and bandwidth usage, and also reduce the demand for vehicle-side computing resources.

*3) Comparison of the algorithm in this paper with the benchmark:* To further verify the effectiveness of the EFWF algorithm, the AB and AP values obtained in section 3.3 under the optimal filtering range were compared with the dataset benchmark to get Tab. II.

TABLE II
Comparison of Different Fusion Algorithms.

| Method | Strategy | $AP_{3D}$ | $AP_{BEV}$ | AB |
|---|---|---|---|---|
| Benchmark | Veh.-Only | 48.06 | 52.24 | 0 |
|  | Late Fusion | 56.06 | 62.06 | 478.61 |
|  | Early Fusion | 62.61 | 68.91 | 1382275.75 |
| Ours | EFWF | 63.72 | 69.57 | 109660.76 |

Tab. II shows that although the late fusion method has a low transmission cost with an AB of 478.61, the $AP_{3D}$ and $AP_{BEV}$ are only 56.06 and 62.06. The early fusion method can increase $AP_{3D}$ and $AP_{BEV}$ to 62.61 and 68.91, but the transmission cost is very high, with an AB of 1382275.75, which consumes much bandwidth and increases the computation on the vehicle-side. The AB of the EFWF algorithm is only 109660.76, which is 92.07% lower than early fusion, while $AP_{3D}$ and $AP_{3D}$ can reach 63.72 and 69.57. The average improvement is 1.37% compared to early fusion and 12.88% compared to late fusion.

EFEW filters out a large amount of irrelevant information from the point cloud at the road-side, significantly reducing transmission costs and bandwidth usage while maintaining high accuracy, reducing the amount of computation on the vehicle-side and making it better able to meet real-time requirements, taking into account the advantages of both algorithms.

## IV. Conclusion

In the vehicle-infrastructure collaboration problem, high detection accuracy should be pursued. At the same time, transmission costs should be kept as low as possible, and the amount of vehicle-side computation should be reduced. Although the late fusion method has low transmission cost and low calculation on the vehicle-side, its detection accuracy could be higher. The early fusion method can significantly improve detection accuracy. However, because it needs to transmit many point clouds, its transmission cost is extraordinarily high, and it is diicult to meet the real-time requirements.

This paper proposes an early fusion with the point cloud filtering method to solve the problem of balancing detection accuracy and transmission cost in vehicle-infrastructure cooperation, in which a 3D object detection is performed to filter the point cloud before transmission. Then the filtered point cloud is transmitted to the vehicle-side for fusion detection. In addition, this method also experiments on the DAIR-V2X dataset. The trend of AP changes under different AB is investigated by changing the filtering range of the road-side point cloud. The optimal filtering range is selected according to this trend. The AB and AP obtained under this optimal filtering range are also compared with the benchmark of the dataset. The experiments show that the method in this paper significantly reduces the transmission cost and shifts part of the vehicle-side computation to the road-side, effectively reducing the amount of vehicle-side computation. Although this increases the amount of calculation on the road-side, it is more efficient than transferring the entire point cloud to the vehicle-side due to more computational resources on the road-side and fewer limitations in terms of power consumption. At the same time, this method maintains the advantage of high accuracy, considering the high accuracy of early fusion and the low cost of late fusion transmission.

However, the method in this paper also has some problems that need to be solved, such as time has been asynchronous. The time asynchrony problem is a diicult problem in vehicle-infrastructure collaboration, and the existing solutions include methods such as time compensation [7] and feature prediction [22]. We hope our work can further promote the research of vehicle-infrastructure collaboration.

## REFERENCES

[1] Ding,Fei,NanZhang,ShengboLi,YougangBian,EnTong,andKeqiang Li. "A survey of architecture and key technologies of intelligent connected vehicle-road-cloud cooperation system." In Acta Automatica Sinica, 48, no. 12 (2022): 2863-2885.

[2] Cai, Zhili, Fengrui Sun, Lingxiang Wei, and Nan Wang. "Design of Vehicle-Road Collaboration System Based on Telematics Technology." In Journal of Shandong Jiaotong University, 19, no. 4 (2011): 17-23.

[3] Yu, Hang, Yongsheng Zhao, Ying Zou, Qian Li, Haiyang Yu, and Yilong Ren. "Multistage Fusion Approach of Lidar and Camera for Vehicle-Infrastructure Cooperative Object Detection." In 2022 5th World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), pp. 811-816. IEEE, 2022.

[4] Wang, Tsun-Hsuan, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." In Computer VisionECCV 2020: 16th European Conference, Glasgow, UK, August 2328, 2020, Proceedings, Part II 16, pp. 605-621. Springer International Publishing, 2020.

[5] Cui, Yuepeng, Hao Xu, Jianqing Wu, Yuan Sun, and Junxuan Zhao. "Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system." IEEE Intelligent Systems 34, no. 3 (2019): 44-51.

[6] Zhao, Junxuan, Hao Xu, Hongchao Liu, Jianqing Wu, Yichen Zheng, and Dayong Wu. "Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors." Transportation research part C: emerging technologies 100 (2019): 68-87.

[7] Yu, Haibao, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo 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, pp. 21361-21370. 2022.

[8] Lang, Alex H., 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, pp. 12697-12705. 2019.

[9] Li, Yangyan, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. "Pointcnn: Convolution on x-transformed points." Advances in neural information processing systems 31 (2018).

[10] Zhou, Yin, and Oncel Tuzel. "Voxelnet: End-to-end learning for point cloud based 3d object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4490-4499. 2018.

[11] Yan, Yan, Yuxing Mao, and Bo Li. "Second: Sparsely embedded convolutional detection." Sensors 18, no. 10 (2018): 3337.

[12] Yang,Zetong,YananSun,ShuLiu,andJiayaJia."3dssd:Point-based3d single stage object detector." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11040-11048. 2020.

[13] Shi, Shaoshuai, Xiaogang Wang, and Hongsheng Li. "Pointrcnn: 3d object proposal generation and detection from point cloud." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 770-779. 2019.

[14] Mao, Jiageng, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "3D object detection for autonomous driving: a review and new outlooks." arXiv preprint arXiv:2206.09474 (2022).

[15] Qian, Rui, Xin Lai, and Xirong Li. "3d object detection for autonomous driving: a survey." Pattern Recognition 130 (2022): 108796.

[16] Qi,CharlesR.,HaoSu,KaichunMo,andLeonidasJ.Guibas."Pointnet: Deep learning on point sets for 3d classification and segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652-660. 2017.

[17] Ioffe,Sergey,andChristianSzegedy."Batchnormalization:Accelerating deep network training by reducing internal covariate shift." In International conference on machine learning, pp. 448-456. pmlr, 2015.

[18] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." In Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807-814. 2010.

[19] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In Computer VisionECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 1114, 2016, Proceedings, Part I 14, pp. 21-37. Springer International Publishing, 2016.

[20] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." In 2012 IEEE conference on computer vision and pattern recognition, pp. 3354-3361. IEEE, 2012.

[21] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In Computer VisionECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740-755. Springer International Publishing, 2014.

[22] Yu, Haibao, Yingjuan Tang, Enze Xie, Jilei Mao, Jirui Yuan, Ping Luo, and Zaiqing Nie. "Vehicle-Infrastructure Cooperative 3D Object Detection via Feature Flow Prediction." arXiv preprint arXiv:2303.10552 (2023).

[23] Li, Yiming, 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 (2021): 29541-29552.

[24] Xu, Runsheng, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. "V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer." In Computer VisionECCV 2022: 17th European Conference, Tel Aviv, Israel, October 2327, 2022, Proceedings, Part XXXIX, pp. 107-124. Cham: Springer Nature Switzerland, 2022.

[25] Li, Yiming, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. "V2X-Sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving." IEEE Robotics and Automation Letters 7, no. 4 (2022): 10914-10921.

[26] Xu, Runsheng, 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), pp. 2583-2589. IEEE, 2022.

[27] Valiente, Rodolfo, Mahdi Zaman, Sedat Ozer, and Yaser P. Fallah. "Controlling steering angle for cooperative self-driving vehicles utilizing cnn and lstm-based deep networks." In 2019 IEEE intelligent vehicles symposium (IV), pp. 2423-2428. IEEE, 2019.