Collaborative 3D Object Detection for Autonomous Vehicles via Learnable Communications

J. Wang, Y. Zeng, *Member, IEEE*, and Y. Gong, *Senior Member, IEEE*

***Abstract*— 3D object detection from LiDAR point cloud is a challenging task in autonomous driving systems. Collaborative perception can incorporate information from spatially diverse sensors and provide significant benefits for accurate 3D object detection from point clouds. In this work, we consider that the autonomous vehicle uses local point cloud data and combines information from neighboring infrastructures through wireless links for cooperative 3D object detection. However, information sharing among vehicles and infrastructures in predefined communication schemes may result in communication congestion and/or bring limited performance improvement. To this end, we propose a novel collaborative 3D object detection framework using an encoder-decoder network architecture and an attention-based learnable communications scheme. It consists of three components: a feature encoder network that maps point clouds into feature maps; an attention-based communication module that propagates compact and fine-grained query feature maps from the vehicle to support infrastructures, and optimizes attention weights between query and key to refine support feature maps; a region proposal network that fuses local feature maps and weighted support feature maps for 3D object detection. We evaluate the performance of the proposed framework on CARLA-3D, a new dataset that we synthesized using CARLA for 3D cooperative object detection. Experimental results and bandwidth consumption analysis show that the proposed collaborative 3D object detection framework achieves a better detection performance and communication bandwidth trade-off than five baseline 3D object detection models under different detection difficulties.**

***Index Terms*— Collaborative perception, learnable communications, 3D object detection, autonomous driving.**

I. Introduction

A CORE component of autonomous driving vehicles or self-driving vehicles is their perceived ability to sense the surrounding environment and make decisions accordingly. The reliability of perception algorithms has improved significantly in the past few years due to the development of deep neural networks (DNNs) that can reason in 3D and intelligently fuse multi-sensor data, such as RGB-D images, LiDAR point clouds, and GPS locations [1], [2], [3]. However, precise and comprehensive perception is still a challenging task, especially when objects are heavily occluded or far away, resulting in sparse observations. This is because the sensors equipped on a vehicle with partial observability and local viewpoints have limited sensing ability in complex driving environments. Recently, collaborative perception, studying how to fuse information from multiple neighboring sensors, has received attention from industry and academia. Collaborative perception enables a driving system to have a longer perception range and reduce blind spots caused by occlusions from one perspective to improve overall accuracy towards perception tasks, such as instance segmentation or object detection. Compared with local perception, cooperative perception has the advantage of augmenting the observation from different perspectives, as well as expanding the perception range beyond the line of sight or field of view up to the boundary of autonomous vehicles [4].

One major challenge for cooperative perception on multiple connected agents is how to design an effective communication scheme to exchange information among agents, since a high bandwidth communication scheme results in network congestion and latency in the autonomous vehicle network. In this work, we formulate a cooperative 3D object detection problem where a vehicle can communicate with other agents to improve its perceptual abilities in its own field of view. Unlike existing works where communication protocol is predefined and/or unified, we consider a vehicle can learn to communicate with other support agents intelligently.

To learn the communication protocol by agents themselves, most works adopt the reinforcement learning approach. They assume all agents are connected and information can be shared across all agents in a centralized manner. The key idea is learning a shared DNN to encode observations into features, then fusing features from all agents based on attention mechanism [5], [6], and finally decoding the fused features for perception or decision-making. Recently, intermediate feature-based fusion has also been developed for autonomous vehicles [7], [8], [9], [10]. Deep neural features are extracted and shared across multiple neighboring agents for a complete and accurate understanding of the driving environment. However, transmitting feature maps across a fully-connected graph would bring high communication costs and delays, especially when the cross-agent bandwidth is limited. We thus face the problem of designing an effective communication

Manuscript received 13 July 2022; revised 15 January 2023 and 5 April 2023; accepted 13 April 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071212 and Grant 62106095, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019B1515130003, and in part by the Guangdong Provincial Department of Education under Grant 2020ZDZX3057. The Associate Editor for this article was G. Mao. *(Corresponding authors: Y. Zeng; Y. Gong.)*

J. Wang and Y. Gong are with the Department of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: gongy@sustech.edu.cn).

Y. Zeng is with the Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zengy3@sustech.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3272027

scheme under bandwidth constraints for cooperative 3D object detection.

To tackle the issue, we propose a multi-agent collaborative 3D object detection framework, where vehicles learn to communicate with support infrastructures in an end-to-end learning manner. To consider the trade-off between detection accuracy and communication bandwidth, we design an attention-based communication mechanism and employ centralized training and decentralized inference. During training, the vehicle broadcasts its query information from all neighboring infrastructures. Then, each neighboring infrastructure computes a learned matching score between its local information and the information received from the vehicle, and sends back the matching score and the encoded feature maps. The vehicle learns attention weights regarding feature importance for cooperative 3D object detection. During inference, the vehicle determines which infrastructure it should communicate with based on the learned attention weights. The communication group dynamically changes and the framework is trained in an end-to-end manner using LiDAR point cloud data.

Furthermore, to evaluate the proposed approach, we build a new cooperative 3D object detection dataset CARLA-3D using the Car Learning to Act (CARLA) simulator [11]. In CARLA-3D, multiple autonomous vehicles are driving in urban environments with diverse 3D models of static objects such as roads, buildings, vegetation, traffic signs, and infrastructure, as well as dynamic objects such as vehicles and pedestrians. All models share a common scale and their sizes correspond to those of objects in the real world. The dataset includes the objects in the vehicle's field of view as the target objects. We compared the proposed framework with a local perception model, three centralized cooperative perception models and a decentralized model on CARLA-3D. Our experimental results show that our cooperative perception approach using the attention-based communication scheme provides a better performance-bandwidth trade-off for 3D object detection. Our main contributions are summarized as follows.

- We consider a challenging task of how to balance detection accuracy and communication bandwidth in multi-agent cooperative 3D object detection and present a novel framework that learns to construct communication groups under bandwidth constraints for cooperative 3D object detection in autonomous vehicle systems.
- We design an attention-based communication mechanism that trained in an end-to-end manner to learn attention weights of all neighboring infrastructures without manually labeling the ground-truth infrastructure for communication. It adaptively fuses feature maps from one of neighboring infrastructures during inference to balance detection performance and bandwidth usage.
- We build a new dataset CARLA-3D that provides LiDAR point clouds of multi-agents for better evaluating cooperative 3D object detection in autonomous vehicle systems.
- We provide comprehensive experiments on CARLA-3D to evaluate the effectiveness of the proposed framework on cooperative 3D object detection and show how the proposed framework outperforms local perception and centralized perception methods in terms of detection accuracy and communication consumption.

II. Related Work

A. 3D Object Detection From Point Clouds

A variety of approaches have been proposed for 3D object detection from point clouds. 3D object detection methods based on point clouds can be categorized into two classes: region proposal-based and single-shot methods. Region proposal-based methods first propose several region proposals containing objects and then determine the category label of each proposal using extracted region-wise features. In [12], a PointRCNN framework was proposed to generate object proposals directly from the raw point cloud and then refine the object proposals for predicting 3D bounding boxes. Shi et al. [13] presented a 3D object detection framework, named PointVoxel-RCNN, which deeply integrates both 3D voxel convolutional neural network (CNN) and PointNet-based set abstraction for accurate 3D object detection from point clouds. Hu et al. [14] proposed a density-aware RoI grid pooling module to aggregate spatially localized voxel features for 3D proposals and probability score refinement.

Single-shot methods directly estimate class probabilities and regress 3D bounding boxes of objects using a single-stage network. Beltrán et al. [15] proposed a biednet architecture to generate 3D proposals through a convolutional neural network. Zhou and Tuzel [16] presented VoxelNet architecture to learn discriminative features from point cloud and detect 3D objects. Later, In [17], a 3D object detection approach named PointPillars was proposed. PointPillars first utilizes PointNet [18] to learn the feature representation of point clouds organized in vertical columns (Pillars), and then encodes the learned feature representation into a pseudo image. After that, a 2D object detection pipeline is performed to predict 3D bounding boxes with the object class. Liu et al. [19] introduced a robust 3D object detection framework, named TANet, consisting of a triple attention module and a coarse-to-fine regression module for accurate detection of the hard objects and robust detection from noisy point cloud data. In contrast to doing 3D object detection from a single point of view, this paper tackles cooperative object detection problems in autonomous driving scenarios where point clouds are gathered from multiple views.

B. Collaborative Perception

Collaborative perception approaches make perception based on information from multiple sensors/agents. Existing cooperative perception approaches can be categorized into three classes according to their information sharing strategies, which are raw-data-based early fusion, feature-based intermediate fusion, and output-based late fusion. Early fusion [20] fuses raw point clouds collected from different positions and angles for enhancing the detection ability of autonomous driving systems. Although early fusion can improve the perception performance by raw data sharing, it requires a lot of communication bandwidth. Late fusion [21], [22] is a bandwidth-efficient method, since it fuses detection results of all agents as

the final outputs. However, it may fail to provide better detection results, since each individual perception error can directly affect the final outputs and cause unsatisfying detection results. Intermediate feature-based fusion can achieve good balance between perception accuracy and communication bandwidth, and has attracted increasing attention in recent years [8], [9], [23], [24]. F-Cooper [7] introduced two feature-level fusion schemes for collaborative perception. Who2com [24] proposed a handshake communication mechanism to determine communication group for cooperative perception. V2vnet [23] presented a message passing mechanism to jointly perceive and predict. Disconet [8] proposed a distilled collaboration graph to model the collaboration among agents. OPV2V [10] proposed a single-head self-attention module to fuse intermediate features and improve perception performances.V2X-ViT [9] proposed a novel holistic attention module to fuse information across heterogeneous agents. Where2comm [25] proposed a novel spatial-confidence-aware communication strategy for communication-efficient collaborative perception. This work tackles the problem of learning to communicate with bandwidth constraints in collaborative perception for 3D object detection.

C. Learning of Communications

Communications is a fundamental aspect of collaborative perception, enabling connected sensors to work as a group for effective perception and decision-making in wireless sensor networks or multi-agent systems. Early works employed predefined communication protocols [26], [27] or a unified communication network [28], [29] instead of automatically learned communication. Recently, a few approaches have involved learning the communication of nodes/agents in multi-agent reinforcement learning. Tan [27] used a neural model, named CommNet, to model the communication scheme between agents. CommNet considers full cooperation across agents, and each agent broadcasts its state to a shared communication channel, and then the other agents use the integrated information for perception. Peng et al. [30] proposed a bi-directional recurrent neural network (BiCNet) based communication model to integrate node-specific information from all connected nodes. Later, Jiang et al. [31] presented an attentional communication model to learn when and how to communicate for cooperative decision-making in large-scale multi-agent systems. Liu et al. [24] designed a multi-stage handshake communication mechanism that learns whom to communicate for reducing bandwidth usage in multi-agent cooperative perception tasks. Those tasks are built on simplistic environments where each agent observes low-dimensional 2D images. We design an attention-based communication block and integrate it into an encoder-decoder-based 3D object detection architecture for a better performance-bandwidth trade-off.

III. Problem Statement

Let us consider an urban driving environment consisting of N infrastructures and an autonomous vehicle. Each infrastructure $i \in \{1, \cdots, N\}$ includes a LiDAR sensor to perceive its surrounding environment and captures LiDAR point cloud data S_i. Fig. 1 illustrates an urban driving scenario for our cooperative perception. To benefit from cooperative perception, we assume that an autonomous vehicle v is equipped with a wireless reception system and a local processor, and the autonomous vehicle and neighboring infrastructures can exchange information through wireless links, using the dedicated short-range communication (DSRC) [32] or the fifth-generation (5G) technology. We consider each agent (vehicle or infrastructure) has prior information about its pose, including position and orientation. For local 3D object detection, the autonomous vehicle uses its local point cloud data S_v and a local processor for perception. For collaborative 3D object detection, the vehicle integrates local point clouds and information received from neighboring infrastructures for better perception. However, there is a trade-off between detection accuracy and communication bandwidth usage in cooperative perception. Bandwidth and latency limitations should be considered in the communication scheme design to provide a better trade-off between detection accuracy and bandwidth usage, and to scale in a bounded way to the number of neighboring sensors. In this work, we assume the wireless communication between the vehicle and infrastructures is ideal, and all sensors are precisely synchronized in time. This work aims to design a novel cooperative perception framework to improve the 3D object detection accuracy of the vehicle with limited transmission bandwidth. The network delay, clock synchronization and communication losses will be considered in future studies.

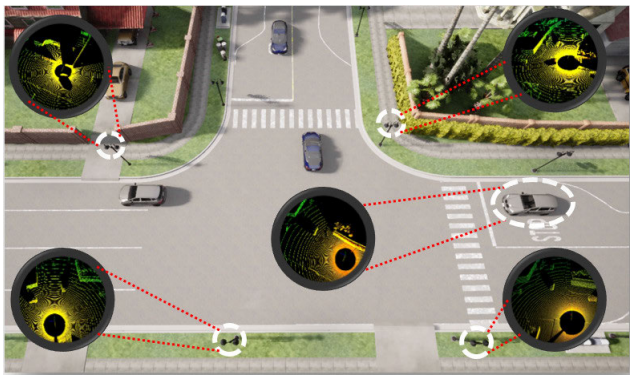


Fig. 1. Illustration of an urban driving scenario for cooperative 3D detection. Our cooperative system aims at improving the 3D object detection ability of a vehicle using information from neighboring infrastructures.

IV. Cooperative 3D Object Detection Under Bandwidth Constraints

In this section, we introduce a framework to tackle our cooperative 3D object detection in detail. The overview of our framework is shown in Fig. 2. Inspired by recent single-shot object detection networks [16], [17], [33], our cooperation detection framework follows an encoder-decoder architecture and consists of three main parts: 1) Feature encoder network; 2) Attention-based communication block; 3) Region proposal network. We use PointPillar [17] as the backbone of the feature encoder network and region proposal network of the proposed

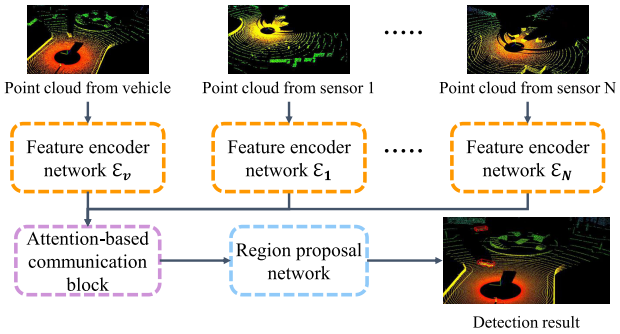


Fig. 2. Overview of our framework for cooperative 3D object detection in autonomous driving systems. Given point clouds from the autonomous vehicle and neighboring infrastructures, the framework first maps point clouds into feature maps locally using feature encoder networks. Later, an attention-based communication block is performed to learn communication protocol using feature maps extracted from the vehicle and infrastructures. After that, a region proposal network is adopted to fuse feature maps and detect objects. The framework is trained in an end-to-end manner.

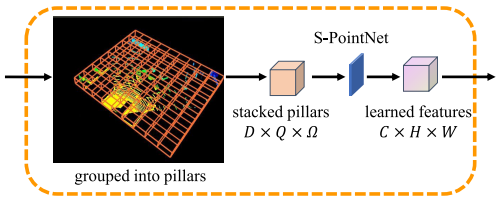


Fig. 3. Architecture of the feature encoder network. The input point clouds are first grouped into pillars, and then a convolution neural network is used to learn feature maps.

framework because of its low optimized memory usage. The proposed attention-based communication block can also be extended to other encoder-decoder-based single and double-stage 3D object detectors. In addition, this work adopts the framework of centralized training and decentralized inference. During training, the inputs of the proposed framework include point clouds collected from the vehicle and infrastructures, and the outputs are object labels and 3D boxes predicted from the vehicle. During inference, the vehicle in our framework performs in a bandwidth-limited manner by only communicating with the selected infrastructure.

A. Feature Encoder Network

To process features for communication and detection, infrastructures and the vehicle first convert their collected point clouds to features, see Fig. 3. Each infrastructure i converts its collected point clouds S_i to feature maps $\mathbf{F}_i = \mathcal{E}_i(S_i)$ using an encoder $\mathcal{E}_i$, and the vehicle encodes its local data S_v into feature maps $\mathbf{F}_v = \mathcal{E}_v(S_v)$ using an encoder $\mathcal{E}_v$. Let us denote a point p_j in the point cloud as $\left[x_j, y_j, z_j, r_j\right]^T$ with coordinates x_j, y_j, z_j and reflectance r_j. In the first step, the point cloud data is discretized into evenly spaced grids, named pillars in the x-y plane with the size of (l_x, l_y, l_z), where l_x, l_y and l_z denote the width, length and height, respectively. Then the set that all pillars are stacked, is defined as $\mathbf{P} = \{P_k\}_{k=1,\cdots,K}$. A pillar $P_k = \left\{p_{kj} = \left[x_{kj}, y_{kj}, z_{kj}, r_{kj}\right]^T \in \mathbb{R}^4\right\}_{j=1,\cdots,\Omega}$ contains a fixed number of points, denoted as Ω. Specifically, Ω points are randomly selected when the number of points in a pillar is greater than Ω, and zero padding is performed to get Ω points when a pillar contains fewer points. A point p_{kj} in the pillar P_k is then augmented with a 9-dimensional ($D = 9$) vector, that is $(x_{kj}, y_{kj}, z_{kj}, r_{kj}, x_{ckj}, y_{ckj}, z_{ckj}, x_{okj}, y_{okj})$, where the c subscript denotes the distance to the arithmetic mean of all points in the pillar, and the o subscript is the offset between this point and center of the pillar in x-axis and y-axis. For each sample of point clouds, this step creates a tensor of size $D \times Q \times \Omega$, where Q is the number of non-empty pillars. Then, a 1×1 convolutional layer followed by batch normalization (BN) [34] and Rectified Linear Unit (ReLU) [35] is adopted to generate a feature map with size of $C \times H \times W$, where C, H, and W denote channel number, length, and width, respectively.

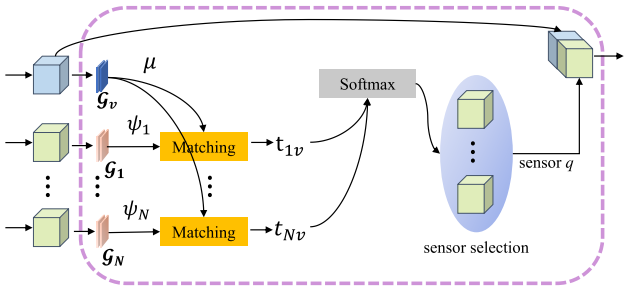


Fig. 4. Illustration of the proposed attention-based communication module. It consists of three steps: 1) the vehicle computes query information and broadcasts it to all neighboring infrastructures; 2) each neighboring infrastructure computes key information and matching score between the local key information and received query information, and sends back the score to the vehicle; 3) the vehicle communicates with neighboring infrastructures based on received attention scores for feature fusion.

B. Attention-Based Communication Module

Fig. 4 illustrates the proposed attention-based communication module. To process query information, the vehicle first encodes feature maps $\mathbf{F}_v$ into compact query features $\mu = \mathcal{G}_v(\mathbf{F}_v)$ using a convolutional network $\mathcal{G}_v$. Then, the vehicle broadcasts the query information to its neighboring infrastructures. After that, each infrastructure encodes feature maps $\mathbf{F}_i$ into key features $\psi_i = \mathcal{G}_i(\mathbf{F}_i)$ using a convolutional network $\mathcal{G}_i$ and computes a matching score t_{iv}, using the received query $\mu \in \mathbb{R}^{M_\mu}$ and its key information $\psi_i \in \mathbb{R}^{M_\psi}$. We design $M_\mu \ll M_\psi$ to save bandwidth usage, since the query information is broadcast from the vehicle through wireless links and the key information is used for calculating matching score locally. Inspired by the attention mechanism presented in [36] and [37], each infrastructure calculates the matching score t_{iv} based on the general attention mechanism [5], that is

$$t_{iv} = \frac{\mu^T \mathbf{W}_a \psi_i}{\left\|\mu^T \mathbf{W}_a\right\| \left\|\psi_i\right\|}, \tag{1}$$

where $\mathbf{W}_a \in \mathbb{R}^{M_\mu \times M_\psi}$ is a learnable matrix. Once all neighboring infrastructures send their scores back to the vehicle, the vehicle normalizes the scores into a probability using a softmax layer. For example, a score t_{iv} can be normalized

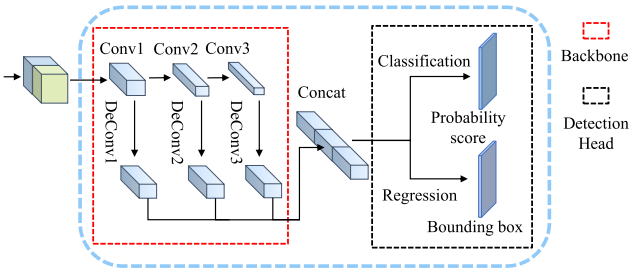


Fig. 5. Region proposal network architecture. The backbone part is performed to generate a multi-scale representation of the input features, and the detection head part is used to estimate a probability score and a bounding box of the proposed region.

using a standard softmax function σ as

$$\sigma(\mathbf{t}_{iv}) = \frac{\exp(t_{iv})}{\sum_{j=1}^{N} \exp(t_{jv})}. \tag{2}$$

During inference, the vehicle selects neighboring infrastructures according to the probability distributions $\sigma(\mathbf{t})$ with $\mathbf{t} = (t_{1v}, \cdots, t_{Nv}.) \in \mathbb{R}^N$. For example, the infrastructure q with the highest probability is selected for cooperative perception, that is

$$q = \arg\max_i \sigma(t_{iv}). \tag{3}$$

Once the vehicle receives feature maps $\mathbf{F}_q$ from the infrastructure q, it uses the corresponding probability $\sigma(\mathbf{t}_{iq})$ to refine $\mathbf{F}_q$ as

$$\mathbf{F}_{\sigma,q} = \sigma(\mathbf{t}_{iq}) \otimes \mathbf{F}_q. \tag{4}$$

where $\otimes$ denotes element-wise multiplication. After that, the vehicle concatenates $\mathbf{F}_v$ and $\mathbf{F}_{\sigma,q}$ along the channel dimension to generate the refined feature maps $\mathbf{F}'_v$, which is given by $\mathbf{F}'_v = [\mathbf{F}_v, \mathbf{F}_{\sigma,q}]$. Note that the neighboring infrastructure selection and feature fusion can be generalized to top-N_s selection and fusion as well. The refined feature maps $\mathbf{F}'_v$ are fed into the region proposal network to get 3D object detection results.

C. Region Proposal Network

In the final stage, a region proposal network $\mathcal{D}$, consisting of a backbone followed by a detection head, is performed to classify objects and predict bounding boxes. The objective of the backbone is to map the refined feature maps $\mathbf{F}'_v$ into multi-scale representation and the detection head is then performed to detect objects as $O_v = \mathcal{D}_v(\mathbf{F}'_v)$. Our backbone follows a CNN architecture similar to the one used in [16], which processes features at three different spatial resolutions. As shown in Fig. 5, the backbone consists of three blocks (Conv1, Conv2, and Conv3). Specifically, Conv1 consists of four 2D convolutional layers with 128 channels and a filter size of 3×3 each. Conv2 consists of six 2D convolutional layers with 256 channels and a filter size of 3×3 each. Conv3 consists of six 2D convolutional layers with 512 channels and a filter size of 3×3 each. The stride size of the first convolutional layer of each block is 2 and the stride size of the following convolutional layers of each block is 1. Each convolution layer is followed by BN [34] and ReLU [35]. Next, we merge the outputs of the three blocks. Since the size of feature maps extracted from the three blocks are ordinarily different, we use a transposed 2D convolutional layer [38] to up-sample each feature map to a fixed size of $(256 \times \frac{H}{2} \times \frac{W}{2})$, see DeConv1, DeConv2 and Deconv3 in Fig. 5. The number of channels of each transposed 2D convolutional layer is 256, and the filter and stride sizes of DeConv1, DeConv2 and DeConv3 are 1×1, 1, 2×2, 2, 4×4, and 4, respectively. After that, we use a convolutional layer with 2 channels and a filter size of 1×1 to get a probability score $\varrho \in [0, 1]$ and a parallel convolutional layer with 14 channels and a filter size of 1×1 to predict a bounding box $A = (x^a, y^a, z^a, w^a, l^a, h^a, \theta^a)$, where (x^a, y^a, z^a) denotes the center of the predicted bounding box and $(w^a, l^a, h^a, \theta^a)$ indicates the width, length, height and rotation angle, respectively.

D. Training

The framework is trained in an end-to-end manner. During training, the vehicle communicates with all neighboring infrastructures and works as a central processor. Specifically, the vehicle receives scores and feature maps from all neighboring infrastructures, e.g., $(t_{iv}, \mathbf{F}_i), \forall i$, and concatenates its local feature map $\mathbf{F}_v$ and attention refined feature maps of all neighboring infrastructures, which is given by

$$\mathbf{F}'_v = [\mathbf{F}_v, \sum_{i=1}^{N} \sigma(\mathbf{t}_{iv}) \otimes \mathbf{F}_i]. \tag{5}$$

Then, given point clouds $\mathbf{S} = \{S_v, S_1, S_2, \cdots, S_N\}$, the parameters of the framework are optimized to detect objects and predict bounding boxes $O_v = \{\varrho, A\} = \mathcal{D}_v(\mathbf{F}'_v)$ by using a joint loss. The joint loss consists of a classification loss $\mathcal{L}_{cls}$, a localization loss $\mathcal{L}_{loc}$ and a direction loss $\mathcal{L}_{dir}$. Let us denote a ground-truth box as $G = (x^{gt}, y^{gt}, z^{gt}, w^{gt}, l^{gt}, h^{gt}, \theta^{gt})$, where (x^{gt}, y^{gt}, z^{gt}) represents the center of the ground-truth box and $w^{gt}, l^{gt}, h^{gt}, \theta^{gt}$ denote the width, length, height and rotation angle, respectively. The object classification loss $\mathcal{L}_{cls}$ is defined as

$$\mathcal{L}_{cls} = -\eta^a (1 - \varrho^a)^\gamma \log \varrho^a, \tag{6}$$

where ϱ^a is the class probability, and η^a and γ are hyper-parameters. The localization difference between a ground-truth box and a predicted bounding box can be defined as

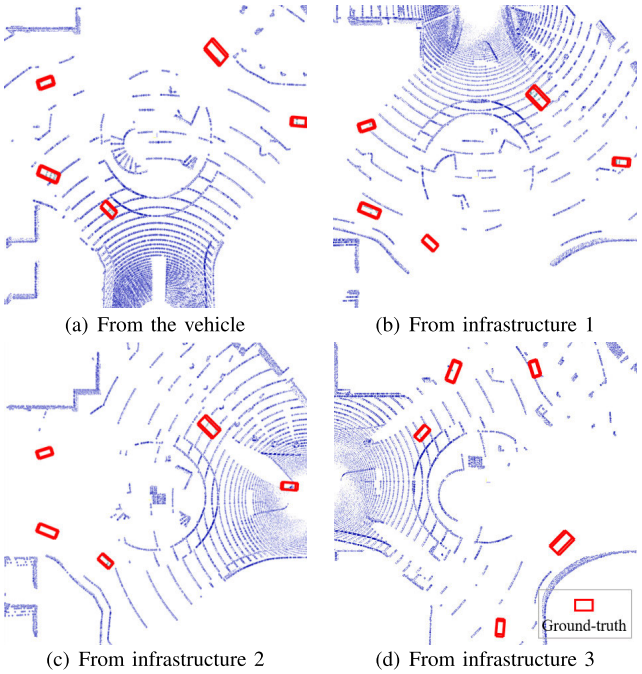
$$\Delta x = \frac{x^{gt} - x^a}{d^a}, \quad \Delta y = \frac{y^{gt} - y^a}{d^a}, \quad \Delta z = \frac{z^{gt} - z^a}{h^a},$$
$$\Delta w = \log \frac{w^{gt}}{w^a}, \quad \Delta l = \log \frac{l^{gt}}{l^a}, \quad \Delta h = \log \frac{h^{gt}}{h^a},$$
$$\Delta \theta = \sin\left(\theta^{gt} - \theta^a\right),$$

with $d^a = \sqrt{(w^a)^2 + (l^a)^2}$. The localization loss between the predicted boxes and ground-truth boxes is given by

$$\mathcal{L}_{loc} = \sum_{b \in (A,G)} \text{Smooth}_{L1}(\Delta b), \tag{7}$$

(a) roundabout (b) T-junction (c) 2-way T-junction

Fig. 6. Bird eye view of three driving scenarios. There are three LiDAR sensors in the roundabout scenario, two in the T-junction scenario, and four in the 2-way T-junction scenario. The white dot circle represents the LiDAR sensors, and the yellow rectangle is the detection range. The autonomous vehicle is driving from bottom to top.



(a) From the vehicle (b) From infrastructure 1

(c) From infrastructure 2 (d) From infrastructure 3

Fig. 7. Examples of point clouds gathered from the vehicle and infrastructures in the roundabout scenario given in Fig. 6(a).

where $\text{Smooth}_{L1}(x)$ [39] is defined as

$$\text{Smooth}_{L1}(x) = \begin{cases} 0.5\ x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{8}$$

Since the localization loss cannot distinguish flipped boxes, a softmax classification loss $\mathcal{L}_{dir}$ is used to classify the predicted bounding boxes on discretized directions [33]. We generate the direction classification targets as follows. If the θ^{gt} is higher than zero, the result is positive; otherwise, it is negative. The direction loss $\mathcal{L}_{dir}$ is defined as

$$\mathcal{L}_{dir} = -\vartheta log(\hat{\vartheta}) + (1 - \vartheta) log(1 - \hat{\vartheta}), \tag{9}$$

where ϑ is the ground-truth heading and $\vartheta \in \{0, 1\}$, $\hat{\vartheta}$ is the estimated heading. The total loss function is then given by

$$\mathcal{L} = \frac{1}{N_{pos}} (\beta_{cls}\mathcal{L}_{cls} + \beta_{loc}\mathcal{L}_{loc} + \beta_{dir}\mathcal{L}_{dir}), \tag{10}$$

where N_{pos} is the number of positive anchors, and β_{cls}, β_{loc} and β_{dir} are scales used to balance these three losses.

V. Experiments

A. Dataset

To evaluate the effectiveness of the proposed model on cooperative perception in autonomous vehicle systems, we use an open-source urban driving simulator Car Learning to Act (CARLA) [11] to generate a new dataset CARLA-3D. CARLA provides open digital assets, including urban layouts, buildings, vehicles, and street infrastructures, and supports flexible specification of sensor suites, environmental conditions, full control of all static and dynamic actors, and maps generation. CARLA enables the simulation of complex driving scenarios as well as datasets, including LiDAR data and camera data, for training and evaluation of autonomous driving systems. We use fixed roadside infrastructure sensors and an autonomous vehicle to generate our dataset. The vehicle and all infrastructures are equipped with LiDAR sensors to capture point clouds. Our dataset consists of three scenarios: a roundabout, a T-junction and a 2-way T-junction. The roundabout scenario includes three infrastructures with LiDAR sensors at 2 meters (2 m) mounting posts placed at intersection. The T-junction scenario uses two infrastructures and the 2-way T-junction scenario includes four infrastructures. Each infrastructure mounts a LiDAR sensor on 2 m high post, and all sensors are placed to fully cover the driving scenarios, as illustrated in Fig. 6. The dataset consists of 1788, 1610, and 1605 frames for the roundabout, T-junction and 2-way T-junction, respectively. Each frame contains two parts: 1) point clouds set and camera images collected from the vehicle and all infrastructures; 2) an object list label annotated each object, which describes the object's ground-truth position, orientation, size, and class. The objects in our dataset include vehicles and pedestrians. Fig. 7 illustrates the point clouds captured from the autonomous vehicle and different infrastructures in the roundabout scenario given in Fig. 6(a). We set the maximum number of objects at any time to 60, including 10 pedestrians and 50 vehicles. The CARLA simulator can design traffic rules and internal collision avoidance mechanisms to manage the motion of the objects. In our dataset, the state of a pedestrian is running or walking, and the probability of running or walking

pedestrian is 0.8. In addition, we treat cars and trucks as vehicles in our dataset, where the probability that a vehicle is set as a car is 0.8. The training set, validation set, and test set are randomly selected at a ratio of 6 : 2 : 2 in CARLA-3D.

Similar to common object detection datasets, e.g., KITTI [40], we consider three difficulty levels of all objects: "Easy", "Moderate" (Mod), and "Hard", depending on the size, occlusion level, and truncation of 3D objects. We define objects in an image with a bounding box height greater than 40 pixels, the occluded area less than 33% and truncated area less than 15% as "Easy" level; objects in an image with a bounding box height greater than 25 pixels, occluded area between 33% and 67% and truncated area less than 30% as "Mod" level; objects in an image with a bounding box height greater than 25 pixels, occluded area greater than 67% and truncated area less than 50% as "Hard" level. To do cooperative object detection using point clouds from LiDAR sensors located in different positions and angles, each sensor needs to map its collected LiDAR data into a unified coordinate system. For example, the vehicle broadcasts its location information to neighboring infrastructures together with query, and then each infrastructure converts point clouds to the positions of the vehicle. The location information of a LiDAR sensor equipped on the vehicle contains its GPS coordinates $C_v = (x_v, y_v, z_v)$ and rotation information, including yaw angles α_v and pitch and roll angles. Given the GPS coordinates of a point cloud j of an infrastructure i and the GPS coordinates of the infrastructure i, denoted as $C_{j,i} = [x_{j,i}, y_{j,i}, z_{j,i}]$ and $C_i = [x_i, y_i, z_i]$, respectively, the point cloud can be transformed into the vehicle's coordinate system as

$$C_{j,v} = \mathbf{R}(\alpha_v - \alpha_i)C_{j,i}^T + \mathbf{R}(\alpha_v)(C_i^T - C_v^T), \tag{11}$$

where $\mathbf{R}$ is a rotation matrix and α_i is the yaw angles of the infrastructure i. Since the pitch and roll angles of autonomous vehicle and infrastructure are 0 in our dataset, $\mathbf{R}$ is defined as

$$\mathbf{R}(\gamma) = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

B. Evaluation Metrics

We use average precision (AP) [41] as a measure to assess the detection performance of the presented 3D object detection framework. AP is derived from precision and recall, which are single-value metrics depending on the probability score and the intersection over union (IoU). Before calculating AP, we first calculate IoU [41]. IoU is given by the ratio of the volume of intersection and volume of the union of the predicted bounding box B_{pred} and ground-truth bounding box B_{gt}:

$$\text{IoU}\left(B_{gt}, B_{pred}\right) = \frac{\text{volume}\left(B_{gt} \cap B_{pred}\right)}{\text{volume}\left(B_{gt} \cup B_{pred}\right)}. \tag{13}$$

IoU is a number in the range of $[0, 1]$, where 0 means no overlap between B_{gt} and B_{pred}, and 1 indicates B_{gt} and B_{pred} are completely overlapped. IoU is used to evaluate the quality of the predicted bounding box position. When the probability score ϱ in Eq. (6) is greater than a threshold δ and $\text{IoU}\left(B_{gt}, B_{pred}\right)$ is greater than a threshold σ, the prediction is positive, otherwise negative. The precision e is defined as the ratio of the number of correct positive predictions in the prediction set, and recall r is defined as the ratio of the number of correct positive predictions in ground-truth set. Then, the corresponding AP for K recall levels [41], [42] can be calculated as

$$\text{AP} = \sum_{k=1}^{K} e_{\text{interp}}(r_{k+1})\left[r_{k+1} - r_k\right], \tag{14}$$

with

$$e_{\text{interp}}(r) = \max_{\tilde{r}:\tilde{r}\geq r} e(\tilde{r}), \tag{15}$$

where K is the number of predicted bounding boxes, and $e(r)$ is the precision as the function of recall r. The k-th recall r_k is calculated by setting the probability score threshold δ equal to the confidence score of the k-th estimated bounding box, sorting by the confidence score in descending order. The $e_{\text{interp}}(r)$ is an interpolated precision that takes maximum precision over all recalls greater than r, which smooths the original precision curve $e(r)$. The mean average precision (mAP) is given by

$$\text{mAP} = \frac{\sum_{i=1}^{M} \text{AP}_i}{M}, \tag{16}$$

where AP_i represents AP value obtained by detecting the i-th class object, M is the number of classes in the detection task.

In addition, we use bandwidth usage (B) per frame and average-precision-improvement-to-bandwidth-usage (AIB) to evaluate the trade-off between detection performance improvement and bandwidth usage of the proposed framework. Bandwidth usage is measured by counting the number of bytes which is sent or received through a wireless link. It is not affected by the properties of sender and receiver. Our experiments are implemented using Python with 32-bit floats. For example, a value of size 1 occupies 4 bytes of memory (a byte contains 8 bits) and can be converted to $B = \frac{4}{1024}$ Kilobytes (Kbytes). AIB is defined as

$$\text{AIB} = \frac{|\upsilon - \upsilon'|}{B} \times 1024, \tag{17}$$

where υ is the mAP of the proposed cooperative perception framework, and υ' is the mAP of the vehicle without communication.

C. Experimental Setup

We conduct four experiments to analyze the 3D object detection performance of the proposed framework, referred to as Learn2com. Firstly, we give an ablation study on size of query and keys. Secondly, we study the effect of different communication learning strategies on cooperative 3D object detection in terms of communication bandwidth usage and detection performance. Then, we compare the detection performance of our method with those of five baseline 3D object detection methods under Easy, Mod, and Hard difficulties. After that, we compare the communication consumption of the proposed method with the five baseline methods in terms of bandwidth usage per frame and AIB. We consider

the following five baseline 3D object detection methods for comparison:

- *LocVehicle*: the autonomous vehicle only uses point clouds collected from its local LiDAR sensors to do 3D object detection.
- *RandSelect*: instead of learning to select one infrastructure to communicate with, in this method, the autonomous vehicle randomly selects one neighboring infrastructure for cooperative detection.
- *CombAll*: the CombAll model considers that the autonomous vehicle communicates with all neighboring infrastructures for cooperative perception and each infrastructure contributes its information with the same weighting factor.
- *AttenAll*: the AttenAll model refines features from all neighboring infrastructures using the attention weights that optimized by the presented attention mechanism, and concatenates the local features and the refined features for cooperative object detection.
- *F-Cooper* [7]: a centralized feature fusion-based method for cooperative perception. Each infrastructure compresses its features using three 3D convolutional layers and sends the compressed features to the vehicle. The vehicle employs maxout fusion operation. We retrain the provided F-Cooper model using the same dataset setting as our model.

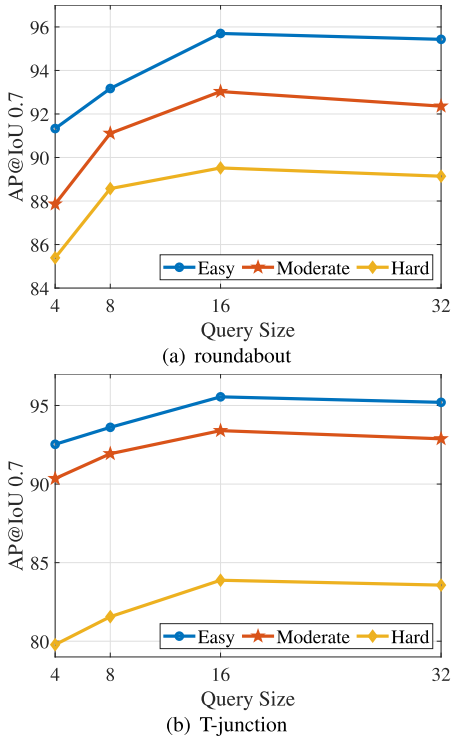


Fig. 8. Ablation study on the size of query under Easy, Mod, and Hard difficulties in two driving scenarios. We use the key size of 128 and vary query size from 4 to 32.

D. Implementation Details

We set pillar size to $l_x = l_y = 0.56$ m, $l_z = 4$ m, and the maximum number of points per pillar $\Omega = 100$ to detect the vehicle. Each class anchor is described by width, length, height, z center, and applied at two orientations: 0 and 90 degrees. Each anchor is matched to a ground-truth and assigned to positive or negative (an object or background). The anchor with the highest IoU that overlaps with a ground-truth or is above the positive match threshold is considered positive, while the anchor is negative when the IoU between the anchor and all ground-truth is below the negative match threshold. The anchors with IoUs between negative match threshold and positive match threshold are ignored during training. For cars, the anchor has width, length, and height of $(1.6, 3.9, 1.56)$ m with a z center of -1.78 m. For trucks, the anchor has width, length, and height of $(1.9, 4.9, 2.05)$ m with a z center of -1.5 m. For pedestrians we set pillars size to $l_x = l_y = 0.28$ m, $l_z = 4$ m, and set the anchor with width, length, and height of $(0.4, 0.4, 1.73)$ m and a z center of -1.5 m. We set positive and negative matching thresholds of class vehicle (class car and truck) anchors to 0.6 and 0.45, respectively, and those of class pedestrian to 0.5 and 0.35. During inference, we measure the AP metric with IoU threshold of $\mathcal{K}$, which can be written as AP@IoU $\mathcal{K}$.

The hyper-parameters η^a, γ, β_{cls}, β_{loc} and β_{dir} in the loss function are set to $\eta^a = 0.25$, $\gamma = 2$, $\beta_{cls} = 1$, $\beta_{loc} = 2$ and $\beta_{dir} = 0.2$. The proposed framework is trained on a PC with four NVIDIA TITAN X GPUs. The models are optimized by the Adam optimizer [43]. The initial learning rate is 0.0002 with an exponential decay factor of 0.8 and decays every 15 epochs. The training process is terminated when the validation loss converges, and the model with the best evaluation performance will be saved for test.

E. Experimental Results

1) Impact of Query and Key Sizes on Detection Performance: In our attention-based communication block, the sizes of query and key are considered to be different, where the size of query is much smaller than those of key to save bandwidth usage. To determine the setup of the sizes of query and key, we do ablation studies on the different sizes of query and key. Since we consider an 80.64 m$\times$71.68 m rectangle detection range with pillar size to $l_x = l_y = 0.56$ m, the size of the feature maps extracted from the feature encoder network is $64 \times 128 \times 144$. We first fix the key size to 128 and analyze the effect of query size on detection performance. Fig. 8 shows detection performance of the proposed framework versus the number of query size under Easy, Mod, and Hard difficulties in two driving scenarios. We observe that the detection performance of the proposed framework is increased with increasing the number of query size under all difficulties. The detection performance is increased dramatically when the query size is increased from 4 to 16, and the performance improvement is flattening when the query size is greater than 16. In addition, we conduct a similar experiment to study the effect of different key sizes on cooperative 3D object detection. We fix the query size as 16 and show the effectiveness of different key sizes in Fig. 9. The experiment results show that increasing key size can improve detection performance, and the best performance is achieved when the key size is

TABLE I

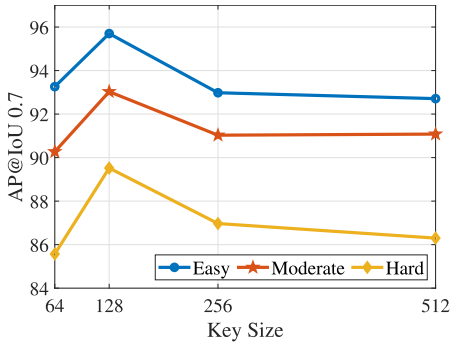
ABLATION STUDY ON COMMUNICATION LEARNING STRATEGIES UNDER EASY, MOD, AND HARD DIFFICULTIES

	(query, key)	roundabout AP@IoU 0.7			T-junction AP@IoU 0.7		
		Easy	Mod	Hard	Easy	Mod	Hard
Backbone-fusion	(4, 64)	91.43	86.87	82.27	90.22	88.65	79.16
Backbone-fusion	(8, 64)	94.53	90.33	85.65	93.60	88.68	80.36
Backbone-fusion	**(16, 64)**	**95.09**	**92.70**	**87.90**	**93.92**	**91.25**	**81.06**
Backbone-fusion	(32, 64)	88.04	87.55	83.00	91.97	89.49	79.35
Backbone-fusion	(16, 128)	90.86	87.10	84.56	91.84	89.26	80.24
Learn2com	**(16, 128)**	**95.70**	**93.03**	**89.52**	**95.55**	**93.40**	**83.88**

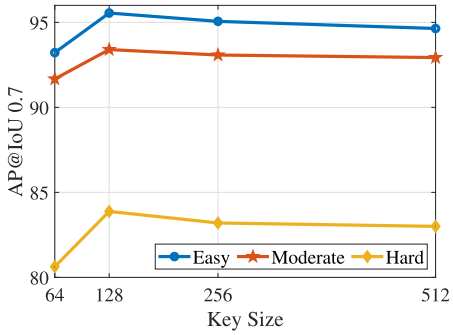
TABLE II

BANDWIDTH USAGE COMPARISON BETWEEN THE PROPOSED METHOD AND THE BACKBONE-FUSION METHOD

	(query, key)	B (Kbytes)	
		roundabout	T-junction
Learn2com	(16, 128)	4608.08	4608.08
Backbone-fusion	(16, 64)	6912.08	6912.08

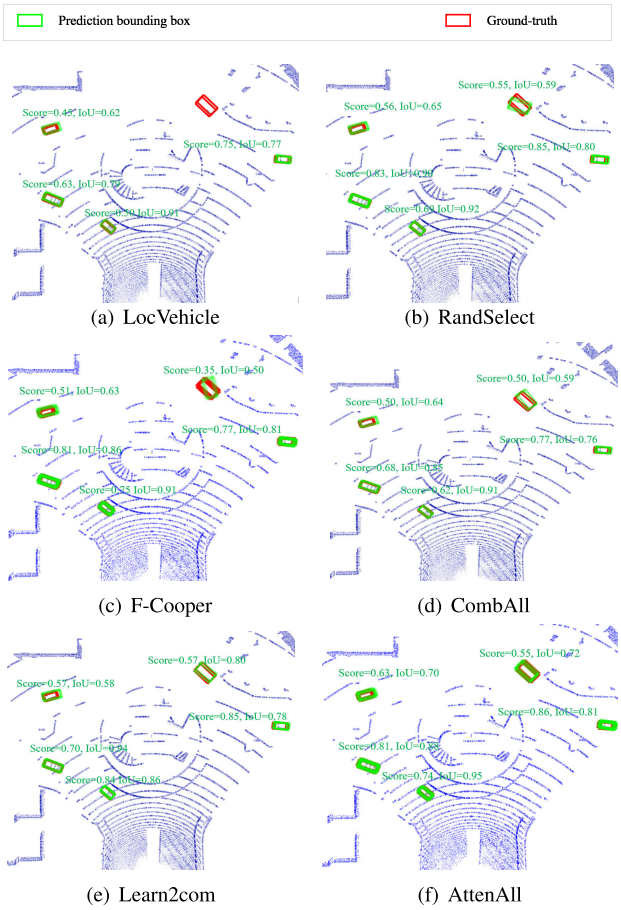


(a) roundabout



(b) T-junction

Fig. 9. Ablation study on the size of key under Easy, Mod, and Hard difficulties in two driving scenarios. We use the query size of 16 and vary key size from 64 to 512.



(a) LocVehicle (b) RandSelect

(c) F-Cooper (d) CombAll

(e) Learn2com (f) AttenAll

Fig. 10. Detection examples of different detection schemes in the roundabout scenario.

128. Once the key size is greater than 128, the detection performance drops and then flattens. This can be explained that the key size of 128 matches the dimension of the input feature maps $64 \times 128 \times 144$. The experiment results indicate that the query size of 16 paired with the key size of 128 can achieve amenable detection performance, we thus set query size to 16 and key size to 128 in our experiments.

2) Impact of Communication Learning Strategies on Detection Performance: We do an ablation study to illustrate the effectiveness of different communication learning strategies: 1) the proposed method, employing features extracted from the feature encoder network for attention-based communication learning; 2) a backbone-fusion method, using the outputs obtained by the backbone of the region proposal network for communication learning. We compare the proposed fusion method with the backbone-fusion method with different sizes of query and key. The comparison results are shown in Table I. We observe that the proposed method gets higher detection accuracy than the backbone-fusion methods. In addition, Table II shows that the backbone-fusion method requires more bandwidth usage than our method. This is because compared with the features extracted from the feature encoder network, the feature maps extracted from the backbone are much deeper.

3) Detection Performance Comparison Under Easy, Mod, and Hard Difficulties: Table III shows the car detection performance comparison between the proposed Learn2com and other baseline models, including LocVehicle, RandSelect,

TABLE III

CARS DETECTION PERFORMANCE COMPARISON UNDER EASY, MOD AND HARD DIFFICULTIES

	Roundabout AP@IoU 0.7			T-junction AP@IoU 0.7			2-way T-junction AP@IoU 0.7		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
LocVehicle	89.17	85.67	82.11	89.91	88.42	78.82	83.46	74.28	60.63
RandSelect	94.57	92.03	86.41	91.28	89.42	79.62	87.27	76.65	65.78
CombAll	92.24	90.16	87.64	94.87	92.63	82.53	86.51	81.93	65.50
AttenAll	98.62	98.02	93.07	96.20	93.45	84.94	90.66	83.88	69.26
F-Cooper	92.54	87.70	85.16	91.74	88.95	79.83	87.50	81.28	62.82
Learn2com	95.70	93.03	89.52	95.55	93.40	83.88	87.99	82.79	67.50

TABLE IV

TRUCKS DETECTION PERFORMANCE COMPARISON UNDER EASY, MOD AND HARD DIFFICULTIES

	Roundabout AP@IoU 0.7			T-junction AP@IoU 0.7		
	Easy	Mod	Hard	Easy	Mod	Hard
LocVehicle	94.86	90.57	78.20	92.91	91.16	78.37
RandSelect	96.95	91.82	83.19	96.78	96.13	82.74
CombAll	96.56	91.98	84.23	98.69	95.93	85.15
AttenAll	99.70	97.48	89.82	99.75	99.72	89.60
F-Cooper	96.96	90.71	83.35	94.28	92.38	82.63
Learn2com	99.64	92.33	84.59	99.70	96.91	86.76

TABLE V

PEDESTRIAN DETECTION PERFORMANCE COMPARISON UNDER EASY, MOD AND HARD DIFFICULTIES

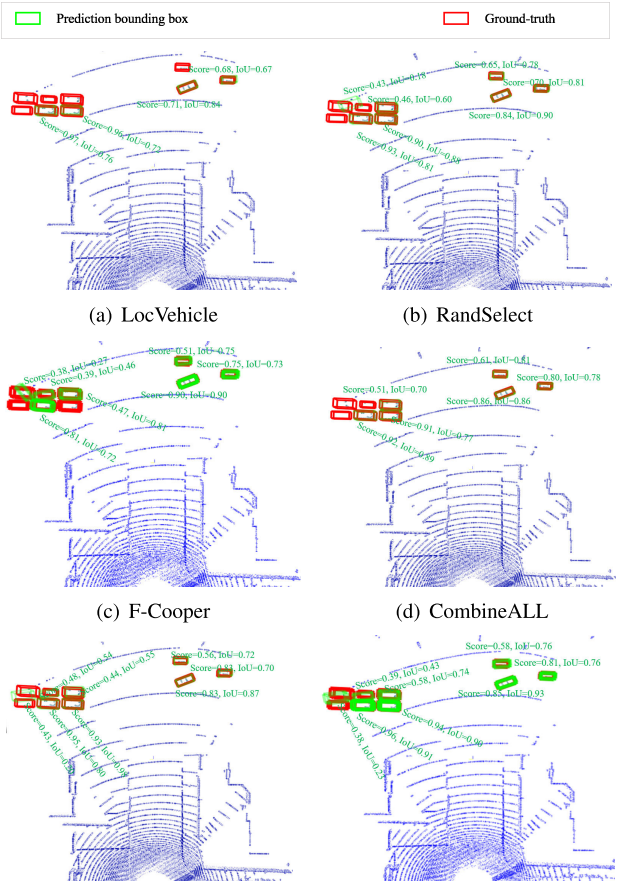
	Roundabout AP@IoU 0.5			T-junction AP@IoU 0.5		
	Easy	Mod	Hard	Easy	Mod	Hard
LocVehicle	85.57	73.21	54.23	72.94	52.55	50.03
RandSelect	86.20	74.92	56.50	74.68	53.65	50.26
CombAll	86.29	76.18	56.86	74.55	53.11	50.80
Learn2com	91.14	79.56	60.30	75.36	57.24	54.83

TABLE VI

THE VEHICLE DETECTION PERFORMANCE COMPARISON IN TERMS OF MAP UNDER MOD DIFFICULTY IN ROUNDABOUT AND T-JUNCTION

	LocVehicle	RandSelect	CombAll	AttenAll	F-cooper	Learn2com
Roundabout	88.12	91.93	91.07	97.75	89.21	92.68
T-junction	89.79	93.03	94.28	96.59	90.67	95.16

CombAll, AttenAll and F-Cooper, under all difficulties. We observe that, all communication-based models have higher detection accuracy than the local processing method LocVehicle under all difficulties. Although the proposed Learn2com performs slightly worse than the attention-based centralized perception method AttenAll, it gets better detection performance than other centralized and decentralized perception methods, including LocVehicle, RandSelect, CombAll and F-Cooper. The detection results of truck and pedestrian are shown in Table IV and Table V, respectively, which are consistent with the experiment results in Table III. The proposed Learn2com achieves better detection performance than LocVehicle, RandSelect, CombAll and F-Cooper, and gets comparable detection accuracy to AttenAll. All communication-based models perform better than the local processing model LocVehicle. In addition, we observe that the pedestrian detection accuracy of all methods is lower than the vehicle detection accuracy, since pedestrian detection is harder than vehicle detection.



(a) LocVehicle (b) RandSelect

(c) F-Cooper (d) CombineALL

(e) Learn2com (f) AttenAll

Fig. 11. Detection examples of different detection schemes in T-junction scenario.

Table VI shows the mean average precision (mAP) of vehicle detection. In the roundabout scenario, Learn2com gets around 4.56 higher mAP than LocVehicle, 0.75 higher mAP than RandSelect, 1.61 higher mAP than CombAll, 3.47 higher mAP than F-Cooper but 5.07 lower mAP than AttenAll under the Mod difficulty level. In the T-junction scenario, compared with LocVehicle, Learn2com gets around 5.37 higher mAP under Mod. Moreover, the mAP of Learn2com is slightly higher than those of CombAll and RandSelect under Mod. Fig. 10 shows detection examples of different detection schemes in the roundabout scenario. Fig. 11 shows detection examples of different detection schemes in the T-junction scenario.

4) The Relationship Between the Final Choice of Infrastructure and Its Distance to the Vehicle: Table VII shows the

TABLE VII

THE RELATIONSHIP BETWEEN THE FINAL CHOICE OF INFRASTRUCTURE AND ITS DISTANCE TO THE VEHICLE IN ROUNDABOUT AND T-JUNCTION

	Roundabout			T-junction	
	infrastructure index			infrastructure index	
	1	2	3	1	2
distance (m)	80	$40\sqrt{2}$	$40\sqrt{2}$	73	73
probability	0.73	0.27	0	0	1

TABLE VIII

PERFORMANCE-BANDWIDTH TRADE-OFF ANALYSIS OF COOPERATIVE DETECTION METHODS IN TERMS OF B PER FRAME AND AIB

	roundabout		T-junction	
	B (Kbytes)	AIB	B (Kbytes)	AIB
RandSelect	**4608**	0.85	**4608**	0.72
CombAll	13824	0.22	9216	0.50
F-Cooper	27648	0.04	18432	0.05
AttenAll	13824.08	0.71	9216.08	0.76
Learn2com	**4608.08**	**1.01**	**4608.08**	**1.19**

relationship between the selection probability of infrastructure and its distance to the vehicle in pedestrian detection in the roundabout and T-junction scenarios. We observe that the final choice of infrastructure is not related to its distance to vehicle. The selection of infrastructure is determined by the training process of the framework. The three parts of the proposed framework, including the feature encoder network, the attention-based communication block and the region proposal network, are trained in an end-to-end manner. During inference, the one with highest attention score is selected for cooperative perception.

5) Performance-Bandwidth Trade-Off Analysis: We compare the bandwidth usage and AIB between our method and the other four communication-based baseline methods, which are CombAll and RandSelect, F-Cooper and AttenAll in vehicle detection. For RandSelect, the feature maps with size of $64 \times 128 \times 144$ are randomly selected from one infrastructure for cooperative detection. For CombAll, feature maps from all N infrastructures, with data size of $N \times 64 \times 128 \times 144$ are transmitted for cooperative detection. For AttenAll, in addition to the data size of feature maps from all N infrastructures, query information with size of 16×1 is required for attention weights calculation and each infrastructure broadcasts its attention score with size of 1×1, which is $N \times 1$ for N infrastructures. For F-Cooper, each infrastructure sends its compressed feature maps with size of $128 \times 128 \times 144$ to the vehicle. The feature maps from all N infrastructures are with size of $N \times 128 \times 128 \times 144$. For our method Learn2com, the data transmission includes feature maps from the selected infrastructure, query information with size of 16×1 for attention score calculation and an attention score with size of 1×1. Table VIII shows the performance-bandwidth trade-off analysis of cooperative detection methods in terms of B and AIB. We observe that the proposed method Learn2com has a better performance-bandwidth trade-off than other baseline models.

VI. CONCLUSION

In this work, we proposed a novel cooperative perception framework (Learn2com) for 3D object detection using an attention-based communication scheme. Our work is based on the fact that an autonomous vehicle can perceive the driving environment better by combining sensing information from neighboring infrastructures. The proposed framework consists of three modules, which are a feature encoder network, an attention-based communication block, and a region proposal network. It first maps point clouds into feature maps using the feature encoder network and then learns to communicate with neighboring infrastructures for a better performance-bandwidth trade-off. After that, the region proposal network produces object classification results and 3D bounding boxes. The proposed framework was trained in an end-to-end manner, and adopted a centralized communication scheme during training and distributed communication scheme during inference.

In addition, we built a new dataset CARLA-3D for cooperative 3D object detection in self-driving scenarios based on CARLA. CARLA is a widely-used open-source simulator for autonomous driving research. We employed it to produce numerically-realistic traffic flow and get realistic sensory streams including LiDAR and camera data. The proposed framework was analyzed and compared with five baseline models. Experimental results showed that the proposed framework achieves comparable detection performance to the attention-based centralized perception method (AttenAll), and performs slightly better than other two centralized perception models (CombAll and F-Cooper). Compared with the centralized perception models where the vehicle communicates with all neighboring infrastructures, the proposed framework consumes much fewer communication costs in terms of transmitted bits and accuracy-improvement-to-bandwidth-usage. In addition, Learn2com gets better 3D object detection performance than the local process model (LocVehicle) and the random selection model (RandSelect), since the communication block learns to fuse features extracted from one of the neighboring infrastructures with a general attention mechanism.

This work gives an extensive study on saving bandwidth for collaborative 3D object detection, which has significant potential in practical autonomous driving systems. The major contribution of this work is to design a learning-based communication scheme and provide a better performance-bandwidth trade-off for collaborative 3D object detection. Although this work did not provide the experiment results on a real-world dataset, it has demonstrated that the proposed framework using the attention-based communication provides a better performance-bandwidth trade-off for cooperative 3D object detection. The effectiveness of the proposed framework in balancing performance gain and bandwidth usage will still hold on real-world datasets. It would be interesting to build a real-world dataset for cooperative perception using autonomous vehicles in our future work. In addition, this work assumes that the wireless communication between the vehicle and infrastructures is ideal, and all sensors are precisely synchronized in time. Moreover, this work considers that the autonomous

vehicle performs cooperative perception to augment the observations from different perspectives/infrastructures and increase detection accuracy of objects in its own field of view. To expand the perception range beyond the field of view of the autonomous vehicle, our future work will reconstruct a large-scale cooperative perception dataset, where all objects in the complementary detection region will be labeled as targets, and redesign the attention mechanism to further balance the detection performance and communication bandwidth in new scenarios. The proposed model was trained in a supervised way. Similar to other supervised detection approaches, the proposed model does not address dynamic communication systems, such as unstable wireless communication channels and sensor drift. How to design a more advanced attention mechanism that leverages top-down information from detection models to help extract robust features during inference is an interesting direction for future work.

REFERENCES

[1] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7337–7345.

[2] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 641–656.

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

[4] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 39–50, Jul. 2015.

[5] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[6] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[7] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proc. 4th ACM/IEEE Symp. Edge Comput.*, Nov. 2019, pp. 88–100.

[8] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 29541–29552.

[9] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2X-VIT: Vehicle-to-everything cooperative perception with vision transformer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2022, pp. 107–124.

[10] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2583–2589.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, vol. 78, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., Nov. 2017, pp. 1–16.

[12] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[13] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10526–10535.

[14] J. S. K. Hu, T. Kuai, and S. L. Waslander, "Point density-aware voxels for LiDAR 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8469–8478.

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

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

[17] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.

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

[19] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "TANet: Robust 3D object detection from point clouds with triple attention," in *Proc. AAAI*, 2020, pp. 11677–11684.

[20] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3D point clouds," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 514–524.

[21] Z. Y. Rawashdeh and Z. Wang, "Collaborative automated driving: A machine learning-based method to enhance the accuracy of shared information," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3961–3966.

[22] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3D object detection in driving scenarios using infrastructure sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1–13, Mar. 2022.

[23] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNet: Vehicle-to-vehicle communication for joint perception and prediction," in *Proc. Eur. Conf. Comput. Vis.* Cham, switzerland: Springer, 2020, pp. 605–621.

[24] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 6876–6883.

[25] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 4874–4886.

[26] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Commun. Mag.*, vol. 44, no. 1, pp. 74–82, Jan. 2006.

[27] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.

[28] A. Singh, T. Jain, and S. Sukhbaatar, "Learning when to communicate at scale in multiagent cooperative and competitive tasks," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–6.

[29] S. Sukhbaatar et al., "Learning multiagent communication with backpropagation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 2244–2252.

[30] P. Peng et al., "Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play StarCraft combat games," 2017, *arXiv:1703.10069*.

[31] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[32] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Dec. 2011.

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

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[35] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, vol. 2010, pp. 807–814.

[36] B. Hurl, R. Cohen, K. Czarnecki, and S. Waslander, "TruPercept: Trust modelling for autonomous vehicle cooperative perception from synthetic data," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 341–347.

[37] A. Das et al., "TarMAC: Targeted multi-agent communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1538–1546.

[38] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*.

[39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[40] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[41] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and W. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.
[42] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

J. Wang received the master's degree in electrical and electronic engineering from the Southern University of Science and Technology in 2022. His research interests include machine learning and autonomous driving.

Y. Zeng (Member, IEEE) received the Ph.D. degree in signal and information processing from the Delft University of Technology, The Netherlands, in 2015. From 2015 to 2018, she was a Research Fellow with the Delft University of Technology. She is currently a Researcher with the Southern University of Science and Technology, Shenzhen, China. Her main research interests are intelligent signal processing, including wireless sensor networks, view synthesis, and image restoration.

Y. Gong (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from Southeast University and the Ph.D. degree in electrical engineering from The Hong Kong University of Science and Technology. He was with the Hong Kong Applied Science and Technology Research Institute, Hong Kong, and Nanyang Technological University, Singapore. He is currently a Professor with the Southern University of Science and Technology, Shenzhen, China. His research interests include cellular networks, mobile computing, and signal processing for wireless communications and related applications. He was on the editorial board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.