This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3390552

Pillar Attention Encoder for Adaptive Cooperative Perception

Zhengwei Bai*,†, *Member, IEEE,* Guoyuan Wu†, *Senior Member, IEEE,* Matthew J. Barth†, *Fellow, IEEE,* Hang Qiu†, Yongkang Liu‡, Emrah Akin Sisbot‡, Kentaro Oguchi‡

***Abstract*—Interest in cooperative perception is growing quickly due to its remarkable performance in improving perception capabilities for connected and automated vehicles. This improvement is crucial, especially for automated driving scenarios in which perception performance is one of the main bottlenecks to the development of safety and efficiency. However, current cooperative perception methods typically assume that all collaborating vehicles have enough communication bandwidth to share all features with an identical spatial size, which is impractical for real-world scenarios. In this paper, we propose *Adaptive Cooperative Perception*, a new cooperative perception framework that is not limited by the aforementioned assumptions, aiming to enable cooperative perception under more realistic and challenging conditions. To support this, a novel feature encoder is proposed and named *Pillar Attention Encoder*. A pillar attention mechanism is designed to extract the feature data while considering its significance for the perception task. An adaptive feature filter is proposed to adjust the size of the feature data for sharing by considering the importance value of the feature. Experiments are conducted for cooperative object detection from multiple vehicle-based and infrastructure-based LiDAR sensors under various communication conditions. Results demonstrate that our method can successfully handle dynamic communication conditions and improve the mean Average Precision by 10.18% when compared with the state-of-the-art feature encoder.**

***Index Terms*—Cooperative Perception, Transformer, Feature Filtering, 3D Object Detection, Connected and Automated Vehicles**

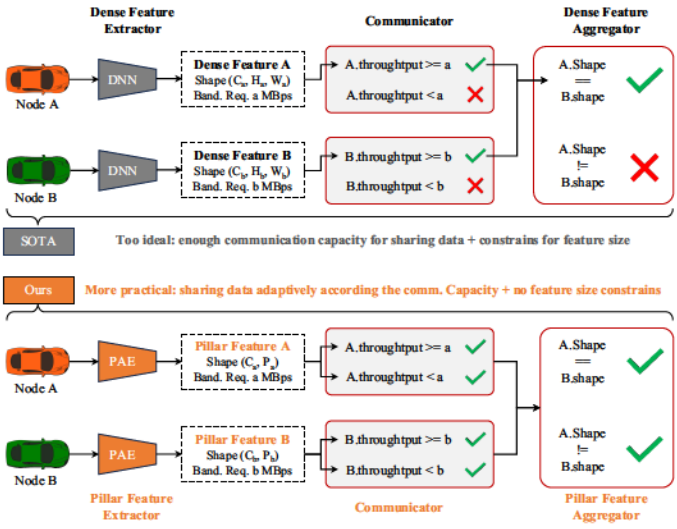


Fig. 1. **The conceptual illustration for the proposed Adaptive Cooperative Perception.** The top part shows the basic pipeline for stat-of-the-art CP methods, which require limited feature-sharing strategies. The bottom part shows our method that can support more adaptive cooperative conditions.

I. Introduction

COMPREHENDING the surrounding environment is one of the key objectives for computer vision systems, which can be used to empower various autonomous systems such as automated driving vehicles [1]. This requires intelligent entities to be able to sense the environment under different conditions with a comprehensive field of view (FOV). To enhance the perception capability, more sensors with different modalities (e.g., RGBD camera, LiDAR, Radar, etc.) tend to be equipped on these entities to build a panoramic ego-perception system [2]. Meanwhile, to support the development of these deep-learning models, various types of datasets have been collected and labeled from sensor platforms with different sensor configurations and modalities [3].

Although remarkable performance has been demonstrated by state-of-the-art perception models to enable a panoramic perception view [4]–[6], it is still a key challenge to unlock the perception bottleneck caused by physical occlusion and limited sensing range [7]. A recent trend to overcome this is to fuse the perception information from spatially separated entities, which is named cooperative perception or collaborative perception (CP). For instance, automated vehicles can enhance safety by receiving detection information for occluded pedestrians from an infrastructure-based perception system [8]. Recent CP approaches [9]–[18] have demonstrated significant potential to enhance perception capabilities by improving perception accuracy and enlarging the field of view.

To fuse the perception data from others, a fundamental process for a CP system is to share the sensing data. Different types of sensing data can be shared, which ends up with different types of fusion methods including early fusion, intermediate fusion, and late fusion [7]. Early fusion requires the sharing of raw sensor data to directly enlarge the sensing range [9]–[11] while late fusion needs the sharing of perception results, e.g., the detected object list [12]. For intermediate fusion, feature data from a specific layer within the perception model is shared and fused [13]–[15]. Among these fusion schemes, intermediate-fusion-based CP approaches [15]–[18] have shown a significant performance improvement by fusing the features generated from Deep Neural Networks (DNNs).

However, these CP approaches bypass a crucial assumption that should not be circumvented in realistic conditions,

* Corresponding Author (Email: zbai012@ucr.edu).
† University of California at Riverside, Riverside, CA 92507 USA.
‡ Toyota North America InfoTech Labs, Mountain View, CA 94043, USA.

i.e., the adaptivity of their CP models. Specifically, feature data requires a large amount of communication bandwidth for transmission. However, as shown in Figure 1, current intermediate-fusion-based CP methods [13], [15]–[18] require that all CP entities must transmit 100% of their feature data with identical spatial shape to enable their fused models, which is nearly impractical due to differences in communication capacities for different entities and uncertainties of wireless communication [19].

In this paper, as shown in Figure 1, we aim to solve the aforementioned issue by designing a CP approach that allows entities to share feature data adaptively based on the actual communication capacity and to fuse the feature data with different spatial shapes. Our contributions can be summarized below:

Proposing a new CP framework called *Adaptive Cooperative Perception* (ACP).

Proposing a new adaptive feature encoder named *Pillar Attention Encoder* (PAE) which extracts the feature data based on the attention mechanism and adaptively reduces the data amount for sharing based on the exact communication bandwidth.

Developing an open-source platform for model training and testing under CP environments.

To the best of our knowledge, there is still no study conducted on this topic, so we introduce several baselines based on the heuristic design to investigate the performance of different methods by numerical analysis and visualization interpretation. The remainder of the paper is organized as follows. Related work is briefly summarized in Section II. Section III presents the methodology, followed by the simulation experiments in Section IV. Section V concludes the paper and gives future insights.

II. Related Work

A. Cooperative Perception

The core idea of cooperative perception is to enhance the single-node perception capacity by leveraging the perception information from another spatially separated entities. These entities can be vehicle-based perception nodes and/or infrastructure-based perception nodes. Hence, in this section, three types of cooperative perception schemes are categorized as 1) vehicle-based CP, 2) infrastructure-based CP, and 3) vehicle-infrastructure-based CP.

1) Vehicle-based CP: Enabled by vehicular networks, Vehicle-to-Vehicle (V2V) cooperative perception has been demonstrated as a promising approach to enhance ego-vehicle perception capabilities through collaborative information sharing among vehicles [20]–[22].

Recent V2V cooperative perception methods significantly explored the usage of deep neural networks for extracting and fusing perception information. For instance, *F-Cooper* [13] achieved cooperation by 1) extracting hidden features from sensor data via Convolutional Neural Networks (CNNs) at each vehicle, i.e., V-PN; and 2) generating perception results based on cross-vehicle feature data sharing. Additionally, Transformers also became an emerging backbone for feature extraction and fusion for cooperative perception [23].

2) Infrastructure-based CP: Equipped with roadside sensors, transportation infrastructure can be a key factor to unlock existing bottlenecks for automated driving, especially in a mixed traffic environment via cooperative perception [24]. Due to the innate attributes of the static and higher pose, infrastructure-based perception entities can achieve better sensing range and field-of-view compared with onboard sensing vehicles [3]. Specifically, a single infrastructure-based perception entity equipped with communication devices can be used for enhancing the perception capacity of vulnerable road users or vehicles with connectivity under certain scenarios, such as the recent real-world prototype system *Cyber Mobility Mirror* [25] and CARMA platform [8].

Furthermore, combining multiple infrastructure entities can significantly improve the perception range. By leveraging the sensing information from multiple roadside cameras with RGB and Depth (RGB-D) information, Arnold et al. [26] proposed a cooperative 3D object detection approach to mainly enhance the sensing range and field of view. Specifically, pseudo-point clouds were generated from the RGB-D camera images and the VoxelNet [27] was applied to fuse all the sensing data to generate the cooperative detection results.

3) Vehicle-Infrastructure-based CP: By leveraging both onboard perception and infrastructure-based perception, vehicle-to-everything (V2X) based cooperative object perception is considered to be the most promising pathway towards tapping the full potential of Cooperative Driving Automation (CDA) [7]. Xu et al. [17] proposed a V2X-based cooperative perception (CP) method considering the heterogeneity of vehicle and infrastructure nodes and multi-scale receptive fields. Lou et al. [8] conducted the Proof-of-Concept of CP in the real world by applying V2X to enable entities to share their sensing results. The program demonstrated the CP system can significantly improve the perception capability of the involved entities.

B. Cooperative Feature Sharing

For cooperative perception, one of the most fundamental components is cooperative feature sharing – sharing the feature data among the entities. In terms of the fusion scheme applied in the cooperative perception method, different types of feature data will be generated and shared [7].

Specifically, cooperative feature fusion can be classified into three categories: 1) *Early Fusion*, which involves fusing raw data during the preprocessing stage [9]–[11]; 2) *Late Fusion*, which entails fusing perception results during the post-processing stage [26], [28]; and 3) *Deep Fusion/Intermediate Fusion*, which involves fusing hidden features from the deep neural networks that designed for feature extraction and fusion.

For early fusion-based CP methods, raw sensor data is designed to be shared with other entities. This enables the direct expansion of the sensing range and field of view. By carefully calculating the relative pose information, both LiDAR data and camera data can be converted to a common coordinate by transformation or projection, such as Cooper [9], AutoCast [11], and AVR [10]. However, transmitting raw data requires an extremely high communication bandwidth to

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3390552

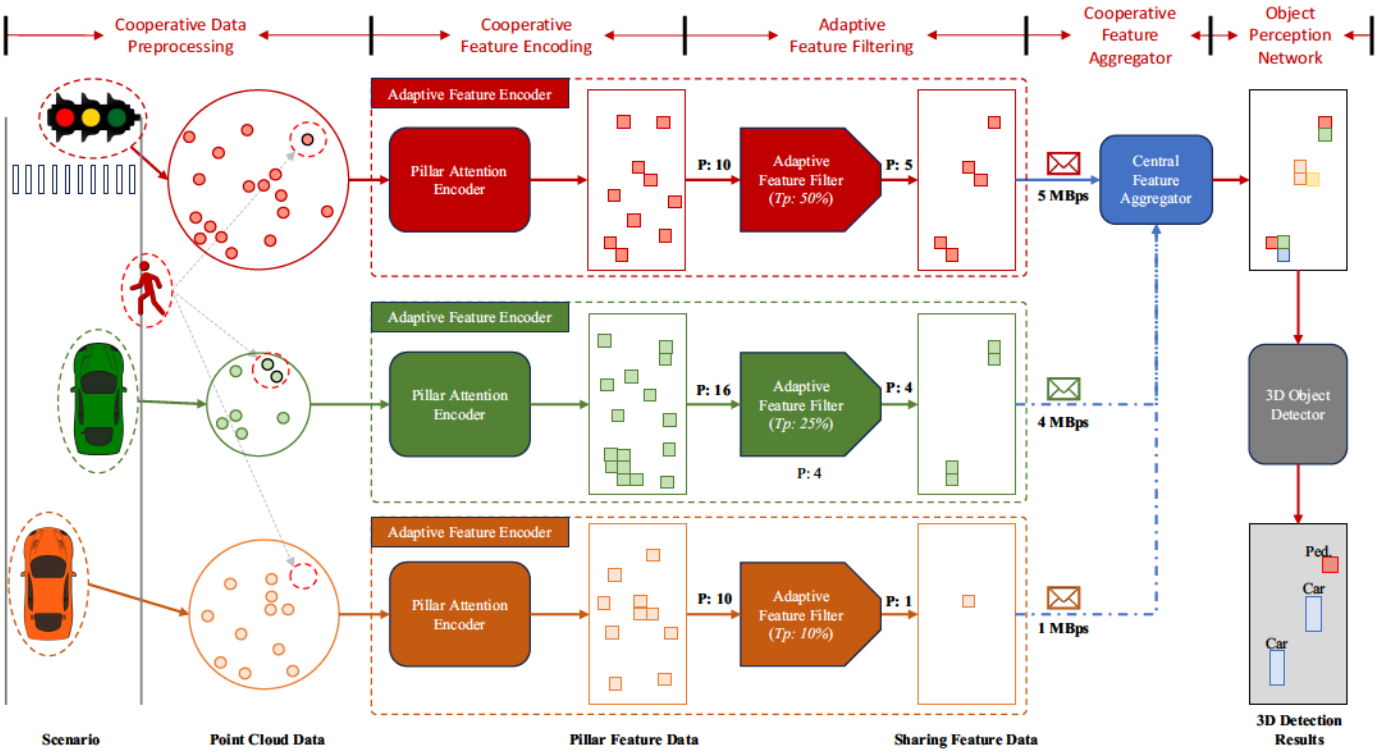


Fig. 2. The diagram depicting the overall framework of the adaptive cooperative perception.

support cooperative perception in the context of automated driving (e.g., A 64-channel LiDAR with 10Hz has a data generating rate of around 20MBps [29]).

Conversely, the data required to be shared by late fusion methods is the semantic textual information, such as the object list containing the 3D object location, rotation, and classification information. For instance, the detection bounding boxes are shared and aligned according to the relative pose estimation, and then these bounding boxes are fused by Non-Maximum Suppression [28] or machine learning-based refining methods [26]. Taking advantage of the semantic data, the bandwidth requirement could be easily achieved for late fusion-based cooperative data sharing. However, these methods can usually provide very limited perception accuracy (e.g. in terms of mean Average Precision or mAP), compared with early fusion and deep fusion as mentioned below.

Deep fusion takes the intermediate features as its input and then outputs the features that combine the hidden feature from different entities. A widely applied methodology is to share the hidden feature extracted from a CNN, such as [13], [15], [16], [30], [31]. Recently, Transformers attracted lots of interest to be the deep feature extractor [17], [18], [23] due to their capability to extract features with larger receptive fields.

Although remarkable perception accuracy is demonstrated by deep fusion-based CP methods, compressing intermediate features is indispensable to deploy the system with realistic communication capacities [7]. Data compression techniques, such as CNN-based channel-wise compression [17] or dedicated Encoder-Decoder methods [15], are commonly used to reduce data volume.

However, these approaches lack the flexibility to support cooperative perception among nodes with distinct communication capacities, where different amounts of data need to be shared for diverse perception nodes. For instance, the fusion of features with varying channels (e.g., a feature map with size (h, w, c_1) is going to be fused with a feature map with size (h, w, c_2)) becomes problematic, and employing all different decoders for decompression from distinct encoders is also impractical.

III. Methodology

A. Adaptive Cooperative Perception Framework

In order to enable cooperative perception with more dynamic scenarios, we propose the Adaptive Cooperative Perception framework which is composed of five main components as shown in Figure 2.

Specifically, the framework can be briefly described as follows:

- **Cooperative Data Preprocessing**: This component aims to apply proper transformation and geo-fencing to prepare the raw sensor data for processing.
- **Cooperative Feature Encoding**: This component aims to extract the feature information from the pre-processed sensor data, which will be used for multi-node feature fusion.
- **Adaptive Feature Filtering**: This component aims to adaptively filter the feature for sharing according to the specific communication bandwidth.
- **Deep Feature Aggregator**: This component aims to fuse the features retrieved from multiple perception nodes and

generate the final feature map for specific downstream tasks, e.g., object detection.

Object Perception Network: This component aims to generate detailed perception results based on specific tasks, such as object detection, tracking, classification, motion detection, etc.

B. Cooperative Data Preprocessing

To fuse perception data from spatially separated sensing nodes, coordinate transformation is a necessary step to align the perception information. This alignment can happen in different stages along the perception pipeline. Sensor data can be aligned from its original coordinate to the collaborator's coordinate by applying transformation according to their relative pose estimation, such as the work in AVR [9], [10], [15]. This alignment can also be designed at the end of the perception pipeline, which is mostly used in late-fusion-based CP methods [28]. For feature-based CP methods, the alignment can also be designed after the feature extraction layer [13].

It is noted that aligning the feature after the feature extraction can lead to a spatially matching issue. Because the extracted feature maps from different entities are grid-like structures, which are hard to be perfectly aligned with others. Thus, specific matching resolution approaches need to be considered [13]. A natural way to circumvent this challenge is to align the raw data before sending it to the feature extractor. Specifically, rather than calculating the hidden feature directly on data collected with respect to (w.r.t.) vehicle's own coordinate, transforming the raw data w.r.t. the coordinate of the cooperative nodes can finally avoid the occurrence of mismatching of the feature maps [17], [18].

However, simply transforming the raw data from cooperative nodes to the ego vehicle's coordinate w.r.t. their relative pose estimation will lead to another severe problem under practical scenarios in which vehicles are not only sending out their feature data but also receiving from others. Transformation according to the relative pose will cause an complexity for those coordinate transformations (i.e., each vehicle needs to apply times coordinate transformations according to the relative pose w.r.t. cooperative nodes).

Thus, for cooperative perception, we propose to use global coordinate transformation rather than relative coordinate transformation to avoid duplicated data transformation [16], [30]. In this study, we consider point cloud data as an example since LiDAR is assumed to be our main sensor for cooperative perception. PCD is assembled on a 3D Cartesian coordinate w.r.t. the geometric center of the LiDAR sensor. So, cooperative data preprocessing aims to align PCD from different local coordinates into a global reference coordinate. Specifically, the PCD can be formulated as:

(1)

Then, the 6-D pose estimation for each sensor is estimated as:

SLaP (2)

where , , , , , and represent the 3D location along axis, axis, axis; and the pitch, yaw, and roll angles of the sensors in the global coordinate, respectively.

Next, alignment to the reference coordinate is calculated by:

(3)

where , , , and represent the rotation matrix along axis, axis, axis, and the translation matrix from ego-coordinate to global reference coordinate, respectively. and represent the raw data before and after the transformation.

C. Pillar Attention Encoder

As mentioned in Section II, existing deep fusion-based CP methods require a fixed spatial size of the feature map to enable the fusion [13], [15], [17], [18], [23]. In other words, the feature data generated from different perception nodes must have the same spatial size so that the deep fusion network is able to fuse them. This significantly limits the flexibility of the cooperation among perception nodes with different computing power and communication bandwidths. This limitation mainly comes from the use of dense feature data to share their cooperative feature, i.e., a deep neural network is applied to extracting the cooperative feature data which has a fixed number of feature channels, height of the feature map, and width of the feature map, respectively.

To unlock this limitation, we propose the Pillar Attention Encoder which extracts the cooperative feature into a format that does not rely on the spatial shape of the feature map. Additionally, a multi-head point attention method is designed to generate stronger representations of the given raw data (i.e., PCD in this paper) which can significantly outperform the original CNN encoder. Figure 3 depicts the main design of the proposed PAE.

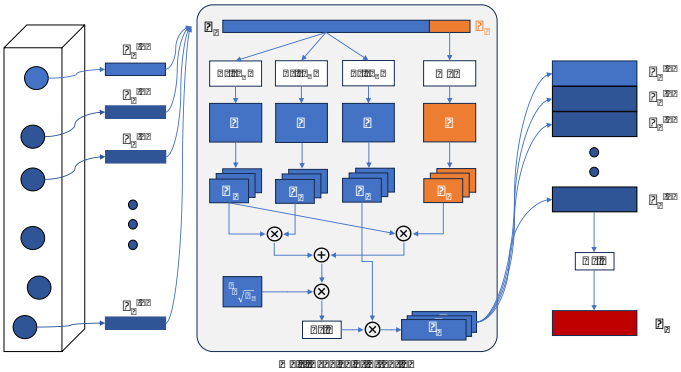


Fig. 3. The diagram depicting the process of the Pillar Attention Encoder.

1) Multi-Head Point Attention: The first step to encode the PCD is to pillarize the 3D point clouds into pillars of points [32] so that the 3D PCD can be reorganized into a pseudo-2D pillar map. The process can be described below.

(4)

(5)

where is the set of pillars of points with the size of . Specifically, and represent the number of pillars and the maximum number of points within a pillar, respectively. at the 2D voxelized spatial location in the range of and , representing the spatial size of the pillar map. Then, for each pillar of points, every point can be formulated as:

(6)

where and are the 3D locations and the intensity of the reflected point, respectively. So, the set of the point feature in a single pillar can be described as:

(7)

For each point feature, we extend its original feature by adding relative geometric features, such as the distance to the arithmetic center of all points in the pillar (the subscript means the -th point in this pillar) and the geometrical center of the pillar , which has the size of and are shown as:

(8)

Specifically, we define the original 3D location feature as which will be mainly used as the position embedding for the point feature vector in the attention calculation.

(9)

Now, the pillar-wise feature can be generated by combining the point feature mentioned above, which can be described as:

(10)

where and have the feature size of and .

Then, three independent linear layers , , and are designed to convert the original pillar feature to , and as below.

(11)

where , and have the size of . Specifically, represents the channels of the hidden feature, which is set as in this paper.

The positional embedding with the size of is then calculated via a multi-layer perceptron (MLP) as described by (for more details about positional embedding, please refer to Section III-C2):

MLP (12)

By applying the multi-head self-attention mechanism , , and positional embedding are then divided into , , , and , , with the size of where represents the number of heads.

Thus, the multi-head point attention is formulated as:

(13)

where is the Softmax operation and the feature from each attention head has the size of . Then attention features from all attention heads are combined and fed into a LayerNorm layer to generate the pillar attention feature with the size of :

LayerNorm (14)

The last step is to perform the channel-wise operation to the pillar feature to achieve the final extracted pillar feature with the size of :

(15)

(16)

2) Positional Embedding: Self-attention model is innately designed for calculating the association between the *Query* and *Key* [33]. To achieve it, the position information for each *Query* is essential to inject the *spatial distance* information into the attention calculation. Thus, most of the state-of-the-art attention-based models have their specifically designed positional encoding or positional embedding methods. To be clear, positional encoding means adding predefined position information (e.g., sinusoidal positional encoding [33]) to the input feature before the MHSA block. However, positional embedding means embedding positional information as hidden features, which is learnable while the positional encoding is fixed.

In the context of computer vision tasks, the designing of positional embedding is relatively more abstract than designing it in natural language tasks. As the latter provides explicit relative positional information between words, while the former has data that are highly structured [34].

However, for LiDAR point cloud data, the positional information is naturally provided from the raw data, which is the 3D location w.r.t. the sensor's coordinate. Thus, the positional embedding design for the proposed multi-head point attention can follow a natural way by decomposing the core of attention weights between the *Query* point and *Key* point , which can be described as:

(17)

Thus, we can embed the positional information to by:

(18)

where means the *MLP* is applied to the natural location feature and the positional information is embedded in the term . Furthermore, by extending the equation above from a single point level to all points within a pillar, we can have:

(19)

where , and have the size of and the has the size of . Considering the multi-head decomposition [33], the attention feature map for multiple attention heads can be formulated as:

(20)

where , and have the size of and the has the size of . which will be multiplied by to get Equation 13.

D. Adaptive Feature Filtering

As mentioned earlier, feature-based CP methods performed remarkable perception accuracy by fusing the hidden features from different perception nodes. However, the massive amount of feature data makes these models hard to implement under real-world conditions where limited and dynamic communication bandwidths exist. Meanwhile, data compression is helpful in reducing the data amount to transmit with low communication bandwidth. But it is nearly impractical to simply use data compression to handle dynamic situations.

Instead of data compression, another straightforward way to reduce the amount of data is filtering out the data points that have less value to the perception task. The main challenging issue for this ideology is how to define the "value" of data points. Based on the process of self-attention calculation, we can use the "the most eminent attention value" generated from the Pillar Attention Encoder for each pillar feature as the "significance value" of this feature in terms of the perception task.

The prerequisite to filter the feature cells within a feature map is the independence of the feature cells. Otherwise, filtering and reassembling dependent feature cells will impact the integrity of the data representation. For instance, most of the current feature-based cooperative perception methods use the feature after a deep neural network backbone – a convolution neural network or a transformer network. Due to the fundamental calculation mechanism, every feature cell in the feature map after the backbone network (i.e., one feature cell with the size of (1, C) out of the whole feature map with the size of (n, C)) will contain the information from the adjacent cells (convolution-based backbone) or the whole feature map (transformer-based backbone).

As shown in Fig.3, the whole process of generating each pillar feature – – is independent of the data outside of this pillar. In other words, each pillar feature is generated solely from its independent data, which gives it the property to be easily reassembled and filtered without impacting the integrity of the data representation. In other words, our proposed Pillar Attention Encoder can inherently support the requirement of adaptive feature filtering.

In this study, we formulate the adaptive feature filtering as a three-step process. The first step is to extract the most eminent attention value for each pillar feature by Eq. 21:

(21)

where represents the *value* of -th pillar feature by max-pooling the most significant value among its channel.

The second step is to generate an affordable number of pillar features, , that can be transmitted with the communication capacity. Specifically, the number of total pillar features generated from the encoder can be defined as .

Thus, the final step is to generate $_{\text{AFF}}$ by selecting the pillar features with the top-k highest value, which is defined as:

(22)

E. Deep Feature Aggregator

In the ACP framework, as shown in Figure 2, a deep feature aggregator is designed to fuse the feature data shared from different perception nodes. In the context of adaptivity, the feature aggregator needs to be able to absorb features with different spatial shapes and features from different numbers of perception nodes. For example, the feature sizes of different vehicles may vary due to the difference in communication capabilities in the onboard devices. Additionally, the number of vehicles or infrastructures that are involved in the cooperative perception system may be dynamic as well.

To enable feature fusion under dynamic, scalable, and heterogeneous conditions, a two-stream neural network is adopted to fuse pillar features from different perception nodes [30]. As shown in Fig. 4, the network consists of three components: an infrastructure-based feature aggregator, a vehicle-based feature aggregator, and an infrastructure-vehicle-based feature aggregator.

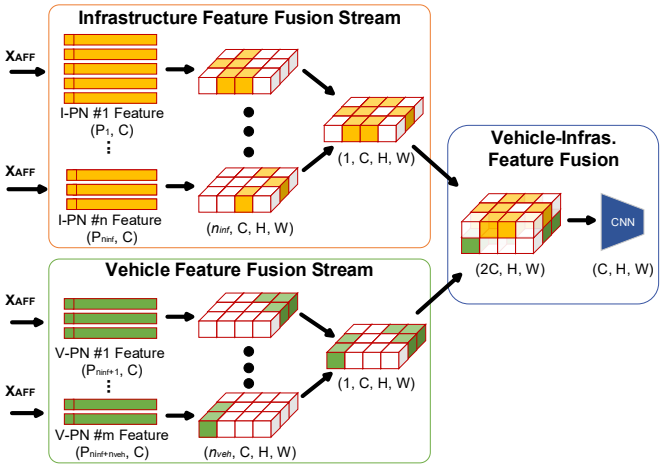


Fig. 4. Diagram illustration for the deep feature aggregator

In this paper, the feature generated for cooperation is the pillar attention feature $_{\text{AFF}}$ which has a spatial shape of where and represent the exact number of pillar features (e.g., ,..., in Fig. 4) and the number of channels for each pillar feature, respectively. Specifically, as is the adaptive threshold according to the communication capacity. Hence, if $_{\text{inf}}$ infrastructure nodes and $_{\text{veh}}$ vehicle nodes are involved in the ACP system, the infrastructure feature fusion stream will take in $_{\text{inf}}$ different $_{\text{AFF}}$ with different spatial sizes. Then these pillar features are projected to a pseudo-bird-eye-view feature map which has a feature shape of $_{\text{inf}}$. Then, *maxpooling* layer is applied to aggregating these data along the first dimension and the aggregated feature will have a spatial shape of . Similarly, the vehicle-based feature fusion

stream will generate another feature map with specifically based on the onboard features. Lastly, these two feature maps are aggregated by a *concatenation* layer followed by a *convolution* layer. So the final output has a normal spatial shape of .

F. Object Perception Network

After the feature fusion, theoretically, perception networks that can be used for single-sensor-based features are also able to work on this fused feature. Meanwhile, the perception network can be designed for different types of downstream perception tasks, such as detection, tracking, segmentation, etc. In this paper, we simply applied a widely used 3D object detector consisting of a feature pyramid network and a 3D anchor-based detection head [35].

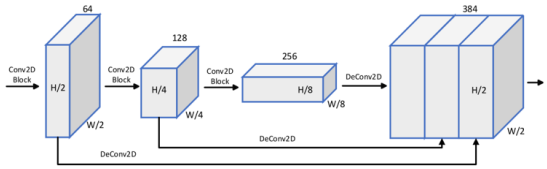


Fig. 5. Deep neural network backbone for hidden feature extraction.

The structure of the detector is briefly introduced below to help illustrate the whole network structure utilized in this paper. Specifically, Figure 5 depicts the structure of the feature pyramid network. The feature downsampling blocks consist of *convolution* layers. Each *Conv2D* block consists of one *convolution* layer with the kernel of , followed by several *convolution* layers with kernels of . Specifically, the numbers of *convolution* layers in each block are , , and , respectively. The *deconvolution* layers (*DeConv2D*) are then applied to upsampling the feature map from different stages of the feature pyramid to include the feature with different receptive fields.

As the main task in this paper is object detection, an anchor-based 3D object detection head [32] is applied to generate the object-level prediction including 3D location, dimensions of the bounding box, yaw angle, and class information.

IV. Experiments

A. Experimental Details

1) Dataset for Adaptive Cooperative 3D Object Detection: The "*CARTI*" (i.e., **CAR**la-ki**TtI**) platform [36] is applied to collect the LiDAR sensor data and 3D ground truth labels for training and testing the models. Specifically, two infrastructure nodes and three vehicle nodes are deployed for data collection. A total of frames of 3D point clouds are collected (samples if counting perception nodes in all frames), including frames for training, frames for validation, and frames for testing.

The specification of the sensors applied is shown in Table I and two different LiDAR settings are used based on our previous work in the real world [37]. To make the simulated point cloud data closer to the realistic conditions, we configure

TABLE I
Sensor specification for the datasets collected in this study.

Sensor Specification	Setting [onboard/roadside]
LiDAR Channels	
LiDAR Height	
LiDAR Sensing Range	
LiDAR Rotation Frequency	Hz
Upper FOV	
Lower FOV	
Noise for Points Per Beam	
Missing Reflection Rate	
Intensity Dropoff Range	

the simulated LiDAR with certain noise settings including standard deviation of the noise for points per beam, missing reflection rate, and intensity dropoff range, which are also specified in Table I. The computing platform for collecting the simulation data and model training is introduced below.

2) Training Details: The training and testing platform consists of an Intel® Core™ i7-10700K CPU and an NVIDIA RTX 3090 GPU. The training pipeline is designed with 160 epochs with *batchsize* of 2. The voxel size is set as [ ] and the maximum number of pillars per node is set as . Specifically, during the training stage, – the number of pillar features that are able to transmit with others – is randomly varying from the range . It is noted that different nodes in the same frame will be assigned various values to emulate the dynamic conditions in the real world.

3) Evaluation Details: It is noticeable that the evaluations under different communication bandwidth limitations (i.e., different) are conducted *WITHOUT* any further fine-tuning. This zero-shot setting allows us to evaluate the model under more critical but realistic conditions.

The detection performance is measured with Average Precision (AP) at Intersection-over-Union (IoU) thresholds of 0.7 for cars and 0.25 for pedestrians. Furthermore, based on the Minimum number of Points (MP) reflected by the ground target, each evaluation class is further divided into three categories: Easy (MP 10), Medium (MP 5), and Hard (MP 1), respectively, to investigate the performance of CP methods at different difficulty levels.

To evaluate the models in a dynamic environment, we tested the models with several different dynamic ranges of communication capacities. To make the evaluation representative and efficient, we evaluate the models under different conditions including: , , , , , and . It needs to be mentioned that all the models are NOT fine-tuned by any of those thresholds designed above while testing.

4) Feature Encoder Baselines: To compare the performance, we choose one of the most popular feature encoders for PCD-based object detectors – Pillar Feature Encoder (PFE) as our baseline, which is proposed in *PointPillar* [32]. Additionally, in the later experiment, our method will be noted as PAE (Pillar Attention Encoder).

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3390552

5) Adaptive Filtering Baselines: Since adaptive filtering is a new concept in this field, we consider several heuristic feature filtering methods including:

K-Nearest Neighbor (KNN): Sorting the features with respect to (w.r.t.) the distance between the pillar feature and the location of the sensor itself. We next select top-K nearest features, since we might assume that the model will have higher confidence in the feature closer to it.

K-Farthest Neighbor (KFN): A converse method w.r.t. the KNN by selecting the farthest feature first.

K-Random Sampling (KRS): A random sampling method among the pillar features.

Specifically, for calculating the distance between the spatial location of the feature cell and the sensor, *Manhattan Distance* is applied to calculating the priorities with better computational efficiency (when compared with *Euclidean Distance*), which is defined as below:

(23)

where is the Manhattan distance between the feature location and the sensor location . Specifically, our adaptive filtering method is noted as KFV (K-Feature Value).

TABLE II
AP PERFORMANCE COMPARISON UNDER DIFFERENT ADAPTIVE TRANSMITTING RATES OF THE ORIGINAL FEATURE SIZE (TR: TRANSMISSION RATE).

TR [ ]%	Methods	Car@IoU=0.7			Pedestrian@IoU=0.25		
		Easy	Medium	Hard	Easy	Medium	Hard
1% - 10%	KNN	18.67	17.56	17.53	11.75	11.89	10.57
	KFN	15.96	16.14	16.16	10.31	11.42	12.24
	KRS	17.01	16.11	16.04	2.20	2.43	2.13
	K-PFE	31.86	30.98	30.54	19.13	19.07	16.57
	K-PAE (Ours)	35.98	34.19	34.04	19.71	19.55	17.01
	Imp.	*12.93%*	*10.36%*	*11.46%*	*3.03%*	*2.52%*	*2.66%*
10% - 20%	KNN	17.92	17.89	17.87	11.06	10.5	9.09
	KFN	33.01	33.06	33.00	9.20	11.90	13.59
	KRS	37.79	35.85	35.75	13.13	13.65	12.57
	K-PFE	50.11	48.51	46.85	21.99	22.63	22.11
	K-PAE (Ours)	62.4	62.11	61.87	25.2	26.68	24.89
	Imp.	*24.53%*	*28.04%*	*32.06%*	*14.60%*	*17.90%*	*12.57%*
20% - 30%	KNN	26.63	26.48	26.4	11.5	11.55	10.45
	KFN	49.90	49.86	49.74	15.62	18.85	21.12
	KRS	51.31	50.60	50.02	15.93	17.15	16.03
	K-PFE	70.89	70.73	70.59	22.11	25.02	26.38
	K-PAE (Ours)	79.84	79.61	79.29	23.85	26.49	27.58
	Imp.	*12.63%*	*12.55%*	*12.32%*	*7.87%*	*5.88%*	*4.55%*
30% - 40%	KNN	34.91	28.75	28.74	13.53	13.31	11.96
	KFN	59.65	59.55	59.42	16.48	19.91	22.95
	KRS	60.41	59.81	59.35	13.32	14.96	14.69
	K-PFE	79.72	79.55	79.35	21.77	24.7	26.74
	K-PAE (Ours)	86.43	86.19	79.76	24.24	27.16	29.56
	Imp.	*8.42%*	*8.35%*	*0.52%*	*11.35%*	*9.96%*	*10.55%*
40% - 50%	KNN	43.55	38.02	37.99	16.35	16.73	14.57
	KFN	68.51	68.23	65.25	12.89	16.68	19.69
	KRS	69.18	68.46	65.72	1456	17.31	17.06
	K-PFE	80.07	79.99	79.91	16.98	20.15	22.49
	K-PAE (Ours)	88.16	87.33	86.87	23.47	27.33	29.81
	Imp.	*10.10%*	*9.18%*	*8.71%*	*38.22%*	*35.63%*	*32.55%*
50% - 100%	KNN	62.14	56.89	56.87	21.63	23.42	21.99
	KFN	74.99	74.92	69.61	14.86	18.92	22.61
	KRS	79.03	78.88	78.73	16.21	19.66	21.46
	K-PFE	86.54	79.99	79.94	16.54	19.96	22.37
	K-PAE (Ours)	87.07	87.08	83.23	23.4	26.41	27.75
	Imp.	*0.61%*	*8.86%*	*4.12%*	*41.48%*	*32.31%*	*24.05%*

B. Evaluation and Analysis

In this section, we evaluate dynamic feature-sharing approaches from two perspectives: 1) quantitative results and analysis to show the numerical results of the methods; and 2) qualitative results and analysis to visualize the performance and interpret the methods. All models are evaluated in terms of 3D object detection for cars and pedestrians. The average precision (AP) is applied to assess the performance.

TABLE III
AP PERFORMANCE COMPARISON UNDER FULLY DYNAMIC TRANSMITTING RATES (1% - 100%).

Methods	Encoders	Car@IoU=0.7			Pedestrian@IoU=0.25		
		Easy	Medium	Hard	Easy	Medium	Hard
K-NN	w/ PFE	26.67	26.58	26.52	8.28	7.58	6.87
	w/ PAE (Ours)	26.78	26.71	26.68	12.84	12.21	11.55
	Imp.	*0.41%*	*0.49%*	*0.60%*	*55.07%*	*61.08%*	*68.12%*
K-FN	w/ PFE	45.40	45.42	45.39	15.19	18.13	19.25
	w/ PAE (Ours)	50.36	50.37	50.28	12.00	15.27	17.37
	Imp.	*10.93%*	*10.90%*	*10.77%*	*-21.00%*	*-15.77%*	*-9.77%*
K-RS	w/ PFE	49.96	45.96	45.85	15.64	16.97	15.74
	w/ PAE (Ours)	52.20	51.73	51.37	15.92	17.20	16.30
	Imp.	*4.48%*	*12.55%*	*12.04%*	*1.79%*	*1.36%*	*3.56%*
K-FV	w/ PFE	65.42	62.07	61.96	22.20	24.08	25.33
	w/ PAE (Ours)	70.91	70.70	70.52	23.64	25.50	26.36
	Imp.	*8.39%*	*13.90%*	*13.82%*	*6.49%*	*5.90%*	*4.07%*

1) Quantitative Results and Analysis: Table II shows the AP performance comparison for different methods under various adaptive transmitting rates of the original feature size. The table also highlights the improvement of our method compared with the K-PFE method which uses the PFE as the encoder and our KFV as filter. It shows that our method (K-PAE) outperforms the baseline under all testing scenarios.

Specifically, with 10% - 20% transmitting rates, our method can improve the AP for car detection by 24% to 32% approximately compared with the baseline. Under the 50% to 100% transmitting rates, our method can improve the AP for pedestrians by 24% to 41% approximately.

For adaptive feature filters, from Table II, our KFV method significantly outperforms others (KNN, KFN, and KRS) by a large margin. The performance improvement can be explained as that features with higher values will have higher dominance for the perception results. Thus, prioritizing those features with higher values and transmitting them before the features with lower values can lead to better performance.

To further compare the performance between PFE and our PAE, we test both encoders for all different filtering methods, as shown in Table III. In Table III, for three of the four feature filtering methods, PAE can improve the performance by up to 68% with testing under the fully dynamic environment with transmitting rates.

TABLE IV
AP PERFORMANCE COMPARISON UNDER DIFFERENT TRANSMITTING RATES.

TR %	Encoders	Car@IoU=0.7			Pedestrian@IoU=0.25		
		Easy	Medium	Hard	Easy	Medium	Hard
1/20	PFE	50.11	48.51	46.85	21.99	22.63	22.11
	PAE (Ours)	52.91	50.23	49.97	21.42	22.18	23.48
	Imp.	*5.59%*	*3.55%*	*6.66%*	*-2.59%*	*-1.99%*	*6.20%*
1/15	PFE	61.24	60.84	60.55	22.02	24.33	25.03
	PAE (Ours)	69.90	69.62	69.40	23.98	25.83	26.74
	Imp.	*14.14%*	*14.43%*	*14.62%*	*8.90%*	*6.17%*	*6.83%*
1/10	PFE	70.54	70.41	70.27	21.64	24.38	26.03
	PAE (Ours)	85.26	79.75	79.67	25.25	27.61	28.36
	Imp.	*20.87%*	*13.27%*	*13.38%*	*16.68%*	*13.25%*	*8.95%*
1/5	PFE	79.77	79.68	79.59	16.99	20.10	22.38
	PAE (Ours)	88.30	87.78	87.64	24.80	27.56	29.90
	Imp.	*10.69%*	*10.17%*	*10.11%*	*45.97%*	*37.11%*	*33.60%*

Besides the dynamic environment in which different nodes

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3390552

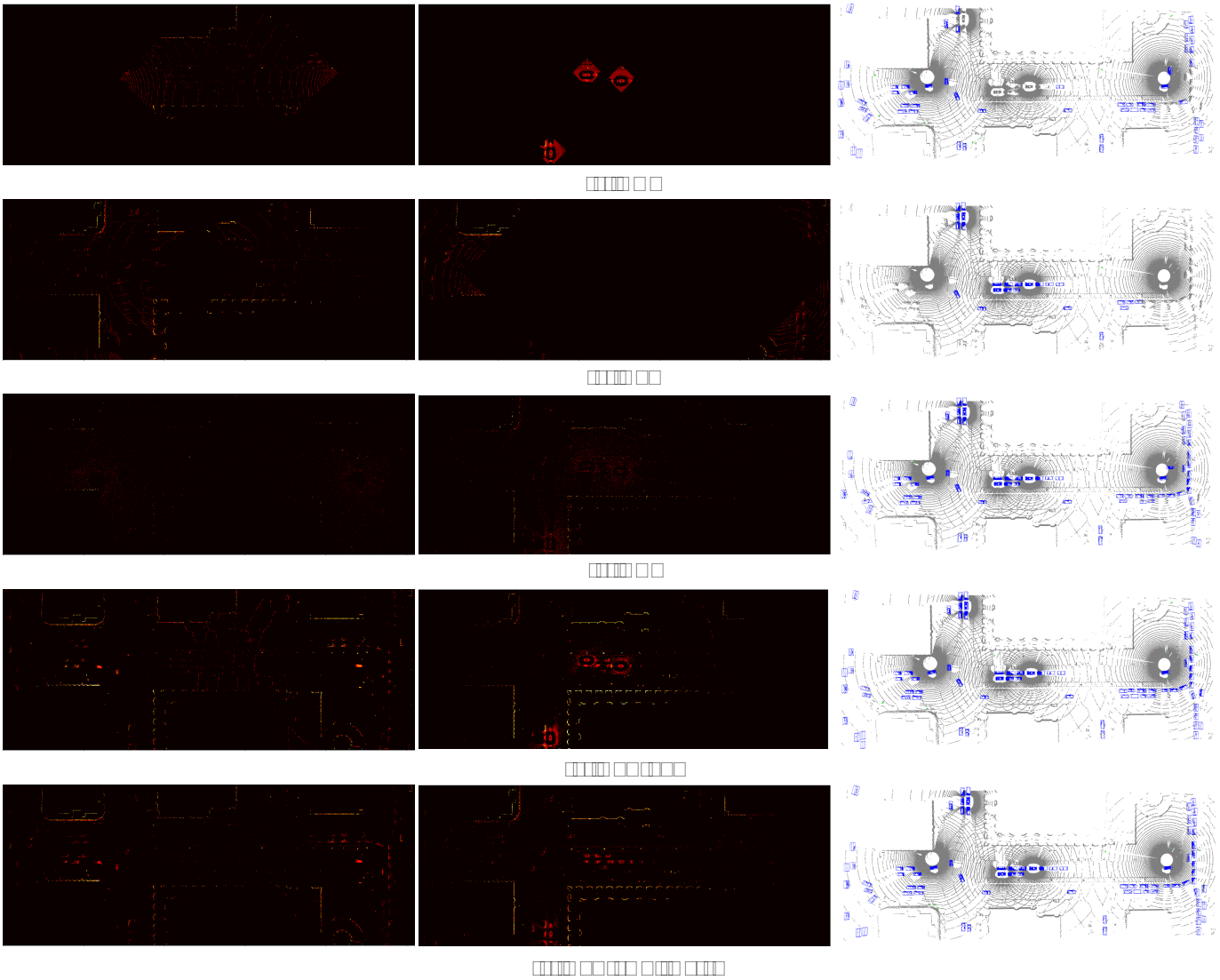


Fig. 6. **Visualization of the filtered feature and 3D object detection results for different ACP methods (better zoom-in).** From the left, the three columns show: 1) the filtered features from infrastructure nodes, 2) the filtered feature from vehicle nodes, and 3) the 3D detection results (blue and green bounding boxes represent cars and pedestrians), respectively. Specifically, different nodes with the same type are grouped together for conciseness.

may have different transmitting rates, we also tested the methods under homogeneous transmitting rates as shown in Table IV. By transmitting out of the original features, our method can outperform the PFE method by 6+% under the *Hard* condition for both car and pedestrian detection. Under the transmitting rate of , , conditions, our method can improve the performance by 10% to 21% for car detection and 6% to 46% for pedestrian detection, respectively.

Additionally, the upper bound of the detection performance of the model is % when , which is in this paper. According to Table IV. we can conclude that our method can achieve 97% and 94% of the AP performance while reducing the transmitting data by and . For the PFE baseline, it can only remain the AP performance by 88% and 78%, respectively.

2) Qualitative Results and Analysis: To further investigate the performance of different ACP methods, Figure 6 shows the visualization of filtered features and the corresponding visualization of 3D object detection results. Heatmaps are applied to visualize the most-eminent feature value of the filtered feature for different methods. Literally, each node should have its own feature map visualization, but for conciseness, we combined the features from nodes with the same type, e.g., features from all vehicle nodes are combined into one feature map for visualization. This operation is only used for efficient visualization and each point in the figure represents a pillar feature.

By comparing the visualized feature maps, several interesting findings can be identified. The distribution of the filtered features shows a strong correlation with the corresponding adaptive feature filtering methods. For instance, filtered features from the KNN method are mainly the features around the sensors, while the KFN-based filtered features are mainly the features that are away from the sensors. But based on the numerical evaluation mentioned above, looking for faraway features seems to have a better performance than looking for

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3390552

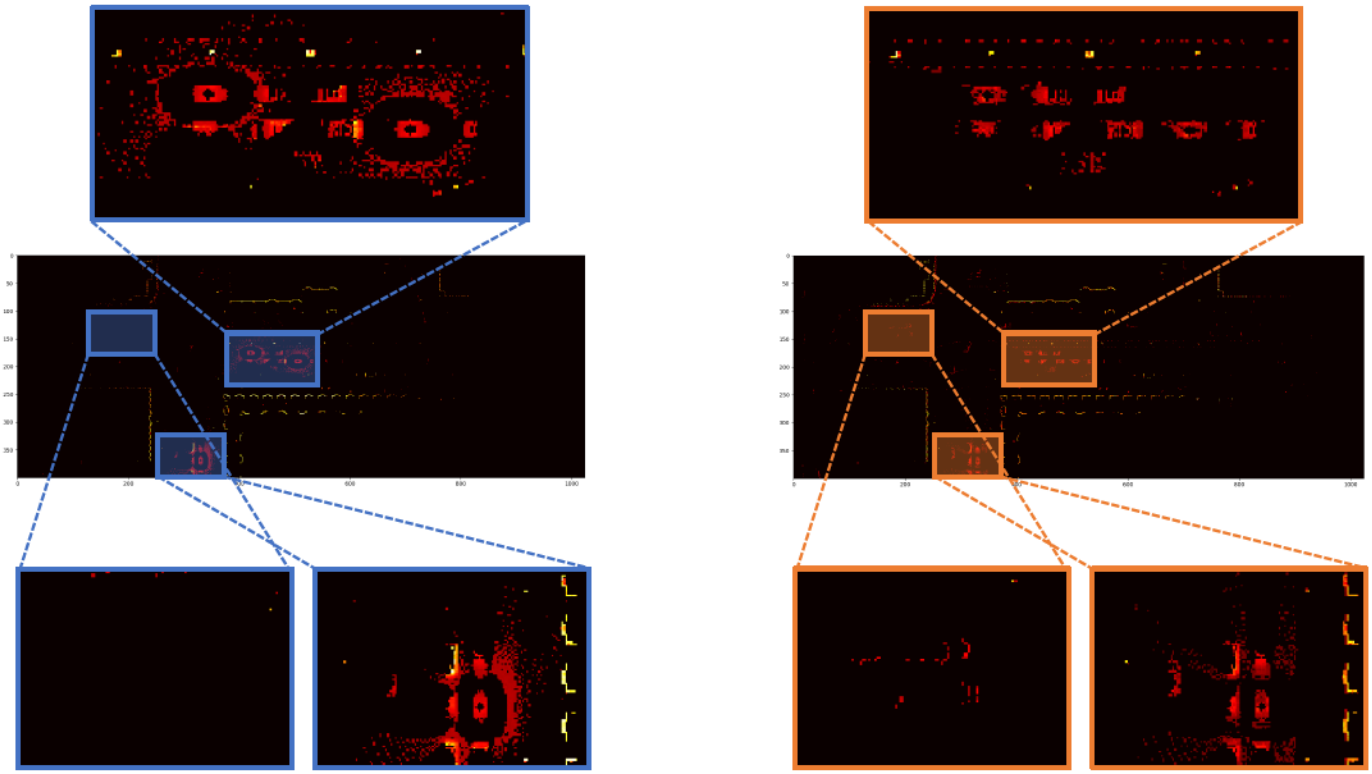


Fig. 7. Visual interpretation for filtered features from the PFE method (left w/ blue windows) and our PAE method (right w/ orange windows).

nearby features, which could be counter-intuitive. The filtered features from KRS are distributed like down-sampled point cloud data.

On the other hand, features filtered by KFV-based methods are more concentrated on the region of interest (RoI), the area where we want the features to be transmitted. Specifically, the features (points in Figure 6) of (d) KFV-PFE and (e) KFV-PAE are mainly the features generated from the PCD reflected by the objects we want to detect. Thus, by looking at the right column in Figure 6, KFV-based methods have evidently better detection results than the three baselines.

To interpret the performance improvement of our PAE compared with the PFE, we visualized the filtered feature map with zoom-in windows for KFV-PAE and KFV-PFE, as shown in Figure 7. These zoom-in windows demonstrate that the PAE method can better differentiate the features that are very close to the RoI features, and the features coming from the objects. It is clearly shown from the top zoom-in window and lower-right zoom-in window that PAE has much fewer features left near the objects. However, the PFE remains significantly more features on the ground that are valueless for the perception task and will waste the bandwidth for sending that worthless information. The effect of this can also be observed from the lower-left zoom-in window of Figure 7. Sending less number of invaluable features, our PAE method can transmit more valuable features to support better perception performance. For instance, in this area, PFE has no features left while our PAE has for RoI objects. In other words, our PAE can enable the ACP methods to filter features better by mainly focusing on the innate significance of the features rather than their spatial distributions.

3) Ablation Study: To further investigate the proposed model, ablation studies are conducted to analyze the time consumption and model complexity., which is shown in Table V. For time consumption, although the PAE consumes approximately 5x more processing time, the overall impact on the whole inference pipeline is trivial. The PFE-based pipeline has an inference speed of 13.84 Hz while PAE only slows the speed by roughly 1Hz. Considering that a common setting for the data rate of LiDAR sensors is 10 Hz [7], both methods can provide real-time inference capability while our method delivers higher mAP performance.

Table V also demonstrates the model complexity. Despite the parameter of PAE is significantly more than the PFE, PAE only increases the total network parameter by 0.31% (since the backbone contributes the major part of parameters). This reveals another advantage of using shallow features for cooperative perception rather than the features after backbone (e.g., dense features used in previous works [13], [15], [17], etc.) – low computation complexity for generating the sharing information and the whole cooperative perception system, which is discussed in depth in one of our previous work [30].

Based on the core design of the attention mechanism [33], multi-head attention enables the model to focus on different parts of the input sequence simultaneously. Each head can learn different aspects or relationships within the data, which improves the performance of the model to capture more complex patterns and dependencies. Thus, PAE involves the multi-head attention design. In this paper, compared with single-head-based PAE, the design of multi-head attention

TABLE V
ANALYSIS FOR TIME CONSUMPTION AND MODEL COMPLEXITY.

Model	Time consumption [ms]	Frame Rate [Hz]	Encoder Parameter	Total Parameter	mean Average Precision [1%-100%TR]
PFE	1.41	13.84	704	4,264,368	43.51
PAE (Ours)	8.17	12.75	7,218	4,277,396	47.94
Difference	+479.43%	-7.88%	+925.28%	+0.31%	+10.18%

improves the mAP performance by 2.83% with the same model complexity.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose *Adaptive Cooperative Perception*, a new framework that is designed to enable cooperative perception under more challenging and realistic conditions. A new feature encoder *Pillar Attention Encoder* (PAE) is proposed for enabling adaptive cooperative perception. An adaptive feature filtering method is proposed to adjust the amount of features for transmitting. Empowered by the specifically designed pillar attention mechanism, the PAE method can better distinguish the features with their significance of the perception task, thus allowing the feature filter to retain more valuable information within a limited feature size. Experiments demonstrate that our method significantly outperforms baseline methods under various testing cases. Specifically, under a full-range adaptive transmitting condition, our method can improve the average precision performance for 3D object detection by the range from 8.4% to 13.8% for cars and the range from 4.1% to 6.5% for pedestrians, when compared with the state-of-the-art feature encoder.

The main limitation of this study is the lack of consideration of the transmission delay and multi-node synchronization issue, which can be omitted in the simulation but are inevitable in real-world conditions. Additionally, the feature aggregator could be further improved by incorporating a channel attention module. Thus, taking account of these factors will be the main focus of our future work.

ACKNOWLEDGMENT

This research was funded by Toyota Motor North America, InfoTech Labs. The contents of this paper reflect the views of the authors only, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of Toyota Motor North America.

REFERENCES

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[2] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.

[3] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Infrastructure-based object detection and tracking for cooperative driving automation: A survey," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 1366–1373.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1907–1915.

[5] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[6] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Springer, 2022, pp. 1–18.

[7] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. A. Sisbot, K. Oguchi, and Z. Huang, "A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation," *ArXiv*, vol. abs/2208.10590, 2022.

[8] Y. Lou, A. Ghiasi, M. Jannat, S. Racha, D. Hale, W. Goforth, P. Bujanovic *et al.*, "Cooperative automation research: Carma proof-of-concept tsmo use case testing: Carma cooperative perception concept of operations," United States. Federal Highway Administration, Tech. Rep., 2022.

[9] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 514–524.

[10] H. Qiu, F. Ahmad, F. Bai, M. Gruteser, and R. Govindan, "Avr: Augmented vehicular reality," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 81–95.

[11] H. Qiu, P.-H. Huang, N. Asavisanu, X. Liu, K. Psounis, and R. Govindan, "Autocast: Scalable infrastructure-less cooperative perception for distributed collaborative driving," in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 128–141. [Online]. Available: https://doi.org/10.1145/3498361.3538925

[12] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 953–959.

[13] 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 *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, ser. SEC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 88–100. [Online]. Available: https://doi.org/10.1145/3318216.3363300

[14] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] 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 *European Conference on Computer Vision*. Springer, 2020, pp. 605–621.

[16] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. A. Sisbot, and K. Oguchi, "Pillargrid: Deep learning-based cooperative perception for 3d object detection from onboard-roadside lidar," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1743–1749.

[17] 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 *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 2022, pp. 107–124.

[18] R. Xu, Z. Tu, H. Xiang, W. Shao, B. Zhou, and J. Ma, "Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers," *arXiv preprint arXiv:2207.02202*, 2022.

[19] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary v2x technologies toward the internet of vehicles: Challenges and opportunities," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 308–323, 2020.

[20] R. Fukatsu and K. Sakaguchi, "Millimeter-wave v2v communications with cooperative perception for automated driving," in *2019 IEEE 89th vehicular technology conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.

[21] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, and O. Altintas, "Value-anticipating v2v communications for cooperative perception," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1947–1952.

[22] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau, and Y. Dupuis, "Survey on cooperative perception in an automotive context," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[23] R. Xu, H. Xiang, X. Xia, X. Han, J. Liu, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," *arXiv preprint arXiv:2109.07644*, 2021.

[24] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, p. 100057, 2021.

[25] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Cyber mobility mirror for enabling cooperative driving automation in mixed traffic: A co-simulation platform," *IEEE Intelligent Transportation Systems Magazine*, 2022.

[26] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[27] Y. Zhou and O. 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*, 2018, pp. 4490–4499.

[28] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2x-based perception in a high-level fusion architecture for cooperative perception systems," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 270–275.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[30] Z. Bai, G. Wu, M. J. Barth, Y. Liu, E. A. Sisbot, and K. Oguchi, "Vinet: Lightweight, scalable, and heterogeneous cooperative perception for 3d object detection," *Mechanical Systems and Signal Processing*, vol. 204, p. 110723, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327023006313

[31] Z. Meng, X. Xia, R. Xu, W. Liu, and J. Ma, "Hydro-3d: Hybrid object detection and tracking for cooperative perception using 3d lidar," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2023.

[32] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

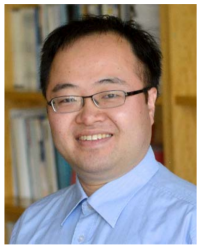
[34] N. Adaloglou, "Transformers in computer vision," *https://theaisummer.com/*, 2021.

[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[36] Z. Bai, "Carti dataset for cooperative perceptiion," Available: https://github.com/zwbai/CARTI_Dataset, 2022.

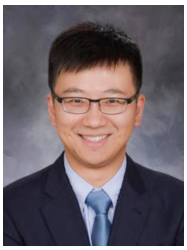
[37] Z. Bai, S. P. Nayak, X. Zhao, G. Wu, M. J. Barth, X. Qi, Y. Liu, E. A. Sisbot, and K. Oguchi, "Cyber mobility mirror: A deep learning-based real-world object perception platform using roadside lidar," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2023.

Zhengwei Bai (Student Member, IEEE) received the B.E. and M.S. degrees from Beijing Jiaotong University, Beijing, China, in 2017 and 2020, respectively. He is currently a Ph.D. student in electrical and computer engineering at the University of California at Riverside. His research focuses on object detection and tracking, cooperative perception, decision making, motion planning, and cooperative driving automation (CDA). He serves as a Review Editor in Urban Transportation Systems and Mobility.



Guoyuan Wu (Senior Member, IEEE) received his Ph.D. degree in mechanical engineering from the University of California, Berkeley in 2010. Currently, he holds an Associate Researcher and an Associate Adjunct Professor position at Bourns College of Engineering – Center for Environmental Research & Technology (CE–CERT) and Department of Electrical & Computer Engineering in the University of California at Riverside. development and evaluation of sustainable and intelligent transportation system (SITS) technologies, including connected and automated transportation systems (CATS), shared mobility, transportation electrification, optimization and control of vehicles, traffic simulation, and emissions measurement and modeling. Dr. Wu serves as Associate Editors for a few journals, including IEEE Transactions on Intelligent Transportation Systems, SAE International Journal of Connected and Automated Vehicles, and IEEE Open Journal of ITS. He is also a member of the Vehicle-Highway Automation Standing Committee (ACP30) of the Transportation Research Board (TRB), a board member of Chinese Institute of Engineers Southern California Chapter (CIE-SOCAL), and a member of Chinese Overseas Transportation Association (COTA). He is a recipient of Vincent Bendix Automotive Electronics Engineering Award.

Matthew J. Barth (Fellow, IEEE) received the M.S. and Ph.D degree in electrical and computer engineering from the University of California at Santa Barbara, in 1985 and 1990, respectively. He is currently the Yeager Families Professor with the College of Engineering, University of California at Riverside, USA. He is also serving as the Director for the Center for Environmental Research and Technology. His current research interests include ITS and the environment, transportation/emissions modeling, vehicle activity analysis, advanced navigation techniques, electric vehicle technology, and advanced sensing and control. Dr. Barth has been active in the IEEE Intelligent Transportation System Society for many years, serving as a Senior Editor for both the Transactions of ITS and the Transactions on Intelligent Vehicles. He served as the IEEE ITSS President for 2014 and 2015 and is currently the IEEE ITSS Vice President of Education.



Hang Qiu is an assistant professor of ECE and CSE at the University of California, Riverside. Previously, he was a postdoctoral scholar in the Platform Lab at Stanford University and a software engineer at Waymo LLC. He received his Ph.D. from the Department of Electrical and Computer Engineering at the University of Southern California and his Bachelor's degree from Shanghai Jiao Tong University. His research focus is on networked cyber-physical systems with edge ML. His work draws upon theories and methods from machine learning, wireless networking, computer vision, and robotics to build robust and cooperative intelligence in edge autonomous systems. He is a recipient of ACM Mobisys Best Paper Runner-up Award, a recipient of MLSys Outstanding Paper Award, a Qualcomm Innovation Fellowship Finalist, and an Outstanding Winner of COMAP ICM.

Yongkang Liu received the Ph.D. and M.S. degrees in electrical engineering from the University of Texas at Dallas in 2021 and 2017, respectively. He is currently a Research Engineer at Toyota Motor North America, InfoTech Labs. His current research interests are focused on in-vehicle systems and advancements in intelligent vehicle technologies.

Emrah Akin Sisbot (Member, IEEE) received the Ph.D. degree in robotics and artificial intelligence from Paul Sabatier University, Toulouse, France in 2008. He was a Postdoctoral Research Fellow at LAAS-CNRS, Toulouse, France, and at the University of Washington, Seattle. He is currently a Principal Engineer with Toyota Motor North America, InfoTech Labs, Mountain View, CA. His current research interests include real-time intelligent systems, robotics, and human-machine interaction.

Kentaro Oguchi received the M.S. degree in computer science from Nagoya University. He is currently a Director at Toyota Motor North America, InfoTech Labs. Oguchi's team is responsible for creating intelligent connected vehicle architecture that takes advantage of novel AI technologies to provide real-time services to connected vehicles for smoother and efficient traffic, intelligent dynamic parking navigation and vehicle guidance to avoid risks from anomalous drivers. His team also creates technologies to form a vehicular cloud using Vehicle-to-Everything technologies. Prior, he worked as a senior researcher at Toyota Central R&D Labs in Japan.