2023 IEEE 26th International Conference on
Intelligent Transportation Systems (ITSC)
24-28 September 2023. Bilbao, Bizkaia, Spain

Dynamic Feature Sharing for Cooperative Perception from Point Clouds

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

2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3503-9946-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITSC57777.2023.10422242

***Abstract*—Perceiving the surrounding environment is critical to enable cooperative driving automation, which is regarded as a transformative solution to improving our transportation system. Cooperative perception, by cooperating information from spatially separated nodes, can innately unlock the bottleneck caused by physical occlusions and has become an important research topic. Although cooperative perception aims to resolve practical problems, most of the current research work is designed based on the default assumption that the communication capacities of collaborated perception entities are identical. In this work, we introduce a fundamental approach – *Dynamic Feature Sharing (DFS)* – for cooperative perception from a more pragmatic context. Specifically, a DFS-based cooperative perception framework is designed to dynamically reduce the feature data required for sharing among the cooperating entities. Convolution-based Priority Filtering (CPF) is proposed to enable DFS under different communication constraints (e.g., bandwidth) by filtering the feature data according to the designed priority values. Zero-shot experiments demonstrate that the proposed CPF method can improve cooperative perception performance by approximately +22% under a dynamic communication-capacity condition and up to +130% when the communication bandwidth is reduced by 90%.**

***Index Terms*—Cooperative Perception, Deature Sharing, Object Detection, Point Clouds, Vehicle-Infrastructure Cooperation**

I. Introduction

One of the central challenges in automated driving is to enable vehicles to comprehend their surrounding environments so that subsequent tasks (e.g., decision-making, planning, control) can be performed appropriately [1]. This requires vehicles to sense the environment with a wide field of view (FOV) and under various environmental conditions, such as lighting issues. To achieve this, current automated driving technologies tend to equip vehicles with more sensors from different modalities for enhancing the sensing ability around the vehicle [2]. Concurrently, different types of datasets from various sensor configurations have been collected and labeled to train the increasingly complicated onboard perception systems [3].

However, no matter how many sensors are utilized on the vehicle, its perception capability will still be limited by occlusions, especially in a complex driving scenario (e.g., at urban intersections surrounded by high-rise buildings). This limitation is mainly caused by sensing the environment from the sensor(s) on a single entity. Recent research has shown that cooperation between sensing systems with multiple spatially-separated locations can inherently overcome these limitations. Cooperative perception has quickly become an emerging solution to unlock the bottleneck of environmental comprehension for automated driving [4]–[8].

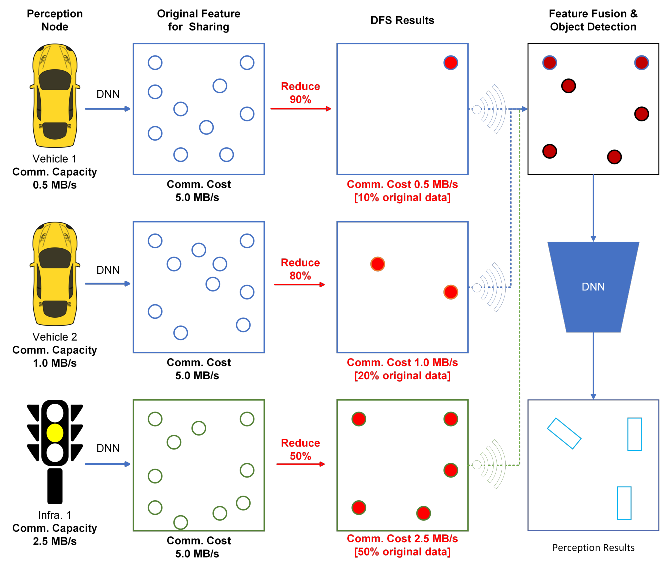

Fig. 1: **Conceptual Overview for Dynamic Feature Sharing.** Cooperative perception entities (e.g., vehicle nodes and infrastructure nodes) have different communication capacities. To be able to transmit the data based on their own available bandwidths, a Dynamic Feature Sharing module is designed to dynamically reduce the amount of shared data based on the original features, without much compromising the perception performance.

Data sharing is a fundamental component of cooperative perception. For instance, in the real world, automated vehicles can increase their perception ranges by receiving the perception information from other sensing entities, such as other connected and automated vehicles [9], or smart infrastructure [10] via vehicle-to-vehicle (V2V) communications or vehicle-to-infrastructure (V2I) communications [11]. Different types of data can be shared in the context of cooperative

Zhengwei Bai, Guoyuan Wu, and Matthew J. Barth are with the Department of Electrical and Computer Engineering, the University of California at Riverside, Riverside, CA 92507 USA (e-mail: zbai012@ucr.edu).

Yongkang Liu, Emrah Akin Sisbot, and Kentaro Oguchi are with Toyota Motor North America, InfoTech Labs, Mountain View, CA 94043, USA.

perception, including raw sensor data, intermediate feature data, and object-list data [12]. Recent cooperative perception methods demonstrated remarkable performance improvements using intermediate feature-based fusion methods, and different feature data were designed for sharing, such as encoder features [6] and backbone features [5].

It is important to note that a commonly-used assumption for those efforts is that the feature-sharing capabilities for the perception nodes are typically identical – sharing the features with the same spatial shape to allow them to be fused together [4]–[7]. This requires all nodes to be able to transmit the same amount of data, which will be problematic when the actual perception nodes have different communication capabilities.

To enable cooperative perception in a more realistic environment, we propose a novel methodology called *Dynamic Feature Sharing* which aims to dynamically reduce the amount of data transmitted to cooperative perception systems without significantly compromising the perception performance. We propose a dynamic feature-sharing framework that can be applied to different types of shared data. Specifically, for intermediate feature sharing, we propose a novel *Convolution-based Priority Filtering* (CPF) method that outperforms other baselines under different communication bandwidth limitations. To evaluate the proposed framework and method, we develop a data acquisition platform to collect training and testing data which includes both vehicles and infrastructure. The main contribution of this work can be summarized below:

- We propose a new fundamental module for the cooperative perception framework called Dynamic Feature Sharing.
- We develop a novel feature filtering method called Convolution Priority Filtering to enable cooperative perception among multiple nodes with dynamic communication capacities.
- We design a specific testing environment for model training and evaluations.
- We investigate 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. 3D Object Detection

Due to a significant amount of interest in automated vehicle R&D over the past decade, onboard object perception techniques have made considerable progress in recent years. There have been various computer vision algorithms proposed for various sensors used (such as monocular/stereo cameras, LiDAR) or perception tasks (e.g., traffic sign recognition, lane detection, road user detection) [13]. The use of convolutional neural networks (CNNs) for camera-based solutions has been widely investigated in recent years, and they also inspire the design of perception pipelines for analyzing point cloud data (PCD) from onboard LiDAR sensors [14].

To support more efficient automated driving in a mixed traffic environment, infrastructure-based surveillance systems can provide additional object-level information to target road users beyond traditional data collection (e.g., volume, point speed) based on loop detectors and radar [3]. Zhao et al. designed a bottom-up pipeline for infrastructure-based object perception using traditional model-based methods [15]. Utilizing data-driven models, Bai et al. demonstrated the concept of *Cyber Mobility Mirror* where a roadside LiDAR was used to enable real-time 3D object perception at an intersection [10].

B. Cooperative Perception

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) [12]. Xu et al. [7] proposed a V2X-based cooperative perception (CP) method considering the heterogeneity of vehicle and infrastructure nodes and multi-scale receptive fields. Lou et al. [16] conducted the Proof-of-Concept of CP in the real world by applying V2X to enable entities to share their sensing results and the program demonstrated the CP system can significantly improve the perception capability of the involved entities.

C. Feature Sharing for Cooperative Perception

Feature data shared in cooperative perception varies according to the data fusion schemes, including early, intermediate, and late fusion [12]. Considering the limited length and main scope of this paper, only intermediate features are discussed below. Feature data at the end of the backbone network, comprised of convolutional layers [4], has attracted significant interest as it contains highly extracted information to benefit cooperation. Another strategy is to share the feature data after simple feature encoders which usually have much smaller neural network structures than the backbone [5], [17].

Since intermediate features usually hold large amounts of information, data compression techniques are often applied to reduce the amount of data to adapt to communication capacities. Examples include CNN-based channel-wise compression [7] or other dedicated Encoder-Decoder methods [5]. However, none of these data compression methods can be utilized to enable cooperative perception for nodes with different communication capacities by sharing different amounts of data for different perception nodes. For example, features with different channels are not able to be fused directly, and it is impractical to apply all different decoders to decompress features from different encoders.

III. Methodology

A. DFS-based Cooperative Perception Framework

We formulate the cooperative perception into five key components and a proposed DFS-based cooperative perception framework is illustrated in Fig. 2. These components include:

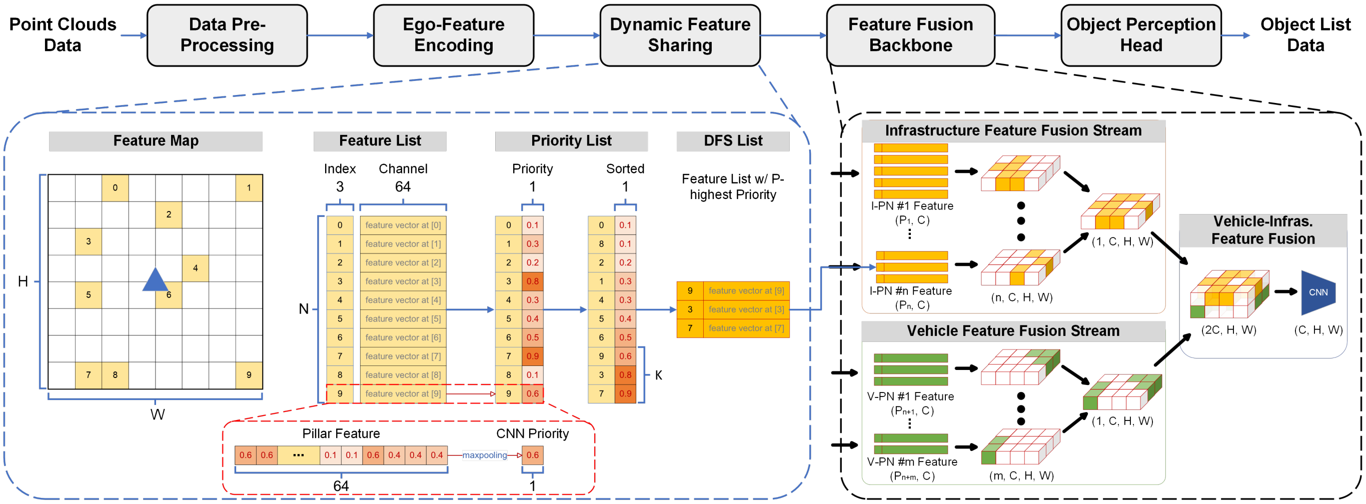

Fig. 2: Systematic diagram for the proposed DFS-based cooperative perception framework.

- **Data Pre-Processing**: This component aims to apply proper transformation and geo-fencing to prepare the raw sensor data for processing.
- **Ego-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.
- **Dynamic Feature Sharing**: This component aims to dynamically reduce the amount of data used for sharing.
- **Feature Fusion Backbone**: 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 Head**: This component aims to generate detailed perception results based on specific tasks, such as object detection, tracking, classification, motion detection, etc.

In the following sections, we discuss these components in more detail.

B. Data Pre-Processing

To align the spatial location of different nodes, a global referencing coordinate (GRC) has been designed for raw data preprocessing. There are two significant advantages of applying a GRC for data transformation: 1) by aligning the data in the early stages, the spatial mismatching problems [4] are circumvented; and 2) in a holistic cooperative perception system [17], every perception node will benefit from each other, and there is no need to transform their data multiple times for different cooperating nodes.

In this study, we consider point cloud data as an example. It will be transformed into the GRC for spatial alignment for all perception nodes, which is defined below.

$$\mathcal{P}^{E\to G} = \begin{bmatrix}\mathcal{R}_X & 0\\ 0 & 1\end{bmatrix}\cdot\begin{bmatrix}\mathcal{R}_Y & 0\\ 0 & 1\end{bmatrix}\cdot\begin{bmatrix}\mathcal{R}_Z & 0\\ 0 & 1\end{bmatrix}\cdot\mathcal{P}^{E}+\mathcal{T}^{E\to G} \quad (1)$$

where $\mathcal{R}_X$, $\mathcal{R}_Y$, $\mathcal{R}_Z$, and $\mathcal{T}^{E\to G}$ represent the rotation matrix along $x-$axis, $y-$axis, $z-$axis, and the translation matrix from ego-coordinate to GRC, respectively. $\mathcal{P}^S$ and $\mathcal{P}^{S\to G}$ represent the raw data before and after the transformation.

C. Ego-Feature Encoding

For encoding the point cloud data, we voxelize it along the x and y axis, which results in a 2D pillar map. Furthermore, we apply the lightweight feature extractor [17] to extract hidden features for every non-empty pillar from heterogeneous perception nodes. Specifically, for each pillar, a $D-$dimensional feature vector V_p is generated as defined below:

$$V_p = \{[x, y, z, i, x_c, y_c, z_c, x_p, y_p]_i\}_{i=1}^{N}, p = 1, \ldots, P \quad (2)$$

where $x_c, y_c, z_c, x_p, y_p, N$, and P represent the distance of each point to the arithmetic center of all points in the $p-$th pillar (the c subscript) and the geometrical center of the pillar (the p subscript), the number of points in the pillar and the number of pillars, respectively. Then, hidden features of V_p are encoded by the following processes.

$$\mathcal{H}_e^{(j)} = \max_{1\le i\le N}(Dense_e(\mathcal{F}_e^{(j)})), j = 1, ..., n \quad (3)$$

$$\mathcal{F}_e^{(j)} = \{\{V_p^{(i)}\}_{i=1}^{P_j}\}_{j=1}^{n} \quad (4)$$

where $\mathcal{H}_e^{(j)}$ presents the feature output from the encoder which has the shape of (P_j, C). Especially, $e \in$ [veh, inf] represents the heterogeneous perception nodes of vehicles and infrastructure. Two decoupled MLP networks are designed as $Dense_e(\cdot)$ to extract V_p from $D-$dimension to $C-$dimension. $\mathcal{F}_e^{(j)}$ presents the feature data from $j-$th infrastructure/vehicle node.

D. Dynamic Feature Sharing

To adjust the difference in communication capabilities of various perception nodes, a dynamic feature sharing (DFS) methodology is proposed to filter the feature data based on certain criteria. As an example shown in Fig 1, if the encoded feature data requires a 5.0 MBps for real-time transmission while the actual communication capability of the node is only 0.5 MBps, a natural way to reduce the bandwidth is to select only 10% of the original data for sharing. However, the method to select the shared features while maintaining perception accuracy is still an open-challenging problem. In this study, we propose a feature-filtering method named *Convolution-based Priority First* (CPF) and several baselines according to the heuristic inspirations which are discussed later in Section IV-A4.

The core concept of CPF is to sift the features according to the convolution values in each pillar feature. A diagram is shown in Fig. 2 to illustrate the detailed process of CPF. There are three main steps for the CPF method:

- **Feature List**: First, we index the pillar features in the feature map generated from Section III-C. In the actual implementation, a natural index is designed based on the 3D spatial location of the pillar. The spatial shape of the Feature List is $(N, 3+64)$.
- **Priority List**: Next, we aggregate the convolution values in each pillar feature by applying *maxpooling* operation – getting the most magnificent convolutional representation, which is named *convolution-based priority* in this paper. The spatial shape of the Priority List is $(N, 3+1)$.
- **DFS List**: Lastly, we sort the priority list and then truncate the list based on the threshold $\mathcal{K}$ which is generated according to the specific communication capability. The spatial shape of the DFS List is $(\mathcal{K}, 3+64)$.

By following the DFS framework (not necessarily the CPF method), it is noted that the threshold $\mathcal{K}$ can be any unsigned integer value, which not only allows different nodes to generate different amounts of feature data for sharing but also enables the node to dynamically generate the feature data at different time according to the varying communication conditions.

E. Feature Fusion Backbone

To enable intermediate feature fusion under dynamic, scalable, and heterogeneous conditions, a two-stream neural network is adopted to fuse pillar features from scalable perception nodes of vehicles and infrastructure [17]. As shown in Fig 2, the feature fusion backbone is composed of three parts: 1) the infrastructure feature fusion stream which fuses the feature data from infrastructure nodes by reprojecting the feature list back to a 2D feature map with the shape of (n, C, H, W) and then aggregating those features via *maxpooling* along each spatial location ending up with a feature map with the shape of $(1, C, H, W)$; 2) the vehicle feature fusion stream which acts similarly to the infrastructure one to generate a $(1, C, H, W)$ feature map representing the features aggregated from all vehicle nodes; and 3) the vehicle-infrastructure feature fusion module which applies the concatenation first to form a $(2C, H, W)$ feature map and then passes the feature into a dense CNN network for further feature extraction.

F. Object Perception Head

In the scope of this study, an anchor-based 3D object detection head [14] is applied to generate 3D bounding boxes with classification and heading direction information. Intersection over Union (IoU) is executed to correspond the prior bounding boxes with the ground truth.

IV. Experiments

A. Experimental Setup

1) Dataset Acquisition: The "*CARTI*" (i.e., **CAR**la-ki**TtI**) platform [18] is applied for collecting the LiDAR sensor data and ground truth labels for CP system model training and testing. Specifically, two infrastructure nodes and three vehicle nodes are deployed for data collection. A total of $9,179$ frames of 3D point clouds are collected ($45,895$ samples if counting perception nodes in all frames), including $3,059$ frames for training, $3,060$ frames for validation, and $3,060$ frames for testing.

TABLE I: Parameter Configuration in the CARTI Platform

Sensor Specification	Setting [onboard/roadside]
LiDAR Channels	64
LiDAR Height	$1.74/4.74m$
LiDAR Sensing Range	$100.0m$
LiDAR Rotation Frequency	10.0Hz
Upper FOV	$+22.5\ /\ +0$
Lower FOV	$-22.5\ /\ -22.5$
Noise for Points Per Beam	$0.01m$
Missing Reflection Rate	45%
Intensity Dropoff Range	$(0, 0.8]$

The specification of sensors applied is shown in Table I and two different LiDAR settings are used based on our previous work in real-world [10]. To make the simulated point cloud data closer to the realistic conditions, we configure 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.

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 $[0.23m, 0.23m, 6.00m]$ and the maximum number of voxels per node $\mathcal{P}$ is set as $15,000$. Specifically, during the training stage, the threshold $\mathcal{K}$ for the number of pillar features per node is randomly varying from the range $\mathcal{K} \in [1,500, 15,000]$ (nodes in one frame are also assigned with different $\mathcal{K}$ to emulate a fully dynamic environment).

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

The detection performance is measured with Average Precision (AP) at Intersection-over-Union (IoU) thresholds of 0.7 for cars. Furthermore, based on the Minimum number of

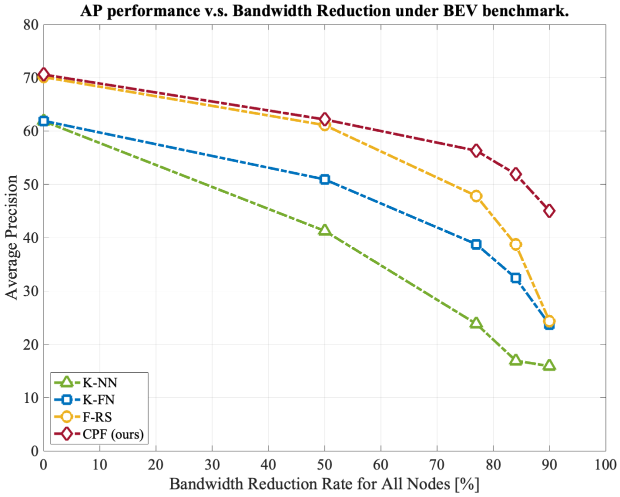

Fig. 3: BEV AP improv. under heterogeneous data-sharing conditions.

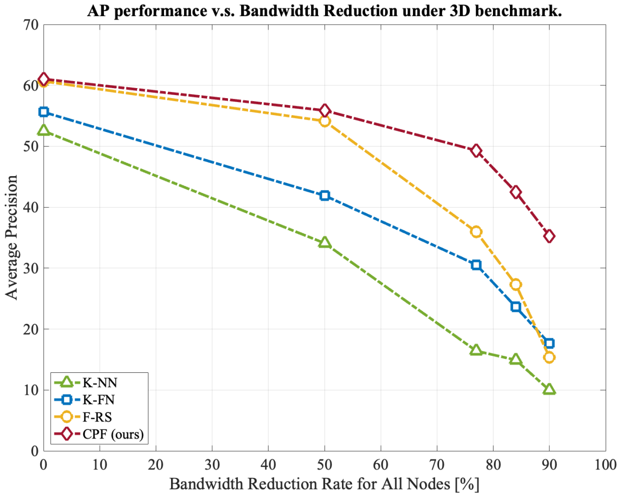

Fig. 4: 3D AP improv. under heterogeneous data-sharing conditions.

Points (MP) reflected by the ground target, each evaluation class is further divided into three categories: Easy (MP$\geq$10), Medium (MP$\geq$5), and Hard (MP$\geq$1), respectively, to investigate the performance of CP methods on different difficulty levels.

For dynamic communication constraints, we evaluated the performance under $\mathcal{K} \in [15000, 7500, 3500, 2500, 1500]$ for vehicle-based perception nodes and infrastructure-based perception nodes representing bandwidth using 100%, 50%, 23%, 16%, and 10%, respectively. It is noted that the vehicle nodes and infrastructure nodes can have different $\mathcal{K}$ to mimic the dynamic real-world environment for communication. Furthermore, those thresholds are selected to balance the representation and complexity of the experiments, and the model is trained for any bandwidth usage and its performance is NOT fine-tuned by any of those thresholds designed above.

4) Comparing Baselines: We considered several heuristic feature filtering methods including:

- K-Nearest Neighbor (K-NN): 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 (K-FN): A converse method w.r.t. the K-NN.
- K-Random Sampling (K-RS): Randomly sampling out K features.

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

$$\mathcal{D}_m = \mid x_p - x_s \mid + \mid y_p - y_s \mid \tag{5}$$

where $\mathcal{D}_m$ is the Manhattan distance between the feature location (x_p, y_p) and the sensor location (x_s, y_s).

TABLE II: Average precision (AP) performance for DFS methods under different benchmarks.

Conditions	Methods	BEV			3D		
		Easy	Medium	Hard	Easy	Medium	Hard
100% Sharing	K-NN	61.87	61.17	57.14	52.52	52.34	52.21
	K-FN	61.92	61.76	61.59	55.62	55.55	52.09
	K-RS	70.31	69.17	66.57	60.67	60.47	60.23
	CPF (Ours)	**70.60**	**69.81**	**66.82**	**61.04**	**60.83**	**60.61**
50% Sharing	K-NN	41.28	37.92	37.89	34.09	33.89	33.76
	K-FN	50.93	50.82	50.59	41.90	41.86	41.73
	K-RS	61.10	60.91	60.72	54.14	50.67	50.50
	CPF (Ours)	**62.17**	**61.96**	**61.78**	**55.88**	**55.72**	**52.16**
23% Sharing	K-NN	23.82	20.49	18.94	16.40	16.29	16.22
	K-FN	38.76	38.99	38.18	30.56	30.72	30.62
	K-RS	47.82	46.50	44.72	35.94	34.33	34.15
	CPF (Ours)	**56.31**	**56.14**	**52.57**	**49.30**	**46.16**	**46.02**
17% Sharing	K-NN	16.92	16.87	16.83	14.92	14.78	14.69
	K-FN	32.41	32.66	32.65	23.63	23.85	23.85
	K-RS	38.71	37.93	37.26	27.33	25.23	25.11
	CPF (Ours)	**51.90**	**51.43**	**51.08**	**42.50**	**42.12**	**41.88**
10% Sharing	K-NN	15.89	15.75	15.64	9.96	9.96	9.96
	K-FN	23.72	24.07	24.11	17.61	17.83	17.87
	K-RS	24.37	24.13	22.33	15.38	14.24	14.16
	CPF (Ours)	**45.20**	**42.25**	**42.02**	**35.26**	**34.98**	**32.68**
Dynamic Sharing	K-NN	34.94	34.74	34.57	28.85	27.69	27.62
	K-FN	46.03	46.10	43.26	38.89	36.98	36.55
	K-RS	51.66	51.40	51.17	41.99	41.72	41.52
	CPF (Ours)	**60.46**	**59.23**	**56.84**	**51.26**	**50.78**	**50.36**
	Improv. (%)	*17.03*	*15.23*	*11.08*	*22.08*	*21.72*	*21.29*

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 illustrate the visualized performance and interpretations of the methods.

1) Quantitative Results and Analysis: Two different benchmarks are applied based on the CARTI dataset – BEV benchmark and 3D benchmark representing the object detection w.r.t. 2D bird-eye-view ground truth and 3D ground truth, respectively. Furthermore, three types of difficulty levels are applied for both benchmarks w.r.t. the minimum points re-

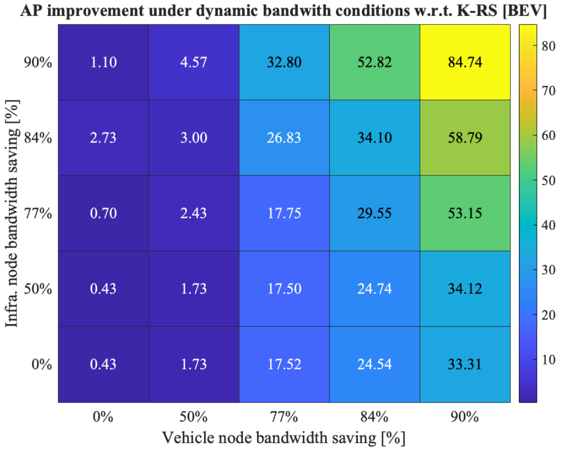

Fig. 5: BEV AP improv. under heterogeneous data-sharing conditions.

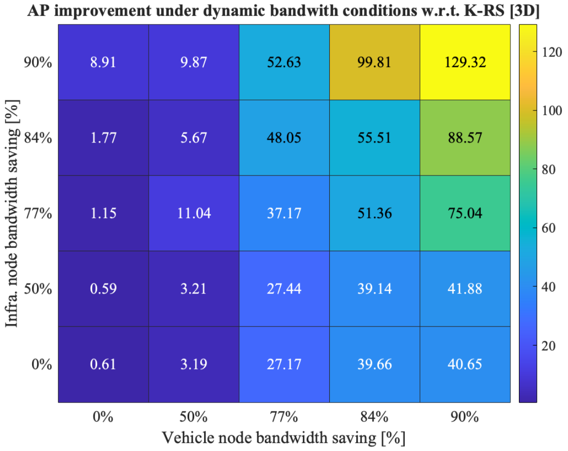

Fig. 6: 3D AP improv. under heterogeneous data-sharing conditions.

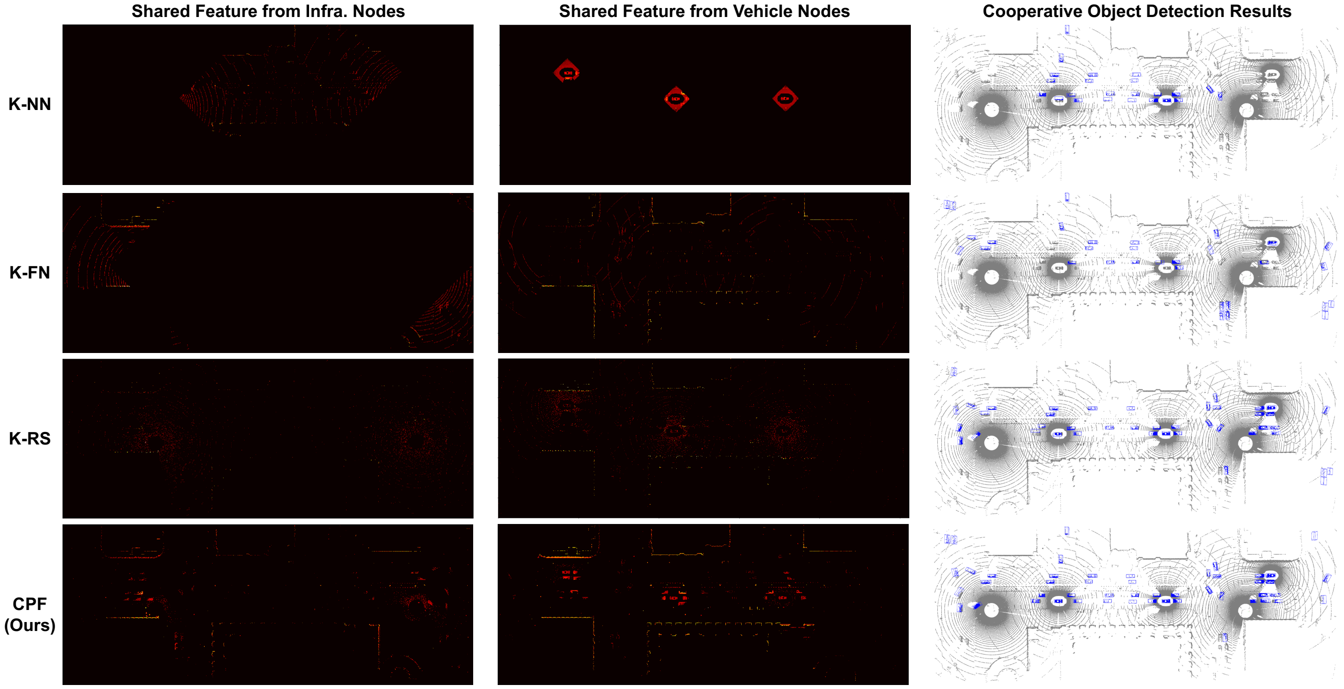
Shared Feature from Infra. Nodes
Shared Feature from Vehicle Nodes
Cooperative Object Detection Results
K-NN
K-FN
K-RS
CPF (Ours)

Fig. 7: **Visual interpretation for the feature data and object detection results (better w/ zoom in).** The first two columns show the visualization of the shared data from infrastructure nodes and vehicle nodes (combining nodes to visualize for space saving). The last column shows the cooperative 3D object detection results from a top-down view.

flected by the ground truth objects (i.e., minimum 10 points, 5 points, 1 point for *Easy*, *Medium* and *Hard*, respectively.)

Table II shows the evaluation results of different DFS methods under various data-sharing conditions. Specifically, for homogeneous data-sharing conditions (i.e., vehicle nodes and infrastructure nodes have the same sharing capabilities), our method, CPF, achieves the best performance among all the testing tracks. Furthermore, the CPF method, compared with other baselines, is more resilient to conditions with reducing sharing limitations, which is demonstrated in Fig. 3 and Fig. 4. Under the conditions of reducing 0% to 50% sharing data, CPF performs slightly better than K-RS while both of these two methods outperform K-NN and K-FN with noticeable margins. Nevertheless, when the available data for sharing keeps shrinking, CPF will significantly outperform all other methods, including K-RS.

For a more realistic scenario in which a different node has its data-sharing capability, CPF can still overtake the best baseline (K-RS) by improving 15.23% and 21.72% AP for BEV and 3D object detection, respectively (under *Medium* difficulty). Furthermore, Fig. 5 and Fig. 6 illustrate the AP improvements of CPF under different data-sharing conditions, compared with the best baseline, K-RS. The relative improvement increases while the amount of data used for sharing decreases. Under the 90% bandwidth reduction condition, CPF can outperform other methods in terms of the AP performance by as much as 84.74% and 129.32%, respectively.

2) Qualitative Results and Analysis: To further investigate the performance of DFS methods, we visualize the feature data from different nodes and the cooperative 3D object detection results, as shown in Fig. 7. To interpret the feature data for sharing, we present the CNN-based priority values within each pillar feature by applying a Heat Map in which the feature with higher priority is shown in a brighter point.

Fig. 7 illustrates the patterns of the feature data after their designed filtering methodologies. For the first row, the K-NN method clearly keeps the feature data around each node, while the K-FN method, shown in the second row of Fig. 7, remains the features that are located far from the node. For the K-RS, the feature map demonstrates its random sampling process and the feature data for sharing looks like down-sampled point cloud data.

For the last row in Fig. 7, we can find that the CPF method can keep the feature data based on their significance to the perception task, regardless of their spatial locations. For instance, most of the feature data passed out from CPF comes from the ground truth objects. By sharing the feature data that essentially comes from ground truth objects, the CPF method will naturally end up with a significant improvement in object detection as shown in the last column in Fig. 7.

V. Conclusions and Future Work

In this paper, we propose a fundamental module for cooperative perception which is *dynamic feature sharing* considering the variety of communication capabilities from different perception nodes. A cooperative perception framework is proposed by considering the dynamic feature sharing and a novel method, convolution-based priority filtering (CPF) is proposed, which can provide state-of-the-art performance for object detection under dynamic data sharing conditions. Specifically, with zero-shot testing, the CPF method can improve the Average Precision for 3D object detection by 21.72% under a fully dynamic feature-sharing condition and 129.32% under a 90% bandwidth-saving condition.

The main limitation of the current work is the lack of large-scale, real-world evaluation, which is also missing in most of the related cooperative perception work. Hence, benchmarking on the real-world dataset, involving real-world communication configurations, and resolving the challenge of spatiotemporal synchronization issues will significantly stimulate research in this field.

Acknowledgments

This research was funded by Toyota Motor North America, InfoTech Labs. The contents of this paper reflect the views of the authors, 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] 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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[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] 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.

[11] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault, “On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 419–10 432, 2017.

[12] 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.

[13] E. Marti, A. Miguel, F. Garcia, and J. Perez, “A review of sensor technologies for perception in automated driving,” *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[14] 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.

[15] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu, “Detection and tracking of pedestrians and vehicles using roadside lidar sensors,” *Transportation research part C: emerging technologies*, vol. 100, pp. 68–87, 2019.

[16] 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.

[17] 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,” *arXiv preprint arXiv:2212.07060*, 2022.

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