IEEE Open Journal of Vehicular Technology

Received 16 October, 2024; revised 30 December, 2024; accepted XX Month, XXXX; Date of publication XX Month, XXXX.

Digital Object Identifier 10.1109/OJVT.2024.XXXXXXX

Real-time Heterogeneous Collaborative Perception in Edge-enabled Vehicular Environments

SAMUEL THORNTON* (Student Member, IEEE), NITHIN SANTHANAM†, RAJEEV CHHAJER† AND SUJIT DEY* (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093, USA
[2]Honda Research Institute USA Inc., Columbus, OH 43212, USA

CORRESPONDING AUTHOR: SAMUEL THORNTON (e-mail: sjthornt@ucsd.edu).

This work was supported in part by the Honda Research Institute and in part by the UCSD Center for Wireless Communications (CWC)

ABSTRACT Vehicular sensing has reached new heights due to advances in external perception systems enabled by the increasing number and type of sensors in vehicles, as well as the availability of on-board computing. These changes have led to improvements in driver safety and have also created a highly heterogeneous environment of vehicles on the road today in terms of sensing and computing. Using collaborative perception, the information obtained by vehicles with sensing capabilities can be expanded and improved, and older vehicles that lack external sensors and computing capabilities can be informed of potential hazards, opening the opportunity to improve traffic efficiency and safety on the roads. However, achieving real-time collaborative perception is a difficult task due to the dynamic availability of vehicular sensing and computing and the highly variable nature of vehicular communications. To address these challenges, we propose a Heterogeneous Adaptive Collaborative Perception (HAdCoP) framework which utilizes a Context-aware Latency Prediction Network (CaLPeN) to intelligently select which vehicles should transmit their sensor data, the specific individual and collaborative perception tasks, and the amount of computational offloading that should be utilized given information about the current state of the environment. Additionally, we propose an Adaptive Perception Frequency (APF) model to determine the optimal end-to-end latency requirement according to the current state of the environment. The proposed CaLPeN model outperforms six implemented comparison models in terms of effective mean average precision (EmAP), beating the next best model's performance by 5.5% on average when tested on the OPV2V perception dataset using two different combinations of wireless communication conditions and vehicular sensor/computing distributions.

INDEX TERMS Connected Vehicles, Collaborative Perception, Edge Computing, Machine Learning.

I. Introduction

Advances in vehicular sensing and perception have paved the way for improved safety systems for users and have opened many new possibilities for further improvements in intelligent transportation systems (ITS). Collaborative vehicular perception is an emerging topic that is showing the potential to produce new heights in vehicular perception performance. Collaborative perception models use sensor data from multiple vehicles as well as sensors on road infrastructure in order to provide a larger perception area than can be achieved by a single vehicle, as well as reducing the impact of environmental hazards such as poor weather and occlusions. From a computational perspective, collaborative perception can also reduce the aggregate computing requirement by utilizing mobile edge computing, which can allow all vehicles to benefit, even if they lack on-board computing. However, achieving collaborative perception in ITS is a challenge considering the data exchange that must occur between vehicles and the dynamic wireless communication channels encountered by moving vehicles, as well as the heterogeneous nature of modern vehicular sensing.

As vehicular perception technology evolves, so has the amount of sensing and computing available in vehicles [1]. As such, there exists a large distribution of sensor and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

computing configurations that appear on the roads today and this distribution will only get larger as time goes on. This heterogeneity among vehicular sensing creates a unique opportunity to utilize collaborative perception in order to continually improve the achievable perception, particularly in high traffic areas where accidents are most likely to occur; as new vehicles with advanced levels of computing and sensing are introduced, the older vehicles will be able to benefit by collaborating with these new vehicles. The emergence of edge computing, which provides computational resources at the edge of the communication network such as base stations on cell towers or mobile access points, provides a potential infrastructure for facilitating data exchange between vehicles and supplementing computational needs. Furthermore, broadcasting the collaborative perception results from these edge nodes allows any vehicle with networking capabilities to benefit with minimal additional communication overhead [2]. This approach provides benefits for all vehicles in terms of perception accuracy while also promoting a future direction towards equity in mobility with the inclusion of pathways for legacy vehicles to gain information from state-of-the-art vehicles.

While collaborative perception can produce numerous benefits for vehicular environments, there are also many potential challenges in its real-world implementation. Some of these challenges include integrating the current street infrastructure while adding additional networking and computing resources needed for edge assisted vehicular collaborative perception, developing communication protocols to standardize the collaborative perception process among all OEMs, and creating security measures to ensure that individual privacy is preserved and adversarial attacks are avoided. For the purpose of this paper, we will focus on the challenge of optimizing the end-to-end collaborative perception process for 3D object detection, both in terms of perception accuracy and execution latency, in edge-based vehicular environments with varying distributions of vehicle types and network conditions. Mobile edge computing can provide an avenue for the vehicular data exchange and computational resources needed, but ensuring that the end-to-end collaborative sensor fusion process, including the data transmissions, completes in the required latency is a considerable challenge considering the highly dynamic nature of vehicular environments. In this work, we explore how to maximize collaborative sensor fusion accuracy in a heterogeneous sensing environment while ensuring that latency requirements are achieved. More specifically, the contributions of this work is as follows:

- We present a **H**eterogeneous **Ad**aptive **Co**llaborative **P**erception (HAdCoP) framework for enabling real-time collaboration between vehicles with diverse sensing capabilities in vehicular edge environments.
- We propose an **A**daptive **P**erception **F**requency (APF) model for dynamically adjusting the latency requirement to increase reliability, minimize perception delays and reduce idle computing time.
- We have created a neural network-based **C**ontext-**a**ware **L**atency **Pre**diction **N**etwork (CaLPeN) which predicts the optimal set of collaborative perception actions that maximize perception accuracy while ensuring the latency requirement is met.

The remainder of this paper will be organized as follows: Section II will be a review of related work in the area of individual and collaborative vehicular perception, as well as vehicular edge computing. In Section III, an overview of the heterogeneous sensing environment with vehicular edge computing is presented, as well as a discussion of the different trade-offs and our problem formulation. In Section IV, we discuss our HAdCoP framework and the three submodels that it consists of as well as our dataset selection process. In Section V, we present the chosen collaborative sensor fusion model, the associated feature extraction models, and the hardware that was used for testing as well as describing how we are evaluating the collaborative sensor fusion performance before presenting an action decision model performance comparison on four associated sets of testing data. Finally, we conclude the paper and discuss our plans for future work in Section VI.

II. Related Work

A. Vehicular Perception

Research in vehicular perception has been rapidly accelerating in the last decade, largely due to the vehicular sensing datasets that have been released during this time. The most well known of these datasets is Kitti [3], but other datasets that are even more comprehensive in terms of environmental diversity and amount of labeled data have emerged such as the NuScenes [4] and Cityscapes [5] datasets. Due to the high quality labeled data these datasets provide along with the rise of machine learning methods, many models have been created for a variety of perception tasks such as motion/trajectory prediction [6] [7], object detection [8] [9], object association [10] [11], object tracking [12], and semantic segmentation [13] that have shown promising levels of performance on these vehicular perception datasets. Object detection, specifically 3D object detection [14], is at the core of many of these tasks and as such has attracted a large amount of research.

The two primary sensors used for vehicular 3D object detection are LiDAR and cameras. LiDAR sensors have become a popular choice, as this sensor can sense the depth of an object more accurately than a camera which leads to more accurate 3D detections [15]. There are four general categories of LiDAR 3D object detectors: Point based models such as PointRCNN [16] and PointFormer [17], Grid based models such as SECOND [18] , PointPillar [19] and PIXOR [20], Point-Voxel based models such as Fast Point R-CNN [21] and Pyramid R-CNN [22], and Range based models such as LaserNet [23] and RangeDet [24]. The voxel-based and point-voxel based LiDAR detection models produce the highest levels of accuracy; however, many grid based

methods are used for real-time vehicular perception due to their exceptional inference latency.

For camera based 3D object detectors, there are five general categories: Image-only monocular based models such as CenterNet [25] and MonoFlex [26], Depth-assisted monocular based models such as Pseudo-LiDAR [27] and MonoDTR [28], Prior-guided monocular based models such as 3D-RCNN [29] and MoNet3D [30], stereo based models such as Stereo R-CNN [31], YOLOStereo3D [32] and PLUMENET [33], and multi-camera based models such as DETR3D [34] and ImVoxelNet [35]. Although there have been improvements in 3D object detection accuracy in camera based models, especially stereo and multi-camera models, their performance still lags behind LiDAR models both in terms of performance and latency [36] [37]. However, since cameras are currently so much more prevalent on vehicles today as compared to LiDAR, additional research for camera based 3D object detection may continue to be of use going forward.

B. Collaborative Vehicular Perception

While the area of individual vehicular perception will continue to advance, a new paradigm of collaborative vehicular perception has also emerged that offers levels of perception that are not achievable by any single vehicle. As in individual vehicular perception, datasets to study collaborative vehicular perception have been created such as OPV2V [38], DAIR-V2X [39], and V2XSet [40]. These datasets provide synchronized sensor data for two or more vehicles all driving within the same area. As such, many new collaborative perception models have been proposed within the last few years. Several different collaboration methods have been proposed, from the more traditional fusion techniques proposed in F-Cooper [41] and CoCa3D [42] to the graph based methods of V2VNet [43], DiscoNet [44] and MP-Pose [45]. However, attention based methods such as AttFusion [38], CoBETV [46] and VIMI [47] have begun to show more optimal levels of performance. Until very recently, collaborative perception research including all work listed up to this point have been homogeneous in terms of sensing capabilities, but more work has begun to emerge which investigate collaborative perception for heterogeneous sensing with models such as HM-ViT [48] and HEAL [49] that can accept different sensor modalities from different vehicles.

One aspect of collaborative perception that we are interested in is ensuring that latency requirements for end-to-end collaborative perception are met given dynamic networking conditions, and there have been some methods that have been created with this task in mind. FPV-RCNN [50] proposes a keypoint feature selection and fusion strategy and Where2Comm [51] proposes a spatial-confidence aware communication mechanism which both aim to reduce the amount of data that is transmitted from vehicles. There are also collaborative perception models such as LCRN [52] and SyncNet [53] that provide methods to mitigate the effects of wireless communication loss or delay on collaborative perception accuracy. However, none of these works consider the variable availability of vehicular computing and sensing in real-world scenarios, which is the focus of this work.

C. Vehicular Edge Computing

With the emergence of these vehicular perception methods, a new need for computing has been created in vehicular environments. Edge computing is one avenue to provide additional computing for vehicles that has shown promise in aiding these perception tasks and this has opened up an entire area of research dedicated to determining how to optimally offload data to the edge in different situations [54].

In order to optimize the edge offloading problem, many works have employed more classical models such as convex optimization [55], mixed integer nonlinear programming [56], game theory [57] [58] [59], Markov decision process [60], heuristics [61], and other numerical optimization methods [62] [63] [64]. More recently more complex machine learning models have been introduced and new methodologies for this task partitioning and offloading problem that involve convolutional neural networks [65], federated learning [66], and reinforcement learning [67] [68] [69] have been created. However, none of these proposed methods consider both accuracy and latency in terms of the optimizations.

In our previous work [70], we have explored real-time collaborative perception in edge enabled vehicular environments for the case of homogeneous sensing and computing. In this paper we aimed to expand that work by introducing a new model which includes variable levels of sensing and computing, individual decisions for each vehicle rather than a single system-level decision for all vehicles, and containing an adaptive latency requirement formulation as opposed to a static one.

III. Heterogeneous Sensor Fusion in Mobile Edge Environments

In this section, an overview of the subject environment will be examined. Then, the different trade-offs that create the decision space for the associated collaborative perception problem will be explained. Finally, the problem formulation will be presented that will serve as the basis for the remainder of this paper.

A. Overview

To begin the discussion of heterogeneous vehicular environments, consider the example shown in Fig. 1. This figure shows vehicles with different levels of sensing and computing driving in the same street environment; vehicles can transmit sensor data to the edge as well as receive action decision and perception results as shown by the dashed lines. Table 1 shows the capacities of the different types of vehicles that are being considered for this work to populate these heterogeneous vehicular environments. By having vehicles communicate with edge computing and communication nodes,

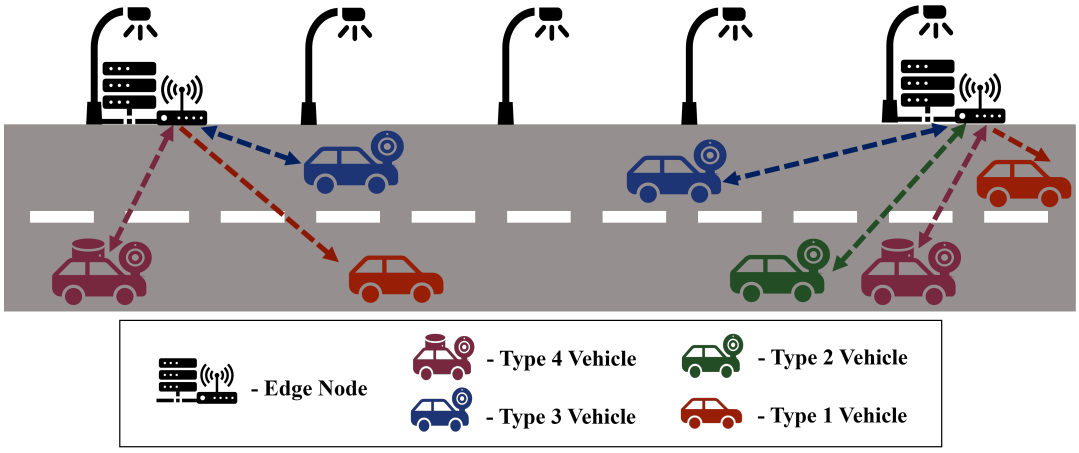


FIGURE 1: Overview of an edge-enabled heterogeneous vehicular environment that contains vehicles with different levels of computing and sensing capabilities as well as edge nodes for vehicles to communicate with and collaboration to occur. Type 1 vehicles, which do not contains external sensors, can still receive the collaborative perception results but do not transmit data to the edge.

information can be shared between vehicles without any sort of vehicle-to-vehicle (V2V) communication. Additionally, decision processes at the edge node can determine which vehicles should be selected to participate in the collaborative perception; all vehicles in the area should have the results of the collaborative perception broadcast to them irrespective of whether they can participate in the sensor fusion process or not. If situations are encountered where the number of vehicles dramatically increases causing congestion on the roads and wireless networks, then less vehicles can be selected to participate in generating the collaborative perception to lighten the networking and computing load. Designing a system in this way allows for every vehicle to potentially benefit while only needing a limited number of vehicles to generate the collaborative perception. This also creates an inherent scalability, as collaborative perception generation can continue to operate even as the total number of vehicles and variations in the network conditions increase.

TABLE 1: The four vehicle types we are considering in this work.

Vehicle Type	Available Sensors	Computing Availability
Type 4	Camera, LiDAR	Yes
Type 3	Camera	Yes
Type 2	Camera	No
Type 1	None	No

B. Trade-offs

In terms of the end-to-end heterogeneous sensor fusion process for collaborative perception in vehicular edge environments, there are a number of different trade-offs that create decisions which can affect the performance of the collaborative perception. The trade-offs discussed in this section will only pertain to participating vehicles, which we define as any vehicle that has at least one sensor and can thus participate in the generation of the collaborative perception. The four specific trade-offs that will be investigated in this work are vehicle selection, feature extraction model, collaborative perception scheme, and the amount of computational offloading. In this section, each one will be discussed as well as its effect on collaborative perception performance.

1) Vehicle Selection

In a heterogeneous vehicular environment, there are vehicles of potentially many different types that coexist in the environment, but not all vehicles need to participate in every second of generating the collaborative perception for a particular edge computing node. There are a number of factors that can affect why one vehicle should be chosen over another, and there are two factors in particular that are going to be considered in this work. One is the sensing available on the vehicle; a vehicle that contains a powerful sensor suite composed of a large number of high-fidelity sensors will likely produce an accurate representation of its surroundings and therefore should have a higher probability of being chosen to participate in the collaborative perception compared to vehicles with less capable sensing suites. The

TABLE 2: Changes in perception accuracy measured in mAP caused by different feature extractors and number of participating vehicles for the HEAL [49] sensor fusion model.

Number of Participating Vehicles	1	2	3	4
PointPillars (LiDAR)	.7532	.9234	.9361	.9394
SECOND (LiDAR)	.7554	.9302	.9426	.9456
ResNet-101 (Camera)	.2043	.4602	.4614	.4656
EfficientNet (Camera)	.3778	.5237	.5344	.5356

other is the wireless communication conditions between the vehicle and the edge node. If there are some vehicles with a weak or non-existent wireless link to the edge nodes, then these vehicles should avoid being chosen despite having high quality sensors available.

To quantify why one particular vehicle might be chosen over another, consider the results presented in Table 2. In this table, the mean average precision (mAP) accuracy values are shown for the HEAL [49] collaborative sensor fusion model when different feature extractors are used for cases of 1-4 participating vehicles. Several conclusions can be drawn from these results. One is that for the 3D object detection task, LiDAR based methods perform much better than camera based methods, which should make LiDAR enabled vehicles more likely to be chosen to be a participating vehicle. Another result to point out from this table is that there are diminishing returns for perception accuracy as the number of participating vehicles increases; each new participating vehicle introduces less new information on average than the one before it and as such every additional mAP gain from an additional vehicle's sensor data is less than the previous. As a result, if there are certain vehicles without sufficient computing or a bad wireless link then those vehicles may not be chosen, as the increase in overall accuracy from the contribution of their sensor data may not be worth the additional latency required especially if there are already 2 or more vehicles already participating.

2) Feature Extraction

Every participating vehicle will have one or more external sensors, and there are many different ways that the visual features needed for perception tasks can be extracted from this sensor data. While some object detection models are created to be efficient, in general the more accurate models are more heavyweight and have higher inference latency and may require powerful computing in order to run in real-time [14]. Including multiple types of feature extraction within the collaborative perception model creates this trade-off between accuracy and latency, which will act as one of the many knobs that can be controlled in this system. As shown in Table 2, the chosen sensor type and the associated feature extraction methods can have a significant effect on the resultant perception accuracy.

3) Collaborative Perception Scheme

Another knob in the realm of accuracy and latency trade-offs is the collaborative perception scheme. For this work, we will consider the collaborative perception schemes of intermediate collaboration and late collaboration. In general, an object detector typically consists of two key components: the feature extractor, which pulls visual features from the raw sensor data, and the detection head, which uses these extracted features to determine the locations of bounding boxes. In

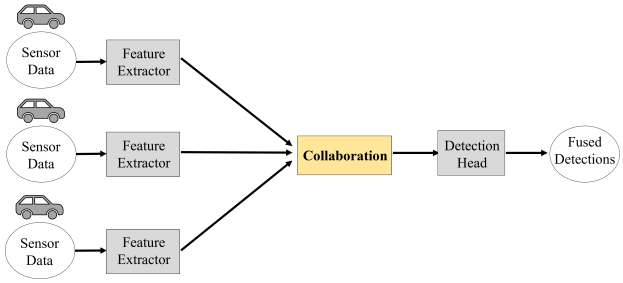


FIGURE 2: Overview of the collaboration scheme of intermediate fusion.

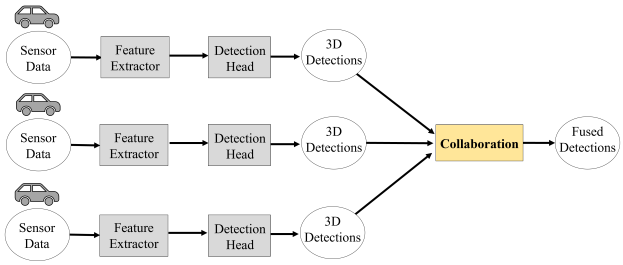


FIGURE 3: Overview of the collaboration scheme of late fusion.

intermediate collaboration, the fusing of data from different sources is performed after extracting features but before the detection head as shown in Fig. 2. The combined features are then processed together to make detection decisions, allowing the model to learn from all data sources simultaneously and leverage complementary information. On the other hand, late collaboration which is shown in Fig. 3 refers to combining the outputs of separate detection models, each working on data from a different source. Intermediate collaboration generally provides richer, more integrated information for decision making, leading to better detection performance on average compared to late collaboration; the latency of late collaboration is less than that of intermediate though as the collaboration method is usually more lightweight, such as non-maximum suppression, compared to the more complex collaborative detection heads in intermediate schemes [71].

4) Computational Offloading

The final trade-off that we are considering in this work is the use of computational offloading. Since an edge node is utilized as the location of the collaboration for collaborative perception, there is always some level of offloading that is required for each time step in this process. In this work, we will consider two different levels of offloading which we term full offloading and minor offloading, and the differences can be seen in Fig. 4. With full offloading, the sensor data is compressed and transmitted to the edge where all computational processes will occur. In minor offloading, some of the computation will occur in the vehicle depending on whether intermediate or late fusion is used. In cases of participating vehicles that do not have computational

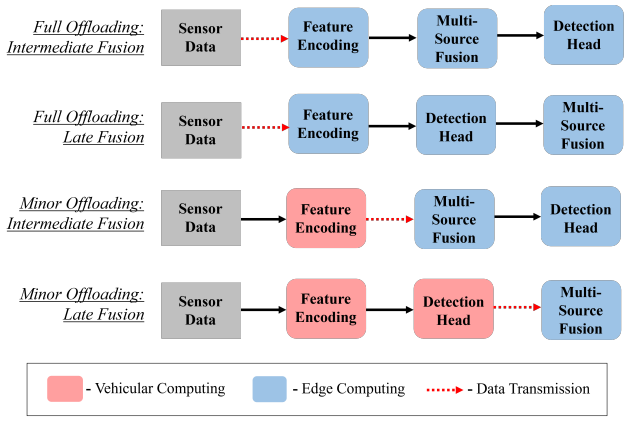


FIGURE 4: The four different cases of computational offloading that are being considered which determine the location of the computing and what data is being transmitted from the vehicle to the edge.

resources, full offloading must be chosen in order to contribute to the collaborative perception. Vehicles with computational resources will have the option to choose either minor or full offloading; the offloading procedure chosen does not affect perception accuracy but can have a large effect on latency. Since the amount of data transmitted from the vehicle to the edge in minor offloading is much less than full offloading, choosing minor offloading will be the best decision in most cases. However, there are some situations where full offloading will be more optimal, such as when there is a significant difference in the computing power between the vehicle and the edge or if there are exceptionally high levels of throughput. Additionally, since the data sizes for the bounding boxes produced by object detection are smaller than the feature extraction results, the transmission latency for late fusion will be less than that of intermediate fusion for minor offloading.

C. Problem Formulation

In this work, the state of the vehicles within the edge-enabled connected vehicle environment is defined as follows:

$$S = \{s_{v_1}, s_{v_2}, ..., s_{v_n}\} \tag{1}$$

Each element in S, s_{v_i}, is a vehicle that contains three states:

$$s_{v_n} = \{r_n, s_n, c_n\} \tag{2}$$

r_n is the current throughput between the vehicle and the edge (in Mbps) and s_n is the current vehicle speed (in Kmph). c_n is the computing and sensing type $\in [1, 2, 3, 4]$, as defined in Table 1. The goal of this work is to select the best action at each time step of the collaborative perception process given the current state information. An action (A) is defined as follows:

$$A = \{a_{cp}, a_{v_1}, ..., a_{v_n}\} \tag{3}$$

A contains a set of instructions for each participating vehicle, a_{v_i}, as well as a set of collaboration parameters, a_{cp}. The only collaboration parameter that we consider in this work is the collaborative perception scheme and as such a_{cp} is defined as follows:

$$a_{cp} = \{0, 1\} \tag{4}$$

In this binary encoding, $a_{cp} = 0$ corresponds to late fusion and $a_{cp} = 1$ corresponds to intermediate fusion. The individual vehicle instructions are defined as:

$$a_{v_n} = \{p_n, o_n\} \tag{5}$$

The action decision for each vehicle contains two instructions. One of these is related to the perception model (p) and the other is related to the offloading level (o). The perception model representation is defined as:

$$p_n = \{0, 1, ..., l\} \tag{6}$$

In this encoding, there are l possible feature extractors for this particular vehicle and the value of l will depend on what sensors the vehicle is equipped with; the more sensors that are available on the vehicle, the more possible values of l there will be since more potential feature extraction models that can be utilized. Each value of l is associated with a particular feature extraction model. The option of $p_n = 0$ correlates with the vehicle not participating in collaborative perception and thus not transmitting its perception data; this option will always be chosen for Type 1 vehicles but may be chosen for Type 2, 3 or 4 vehicles if the conditions warrant it. The other action for each vehicle is the offloading level, which is defined as:

$$o_n = \{0, 1\} \tag{7}$$

In this binary encoding, we define $o_n = 0$ as minor offloading and $o_n = 1$ as full offloading. In cases of vehicles that do not have on-board computing, this value will not factor into the action decision for this vehicle since $o_n = 1$ automatically.

For every time step, the set of all possible actions that can be executed is defined as follows:

$$\mathcal{A} = \{A_1, A_2, ..., A_k\} \tag{8}$$

In this equation, $\mathcal{A}$ contains all possible actions (A_i) that can be chosen given the current distribution of vehicles that are available to participate. Every action selected will produce an associated end-to-end latency value and collaborative perception accuracy value that we will define as follows:

$$f_{AP}(A_i \mid S) = mAP_i \tag{9}$$

$$f_L(A_i \mid S) = L_i \tag{10}$$

Essentially, every possible action (A_i) will have a resulting latency L_i (measured in seconds) and collaborative performance accuracy mAP_i (measured in mean average precision). Although accuracy should be maximized, a real-time constraint should also be applied to the maximization formulation to ensure that the chosen actions can meet a given latency requirement τ. As such, we will define the proposed optimization problem as follows:

This article has been accepted for publication in IEEE Open Journal of Vehicular Technology. 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/OJVT.2025.3533368 IEEE Open Journal of Vehicular Technology

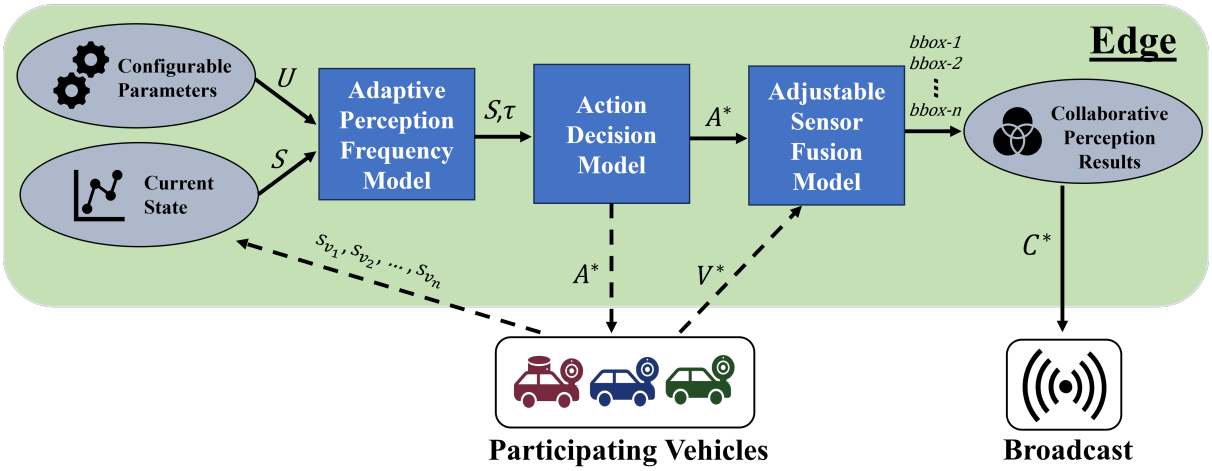


FIGURE 5: An overview of our proposed HAdCoP framework which consists of three sub-models: the APF model, the action decision model, and the adjustable sensor fusion model. Each sub-model can be chosen and configured by the user to match the desired use case.

$$\max_{\mathcal{A}|S} \quad mAP_i \tag{11a}$$

$$s.t. \quad L_i < \tau \tag{11b}$$

In order to define a singular metric that can be used for performance evaluation, a service delivery value is first defined as follows:

$$D_i = \begin{cases} 1, & \text{if } L_i < \tau \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

In this equation, we have defined successful service delivery (D_i) as the end-to-end collaborative perception process, including all data transmissions, completing in an elapsed time less than τ. Subsequently, we define the metric of effective mean average precision (EmAP) as follows:

$$EmAP_i = (D_i) \times (mAP_i) \tag{13}$$

Since $D_i \in [0, 1]$, optimizing this new metric produces an equivalent optimization formulation as the one presented in Equation (11) can be restated as follows:

$$\max_{\mathcal{A}|S}(EmAP_i) \tag{14}$$

This $EmAP$ metric accurately reflects this optimization problem considering that the goal is to produce the most accurate perception possible while still ensuring the latency requirement is met which also maximizes reliability since any failure to deliver makes $EmAP = 0$.

IV. Methodology

In this section, the core methodology for heterogeneous collaborative perception in real-time vehicular edge environments will be presented. The HAdCoP framework is the center of this methodology, as this is what defines the data flow and fusion process and is where the discussion will begin. Afterwards, each component of this model will be discussed individually. Finally, a discussion on the dataset selection and generation used to create the training and testing data for the performance evaluation is provided.

A. Heterogeneous Adaptive Collaborative Perception

An overview of the **H**eterogeneous **Ad**aptive **Co**llaborative **P**erception (HAdCoP) framework can be seen in Fig. 5. There are three submodels within the HAdCoP framework that control the three main processes. The APF model first takes the set of configurable parameters (U) and the current state (S) as input. Then, the latency threshold for the current time step (τ) is computed. This threshold, along with the state information, is passed to the action decision model, which determines the action (A^*) to be executed for the current time step. As defined in Section III-C, an action contains the overall collaborative perception scheme as well as the instructions for each vehicle which determine what object detection or feature extraction tasks if any need to be computed on the vehicle and what type of data to be transmitted to the edge; the combination of all data the participating vehicle have been instructed to offload to the edge (V^*) is immediately ingested by the adjustable sensor fusion model once it is received at the edge.

The output of the adjustable sensor fusion model is the collaborative perception results (C^*) for the current time step and this is broadcast to all vehicles in the area. At the next time step, the process will start over again with the current state information being transmitted to the edge. This framework allows for dynamic adaptation of the collaborative perception model and corresponding frequency of results generation to match the environment as the vehicle sensor and computing distributions and wireless conditions change over time.

This proposed framework defines the data transfer process between each of the different submodels, but the selection of each particular model can be chosen by the user. This

modular approach allows the proposed framework to have the potential to be applied to any number of vehicular use cases by changing the three models within it. For the subject heterogeneous collaborative perception problem, we utilize a previous work for the adjustable sensor fusion model, but have proposed new models to serve as the APF model and the action decision model which will be discussed in the subsequent subsections.

B. Adaptive Perception Frequency Model

TABLE 3: Summary of Key Notations with Descriptions

Notation	Description
T	Estimated Latency Lower Bound
τ	Perception Latency Threshold
T_{MO}	T for cases of minor offloading
T_{FO}	T for cases of full offloading
L_{MO}	Expected computing latency for minor offloading
L_{FO}	Expected computing latency for full offloading
L_C	Context Latency Estimation
R_{A2}	Maximum average throughput between all sets of two vehicles
D_C	Data size of the compressed sensor data
D_E	Data size of the encoded sensor data
C	Number of vehicles with on-board computing
n	Number of participating vehicles
o	Number of vehicle objects (3D bounding boxes) in previous frame
v	Current average speed
c	Ratio of vehicles with on-board computing
s	Ratio of vehicles with LiDAR sensors
α	Object weight parameter
β	Vehicle speed parameter
γ	Vehicular computing parameter
δ	Vehicular sensing parameter

The first submodel within the HAdCoP framework is the **A**daptive **P**erception **F**requency (APF) model. The purpose of this model is to compute the lowest realistically achievable latency threshold that the end-to-end collaborative perception process can execute in given information about the current state. The motivation for this is that vehicular environments are highly dynamic and can experience very sudden changes due to the high-speed nature of vehicles. To this end, the latency threshold should be as low as it can be at all times. However, there are situations where a lack of vehicles with computing, advanced sensing, or poor wireless communication conditions causes the amount of time required for the end-to-end collaborative perception process to increase. In these cases, the latency threshold should be increased so that there is sufficient time to compute the collaborative perception results for the current time step before sensor data ingestion begins for the next time step. In this way, the optimal amount of data can be processed in a useful way and perception delays as well as idle time are reduced.

The APF model contains two steps: computing a weighted lower bound threshold value (T) and binning this threshold to produce the final latency threshold τ. The main reason why T is being binned is due to the nature of sensors operating on fixed frequencies (e.g. 1Hz, 10Hz, 30Hz, etc.); in order to keep the different types of sensor synchronized, only certain frequencies that all sensors can utilize should be chosen.

There are two terms that will be defined which create the framework of the APF model and these are the lower bounds for cases involving vehicular computing (minor offloading) T_{MO} and for cases involving no vehicular computing (full offloading) T_{FO}. These are defined as follows:

$$T_{FO} = L_{FO} + \frac{2D_C}{R_{A2}} \tag{15}$$

$$L_C = \frac{\alpha}{(o+1)} + \frac{\beta}{(v+1)} + \frac{\gamma}{(c+1)^n} + \frac{\delta}{(s+1)^n} \tag{16}$$

$$T_{MO} = L_{MO} + \frac{2D_E}{R_{A2}} + L_C \tag{17}$$

The variables mentioned in the terms of both equations can be found in Table 3. T_{FO} is fairly straightforward, since it is just the expected value of the computation latency added to a best-case estimation of the time it would take for data transmission. T_{MO} has a similar set of first terms, though their values will be much lower than T_{FO} since $L_{MO} < L_{F0}$ and $D_E < D_C$. This case of having the potential to utilize minor offloading results in more potential actions that can be chosen which can lead to more optimal action decisions if the state information can be effectively represented in determining an achievable latency threshold. As such, four configurable terms are included to represent the effects of object density, vehicle speeds, and the distribution of sensors and computing in participating vehicles, and these terms are summed to create an estimated context latency L_C. The associated parameter values for these terms form the set of configurable parameters for the HAdCoP framework ($U = [\alpha, \beta, \gamma, \delta]$). Now that T_{MO} and T_{FO} have been defined, the APF model is defined as follows:

$$T = \begin{cases} T_{FO}, & \text{if } C < 2 \\ min(T_{MO}, T_{FO}), & \text{otherwise} \end{cases} \tag{18}$$

$$\tau = \begin{cases} .1, & \text{if } T < .1 \\ .2, & \text{if } .1 \leq T < .2 \\ .5, & \text{if } .2 \leq T < .5 \\ 1, & \text{otherwise} \end{cases} \tag{19}$$

There are two terms in Equation (18) that correspond to two cases that can be encountered in connected vehicle environments. The top term in Equation (18) is used when there are fewer than two vehicles with on-board computing capacities (C) available. In this case for the collaborative perception results to be generated, at least one participating vehicle must utilize full offloading, and as such, the data transmission will dominate the required total latency and T_{FO} will best reflect the lower bound for the latency. The bottom term in Equation (18) shows what will happen in cases where minor offloading could be utilized and in most cases T_{MO}

will be chosen, but in certain cases with very high wireless throughput T_{FO} may produce a lower value and be chosen instead. Equation (19) states the four perception frequency values we are considering for HAdCoP and specifies which values of T correspond to which values of τ. Since T is a theoretical lower bound, τ is just the value of T that has been rounded up to the nearest bin value.

C. Action Decision Model

The action decision model is the second submodel. It is the core of the HAdCoP framework as this is what determines the selected set of actions to produce accurate, real-time collaborative perception. As is consistent with the problem formulation presented in Section III-C, the goal of this model is to maximize EmAP. To accomplish this, we have created a **C**ontext-**a**ware **L**atency **Pre**diction **N**etwork (CaLPeN) which is shown in Fig. 6. The idea behind this network is to have multiple individual neural networks, which we have termed I-Nets, for each vehicle's set of state and potential action instruction combinations to produce a latent space representation for each combination. These latent space representations are then concatenated and have an additional feature appended to them, which is the collaborative perception scheme a_{cp}, to form the set of intermediate data (ID). This ID is the input to the collaborative neural network, or C-Net, which classifies the ID to predict which of the associated actions will meet the required latency. The action within this set of actions that were predicted to meet the latency requirement that has the highest expected perception accuracy is chosen as A^*.

The input to the CaLPeN model is a (1×12) element vector $S = \{s_{v_1}, s_{v_2}, s_{v_3}, s_{v_4}\}$ that contains the state vehicle state information for the four vehicles from the current time step as described in Section III-C. Each vehicle's associated I-Net input is created by concatenating copies of that particular vehicle's state to every one of the $2l$ combinations of possible instructions that the vehicle can execute to create the batch of data vectors $\{[s_{v_n}, a_1], [s_{v_n}, a_2], ..., [s_{v_n}, a_b]\}$ that is of total size $(2l \times 5)$ for each vehicle. Each I-Net will produce an output of size $(2l \times 1)$ and these are concatenated together to form an output of size $(2l \times 4)$; copies of this concatenated output are created with each having a possible value of a_{cp} appended onto it and then each of these copies are concatenated together. Since we have defined a_{cp} to contain only binary values, the ID that is the input to the C-Net is of size $(4l \times 5)$. The output of the C-Net is the predicted optimal action A^* which is of size (1×9): four sets of vehicle instructions which contain an offloading and feature extraction instruction and one element to indicate the chosen collaborative perception scheme.

The value of l that is used is that of the participating vehicle with the highest amount of sensors; we describe our chosen sensor and model configurations in Section V-A which lead to a value of $l = 4$, but the batch sizes that define the first dimension of the data that moves through the CaLPeN

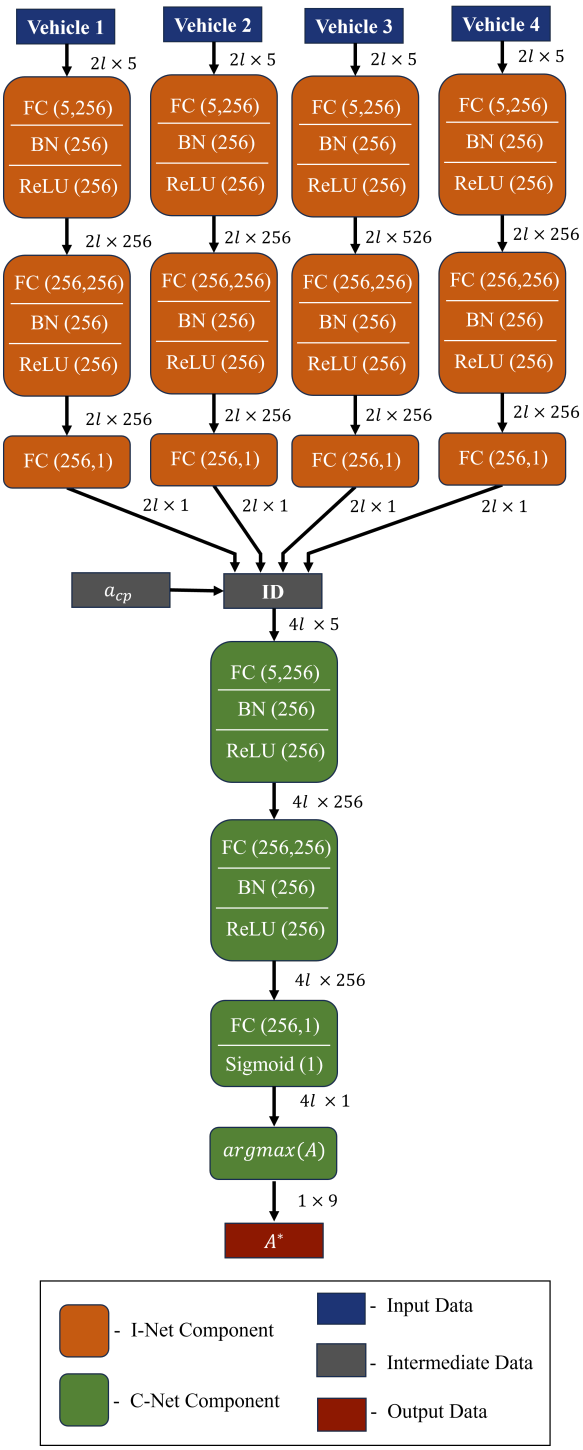


FIGURE 6: An overview of our proposed CaLPeN model that will act as the action decision model in HAdCoP. Each vehicle has its state/action combinations processed on a I-Net before the final C-Net determines which action should be chosen. Each computational block depicted contains the input/output size for the fully connected (FC) layers and channel size for any batch normalization (BN) layers or activation functions and the data sizes are listed before/after each block.

have been described in terms of l to be generalizable to other selections of sensors and associated feature extraction models. While each of the I-Nets have the same neural network architecture, they are trained separately and as such do not share parameter weights. This entire network is trained end-to-end for 5 epochs using the Adam optimizer [72] and the binary cross-entropy loss function. For the proposed CaLPeN, it is assumed that there are at least 4 vehicles available to participate in the collaborative perception; if there are ever less than 4 vehicles available, then null states will be used in place of the missing vehicles.

D. Adjustable Sensor Fusion Model

The final submodel of HAdCoP is the adjustable sensor fusion model. Any multi-source sensor fusion model that can be configured to accept any type of sensor input would be able to act as the adjustable sensor fusion model. Fortunately, some collaborative perception models have been created in recent years that fit this criteria. For this work, we are not attempting to create our own adjustable sensor fusion model but have leveraged a recent work in this area instead. We are using the Heterogeneous Alliance (HEAL) framework [49] as the adjustable sensor fusion model. This model has been shown to produce state-of-the-art results in collaborative 3D object detection and has been designed specifically to accept any type of sensor data as input. For each sensor input, multiple feature extractors can be made available due to the backward alignment of new agents in the collaborative training process. Additionally, this model is also lightweight enough that it can be executed in real-time on most modern GPUs.

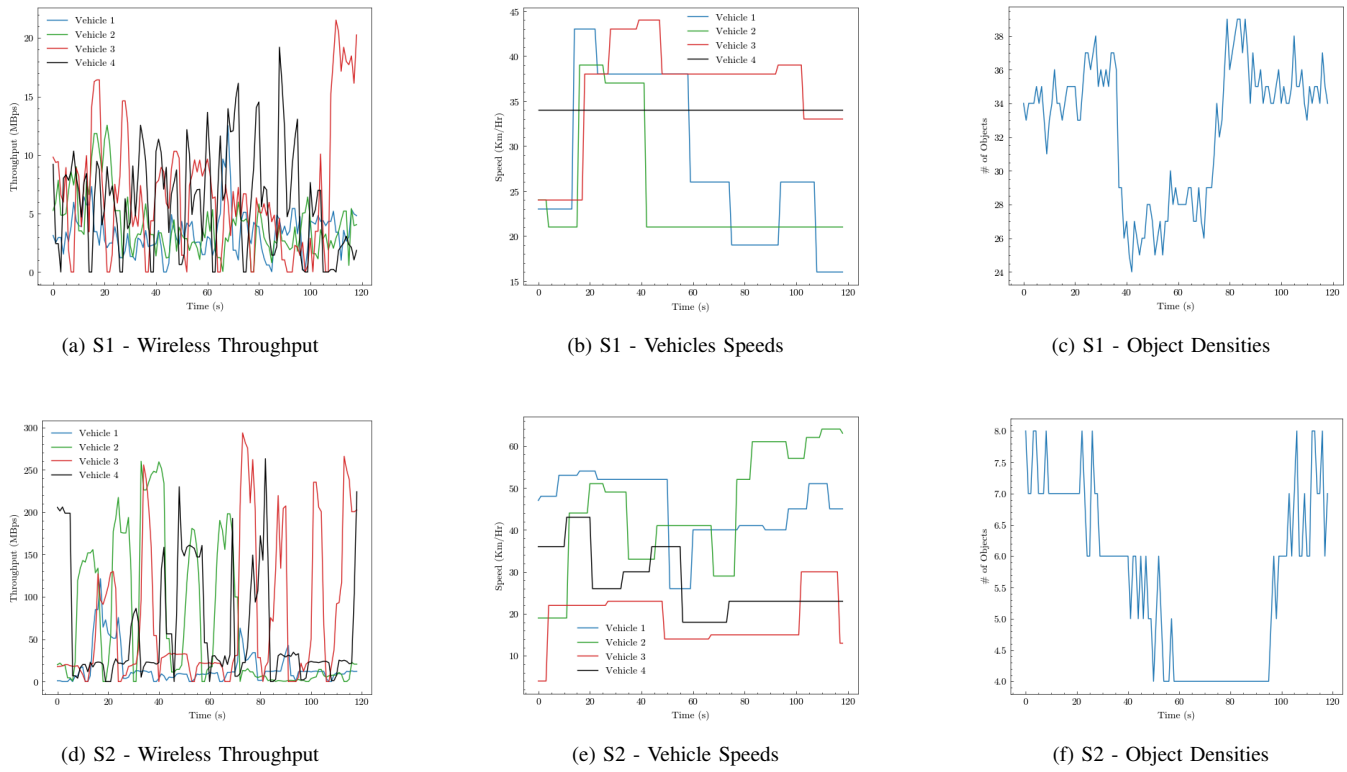
E. Dataset Selection

There are no datasets that contain perception data (real or synthetic) from multiple moving vehicles in the same area along with the corresponding wireless communication data for the links from the vehicles to the edge or cell towers. However, there are datasets that contain multi-source vehicular sensor data and vehicular communications separately. As such, our approach to creating training and testing datasets that could be used to study this heterogeneous collaborative perception task was to combine information from a perception dataset and information from a wireless communication dataset to simulate the intended environment. The chosen perception dataset is OPV2V [38]. This is one of the largest collaborative perception datasets (11,464 frames) with a focus on 3D object detection that contains 4 or more vehicles driving simultaneously equipped with both camera and LiDAR sensors. For the wireless dataset, we utilized 5G wireless traces from moving vehicles that have been published for research purposes [73]. This dataset contains throughput, channel and context information for 5G networks and contains over 50 unique wireless traces from moving vehicles.

To generate training and testing data for this work, segments from both perception and wireless communication datasets were selected to form two distinct scenarios that can demonstrate the robustness of our methodology. The wireless communication dataset contains 16 traces from a moving vehicle performing a file download/upload, and 8 traces were selected to form two datasets of 4 vehicles. From each wireless trace, 600 data points of wireless throughput were chosen, each containing the vehicle's wireless throughput and speed; for each of these data points of the wireless data, a corresponding set of OPV2V perception data is associated with it forming a complete dataset needed to explore collaborative perception in real time in vehicular edge environments. The two sets of wireless traces along with the vehicle speeds and object counts for the testing datasets can be seen in Fig. 7. We name these two testing datasets Scenario 1 (S1) and Scenario 2 (S2) respectively. These two segments of perception and networking data were specifically chosen to simulate two distinct scenarios with S2 having higher average throughput and vehicle speeds compared to S1 but with higher variance and lower levels of surrounding object density.

Each vehicle in the OPV2V dataset contains the same sensor suite that contains both LiDAR and cameras, but we are interested in studying environments that are closer to the real world and the heterogeneous sensing that is encountered on the streets today and that may be on the roads in the future decades. To combat this, we have defined two different distributions of vehicular sensing and computing to simulate what may be encountered in vehicular environments. Currently, there are few vehicles that contain high-definition 360-degree LiDAR sensors on the road today, but this type of sensing has become a staple of vehicular perception research over the last decade due to its superior performance in 3D perception tasks compared to camera or radar/ultrasound [74]. As such, LiDAR sensors are expected to appear in production vehicles in the next decade as demand for continued improvements in automotive safety increases [75]. Although it is unknown what the distribution of sensors will look like on the roads of the future, we model two different scenarios of vehicular sensing distributions to use for testing purposes: One for near-future use cases where LiDAR sensing and powerful computing suites are still sparsely populated in real-world street environments which we term computing and sensing conditions 1 (CSC1) and another for a level of sensing and computing for the more distant future where advanced sensing and computing on vehicles is more ubiquitous which we have termed computing and sensing conditions 2 (CSC2). A visual representation of these two distributions can be seen in Fig. 8, using the vehicle type definitions from Table 1. For each data point in the networking and perception data S1/S2, 4 vehicles are sampled from the chosen computing and sensing distribution to determine the set of vehicles for each data point. By taking the combinations of the two different vehicle distributions and two different sets of associated wireless and perception data, 4 total datasets are produced: CSC1-S1, CSC1-S2, CSC2-S1, and CSC2-S2. Each of these datasets contain 600 data

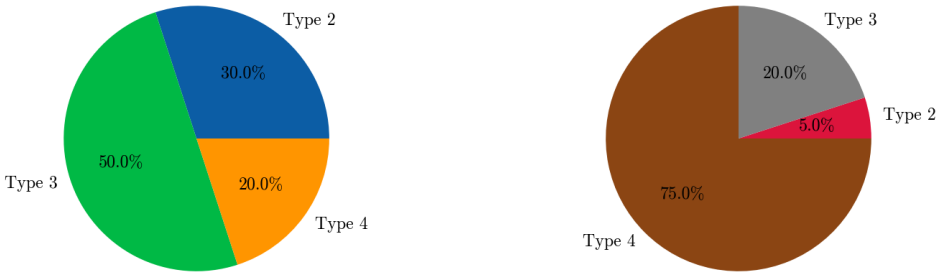
This article has been accepted for publication in IEEE Open Journal of Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJVT.2025.3533368



(a) S1 - Wireless Throughput (b) S1 - Vehicles Speeds (c) S1 - Object Densities

(d) S2 - Wireless Throughput (e) S2 - Vehicle Speeds (f) S2 - Object Densities

FIGURE 7: The wireless throughput and associated vehicle speeds and object densities for the S1 and S2 testing datasets.



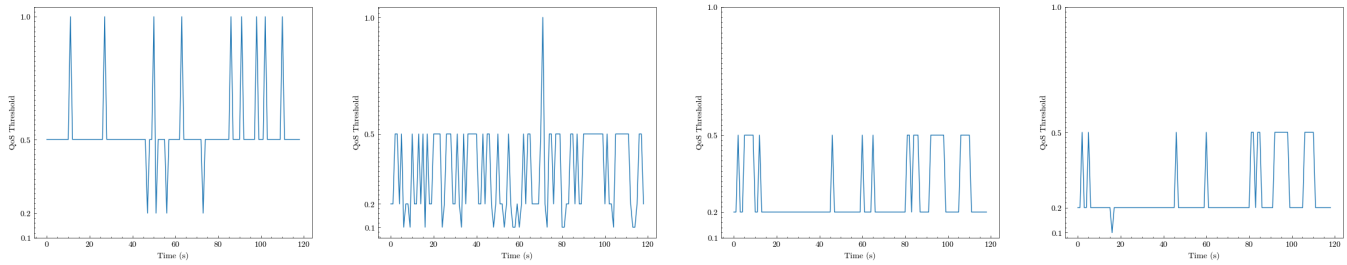
(a) Computing and Sensing Conditions 1 (CSC1) - Vehicle Type Distribution

(b) Computing and Sensing Conditions 2 (CSC2) - Vehicle Type Distribution

FIGURE 8: Distribution of vehicle types for the two created computing and sensing conditions.

points which are split 80%/20% (480/120 data points) to form training and testing sets respectively for the action decision models.

Additionally, an associated set of latency thresholds have been produced by the proposed APF model for each of these 4 testing datasets. The values for τ for each of these cases can be seen in Fig. 9 and will be used for the performance evaluation in the action decision model comparison. As is consistent with the APF model formulation discussed in Section IV-B, the cases of higher wireless throughput produce lower threshold values than those with lower throughput. The values for CSC1 are less than that of CSC2 as well due to the increased presence of LiDAR sensors and on-board computing in CSC2. The parameter values used to generate these values of τ were $\alpha = .25$, $\beta = .25$, $\gamma = .75$ and $\delta = .25$. These values were generated by doing a parameter sweep for all for parameters values in the range [0, .25, .50, .75, 1] and choosing the associated parameter values that performed best over the entire training set in terms of



(a) τ for the CSC1-S1 testing dataset (b) τ for the CSC2-S1 testing dataset (c) τ for the CSC1-S2 testing dataset (d) τ for the CSC2-S2 testing dataset

FIGURE 9: Plots shows how the perception latency threshold value τ changes over time for the 4 testing sets.

minimizing the distance between T and the true latency value T_{true} under the constraint that $T > T_{true}$.

V. Experimental Results

In this section, the remaining details of the research setup as well as the experimental results will be presented. These remaining details include the chosen collaborative sensor fusion models as well as information about how we are evaluating accuracy and latency and the specific hardware that was used. A comparison study is also presented providing the performance values for different action decision models including our proposed CaLPeN model in addition to other machine learning and heuristic models.

A. Collaborative Sensor Fusion Evaluation

Now that the datasets have been generated, the performance results for the collaborative sensor fusion are evaluated in terms of accuracy and latency. As mentioned in Section IV-D, we are using HEAL [49] as the adjustable sensor fusion model, which is based on a neural network architecture. However, each feature extractor needs to correspond to a particular 3D object detection method and two methods were chosen for each sensor type. For LiDAR sensors, the PointPillars [19] ($p_n = 3$) and SECOND [18] ($p_n = 4$) 3D object detection models were used. For camera sensors, two instances of the lift-splat-shoot [76] 3D object detection model were used: one using a modified ResNet-101 [77] ($p_n = 1$) feature extractor and another using an EfficientNet-B0 [78] ($p_n = 2$) feature extractor. Since there are two different LiDAR feature extractors and two different camera feature extractors, the value of l as defined in Section III-C will be $l = 4$ for type 4 vehicles, $l = 2$ for types 2 and 3, and $l = 0$ for type 1. To evaluate the accuracy, we tested all combinations of feature extractors for each individual vehicle and averaged the results over the entire OPV2V testing dataset to mitigate the performance differences between different segments of the dataset and establish the general trends that appear between different sensing and feature extraction combinations on overall accuracy.

The execution latency of the adjustable sensor fusion model is determined by the hardware it is executed on as well as the parameters of inference (i.e. what feature extractors are being used and how many vehicles are participating). By including computational offloading to a mobile edge computing node, it is assumed that the computational power of the edge is greater than that of the vehicles in order to make this trade-off feasible. While some vehicles do have some very limited amount of computing, we are only going to consider vehicles with > 1 TFLOPS of computing to be considered as having computing in terms of the vehicle types to ensure that the vehicle can at least execute the lightest of the collaborative perception models in real time. To model the computing power of vehicles with available computing for the testing data, a NVIDIA RTX 1080Ti GPU (11.3 TFLOPS) is used to model the computing capacity of a vehicle that has computing and for the computing power of the edge, an NVIDIA RTX 4090 GPU (82.6 TFLOPS) is used. For the HEAL model, there are two main steps to the intermediate collaborative perception inference which is 1) individual vehicle feature extraction and alignment and 2) collaborative pyramid fusion and detection head. Step 1) can be executed on the vehicle or on the edge while step 2) always occurs at the edge. Inference latency results for step 1) for both the vehicle and the edge can be seen in Fig. 10 and the inference latency results for step 2) at the edge as a function of how many vehicles participate is seen in Fig. 11. For late collaborative perception, step 1) is just individual vehicle 3D object detection while step 2) is

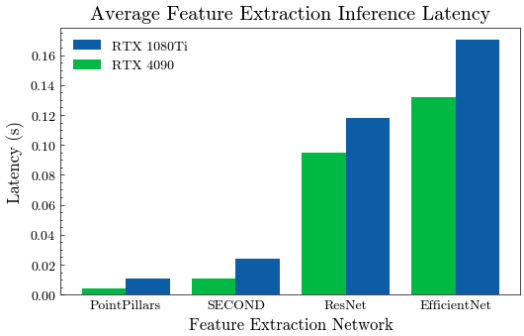


FIGURE 10: Average inference latency values for various feature extractors tested on NVIDIA RTX 1080Ti (vehicle) and RTX 4090 (edge) GPUs.

TABLE 4: The EmAP performance values for each of the potential action decision models.

Action Decision Model	CSC1-S1	CSC2-S1	CSC1-S2	CSC2-S2
CaLPeN	**.7780**	**.9283**	**.7695**	**.9328**
RF	.7313	.8749	.7272	.9001
LR	.6468	.7753	.6529	.8385
VIA [60]	.7136	.8572	.6908	.8577
HMAOA [61]	.3890	.5337	.2255	.4432
LL	.7359	.8770	.7139	.8697
HA	.4774	.4989	.0942	.2622

non-maximum suppression on the union of all vehicles 3D detections which was estimated to take 2ms.

B. Action Decision Model Comparison

With all the accuracy and latency values for the collaborative sensor fusion established, it is now possible to generate performance values for our CaLPeN model in terms of EmAP. To demonstrate the robustness of our model, we have implemented several other methods and baselines to compare against the performance of our model. All models that have been tested are classifiers and for this comparison they receive the same input data of size (1×12) as described in Section IV-C; this input data will be copied and concatenated with the $4l$ possibilities of the nine element combinations within an action selection to create a total input data size of $(4l \times 21)$. The output of these action decision models will then be a set of binary labels of size $(4l \times 1)$ which subsequently has the same $argmax$ function that is the last block of the C-Net within CaLPeN applied to it in order to select the associated action from the output with the positive label that has the maximum sensor fusion accuracy. The result of this process will be a predicted action A^* for each of the comparison action decision models that produces an associated EmAP value. This process is repeated over all 4 testing cases and the results are presented in the remainder of this section.

The CaLPeN model we proposed utilizes machine learning and we wanted to use other models that also utilize machine learning to compare against. The first of these that we tested is logistic regression (LR), which serves as the most lightweight option in this category, but one that still fits the problem formulation well; logistic regression tends to work better in classification problems compared to linear regression. Additionally, we wanted to employ an ensemble model that can capture some of the nonlinearity and patterns in the training data that more complex machine learning models are able to achieve and for this we chose the random forest (RF).

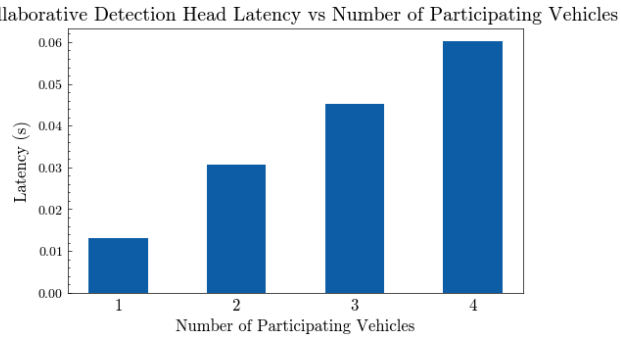


FIGURE 11: Average inference latency values for the collaboration detection head as a function of participating vehicles tested on NVIDIA RTX 4090 (edge) GPU.

In addition to machine learning models, we also created two heuristic baselines to compare against. One of these baselines is termed highest accuracy (HA), since this baseline's heuristic is to choose the action from the set of all possible actions that has the highest expected accuracy. The other baseline is termed lowest latency (LL) and this baseline's heuristic attempts to choose the action which will produce the lowest latency. It cannot be directly inferred from the state information which action will definitively produce the lowest resultant latency, but choosing only the two vehicles with the highest throughput and having each of these two vehicles use the feature extractor with the lowest expected latency will produce the correct action in the majority of cases and as such that is what the LL heuristic does.

The final type of comparisons we conducted were for methodologies from a related work that were created for an analogous task. In this way, by updating the objective function and parameters to match our problem formulation, these related methodologies can be used as action decision models and compared against our proposed model. We chose two works focused on optimizing offloading decision in mobile edge computing environments. One work proposes a Value Iteration Algorithm (VIA) [60] method based on a Markov decision process and the other work proposed a heuristic mobility aware offloading algorithm (HMAOA) [61] to determine the optimal offloading decision.

The results of this action decision model comparison can be seen in Table 4. A visual representation of these results can also be seen in Fig. 12. What sticks out most in these results is that methods which focus on maximizing a utility function, such as accuracy, without properly factoring in a latency constraint will not perform well for this particular

This article has been accepted for publication in IEEE Open Journal of Vehicular Technology. 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/OJVT.2025.3533368

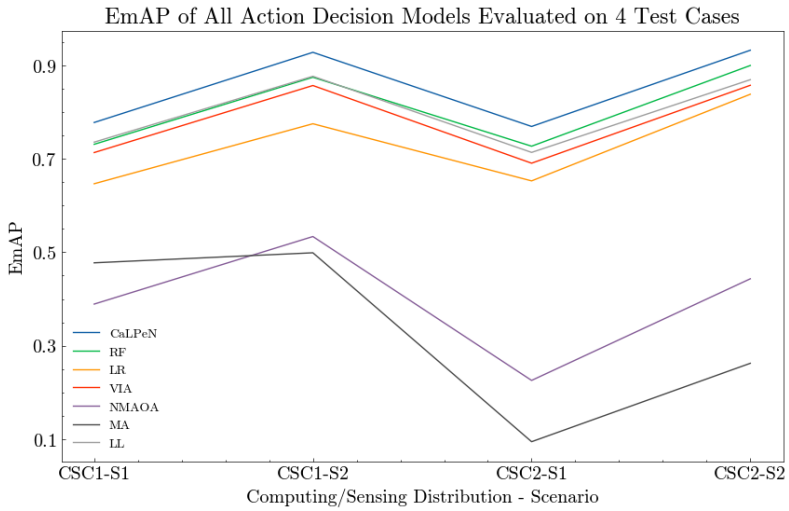


FIGURE 12: Results of the action decision model comparison over the 4 testing sets measured in EmAP.

TABLE 5: The values for four major categories of action decisions averaged over the four testing datasets.

Action Decision Model	CPS	NVS	OP	LSP
CaLPeN	.918	3.321	.095	.553
RF	.788	3.031	.068	.575
LR	.357	2.109	.079	**.750**
VIA [60]	.025	2.0	.022	.623
HMAOA [61]	.002	**4.0**	**.224**	.514
LL	0	2.0	.021	.731
HA	**1.0**	**4.0**	.124	.514

problem. However, with the HAdCoP framework, many different choices are viable to use as action decision models. With an APF model choosing achievable perception frequencies, even using the LL heuristic will produce acceptable performance values for the chosen adjustable sensor fusion model. However, the machine learning models are the highest performing on this task with our proposed CaLPeN model performing the best in all 4 test cases.

To further explore how the decisions made differ between each of the implemented action decision models, statistics were recorded for various categories of actions taken, and the average values for these in the four testing datasets can be seen in Table 5. The four action categories that are presented in this table are the Collaborative Perception Scheme (CPS), the Number of Vehicles Selected (NVS), the Offloading Percentage (OP) and the LiDAR Selection Percentage (LSP). There are several observations that can be made from this table that help explain the performances of each of the methods tested. One thing to note is that HMAOA and HA always selected all four vehicles (NVS = 4.0), making it impossible to complete the collaborative perception process in a time less than τ in all cases, which is the main reason why these methods perform so much worse than all the others. On the other end of the spectrum, VIA and LL always choose only two vehicles (NVS = 2.0) which is a safe option in terms of adhering to the latency requirement, but will never produce optimal accuracy values. While CaLPeN, RF, and LR all produce an NVS between two and four showing there is dynamic selection taking place, CaLPeN is able to make far more optimal decisions shown by the increase in CPS, NVS and OP while still producing a higher EmAP. Even though LR had the highest LSP, it lost out on collaborative perception accuracy increase by not including a third or fourth vehicle enough of the time.

VI. Conclusion and Future Work

In this work, we have presented HAdCoP, a Heterogeneous Adaptive Collaborative Perception framework that contains an APF model, an action decision model, and an adjustable sensor fusion model. We have proposed a novel APF model as well as CaLPeN, a neural network-based Context-aware Latency Prediction Network, which is used as the action decision model. Using HEAL [49] as the adjustable sensor fusion model, we show that CaLPeN is capable of outperforming the six comparison models implemented using our four generated test datasets in terms of EmAP, beating the next best model's performance by 5.5% on average.

With these results, we have shown that as the amount of vehicular sensing and computing increases, a collaborative perception system's potential does as well assuming ITS infrastructure continues to grow to allow for the additional computing and communication needs. There are additional features that can be incorporated into the collaborative perception process, as well as new topics that have not yet been fully explored. In the realm of collaborative perception, there is still a need for protocols, both in terms of networking to ensure all vehicles are able to transmit and receive data in addition to security and privacy protocols to protect the identity of road users and their data as well as prevent potential malicious activity. New methods for multi-source sensor fusion can help further reduce latency and increase the accuracy of the end-to-end collaborative perception process. Additionally, investigating how to enhance sensor data robustness to improve collaborative perception performance in the limited sensing and computing environments that exist today may help propel the adoption of such technologies.

In terms of improving the performance of the proposed HAdCoP framework, adding additional context features, such as the expected field of view of each vehicle's sensors, can aid in the vehicle selection process so that vehicles that can contribute sensor data of areas that have not been seen by other vehicles are more likely to be chosen. Additionally, creating a new machine learning architecture for HAdCoP that combines all three submodels into one may help improve both perception accuracy as well as reducing latency overhead by improving algorithmic efficiency. The potential to include sensor data from street infrastructure can reduce the number of vehicles needed to participate in collaborative perception in each time step while maintaining or even improving the overall perception accuracy with the inclusion of new viewpoints. Finally, adding or creating new testing data with increased diversity in terms of weather and street environments as well as the inclusion of real-world data will help further validate the robustness of the proposed framework.

REFERENCES

[1] S. Lu and W. Shi, "Vehicle computing: Vision and challenges," *Journal of Information and Intelligence*, vol. 1, no. 1, pp. 23–35, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949715922000038

[2] R. Yu, D. Yang, and H. Zhang, "Edge-assisted collaborative perception in autonomous driving: A reflection on communication design," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, 2021, pp. 371–375.

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2020.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, pp. 1–14, 2014.

[7] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[8] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[9] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.

[10] L. Rakai, H. Song, S. Sun, W. Zhang, and Y. Yang, "Data association in multiple object tracking: A survey of recent techniques," *Expert systems with applications*, vol. 192, p. 116300, 2022.

[11] S. Thornton, B. Flowers, and S. Dey, "Multi-source feature fusion for object detection association in connected vehicle environments," *IEEE Access*, vol. 10, pp. 131 841–131 854, 2022.

[12] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[13] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019.

[14] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[15] G. Zamanakos, L. Tsochatzidis, A. Amanatiadis, and I. Pratikakis, "A comprehensive survey of lidar-based 3d object detection methods with deep learning for autonomous driving," *Computers Graphics*, vol. 99, pp. 153–181, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849321001321

[16] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 770–779.

[17] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7459–7468.

[18] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3337

[19] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 689–12 697.

[20] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652–7660.

[21] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast point r-cnn," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9774–9783.

[22] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, "Pyramid r-cnn: Towards better performance and adaptability for 3d object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2703–2712.

[23] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 669–12 678.

[24] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "Rangedet: In defense of range view for lidar-based 3d object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2898–2907.

[25] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6568–6577.

[26] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3d object detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3288–3297.

[27] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8437–8445.

[28] K. Huang, T. Wu, H. Su, and W. H. Hsu, "Monodtr: Monocular 3d object detection with depth-aware transformer," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4002–4011.

[29] A. Kundu, Y. Li, and J. M. Rehg, "3d-rcnn: Instance-level 3d object reconstruction via render-and-compare," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3559–3568.

[30] X. Zhou, Y. Peng, C. Long, F. Ren, and C. Shi, "Monet3d: towards accurate monocular 3d object localization in real time," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.

[31] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7636–7644.

[32] Y. Liu, L. Wang, and M. Liu, "Yolostereo3d: A step back to 2d for efficient stereo 3d detection," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 018–13 024.

[33] Y. Wang, B. Yang, R. Hu, M. Liang, and R. Urtasun, "Plumenet: Efficient 3d object detection from stereo images," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3383–3390.

[34] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, , and J. M. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *The Conference on Robot Learning (CoRL)*, 2021.

[35] D. Rukhovich, A. Vorontsova, and A. Konushin, "Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 1265–1274.

[36] J. Mao, S. Shi, X. Wang, and H. Li, "3d object detection for autonomous driving: A comprehensive survey," *International Journal of Computer Vision*, vol. 131, no. 8, pp. 1909–1963, 2023.

[37] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: A survey," *Pattern Recognition*, vol. 130, p. 108796, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320322002771

[38] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2583–2589.

[39] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, and Z. Nie, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 21 329–21 338.

[40] Z. T. X. X. M.-H. Y. J. M. Runsheng Xu, Hao Xiang, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[41] 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

[42] Y. Hu, Y. Lu, R. Xu, W. Xie, S. Chen, and Y. Wang, "Collaboration helps camera overtake lidar in 3d detection," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9243–9252.

[43] 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 *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2020, p. 605–621. [Online]. Available: https://doi.org/10.1007/978-3-030-58536-5_36

[44] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," in *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

[45] Y. Zhou, J. Xiao, Y. Zhou, and G. Loianno, "Multi-robot collaborative perception with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2289–2296, 2022.

[46] H. X. W. S. B. Z. J. M. Runsheng Xu, Zhengzhong Tu, "Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers," in *Conference on Robot Learning (CoRL)*, 2022.

[47] Z. Wang, S. Fan, X. Huo, T. Xu, Y. Wang, J. Liu, Y. Chen, and Y.-Q. Zhang, "Vimi: Vehicle-infrastructure multi-view intermediate fusion for camera-based 3d object detection," *ArXiv*, vol. abs/2303.10975, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257632160

[48] H. Xiang, R. Xu, and J. Ma, "Hm-vit: Hetero-modal vehicle-to-vehicle cooperative perception with vision transformer," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 284–295.

[49] Y. Lu, Y. Hu, Y. Zhong, D. Wang, Y. Wang, and S. Chen, "An extensible framework for open heterogeneous collaborative perception," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=KkrDUGIASk

[50] Y. Yuan, H. Cheng, and M. Sester, "Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3054–3061, 2022.

[51] Z. L. Y. Z. S. C. Yue Hu, Shaoheng Fang, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," in *Thirty-sixth Conference on Neural Information Processing Systems (Neurips)*, November 2022.

[52] J. Li, R. Xu, X. Liu, J. Ma, Z. Chi, J. Ma, and H. Yu, "Learning for vehicle-to-vehicle cooperative perception under lossy communication," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 2650–2660, 2023.

[53] Z. Lei, S. Ren, Y. Hu, W. Zhang, and S. Chen, "Latency-aware collaborative perception," in *European Conference on Computer Vision*. Springer, 2022, pp. 316–332.

[54] R. Meneguette, R. De Grande, J. Ueyama, G. P. R. Filho, and E. Madeira, "Vehicular edge computing: Architecture, resource management, security, and challenges," *ACM Comput. Surv.*, vol. 55, no. 1, nov 2021. [Online]. Available: https://doi.org/10.1145/3485129

[55] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. S. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62 624–62 632, 2019.

[56] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2019.

[57] Y. Liu, S. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[58] W. Fan, M. Hua, Y. Zhang, Y. Su, X. Li, B. Tang, F. Wu, and Y. Liu, "Game-based task offloading and resource allocation for vehicular edge computing with edge-edge cooperation," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 6, pp. 7857–7870, 2023.

[59] Q. Luo, C. Li, T. H. Luan, and W. Shi, "Minimizing the delay and cost of computation offloading for vehicular edge computing," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2897–2909, 2022.

[60] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via markov decision process in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2483–2493, 2021.

[61] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3341–3356, 2020.

[62] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[63] Y. Liu, S. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[64] B. Cao, Z. Li, X. Liu, Z. Lv, and H. He, "Mobility-aware multiobjective task offloading for vehicular edge computing in digital twin environment," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3046–3055, 2023.

[65] X. Xu, S. Tang, L. Qi, X. Zhou, F. Dai, and W. Dou, "Cnn partitioning and offloading for vehicular edge networks in web3," *IEEE Communications Magazine*, vol. 61, no. 8, pp. 36–42, 2023.

[66] M. K. Hasan, N. Jahan, M. Z. A. Nazri, S. Islam, M. Attique Khan, A. I. Alzahrani, N. Alalwan, and Y. Nam, "Federated learning for

computational offloading and resource management of vehicular edge computing in 6g-v2x network," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3827–3847, 2024.
[67] W. Zhan, C. Luo, J. Wang, C. Wang, G. Min, H. Duan, and Q. Zhu, "Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5449–5465, 2020.
[68] Z. Ning, P. Dong, X. Wang, J. J. P. C. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, oct 2019. [Online]. Available: https://doi.org/10.1145/3317572
[69] J. Lin, S. Huang, H. Zhang, X. Yang, and P. Zhao, "A deep-reinforcement-learning-based computation offloading with mobile vehicles in vehicular edge computing," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15 501–15 514, 2023.
[70] S. Thornton and S. Dey, "Multi-modal data and model reduction for enabling edge fusion in connected vehicle environments," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 11 979–11 994, 2024.
[71] Y. Han, H. Zhang, H. Li, Y. Jin, C. Lang, and Y. Li, "Collaborative perception in autonomous driving: Methods, datasets, and challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 6, pp. 131–151, 2023.
[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
[73] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: A 5g dataset with channel and context metrics," in *Proceedings of the 11th ACM Multimedia Systems Conference*, ser. MMSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 303–308. [Online]. Available: https://doi.org/10.1145/3339825.3394938
[74] A. Pandharipande, C.-H. Cheng, J. Dauwels, S. Z. Gurbuz, J. Ibanez-Guzman, G. Li, A. Piazzoni, P. Wang, and A. Santra, "Sensing and machine learning for automotive perception: A review," *IEEE Sensors Journal*, vol. 23, no. 11, pp. 11 097–11 115, 2023.
[75] R. Roriz, J. Cabral, and T. Gomes, "Automotive lidar technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6282–6297, 2022.
[76] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Proceedings of the European Conference on Computer Vision*, 2020.
[77] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[78] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

Samuel Thornton (Student Member, IEEE) received his B.S. in Electrical Engineering from the University of Southern California in 2016 and his M.S. in Electrical Engineering with a focus in Intelligent Robotics, Systems, and Control from the University of California, San Diego in 2020. He is currently a Ph.D. candidate in Electrical Engineering at the University of California, San Diego.

He was an undergraduate research assistant in the Magnetic Resonance Engineering Laboratory at the University of Southern California from 2014-2015 and has been a graduate student researcher at the Mobile Systems Design Laboratory at the University of California, San Diego since 2017. His research focus is in machine learning, computer vision, and edge computing with a focus in collaborative vehicular applications.

Nithin Santhanam is a research engineer at Honda Research Institute, USA. He graduated with a B.S. in Applied Mathematics with a concentration in Computer Science from the University of Pittsburgh in 2017 and has been with Honda since 2020. He currently works on the 99p Labs Software-defined Intelligence Research team in Columbus Ohio. He specializes in perception, environment recreation, simulation and Digital Twin research through software development, data insights and robotics.

Rajeev Chhajer is a technical expert and an active contributor to the mobility industry with 20 years of experience including having worked on his own startups. Currently, Rajeev leads the Software-defined Intelligence research domain at Honda Research Institute. His research involves thinking about technologies and systems in a smart city ecosystem from the viewpoint of the computing continuum and connectivity, embedded systems and informative data to enable convenient, accessible, sustainable, and efficient mobility solutions for the future. Rajeev is a founding member of 99P Labs jointly formed between Honda and The Ohio State University with the idea to do collaborative research and innovation with public partners, startups and Universities in the broad domains of Mobility, Energy and Data.

Sujit Dey (Fellow, IEEE) is a Professor in the Department of Electrical and Computer Engineering, the Director of the Center for Wireless Communications, and the Director of the Institute for the Global Entrepreneur at University of California, San Diego. He heads the Mobile Systems Design Laboratory, developing innovative and sustainable edge computing, networking and communications, multi-modal sensor fusion, and deep learning algorithms and architectures to enable predictive personalized health, immersive multimedia, and smart transportation applications. He has created inter-disciplinary programs involving multiple UCSD schools as well as community, city and industry partners; notably the Connected Health Program in 2016 and the Smart Transportation Innovation Program in 2018. In 2017, he was appointed as an Adjunct Professor, Rady School of Management, and the Jacobs Family Endowed Chair in Engineering Management Leadership.

Dr. Dey served as the Faculty Director of the von Liebig Entrepreneurism Center from 2013-2015, and as the Chief Scientist, Mobile Networks, at Allot Communications from 2012-2013. In 2015, he co-founded igrenEnergi, providing intelligent battery technology and solutions for EV mobility services. He founded Ortiva Wireless in 2004, where he served as its founding CEO and later as CTO and Chief Technologist till its acquisition by Allot Communications in 2012. Prior to Ortiva, he served as the Chair of the Advisory Board of Zyray Wireless till its acquisition by Broadcom in 2004, and as an advisor to multiple companies including ST Microelectronics and NEC. Prior to joining UCSD in 1997, he was a Senior Research Staff Member at NEC C&C Research Laboratories in Princeton, NJ. He received his Ph.D. in Computer Science from Duke University in 1991.

Dr. Dey has co-authored more than 250 publications, and a book on low-power design. He holds 18 U.S. and 2 international patents, resulting in multiple technology licensing and commercialization. He has been a recipient of nine IEEE/ACM Best Paper Awards, and has chaired multiple IEEE conferences and workshops. Dr. Dey is a Fellow of the IEEE.