

Load-Aware Agent Selection via Delay Prediction in V2X Cooperative Perception

1st Sumiko Miyata

*School of Engineering
Institute of Science Tokyo
Tokyo, Japan
sumiko@ict.eng.isct.ac.jp*

2nd Takamichi Miyata

*Faculty of Advanced Engineering
Chiba Institute of Technology
Chiba, Japan
takamichi.miyata@it-chiba.ac.jp*

3rd Yasir Ali

*School of Architecture, Building, and Civil Engineering
Loughborough University
Loughborough, UK
y.y.ali@lboro.ac.uk*

4th Irfanullah Khan

*School of Computing Science
The University of Glasgow
Glasgow, UK
irfanullah.khan@glasgow.ac.uk*

5th Paul Harvey

*School of Computing Science
The University of Glasgow
Glasgow, UK
paul.harvey@glasgow.ac.uk*

Abstract—In V2X cooperative perception, vehicles and roadside units share LiDAR features to overcome occlusion and limited sensing range. However, existing frameworks, such as V2X-ViT, do not account for computational delays in processing this data caused by GPU queue congestion at surrounding agents. Sharing delayed features can degrade perception accuracy and waste limited network bandwidth. In this paper, we propose a load-aware agent selection mechanism that predicts feature latency using an M/G/1 queueing model from GPU load information obtained before full feature transmission. The method requests features only from agents with predicted latency below a predefined threshold. Experiments on the V2XSet dataset show that it reduces unnecessary communication traffic while maintaining or improving 3D object detection performance in high-load scenarios.

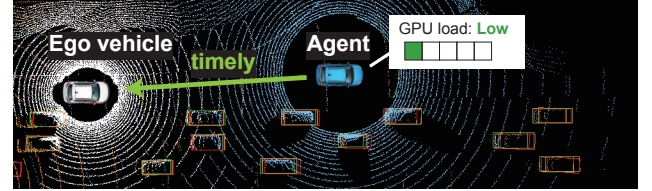
Index Terms—V2X cooperative perception, LiDAR-based 3D object detection, queueing theory, M/G/1 queue, load-aware agent selection

I. INTRODUCTION

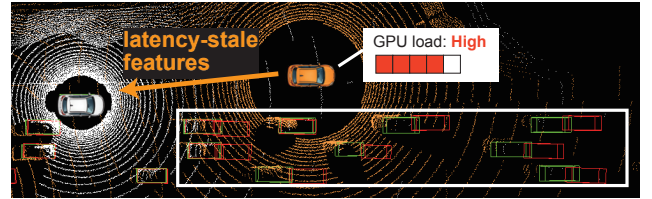
The Internet of Things (IoT) is revolutionizing intelligent transportation systems (ITS) by enabling seamless connectivity among vehicles, infrastructure, and pedestrians [1]. This IoT ecosystem facilitates the exchange of critical sensor data, essential for ensuring the safety of autonomous vehicles through precise environmental perception. However, single-vehicle perception faces physical limits such as occlusions and sparse long-range observations, which pose significant risks [2]. Such gaps are further exacerbated by behavioral uncertainties in mixed traffic, necessitating robust information sharing [3]. To address these challenges, Vehicle-to-Everything (V2X) cooperative perception has emerged as a key solution [1], [4], enhancing environmental understanding by integrating diverse visual perspectives shared among vehicles (V2V) and infrastructure (V2I).

While early frameworks such as F-Cooper [5] and V2VNet [6] pioneered intermediate feature fusion, they generally operated under the assumption of ideal communication without

Timely intermediate features **enhance** detection accuracy



However, **high** GPU load at agent increases queueing delay



Latency-stale features may **degrade** detection accuracy

Fig. 1: Effect of Latency-Stale Features. Green and red bound boxes show ground truth and predictions, respectively.

significant latencies. However, asynchronous information sharing and communication latencies present formidable barriers to realizing robust cooperative systems. To mitigate these issues, the state-of-the-art framework, V2X-ViT, incorporates Delay-aware Positional Encoding (DPE) to adaptively compensate for spatial mismatches caused by mild temporal delays within feature maps [7].

Despite its efficacy, recent studies including V2X-Real [8] have highlighted that V2X-ViT and other existing frameworks operate under the assumption of instantaneous feature extraction by cooperating agents (e.g. ego vehicles or road side units (RSU)). This assumption overlooks dynamic computational delays stemming from GPU processing queue congestion [7], [8]. In latency-sensitive tasks like autonomous driving, significantly delayed features fail to contribute positively to perception accuracy. Requesting features from all agents within range

TABLE I: Comparison of V2X cooperative perception methods across key functional aspects.

Features	F-Cooper [5] V2VNet [6]	V2X-ViT [7]	Ours
Intermediate Feature Fusion	✓	✓	✓
Resilience to Mild Transmission Delay		✓	✓
Load-aware Agent Selection for Bandwidth Efficiency			✓

without accounting for their computational load can result in requests to agents with congested GPU queues. Consequently, this results in the wastage of critical network bandwidth on features with reduced utility due to latency, referred to hereafter as latency-stale features, that either remain unused or degrade overall accuracy due to excessive latency. Fig. 1 illustrates this key motivation: timely intermediate features can enhance cooperative perception, whereas latency-stale features from congested agents may misalign with the current scene and degrade detection accuracy.

Preventing such bandwidth inefficiency and cascaded delay-induced accuracy degradation requires a mechanism that dynamically assesses whether data can be received within an allowable latency threshold prior to transmission. This study introduces a dynamic load-aware agent selection method by integrating queueing theory [9], [10] into V2X cooperative perception. Specifically, the proposed method formulates the GPU-based inference process of each agent as an M/G/1 non-preemptive priority queueing model. Prior to full feature transmission, the ego vehicle acquires the current load status of potential collaborators to predict processing overhead. This predictive model enables the system to proactively refrain from requesting features from agents whose total projected latency exceeds a predefined threshold.

Table I provides a comparative summary of the proposed method against conventional intermediate fusion methods across several evaluation axes. Unlike those methods [5]–[7], the proposed approach achieves both bandwidth efficiency and load-aware agent selection via queueing-based delay prediction while maintaining superior perception accuracy.

The primary contributions of this work are as follows:

- **Cross-disciplinary Modeling:** Application of an M/G/1 queueing model to the V2X-ViT framework establishes a delay prediction model that explicitly incorporates both communication and processing overhead.
- **Threshold-based Dynamic Control:** A proactive selection mechanism coordinates communication by filtering redundant bandwidth consumption and suppresses cascaded frame delays and accuracy degradation under high-load conditions.
- **Robustness in High-Load Scenarios:** Performance evaluations using the V2XSet dataset demonstrate that the proposed method significantly reduces communication traffic while maintaining or improving 3D object detection accuracy compared to V2X-ViT in environments

with congested agents.

II. RELATED WORK

A. Cooperative Perception and Intermediate Fusion

Single-agent perception faces physical constraints like occlusions and sparse long-range observations, increasing accident risks [4]. To mitigate these, cooperative perception enables information sharing among vehicles and infrastructure [4]. Approaches vary by data granularity: *early fusion* shares raw data but consumes excessive bandwidth [11], while *late fusion* shares only detection results, which minimizes traffic but lacks contextual richness for robust integration [12].

Intermediate fusion shares encoder-extracted features, balancing accuracy and bandwidth [5], [6]. For instance, V2X-ViT introduced Delay-aware Positional Encoding (DPE) to compensate for spatio-temporal mismatches by embedding latency into feature maps [7]. However, these methods focus on post-hoc correction *after* features are received [7]. They lack proactive mechanisms to manage structural delays caused by GPU congestion at source agents, leading to the inefficient acquisition of “latency-stale” features that have already lost their utility.

B. Queueing Theory and Task Control in Edge Environments

Transmitting large-scale LiDAR data in bandwidth-constrained edge environments introduces significant communication and processing overhead [13]. Queueing theory provides a computationally efficient tool to analyze these latencies by modeling arrival rates, service times, and priorities [9], [14]. Theoretical analysis shows that residence time responds non-linearly to resource utilization, with processing delays escalating exponentially as agents approach maximum capacity [10].

Prior research has utilized queueing models, such as M/G/1, for task allocation and offloading (e.g., DSO [13] and TTS [14]) to optimize latency for real-time tasks coexisting with delay-tolerant ones. While effective for offloading, these approaches have not yet been integrated into cooperative perception to regulate feature sharing. This study bridges this gap by proposing a load-aware agent selection mechanism. By predicting processing overhead via M/G/1 theory, our method enables the ego vehicle to proactively refrain from requesting features from congested agents, thereby ensuring communication efficiency.

III. SYSTEM MODEL

This section presents the proposed load-aware V2X perception framework, which determines whether each cooperating agent should participate in feature fusion before full feature transmission. As illustrated in Fig. 2, the framework consists of four steps: Step 1, metadata sharing; Step 2, load-aware agent selection with delay prediction; Step 3, feature extraction and sharing; and Step 4, 3D object detection. Algorithm 1 summarizes the overall procedure.

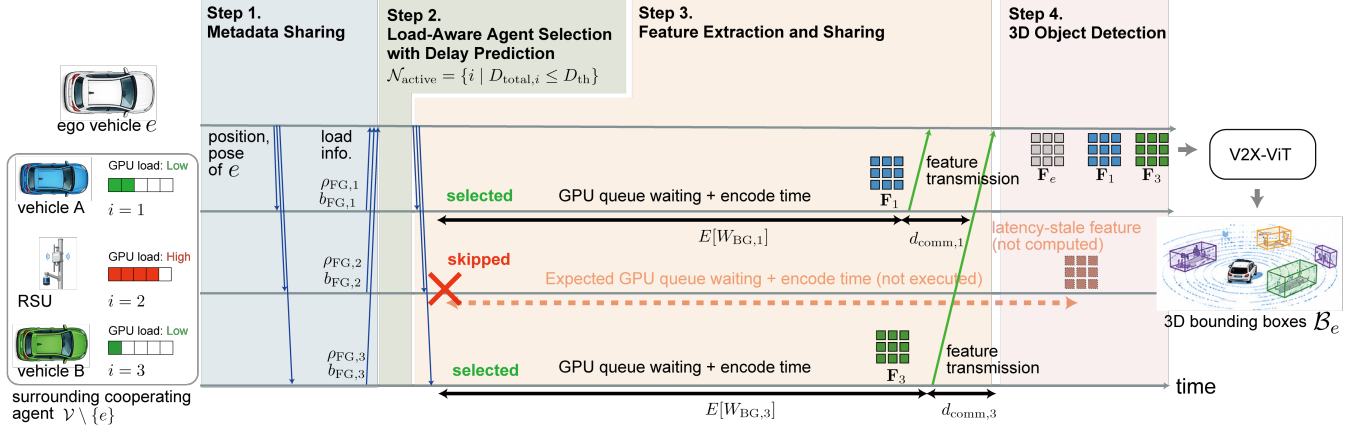


Fig. 2: Overview of the proposed dynamic load-aware agent selection. Our proposed method proactively assesses the GPU processing load status of each agent prior to feature transmission. If the predicted average latency based on the load exceeds a predefined threshold, the system refrains from requesting features, thereby achieving both perception accuracy maintenance and communication traffic reduction.

TABLE II: Summary of major notations.

Notation	Description
$\mathcal{V}, N$	Set and total number of agents within range
e, i	Ego vehicle and a specific cooperating agent
P_e, P_i	Raw LiDAR point clouds at ego and agent i
F_e, F_i	Intermediate feature maps extracted by encoders
$\lambda_{FG,i}, \lambda_{BG}$	Arrival rates of FG and BG tasks at agent i
$b_{FG,i}, b_{BG}$	Mean service times for FG and BG tasks
$b_{FG,i}^{(2)}, b_{BG}^{(2)}$	Second moments of service times
$\rho_{FG,i}, \rho_{BG}$	Load factors for FG and BG tasks
$E[W_{BG,i}]$	Expected system residence time for BG tasks
$d_{comm,i}$	Communication latency between agent i and e
$D_{total,i}$	Predicted end-to-end latency for features from i
D_{th}	Maximum allowable latency threshold
$\mathcal{N}_{active}$	Set of agents selected for feature fusion
B_e	Set of final detected 3D bounding boxes

A. Step 1: Metadata Sharing

Let $\mathcal{V}$ denote the set of N agents within the communication range, including vehicles and roadside units (RSUs), and let $e \in \mathcal{V}$ denote the ego vehicle. The major notations used throughout this section are summarized in Table II. For each candidate cooperating agent $i \in \mathcal{V} \setminus \{e\}$, the metadata sharing process starts with the ego vehicle broadcasting its position and pose before requesting intermediate features. As shown in Fig. 2 and Algorithm 1, this lightweight exchange also includes compact load-related information required for delay prediction.

The position and pose of the ego vehicle e are used by each cooperating agent to align its local point cloud with the ego coordinate frame before feature extraction. The load-related information is then used to estimate whether the feature map from agent i can be delivered within the allowable latency threshold D_{th} . At this stage, the agents exchange only scalar metadata, rather than the full intermediate feature map F_i , thereby enabling agent selection before feature transmission. The queueing-based delay prediction model that uses this load

information is described in the next subsection.

B. Step 2: Load-Aware Agent Selection with Delay Prediction

Using the load-related information obtained through metadata sharing, the ego vehicle predicts the end-to-end latency of each candidate cooperating agent before requesting its intermediate feature map. The computational resource of each cooperating agent $i \in \mathcal{V} \setminus \{e\}$ is modeled as a single-server M/G/1 non-preemptive priority queue. Tasks are classified into high-priority foreground (FG) tasks, which represent the agent's own perception processing, and low-priority background (BG) tasks, which are generated by cooperation requests from the ego vehicle.

We assume that FG and BG tasks arrive at agent i according to Poisson processes with average arrival rates $\lambda_{FG,i}$ and λ_{BG} , respectively. Let $b_{FG,i}$ and b_{BG} denote the mean service times, and let $b_{FG,i}^{(2)}$ and $b_{BG}^{(2)}$ denote the second moments of the service times. The average system residence time $E[W_{BG,i}]$ of a BG task, defined as the time from the reception of a cooperation request at agent i to the completion of feature computation, is given by [10]:

$$E[W_{BG,i}] = b_{BG} + \frac{\frac{1}{2} (\lambda_{FG,i} b_{FG,i}^{(2)} + \lambda_{BG} b_{BG}^{(2)})}{(1 - \rho_{FG,i})(1 - \rho_{FG,i} - \rho_{BG})}, \quad (1)$$

where $\rho_{FG,i} = \lambda_{FG,i} b_{FG,i}$ and $\rho_{BG} = \lambda_{BG} b_{BG}$ are the load factors of FG and BG tasks, respectively.

From the perspective of the ego vehicle, the total predicted delay for receiving the feature map from agent i is obtained by adding the communication latency $d_{comm,i}$ to the queueing-based residence time:

$$D_{total,i} = d_{comm,i} + E[W_{BG,i}]. \quad (2)$$

This delay prediction is used to determine whether the feature map from agent i is expected to arrive within the allowable latency threshold D_{th} .

Algorithm 1 Queueing-Based Load-Aware Agent Selection for V2X-ViT

Require: Agent set $\mathcal{V}$ including ego e and latency threshold D_{th}

Ensure: Detected 3D bounding boxes $\mathcal{B}_e$

```

1: Initialize the collaborator set  $\mathcal{N}_{active} \leftarrow \emptyset$ 
2: Extract ego feature map:  $\mathbf{F}_e = \text{Encoder}(\mathbf{P}_e)$ 
3: for each candidate agent  $i \in \mathcal{V} \setminus \{e\}$  do
4:   Step 1, Metadata Sharing: Ego vehicle  $e$  sends its position
     and pose to each  $i$ . Each agent  $i$  sends  $\{\rho_{FG,i}, b_{FG,i}\}$  to  $e$ 
5:   Step 2, Load-Aware Agent Selection and Delay Prediction:
     estimate expected total delay  $D_{total,i}$  from  $\rho_{FG,i}$  and  $b_{FG,i}$ 
6:   if  $D_{total,i} \leq D_{th}$  then
7:     Add  $i$  to the collaborator set:  $\mathcal{N}_{active} \leftarrow \mathcal{N}_{active} \cup \{i\}$ 
8:     Step 3, Feature Extraction and Sharing: Each agent  $i \in$ 
        $\mathcal{N}_{active}$  extracts  $\mathbf{F}_i = \text{Encoder}(\mathbf{P}_i)$  and transmits  $\mathbf{F}_i$  to ego
       vehicle  $e$ 
9:   else
10:    Skip agent  $i$  and suppress feature transmission
11:   end if
12: end for
13: Step 4, 3D Object Detection:
     $\mathcal{B}_e = \text{V2X-ViT}(\{\mathbf{F}_e\} \cup \{\mathbf{F}_i \mid i \in \mathcal{N}_{active}\})$ 
14: return  $\mathcal{B}_e$ 

```

The ego vehicle then constructs the active collaborator set as

$$\mathcal{N}_{active} = \{i \in \mathcal{V} \setminus \{e\} \mid D_{total,i} \leq D_{th}\}. \quad (3)$$

Only agents in $\mathcal{N}_{active}$ are selected for subsequent feature exchange and fusion. By excluding agents whose predicted delay exceeds D_{th} , the proposed method avoids latency-stale features from highly loaded agents and reduces redundant communication traffic.

C. Step 3: Feature Extraction and Sharing

Each selected agent $i \in \mathcal{N}_{active}$ transforms its local LiDAR point cloud $\mathbf{P}_i$ according to the ego vehicle's position and pose shared in the metadata exchange step, and then extracts an intermediate feature map $\mathbf{F}_i$ using the PointPillars backbone [15]. The extracted feature map is compressed along the channel dimension and transmitted to the ego vehicle via wireless communication.

D. Step 4: 3D Object Detection

Once the ego vehicle receives the intermediate feature maps from the selected agents, it performs feature fusion using V2X-ViT. Let $\mathbf{F}_e$ denote the ego vehicle's own feature map extracted from its LiDAR point cloud $\mathbf{P}_e$. The input to the fusion network is then given by the ego feature map and the feature maps received from the active collaborators, i.e., $\{\mathbf{F}_e\} \cup \{\mathbf{F}_i \mid i \in \mathcal{N}_{active}\}$. The set of 3D bounding boxes $\mathcal{B}_e$ is obtained as

$$\mathcal{B}_e = \text{V2X-ViT}(\{\mathbf{F}_e\} \cup \{\mathbf{F}_i \mid i \in \mathcal{N}_{active}\}). \quad (4)$$

Because the fusion process uses only agents satisfying the latency constraint, the proposed framework suppresses the adverse effect of delayed features while preserving the benefit of cooperative perception.

IV. NUMERICAL RESULTS

This section evaluates the effectiveness of the proposed method through comprehensive numerical experiments.

TABLE III: Parameter Settings for Delay Prediction and Simulation

Category	Parameter	Value / Setting
FG Tasks	$\lambda_{FG,i}$	10 tasks/s (with prob. p)
	$b_{FG,i}$	5 tasks/s (with prob. $1 - p$)
	p (High load prob.)	65 ms Range [0, 1] in steps of 0.1
BG Tasks	b_{BG}	10 ms
	λ_{BG}	10 tasks/s
Calculated	$E[W_{BG,i}]$	499 (High) / 66 (Low) ms
	$d_{comm,i}$	50 ms
	D_{th}	500 ms

A. Experimental Setup

Dataset. To validate the proposed mechanism, we employ the V2XSet dataset [7], which was constructed using the high-fidelity simulators. The dataset contains diverse scenes featuring at least two and up to seven agents, encompassing various combinations of ego vehicles, surrounding cooperative vehicles, and RSUs. Each agent is equipped with a 32-channel LiDAR sensor with a sensing range of 120 meters, recording point clouds at a frequency of 10 Hz. Given this range, the experimental scenarios are designed within a geographical space where the sensing fields of multiple agents significantly overlap, enabling the evaluation of feature-level fusion in resolving occlusions and enhancing distant object detection. The scenarios cover various road layouts such as straight roads, curves, and intersections. For evaluation, we use the test set consisting of 2,833 frames.

Baselines. The performance of the proposed method is compared against two baseline approaches:

- **Non-fusion:** Performs single-agent perception using only an ego vehicle's LiDAR data.
- **V2X-ViT [7]:** Serves as the state-of-the-art cooperative framework that unconditionally communicates ($\mathcal{N}_{active} = \mathcal{N} \setminus \{e\}$) and fuses features from all agents within the communication range.

The proposed method dynamically excludes agents whose predicted total latency $D_{total,i}$ exceeds the threshold D_{th} based on the M/G/1 queueing model.

Parameter Settings. The specific parameters used for the M/G/1 queueing model and delay calculations are summarized in Table III. The foreground (FG) task service times are modeled with an exponential distribution, while the background (BG) task for point cloud encoding is set as a constant service time based on empirical measurements from V2X-ViT [7]. Under these configurations, the average system residence time $E[W_{BG,i}]$ and the total delay $D_{total,i}$ are calculated for both high-load and low-load states. The ego vehicle then performs agent selection relative to the predefined threshold D_{th} , which is set to exclude significantly latency-stale features from high-load agents.

Other parameters, including communication noise models, follow the configurations established in [7]. Since the pre-transmission exchange only a small number of scalar load

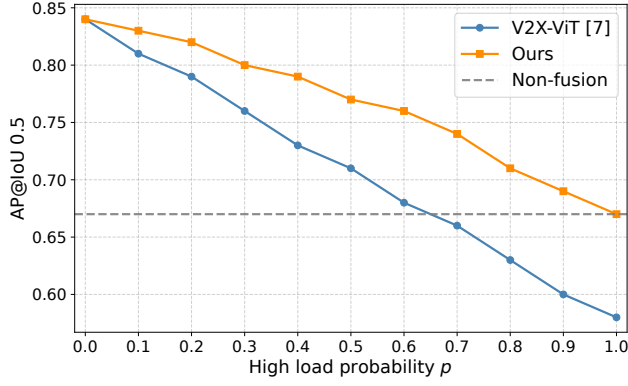


Fig. 3: Object detection accuracy (AP@IoU0.5) vs. high-load probability p . Higher is better.

parameters, its communication overhead is negligible and is not included in the traffic measurements.

Metrics. Following the evaluation protocol of V2X-ViT [7], we assess 3D object detection performance for vehicle objects using *Average Precision (AP)*. A detection is considered correct if the *Intersection over Union (IoU)*, which measures the overlap ratio between the predicted 3D bounding box and the ground truth, exceeds 0.5. The AP@IoU0.5 is then calculated as the area under the Precision-Recall curve by varying the threshold on detection confidence scores.

B. Impact of System Load on Perception Accuracy

Fig. 3 illustrates the impact of the high-load probability p on the object detection accuracy (AP@IoU0.5). Here, p is defined as the probability that each cooperating agent is in a high-load state, which we refer to as the high-load probability. For the baseline V2X-ViT, an increase in p raises the probability that features arriving at the ego vehicle experience significant latencies. This excessive latency exceeds the compensation capability of Delay-aware Positional Encoding (DPE), a key feature of V2X-ViT, resulting in a sharp decline in AP. Specifically, in regions where $p > 0.7$, the perception accuracy of V2X-ViT falls below that of the non-fusion baseline.

In contrast, the proposed method effectively mitigates this degradation by selectively fusing only low-latency features and refraining from communication with agents predicted to be in high-load states. This strategy enables the system to maintain high perception accuracy while simultaneously suppressing redundant bandwidth consumption even under high-load conditions. As expected, at $p = 1$, the AP of the proposed method converges with the non-fusion baseline (grey dotted line), since all agents are excluded from collaboration.

C. Communication Traffic Reduction

Fig. 4 shows the average communication traffic as the high-load probability p varies, measured as the amount of feature-map data received by the ego vehicle per frame. In V2X-ViT, which unconditionally communicates with all agents, the

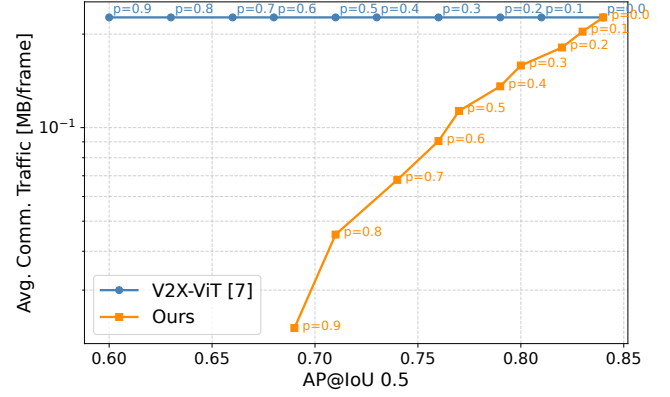


Fig. 4: Average Communication Traffic per frame versus AP@IoU0.5 with different p .

communication volume remains constant regardless of the network load. In contrast, the proposed method adaptively reduces the communication volume as p increases.

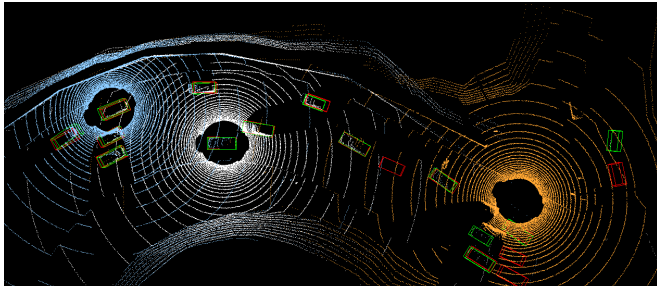
Importantly, as shown in Fig. 4, this reduction in communication traffic is achieved simultaneously with improved perception accuracy. This result suggests that the proposed method effectively utilizes limited communication bandwidth by selectively excluding significantly delayed data that does not contribute to accuracy improvement. In particular, under the condition of $p = 0.5$, the proposed method improves detection accuracy by approximately 6 percentage points compared with V2X-ViT, while reducing communication traffic by approximately 50%.

D. Qualitative Results

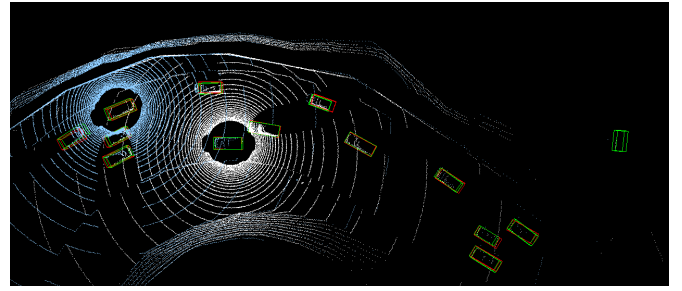
Fig. 5 presents qualitative detection results under latency-aware collaborative perception. The visualization highlights how delayed information from cooperative agents affects object localization in feature-level fusion. In particular, V2X-ViT (Fig. 5a) shares features from all cooperative agents regardless of their latency, which can introduce spatial misalignment and lead to inaccurate 3D bounding-box predictions. In contrast, our method (Fig. 5b) selectively excludes agents whose average latency exceeds the predefined threshold. This design prevents unreliable delayed features from being fused, thereby reducing communication cost while improving detection accuracy.

V. CONCLUSION

This paper proposed a dynamic load-aware agent selection mechanism to mitigate computational delays and bandwidth waste in V2X cooperative perception. By predicting feature latency via an M/G/1 queueing model, our method selectively requested features only from agents capable of meeting the latency threshold. Evaluation on the V2XSet dataset demonstrated that while conventional unconditional cooperation suffered significant AP degradation under high



(a) V2X-ViT [7]



(b) Ours

Fig. 5: Qualitative comparison with latency-aware collaborative perception. Green and red 3D bounding boxes represent the ground truth and prediction, respectively. White, pale-blue, and orange point clouds denote measurements captured by the ego vehicle, cooperative agents, and cooperative agents whose average latency exceeds the threshold, respectively.

server loads, the proposed approach maintained stable accuracy while reducing network traffic. These results highlighted the necessity of latency-aware feature acquisition when computational overhead was non-negligible. Furthermore, the proposed load-aware criterion is orthogonal to existing communication-efficient frameworks such as When2com [16] and Where2comm [17]. By integrating our computational-load perspective with their spatial and temporal selection strategies, a synergistic reduction in both latency and network traffic can be achieved.

Future work includes evaluating the method under dynamic wireless environments, extending it to resource contention among multiple ego vehicles, and integrating latency-aware selection with visual utility and energy-aware resource allocation, such as service function chaining [18].

DECLARATION OF AI TOOL USAGE

ChatGPT, Gemini, and Claude were used to assist in writing parts of the experimental code and in revising the English manuscript. The authors take full responsibility for the accuracy and integrity of the reported work.

ACKNOWLEDGMENT

This research is partially supported by JSPS KAKENHI 23K03871, the NICT (24101), and EPSRC grant EP/Z533221/1.

REFERENCES

- [1] S. Miyata, T. Miyata, Y. Ali, and A. Filtness, "Nucleolus-Based Incentive Mechanism for V2X Cooperative Perception toward Road Safety," in *Proceedings of the IEEE International Conference on Communications (ICC), IoT and Sensor Networks Symposium*, 2026 (to appear).
- [2] E. Papazikou, R. Talbot, L. Brown, S. Maynard, and A. Filtness, "Unpacking the relationship between task complexity and driving risk: Insights from a uk on-road trial," *IATSS Research*, vol. 49, no. 2, pp. 127–136, 2025.
- [3] M. T. Kashifi, A. Sharma, Y. Ali, and H. Haitao, "Benchmarking intra-driver heterogeneity car-following models using a behavioural and numerical evaluation framework," *Transportation Research Part B: Methodological*, vol. 205, p. 103399, 2026.
- [4] C. Wei, G. Wu, and M. J. Barth, "Cooperative Perception for Automated Driving: A Survey of Algorithms, Applications, and Future Directions," *Proceedings of the IEEE*, pp. 1–27, 2025.
- [5] 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 ACM/IEEE Symposium on Edge Computing*. Association for Computing Machinery, 2019, pp. 88–100.
- [6] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction," 2020.
- [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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 107–124.
- [8] H. Xiang, Z. Zheng, X. Xia, R. Xu, L. Gao, Z. Zhou, X. Han, X. Ji, M. Li, Z. Meng, L. Jin, M. Lei, Z. Ma, Z. He, H. Ma, Y. Yuan, Y. Zhao, and J. Ma, "V2X-Real: A Large-Scale Dataset for Vehicle-to-Everything Cooperative Perception," 2024.
- [9] S. Miyata, T. Miyata, I. Khan, and P. Harvey, "Bidirectional Communication-Aware Offloading for Federated Prompt Learning at the Edge," in *Proceedings of the IEEE International Conference on Communications (ICC), Communication QoS, Reliability and Modeling Symposium*, 2026 (to appear).
- [10] D. Bertsekas and R. Gallager, *Data Networks (2nd ed.)*. USA: Prentice-Hall, Inc., 1992.
- [11] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative Perception for Connected Autonomous Vehicles Based on 3D Point Clouds," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 514–524.
- [12] 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 *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2583–2589.
- [13] J. Wang, W. Zhang, Z. Lin, and Y. Zheng, "Optimal Sequential Computation Offloading and Migration in Mobile Edge Computing with Multi-priority Tasks," *Computing*, 2025.
- [14] S. Miyata, T. Miyata, P. Harvey, and I. Khan, "On the Effectiveness of Task Off-loading in Edge based Federated Learning Environment," in *Proceedings of IEEE ICDCS International Workshop on ANMS*, 2025.
- [15] 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 (CVPR)*, 2019, pp. 12 689–12 697.
- [16] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "When2com: Multi-agent Perception via Communication Graph Grouping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3042–3052.
- [17] Y. Hu, S. Fang, Z. Xie, Y. Chen, and H. Li, "Where2comm: Communication-Efficient Collaborative Perception via Spatial Confidence Maps," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 4874–4886.
- [18] T. Krishnamohan and P. Harvey, "Energy-aware optimal service function chain embedding in emulated multi-access edge computing for internet of things," in *Proceedings of the International Workshop on Testing Distributed Internet of Things Systems (TDIS)*, 2026.