To Optimize Edge-Intelligent Cooperative Perception in Heterogeneous Vehicular Networks

Journal:	*Transactions on Mobile Computing*
Manuscript ID	TMC-2025-06-1755
Manuscript Type:	Regular
Keywords:	C.2.1 Network Architecture and Design < C.2 Communication/Networking and Information Technology < C Computer Systems Organization, C.2.7.c Sensor networks < C.2.7 Wide-area networks < C.2 Communication/Networking and Information Technology < C Computer Systems Organization, I.2.11 Distributed Artificial Intelligence < I.2 Artificial Intelligence < I Computing Methodologies, J.9 Mobile Applications < J Computer Applications

To Optimize Edge-Intelligent Cooperative Perception in Heterogeneous Vehicular Networks

Guozhi Yan, Kai Liu, *Senior Member, IEEE*, Chunhui Liu, and Lingjie Duan, *Senior Member, IEEE*

***Abstract*—Cooperative Perception (CP) has been a promising paradigm to enhance single-vehicle awareness by enabling perception sharing among connected vehicles. However, existing studies often overlook practical challenges such as resource limitation and edge heterogeneity, which lead to synchronization bottlenecks and limit deployment efficiency. To bridge this gap, we propose EI-Cooper, a novel Edge Intelligence (EI)-enhanced cooperative framework for efficient and adaptive CP in heterogeneous vehicular networks. The novelty of EI-Cooper is fourfold. First, we leverage key EI techniques including selective cooperation, model pruning and bandwidth allocation to jointly coordinate the perception, computation, communication within the CP pipeline. To the best of our knowledge, EI-Cooper represents the first attempt to extend CP with EI capabilities. Secondly, we formulate a Synchronization-Efficient Cooperative Perception (SECP) problem, which jointly determines edge selection, pruning ratios and bandwidths to balance end-to-end synchronization efficiency and perception accuracy. Thirdly, to tackle the closed-box nature and computational NP-hardness of SECP, we decompose it into two interpretable subproblems, respectively capturing macro-level spatial completeness and micro-level semantic retention. Finally, we develop a Two-Stage Hierarchical Optimization (TSHO) algorithm, where the first stage maximizes coverage via submodular node selection with a $(1-1/e)$ approximation, and the second stage performs alternating optimization of pruning and bandwidth allocation under convergence guarantees. Extensive experiments on public datasets and a real-world prototype demonstrate the superiority of EI-Cooper.**

***Index Terms*—Cooperative perception, edge intelligence, heterogeneity, perception-computation-communication coordination.**

I. Introduction

THE prevailing perception paradigm in modern vehicles primarily relies on onboard sensors to understand their surrounding environment. However, such single-vehicle perception is inherently limited by restricted sensor range and occlusions, particularly in complex traffic scenarios. With the emergence of advanced vehicular communication technologies like Cellular vehicle-to-everything (C-V2X) [1] and vehicle computing [2], cooperative perception (CP) has emerged as a promising solution, where nearby vehicles or roadside infrastructures share complementary perception information to enhance situational awareness and improve driving safety. Fig 1 illustrates an example scenario demonstrating CP importance.

Corresponding author: Kai Liu.
Guozhi Yan, Kai Liu and Chunhui Liu are with the College of Computer Science, Chongqing University, Chongqing 400040, China (email: {yanguozhiup, liukai0807, chhliu0302}@cqu.edu.cn).
Lingjie Duan is with the Pillar of Engineering Systems and Design, Singapore University of Technology and Design, Singapore 487372 (email: lingjie_duan@sutd.edu.sg).

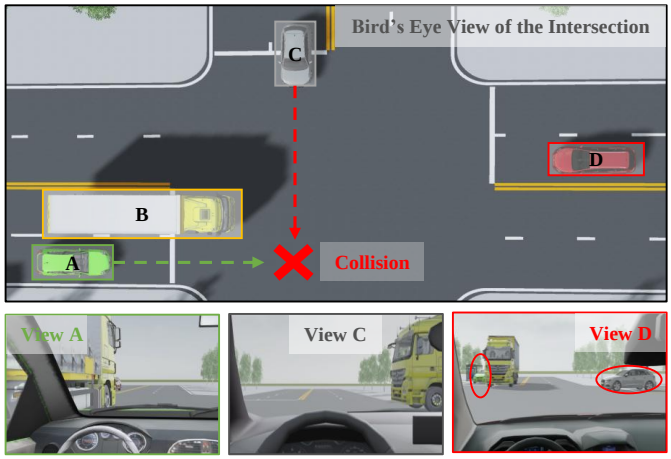


Fig. 1. An illustrative intersection scenario demonstrating CP importance. Due to the occlusion by a large truck (Vehicle B), Vehicle A and C cannot detect each other from their local viewpoints. In contrast, Vehicle D has a clear view of the intersection and can share its perception to help A or C anticipate potential collisions, demonstrating the critical role of CP in enabling comprehensive situational awareness.

According to the extent of information sharing, existing CP strategies can be broadly categorized into three types: early, late, and intermediate cooperation [3]. Early cooperation [4], [5], [6] shares raw sensory data (e.g., LiDAR or camera streams) to retain straightforward environment context, but imposes heavy communication burdens. For instance, a single vehicle may generate up to 2.3 GB of sensor data per second [7], leading to substantial transmission delays. Late cooperation [8], [9] transmits only high-level perception outputs (e.g., bounding boxes), offering low overhead but often at the cost of degraded accuracy due to incomplete or noisy individual detections. Intermediate cooperation [10], [11] strikes a balance by sharing deep feature representations, which are compact (typically under 1 MB after compression) yet informative for effective fusion. Accordingly, this work focuses on the design and optimization of intermediate cooperation strategies. Unless otherwise specified, CP refers to intermediate cooperation throughout the rest of this paper.

The above approaches each exhibit strengths in enhancing CP performance. However, most existing intermediate CP studies have primarily focused on designing fusion architectures or neural network models to improve accuracy, while overlooking practical deployment challenges in real-world vehicular networks: (1) *Scarce Communication Bandwidth*. Although feature representations can be efficiently encoded to reduce the transmission overhead, limited bandwidth in

vehicular networks still renders it infeasible to involve all cooperative vehicles simultaneously. (2) *Limited Computation Resources*. Most feature extraction models such as Deep Neural Networks (DNNs) or Transformers entail high computational complexity, posing challenges for resource-constrained vehicles. Moreover, vehicular nodes typically exhibit substantial heterogeneity in computational resources, ranging from embedded units to high-performance GPUs, leading to severe workload imbalance. (3) *Heterogeneous Synchronization Bottlenecks*. The joint impact of heterogeneous computation and communication introduces extra synchronization delays under the widely adopted Bulk Synchronous Parallel (BSP) model. In BSP-based fusion architectures, the requesting vehicle must wait for all cooperative features before fusion, rendering overall responsiveness dictated by the slowest participant, thus significantly degrading cooperation efficiency.

Some previous works have made efforts to separately address the above challenges. Firstly, numerical studies have considered task-oriented feature encoding [10], [11], node selection and bandwidth allocation [4], [12], [13] to improve the overall CP transmission efficiency. However, these approaches do not fundamentally address the underlying model complexity. The substantial computational burden imposed by deep models remains a key barrier to efficient onboard processing. To address computational constraints, edge intelligence (EI) [3], has emerged as a promising paradigm for enabling intelligent computation and data processing at the network edge. Techniques such as cooperative inference [14], [15] and model pruning [16], [17], [18] have been widely adopted to enhance the responsiveness and efficiency of edge devices. Nevertheless, most EI research has largely focused on individual devices or centralized servers, offering limited support for the distributed perception and multi-agent collaboration inherent in CP. Moreover, recent studies have also explored dynamic aggregation [19], adaptive resource allocation [20], and selective cooperation [21] to mitigate the system heterogeneity. Yet, these methods are tailored for general edge computing contexts and cannot be directly applied to CP, which imposes additional demands such as unified perception-computation-communication trade-offs, heterogeneous resource coordination, and accuracy-driven optimization.

Against this background, we propose to integrate EI into the CP pipeline, and introduce EI-Cooper, *the first attempt that extends EI capabilities to support efficient CP deployment in heterogeneous vehicular networks.* This integration is motivated by the inherent synergy between CP and EI: (1) *Model Reliance*. CP tasks, such as feature extraction and detection, rely heavily on the efficient execution of EI models at the network edge. (2) *Data Driven*. EI effectiveness in both training and inference critically depends on the distributed data collected from multiple vehicles [22]. Building on this synergy, the key contributions of this work are summarized as follows:

EI-Cooper Framework: We propose EI-Cooper, a novel EI-enhanced framework to support adaptive and efficient CP deployment in heterogeneous vehicular networks. We begin by exploring on an open benchmark dataset to reveal the impact of system heterogeneity—including disparities in computational capacity, communication bandwidth, and spatial completeness—on CP accuracy and delay. These findings motivate a unified coordination of perception, computation, and communication within the CP pipeline. EI-Cooper integrates key EI techniques, including selective cooperation, adaptive model pruning, and bandwidth allocation, to enable heterogeneity-aware and performance-efficient CP deployment.

SECP Problem: We formulate the Synchronization-Efficient Cooperative Perception (SECP) problem, which is the first to explicitly address the heterogeneous synchronization bottleneck in CP deployment. SECP jointly captures node-level computation and communication delays as well as the overall system response delay under constrained and heterogeneous resources. Moreover, the CP accuracy is modeled as a closed-box function affected by selective cooperation and feature extraction. The resulting SECP is formulated as a mixed-integer nonlinear programming (MINLP) problem to balance perception accuracy and synchronization efficiency.

SECP Transformation: To enable tractable optimization, we decouple the closed-box accuracy into two interpretable and orthogonal components: *perception coverage*, reflecting macro-level spatial completeness, and *feature quality*, representing micro-level semantic retention. The SECP problem is then transformed into two sequential subproblems: coverage maximization and feature quality–delay trade-off. We further prove the submodularity of the aggregated coverage with respect to node selection and derive an analytical approximation of feature quality as a function of pruning ratios.

TSHO Algorithm: We propose a Two-Stage Hierarchical Optimization (TSHO) algorithm to efficiently solve the transformed subproblems. In stage I, we estimate node-wise perception coverage using a lightweight BEV occupancy grid and perform coverage-aware node selection via greedy submodular maximization with a $(1 - 1/e)$ approximation guarantee. In stage II, we optimize model pruning and bandwidth allocation under heterogeneous resource constraints via an alternating optimization strategy, which provides convergence guarantee for the continuous non-convex objective.

The rest of this paper is organized as follows. Section II reviews the related work. Section III presents the motivation and design of EI-Cooper framework. Section IV give an analytical model and formulates the SECP problem. Section V introduces the problem transformation. Section VI details the design of TSHO algorithm. Section VII presents the performance evaluation. Finally, section VIII concludes this paper.

II. Related Work

Extensive research has explored CP for enhanced situational awareness. Nonetheless, constrained resources and edge heterogeneity introduce substantial computation and communication overheads, rendering CP systems especially susceptible to heterogeneous synchronization bottlenecks. Table I summarizes representative solutions targeting these challenges.

Firstly, substantial efforts have been devoted to alleviating the *communication issue* in CP[1]. Hou *et al.* [4] proposed

[1]This work focuses on network-level optimization techniques such as selective communication and bandwidth allocation for latency reduction. Other methods, such as task-oriented communication [10], [11] and feature quantization [17], are orthogonal and can be integrated into our framework.

TABLE I
ANALYSIS OF OUR WORK AGAINST EXISTING LITERATURE

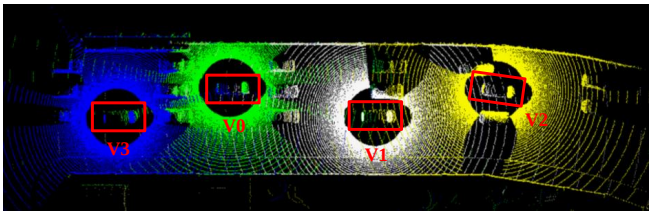
Reference	CP	COMM	COMP	HETERO
TITS'25 [4]	✓	✓	×	✓
TMC'24 [12]	✓	✓	×	×
ICRA'24 [13]	✓	✓	×	×
TON'23 [14]	×	×	✓	✓
TMC'24 [16]	×	✓	✓	✓
JSAC'25 [17]	×	✓	✓	×
TVT'22 [19]	×	×	×	✓
TMC'23 [20]	×	✓	✓	✓
TMC'24 [21]	×	✓	×	✓
Ours	✓	✓	✓	✓

△ COMM, COMP, and HETERO represent the communication, computation, and heterogeneous synchronization bottlenecks, respectively.

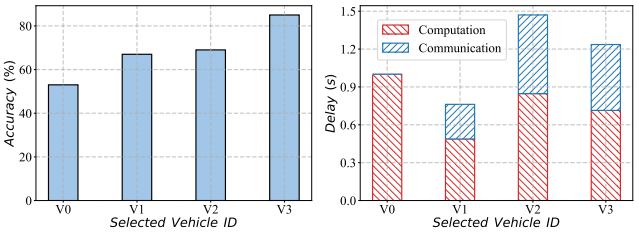
ECOP, an edge-coordinated scheme that employs perceptual gain estimation and online learning to jointly optimize node selection and bandwidth allocation. Fang *et al.* [12] developed PACP, a priority-aware framework leveraging BEV-based matching to determine selection priorities, transmission rates, and compression ratios. Similarly, Zhang *et al.* [13] introduced SmartCooper, which integrates channel-aware compression and coverage-aware vehicle selection to enable accurate yet communication-efficient CP. While these studies have made progress in optimizing communication by addressing *who to cooperate with*, they largely overlook the computational efficiency on the cooperative nodes. In practice, many vehicles are equipped with resource-constrained edge processors, and the high computational cost of deep feature extraction remains a significant barrier to efficient CP execution.

Recent studies have investigated EI techniques such as cooperative inference and model pruning to alleviate the *computation burden* in vehicular systems. Liu *et al.* [14] explored DNN inference partition and offloading to balance workloads among heterogeneous cooperative vehicles. Jiang *et al.* [16] proposed FedMP, which employs adaptive pruning to reduce both computation and communication costs during model training. Yao *et al.* [17] integrated model splitting and pruning to jointly achieve efficient and accurate edge inference. However, these EI studies primarily focus on individual devices and neglect the unique CP challenges, which entails distributed data collection and multi-agent coordination.

A few efforts have also targeted the *synchronization challenge* caused by system heterogeneity. Liang *et al.* [19] proposed Semi-SynFed, which adaptively selects participants and adjusts waiting times to mitigate synchronization delays during distributed model aggregation. Zhang *et al.* [20] introduced MADCA-FL, enabling resource-aware participant configuration for faster convergence in dynamic and heterogeneous networks. Luo *et al.* [21] jointly optimized client sampling and bandwidth allocation to address both data and resource-level heterogeneity. Despite their merits, these methods are tailored for generic edge computing or federated learning (FL) settings and cannot be directly extended to vehicular CP systems, which tightly integrates perception, communication, and computation within a unified framework.



(a) An example scenario from OPV2V with four vehicles



(b) Heterogeneous AP contribution (c) Heterogeneous delay distribution

Fig. 2. Case-study of CP in heterogeneous vehicular networks.

III. FRAMEWORK DESIGN

In this section, we present a motivating case-study to illustrate the impact of system heterogeneity on CP accuracy and efficiency. Inspired by the experimental observations, we propose EI-Cooper, an EI-enabled optimization framework that integrates selective cooperation, model pruning, and bandwidth allocation to jointly coordinate perception, computation, and communication of CP.

A. Motivation

We extract a representative scenario from OPV2V, a large-scale open-source CP benchmark dataset [23]. As illustrated in Fig. 2(a), the scenario involves four vehicles, where three cooperative vehicles (V_1, V_2, and V_3) transmit their locally extracted perception features to the CP requesting vehicle (V_0), which then performs feature fusion to enhance situational awareness. Since the original OPV2V dataset does not explicitly model resource constraints or heterogeneity, we manually configure each vehicle with distinct computation and communication capabilities to emulate realistic deployments.

We adopt object detection as the benchmark task and evaluate CP performance from two perspectives: perception accuracy and cooperation efficiency. Accuracy is quantified by the average precision at an Intersection over Union (AP@IoU) threshold of 0.7. A prediction is considered a true positive (TP) if its IoU—the ratio between the intersection and union of predicted and ground-truth bounding boxes—exceeds the threshold. This metric captures both false positives and missed detections, thereby reflecting the system's ability to accurately detect objects while maintaining spatial coverage of the environment [10]. Cooperation efficiency is measured by the task response delay, defined as the elapsed time from data acquisition at the cooperative vehicle to the feature fusion at the requesting vehicle. This delay includes both individual computation and V2V communication latency. Based on the experimental results, we derive the following observations:

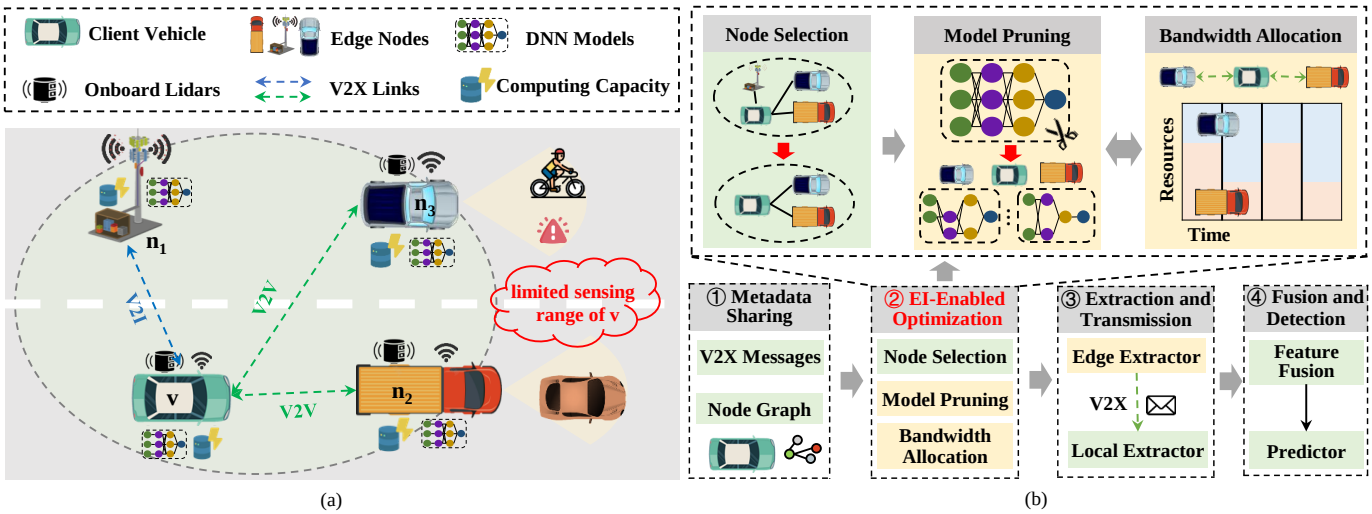


Fig. 3. EI-Cooper framework. (a) Scenario; (b) Workflow.

Observation 1. *The contribution of each cooperative vehicle to the overall CP accuracy is inherently heterogeneous.* We individually select each vehicle to cooperate with V_0 and measure the resulting accuracy. As illustrated in Fig. 2(b), V_0 serves as the baseline using only its local perception without cooperation. Among the three candidates, V_3 provides the most substantial accuracy improvement, whereas V_1 and V_2 yield relatively modest and similar gains. This variation stems from differences such as spatial coverage, and the semantic richness of shared features across the cooperative vehicles.

Observation 2. *Within a given cooperative group, individual vehicles exhibit significant variance in end-to-end delay, which includes both computation (i.e., feature extraction) and communication (i.e., feature sharing) stages.* We select V_1, V_2 and V_3 vehicles as a cooperative group and evaluate their delay distribution. As shown in Fig. 2(c), V_1 achieves the lowest latency due to its superior computational and communication capabilities. In contrast, V_2 and V_3 suffer from higher delays caused by limited onboard resources and bandwidth. Under the BSP fusion mechanism, the overall system latency is dictated by the slowest node (i.e., V_2), thus introducing a synchronization bottleneck that degrades cooperation efficiency.

These observations collectively underscore that cooperative vehicles exhibit significant heterogeneity in both perception data distributions and resource capabilities, which motivates the need for joint perception-computation-communication coordination to optimize CP accuracy and efficiency in real-world deployments. Specifically, observation 1 reveals that not all cooperative vehicles contribute equally to the final detection results. Involving all available nodes may introduce redundant or low-value features, leading to marginal accuracy improvements at the cost of increased communication and computation overhead. This highlights the importance of intelligent vehicle selection strategies that identify participants offering the most complementary perception perspectives to enhance situational awareness. Observation 2 further emphasizes the detrimental effect of resource heterogeneity on end-to-end CP delay. Due to disparities in computation and communication resources, slower nodes can become bottlenecks. This motivates the resource-aware coordination strategies, where model complexity (e.g., via adaptive pruning) and transmission scheduling (e.g., via bandwidth allocation) are tailored to each vehicle's resource profile to enhance overall cooperation efficiency.

B. EI-Cooper Framework

Motivated by the above observations, we design the EI-Cooper frmaework, as illustrated in Fig. 3. Consider a typical CP scenario, where the vehicle requesting assistance is referred to as the *client vehicle* v, while its surrounding vehicles or roadside units are termed as *edge nodes*, represented by the set $\mathcal{N} = \{n_1, n_2, \cdots, n_{|\mathcal{N}|}\}$. All participating nodes are equipped with LiDAR sensors[2] and preloaded AI models (e.g., DNN-based extractors) to perform local perception. Communication among nodes is enabled via V2X links with limited range.

The workflow of EI-Cooper consists of the following four stages: ① *Metadata Sharing.* At each time step, edge nodes periodically broadcast their metadata (e.g., location, sensor parameters, resource state) via V2X beacons. The client vehicle constructs a local spatial graph to track available candidates for cooperation. ② *EI-Enabled Optimization.* Based on the constructed graph, the client vehicle jointly optimizes node selection, model pruning, and bandwidth allocation, aiming to balance perception accuracy and cooperation efficiency. Details are provided in Section VI. ③ *Feature Extraction and Transmission.* According to the CP request, each selected edge node projects its point clouds into the client's coordinates and extracts features using its local DNN models[3]. The features are then encoded and transmitted to the client vehicle. ④ *Fusion and Detection.* Upon receiving all extracted features, the client vehicle aggregates both local and remote information using a fusion network to generate final detection outputs, including object location, size, and confidence score.

[2]The framework also supports other sensing modalities such as cameras or multi-modal fusion, which are left for future exploration.

[3]Following prior works [11], [12], [23], we assume homogeneous models but heterogeneous pruning configurations across edge nodes. Techniques such as cross-model alignment [24] are complementary and can also be integrated.

IV. System Model

In this section, we first introduce an analytical model that characterizes the system-wide response delay, and perception accuracy within the EI-Cooper framework. We then formally define the Synchronization-Efficient Cooperative Perception (SECP) problem, which seeks to balance perception accuracy and cooperation efficiency. Finally, we discuss the closed-box nature and computational NP-hardness of SECP.

A. Analytical Model

We consider a time-slotted system where the client vehicle v initiates a sequence of $|\mathcal{T}|$ CP requests over a planning horizon $\mathcal{T}$ (e.g., a navigation segment). For each request $t \in \mathcal{T}$, an edge node $n \in \mathcal{N}$ is characterized by the tuple $< l_n, c_n, \phi_n, \psi_n >$, where $l_n = (x_n, y_n, z_n)$ represents its location coordinate, c_n denotes its computation capacity, ϕ_n states its V2X communication coverage, and ψ_n contains the shared metadata. The Euclidean distance between the pair of v and n is computed by $L_{vn} = ||l_v, l_n||_2$. V2X communication between v and n is feasible only if $L_{vn} \leq \min(\phi_v, \phi_n)$.

Let a binary variable $\alpha_n \in \{0, 1\}$ indicate whether n is selected by v for perception fusion. Accordingly, the selected node set is defined as $\mathcal{A}_v = \{n \in \mathcal{N} \mid \alpha_n = 1\}$. Each selected edge node n captures and processes its local point cloud data $\mathbf{x_n}$. To extract high-level semantic features, a DNN-based feature extractor m, such as PointPillars [25] is employed, incurring a computational overhead quantified by F_m (e.g., FLOPs). The extracted deep feature vector is given by $\mathbf{f}_n = m(\mathbf{x}_n)$, with an associated data size of $d_{\mathbf{f}_n}$.

The feature extractor typically incur significant computation delays on edge devices due to their limited processing capabilities. Additionally, the large volume of intermediate features imposes substantial communication overhead on the wireless channel. To mitigate these challenges, we adopt adaptive model pruning [16], which reduces both computational and communication costs by allowing each edge node to retain only a fraction of the original model. Let $\rho_n \in [0, 1)$ denote the pruning ratio for node n, where $\rho_n = 0$ indicates use of the full model, and higher values of ρ_n represent more aggressive pruning, reducing computational complexity at the potential expense of feature degradation. We adopt a magnitude-based pruning scheme [18], in which model parameters are sorted by absolute value, and the smallest ρ_n fraction is removed. This method proportionally reduces FLOPs while preserving the most informative weights. Let $P(\cdot, \cdot)$ denote the pruning function, such that the pruned model of each node n is $\tilde{m}_n = P(\rho_n, m)$. Accordingly, edge node n extracts its feature representation as $\tilde{\mathbf{f}}_n = \tilde{m}_n(\mathbf{x}_n)$. The computation delay CD_n incurred at node n is given by [26]:

$$CD_n = \frac{F_m(1 - \rho_n)}{c_n} \tag{1}$$

This equation also applies to the extraction stage on the client vehicle v, where its local computing delay is denoted as CD_v.

Following local extraction, each node n transmits its intermediate feature vector $\tilde{\mathbf{f}}_n$ to the client vehicle v. The transmitting delay CD_n is computed by [26]:

$$TD_n = \frac{d_{\mathbf{f}_n}(1 - \rho_n)}{r_{vn}} \tag{2}$$

where r_{vn} denotes the transmission rate between v and n.

According to the 3GPP 5G standard [27], the V2X network adopts an Orthogonal Frequency Division Multiplexing (OFDM)-based access scheme, where all edge nodes share a total bandwidth B via orthogonal sub-channels. Each selected edge node is allocated a independent bandwidth segment b_n for feature transmission, subject to the constraint $\sum b_n \leq B$. The available transmission rate r_{vn} is computed as [28]:

$$r_{vn} = b_n \log_2 \left(1 + \frac{P_n \cdot g_{vn}}{\sigma^2}\right) \tag{3}$$

where P_n is the transmit power, σ^2 is the received noise power, and $g_{vn} = h^S_{vn} h^L_{vn}$ captures both small-scale and large-scale channel effects. Specifically, $h^S_{vn} \sim \mathcal{CN}(0, 1)$ represents small-scale Rayleigh fading, and h^L_{vn} accounts for large-scale path loss and log-normal shadowing. Following the 3GPP TR 37.885 V2X protocol [29], the pass-loss is computed as $38.77 + 18.2 \log f_c + 16.7 \log L_{vn}$, where f_c denotes the carrier frequency in GHz and L_{vn} is the Euclidean distance. Since the client vehicle v retains its local features for further fusion without transmission, its transmission delay is $TD_v = 0$.

On this basis, the response delay for each node (v or n) from local feature extraction to sharing with the client vehicles is:

$$RD_i = CD_i + TD_i \quad \forall i \in \mathcal{A}_v \cup \{v\} \tag{4}$$

The client vehicle then aggregates both its local features and those received from selected nodes. This fusion process can employ conventional operations such as element-wise pooling [30], or advanced techniques like attention-based weighted summation [11], [23]. The fusion operation is modeled as:

$$\widehat{\mathbf{f}_v} = \text{Fusion}\Big(\sum_{i \in \mathcal{A}_v \cup \{v\}} \tilde{\mathbf{f}}_i\Big) \tag{5}$$

where $\widehat{\mathbf{f}_v}$ is the aggregated feature vector at the client vehicle v. This fused feature is then fed into the prediction head for downstream tasks such as detection and segmentation. Thus, the overall perception accuracy of the CP request is denoted as AP_t, evaluated using AP@IoU. In this fusion pipeline, final results are generated only after all required features from the selected edge nodes are received, following BSP mechanism. Consequently, the response delay for a CP request t, denoted by RD_t, is determined by the slowest responding node [4]:

$$RD_t = \max_{i \in \mathcal{A}_v \cup \{v\}} RD_i \tag{6}$$

It is worth noting that for complex DNN models, prior empirical studies [4], [15] have shown that the feature extraction and transmission stages dominate the total latency, while the delay introduced by the detection head is relatively minor. Therefore, our analysis focuses primarily on the synchronization time from feature extraction at edge nodes to aggregation at the client vehicle, and omits the negligible delay of fusion head.

B. SECP Problem

The SECP problem is formulated as follows: for each CP request $t \in \mathcal{T}$, given a set of edge nodes $\mathcal{N}$ with attributes $< l_n, c_n, \phi_n, \psi_n >$, the client vehicle v aims to jointly optimize the cooperative edge node selection $\mathcal{A}_v$, model pruning ratio ρ_n, ρ_v, and bandwidth allocation b_n, with the goal of maximizing the CP task utility, which balances perception accuracy AP_t and system response delay RD_t.

$$\mathcal{P}: \max_{\{\alpha_n\},\{\rho_n,\rho_v\},\{b_n\}} \quad AP_t - \varphi \cdot RD_t \tag{7a}$$

$$\text{s.t.} \quad \alpha_n \in \{0,1\}, \quad \forall n \in \mathcal{N} \tag{7b}$$

$$|\mathcal{A}_v| \le K, \quad \forall \alpha_n = 1 \tag{7c}$$

$$\rho_n \in [0,1), \quad \forall n \in \mathcal{A}_v \cup \{v\} \tag{7d}$$

$$b_n \in (0,B], \quad \forall n \in \mathcal{A}_v \tag{7e}$$

$$\sum b_n \le B, \quad \forall n \in \mathcal{A}_v \tag{7f}$$

where $\varphi > 0$ is a tunable trade-off weight controlling perception accuracy and response delay. Constraint (7b) ensures that the selection decision for each edge node is represented as a binary variable. Constraint (7c) implies that the number of selected edge nodes must not exceed a predefined threshold K. In practical vehicular environments, this threshold can be determined by the number of available V2X subchannels, or edge node occupancy constraints [12], [13]. Constraints (7d) and (7e) ensure that pruning ratios and bandwidth allocations are continuous variables, allowing flexible resource assignment. Constraint (7f) enforces the total bandwidth budget.

This optimization problem poses two main challenges: (i) the CP contribution of each node n and their fusion accuracy AP_t exhibit high nonlinearity and combinatorial dependencies across multiple stages, including feature extraction, aggregation, and detection. Due to the closed-box nature of DNNs, AP_t cannot be explicitly formulated as a closed-form function of decision variables such as node selection, pruning ratios, or bandwidth allocations. (ii) Even with a prior known accuracy model, the formulation still involves both discrete (edge node selection) and continuous (model pruning ratio and bandwidth allocation) variables, resulting in a mixed-integer nonlinear programming (MINLP) problem, which is generally NP-hard.

V. Problem Transformation

To enable tractable optimization of SECP, we first decompose the closed-box accuracy AP_t into two interpretable and orthogonal components: *perception coverage* and *feature quality*, inspired by AP@IoU evaluation. Accordingly, the original problem is further transformed into two sequential subproblems, with the aim of coverage maximization and feature quality-response delay trade-off, respectively.

A. Accuracy Decomposition

Since perception accuracy cannot be analytically characterized prior to execution, we analyze it via the widely adopted AP@IoU metric, as introduced in Sec.III-A. Fig.4 illustrates an example where the rectangular region denotes the Maximum Cooperative Perception Coverage (MCPC) for a given CP request. Prediction p_1 is derived from the client vehicle's local observation, whereas $\{p_2, p_3\}$ are attributed to the extended spatial coverage from n. According to the AP@IoU metric, only $\{p_1, p_2\}$ are counted true positives. Prediction p_3, although within coverage, fails the IoU threshold due to degraded localization quality. This example highlights two key factors governing AP@IoU:

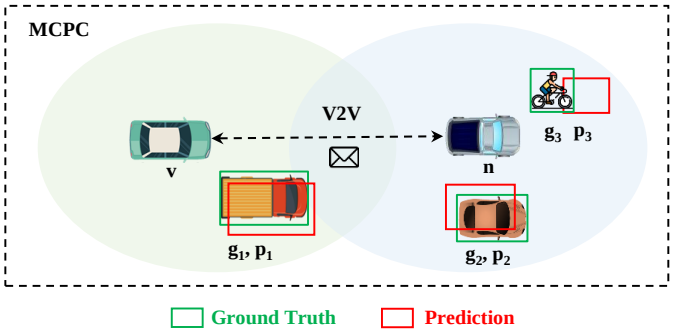


Fig. 4. Illustrative example of AP@IoU evaluation. The client vehicle v cooperates with edge node n, whose sensing ranges are shown as colored circles. Green boxes indicate ground truth objects $\{g_1, g_2, g_3\}$, while red boxes denote the corresponding predictions $\{p_1, p_2, p_3\}$.

- *Perception Coverage*. From a macro-level spatial completeness, this quantifies the extent of environmental regions jointly observed by the cooperative vehicles. A broader range within the MCPC enables to observe beyond the ego vehicle's local field of view (e.g., g_2, g_3), thereby enhancing recall and reducing missed detections.
- *Perception Quality*. From a micro-level semantic retention, this measures the fidelity of shared features, which directly affects the alignment between predicted bounding boxes and ground truth. In feature-level CP systems, perception quality is highly sensitive to the model pruning ratio—while higher pruning reduces computation, it may degrade feature semantics, especially for distant or sparse objects (e.g., g_3), thereby impairing detection precision.

B. Problem Decomposition

Based on the accuracy decomposition, we further decompose the original problem into two sequential subproblems, aiming to optimize perception coverage and feature quality, respectively. Specifically, the selection of edge nodes determines the extent of perceivable coverage. As the first step, we focus on selecting a candidate set of cooperative nodes $\mathcal{A}_v$ to maximize spatial coverage within the MCPC, subject to a selection constraint of K. This stage is performed independently of model pruning and bandwidth allocation. Denote δ_n as the perception area of edge node n, and δ_v as that of the client vehicle v. We define a set function $\mathcal{F}(\mathcal{A}_v) = \text{Area}\big(\delta_v \bigcup (\cup_{n\in\mathcal{A}_v} \delta_n)\big)$ to represent the total perception area jointly covered by the client and its selected edge nodes. The coverage maximization problem is formulated as:

$$\mathcal{P}_1: \max_{\mathcal{A}_v \subseteq \mathcal{N}} \quad \mathcal{F}(\mathcal{A}_v) \tag{8a}$$

$$\text{s.t.} \quad \alpha_n \in \{0,1\}, \ \forall n \in \mathcal{N} \tag{8b}$$

$$|\mathcal{A}_v| \le K, \quad \forall \alpha_n = 1 \tag{8c}$$

We briefly review the definition and primary characteristics of submodularity and give some properties of $\mathcal{P}_1$.

Definition 1. ***Non-negativity, Monotonicity and Submodularity*** *[31]. Let $\mathcal{N}$ be a finite ground set, the set function $\mathcal{F}: 2^{\mathcal{N}} \to \mathbb{R}$ satisfies the following properties:*

- *Non-negativity: $\mathcal{F}$ is non-negative if $\mathcal{F}(\mathcal{A}_v) \geq 0$ for all $\mathcal{A}_v \subseteq \mathcal{N}$;*
- *Monotonicity: $\mathcal{F}$ is monotone if $\mathcal{F}(\mathcal{A}_v^1) \leq \mathcal{F}(\mathcal{A}_v^2)$ for all $\mathcal{A}_v^1 \subseteq \mathcal{A}_v^2 \subseteq \mathcal{N}$;*
- *Submodularity: $\mathcal{F}$ is submodular if, for all $\mathcal{A}_v^1 \subseteq \mathcal{A}_v^2 \subseteq \mathcal{N}$ and $n \in \mathcal{N} \setminus \mathcal{A}_v^2$, the following holds:*

$$\mathcal{F}(\mathcal{A}_v^1 \cup \{n\}) - \mathcal{F}(\mathcal{A}_v^1) \geq \mathcal{F}(\mathcal{A}_v^2 \cup \{n\}) - \mathcal{F}(\mathcal{A}_v^2).$$

The following theorem shows that the set function $\mathcal{F}(\mathcal{A}_v)$ in $\mathcal{P}_1$ satisfies the submodular conditions in Definition 1.

Theorem 1. ***Submodularity of Coverage Function****. The set function $\mathcal{F}(\mathcal{A}_v)$ is non-negative, monotone, and submodular.*

Proof. Please refer to Appendix A. □

Further, given a candidate edge node set $\mathcal{A}_v$, the CP accuracy is further affected by the quality of extracted features, which depends on the pruning ratio ρ_n and ρ_v. In addition, both pruning ratio ρ_n and allocated bandwidth b_n impact the computation and communication overhead at each node. To address this, we formulate a joint pruning and bandwidth allocation problem to balance aggregated feature quality and system-wide cooperation efficiency.

Formally, the degradation of feature quality caused by pruning can be quantified using the mean squared error (MSE) between the features extracted from full and pruned models [18]. Therefore, we adopt a feature quality function derived from the MSE analysis of magnitude-based pruning schemes. Let $Q_i(\rho_i) \in (0, 1]$ denote the retained feature quality of node i under pruning ratio ρ_i. The overall feature quality aggregated from all contributors (including the client vehicle v) is defined as $Q(\mathcal{A}_v) = \sum_{i \in \mathcal{A}_v \cup \{v\}} \omega_i \cdot Q_i(\rho_i)$, where $\omega_i = \frac{\delta_i}{\sum_{i \in \mathcal{A}_v \cup \{v\}} \delta_i}$ is a coverage-aware weight reflecting the spatial contribution of each node and $\sum_{i \in \mathcal{A}_v \cup \{v\}} \omega_i = 1$. The feature quality system delay trade-off is formulated as:

$$\mathcal{P}_2: \max_{\{\rho_n, \rho_v\}, \{b_n\}} \quad Q(\mathcal{A}_v) - \varphi \cdot RD_t \tag{9a}$$

$$\Updownarrow$$

$$\min_{\{\rho_n, \rho_v\}, \{b_n\}} \quad -Q(\mathcal{A}_v) + \varphi \cdot RD_t \tag{9b}$$

$$\text{s.t.} \quad \rho_n \in [0, 1), \ \ \forall n \in \mathcal{A}_v \cup \{v\} \tag{9c}$$

$$b_n \in (0, B], \ \ \forall n \in \mathcal{A}_v \tag{9d}$$

$$\sum b_n \leq B, \ \forall n \in \mathcal{A}_v \tag{9e}$$

The objective in (9a) is equivalent to (9b). The following lemma characterizes how the feature quality degrades as a function of the pruning ratio.

Lemma 1. ***Feature Quality Approximation****. Under the weight magnitude-based pruning scheme, the retained feature quality can be theoretically approximated as:*

$$Q(\rho) = 1 - \left(\rho - \frac{2}{\sqrt{\pi}} \cdot \text{erf}^{-1}(\rho) \cdot e^{-(\text{erf}^{-1}(\rho))^2}\right) \tag{10}$$

where $\text{erf}^{-1}(\cdot)$ is the inverse error function [32].

Proof. Please refer to Appendix B. □

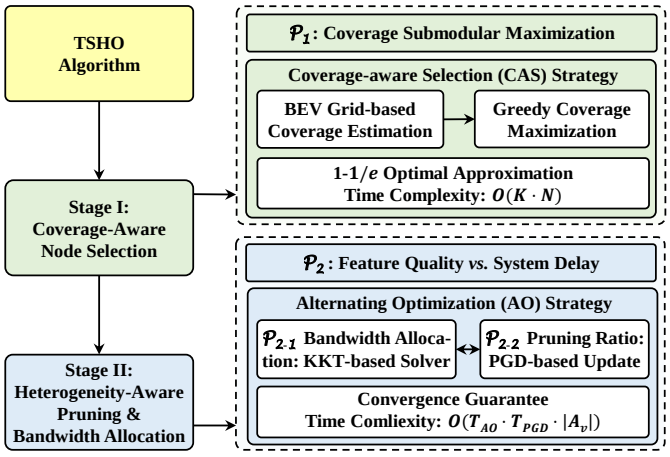


Fig. 5. Overview of the TSHO algorithm.

VI. Algorithm Design

A. Overview

Following the problem transformation, we develop a Two-Stage Hierarchical Optimization (TSHO) algorithm to sequentially solve the decoupled subproblems. An overview of TSHO is shown in Fig. 5, which consists of the following two stages.

Stage I: Coverage-Aware Node Selection. As illustrated in Stage I of Fig. 5, the first stage address problem $\mathcal{P}_1$ by selecting edge nodes to maximize the cooperative perception range and achieve complementary spatial awareness. Specifically, each node's coverage is first estimated using a BEV occupancy grid-based method. Based on the estimation, the node selection task is formulated as a submodular maximization problem, where a greedy algorithm is employed to iteratively select nodes that yield the highest marginal coverage gain. This coverage-aware selection (CAS) strategy provides a $(1-1/e)$-approximation to the optimal solution, while maintaining low time complexity suitable for efficient onboard execution.

Stage II: Heterogeneity-Aware Pruning and Bandwidth Allocation. As shown in Stage II of Fig. 5, the second stage addresses problem $\mathcal{P}_2$ by enabling a heterogeneity-aware pruning and bandwidth allocation under heterogeneous resource constraints. Given the selected node set, this feature quality and system delay trade-off is formulated as a continuous non-convex problem and solved via an alternating optimization (AO) approach. In each iteration:

- A convex bandwidth allocation subproblem is solved via the Karush-Kuhn-Tucker (KKT) conditions to minimize communication delay;
- Pruning ratios are updated using projected gradient descent (PGD), balancing feature degradation and computational efficiency.

These subproblems are solved iteratively with guaranteed convergence to a stationary point, ensuring a balanced trade-off between cooperation efficiency and perception accuracy.

B. TSHO Algorithm

1) Coverage-Aware Node Selection (CAS): Before solving the coverage maximization problem $\mathcal{P}_1$, one major challenge is how to accurately determine the perception coverage δ_n of each node. Existing studies typically adopt one of two approaches: (i) estimating coverage from physical and inertial sensor parameters(e.g., LiDAR scanning range, field of view) [13], or (ii) leveraging deep models to generate semantic visibility maps from learned representations [33]. However, both approaches present notable limitations. The geometry-based estimation methods are often idealized and fail to account for occlusions and dynamic obstacles in real-world environments. In contrast, deep model-based estimators, while more expressive, are computationally intensive and less suitable for efficient onboard deployment.

To address the above limitations, we adopt a lightweight and scalable approximation method for coverage estimation based on LiDAR point clouds and BEV occupancy grids, inspired by the voxelization process in PointPillars [25]. Specifically, we discretize the MCPC area into a uniform BEV grid. Let $\mathcal{G} = \{g_1, g_2, \cdots, g_{|\mathcal{G}|}\}$ denote the set of all grid cells. For each node n, its point clouds in the current frame is denoted by $\mathbf{x}_n$. The point density in cell g_i is computed as $D_n(g_i) = \frac{|\mathbf{x}_n \cap g_i|}{A_g}$, where $|\mathbf{x}_n \cap g_i|$ represents the number of points falling within cell g_i, and A_g denotes the area of a single cell (i.e., grid resolution). A cell is considered perceptually covered by node n if its density exceeds a predefined threshold τ_d. Accordingly, the perception coverage of node n is defined as $\delta_n = \{g_i \in \mathcal{G} | D_n(g_i) \geq \tau_d\}$. This formulation is consistent with prior work such as [5], and the threshold can be empirically tuned based on sensor parameters and environmental sparsity. Notably, this estimation is performed in a distributed manner at each edge node as part of LiDAR preprocessing. During the *Metadata Sharing* stage, each node n transmit this coverage information δ_n to the client vehicle. We illustrate this grid-based coverage estimation in Fig 6.

Following this estimation, problem $\mathcal{P}_1$ is identified as a monotone submodular maximization problem with respect to selective cooperation, as established in Theorem 1. We therefore adopt a greedy algorithm that iteratively selects nodes with the highest marginal coverage gain until the selection budget K is exhausted. The procedure is detailed in Stage I of Algorithm 1, with a performance guarantee remarked below.

Remark 1. ***Approximation Guarantee of CAS for*** $\mathcal{P}_1$. *According to [31], the greedy solution to the submodular maximization achieves a* $(1 - 1/e)$*-approximation to the optimal coverage, ensuring a performance bound for* $\mathcal{P}_1$.

2) Alternating Optimization (AO): Problem $\mathcal{P}_2$ is a continuous, non-convex optimization problem due to the max-delay RD_t and non-linear feature quality $Q(\rho)$. Moreover, both decision variables ρ_n and b_n are tightly coupled: ρ_n influences both feature quality and task response delay, while

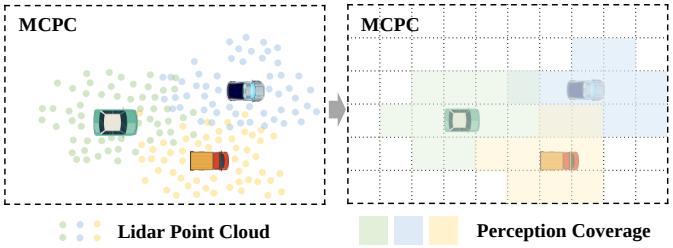


Fig. 6. Illustrative example of BEV occupancy grid-based coverage estimation. Colored points denote raw LiDAR point clouds; dashed cells represent the discretized BEV occupancy grid; shaded cells indicate the estimated perception coverage of each vehicle.

b_n affects communication delay. To address this, we adopt an AO approach that iteratively updates pruning ratios and bandwidth allocations. In each iteration, two subproblems are solved: (i) *Bandwidth Allocation:* Given fixed ρ_n, solve a convex bandwidth allocation problem to minimize RD_t; (ii) *Pruning Optimization:* Given fixed b_n, update ρ_i (including the client vehicle) to balance feature quality and delay. These subproblems are alternately solved until convergence. The procedure is provided in Stage II of Algorithm 1.

Step 1: Bandwidth Allocation (Fixed ρ_n*)*: Given fixed pruning ratios, the aggregated feature quality $Q(\mathcal{A}_v)$ and computing delay CD_n remain constant. The subproblem $\mathcal{P}_2$ thus reduces to a convex min–max optimization over $\{b_n\}$ to minimize the maximum response delay RD_t. We introduce an auxiliary variable T to represent the upper bound of RD_t, and reformulate the subproblem as:

$$\mathcal{P}_{2-1}: \quad \min_{T,\{b_n\}} \quad T \tag{11a}$$

$$\text{s.t.} \quad b_n \in (0, B], \quad \forall n \in \mathcal{A}_v \tag{11b}$$

$$\sum b_n \leq B, \quad \forall n \in \mathcal{A}_v \tag{11c}$$

$$\frac{F_m(1-\rho_n)}{c_n} + \frac{\kappa_n(1-\rho_n)}{b_n} \leq T, \quad \forall n \in \mathcal{A}_v \tag{11d}$$

where $\kappa_n = \dfrac{d_{\mathbf{f}_n}}{\log_2\left(1 + \dfrac{P_n \cdot g_{vn}}{\sigma^2}\right)}$ denotes the effective transmission factor of node n. The following theorem indicates that problem $\mathcal{P}_{2-1}$ is convex and can be efficiently solved by deriving the Karush-Kuhn-Tucker (KKT) conditions [34].

Theorem 2. ***KKT-based Solution for*** $\mathcal{P}_{2-1}$. *Under fixed pruning ratios, the optimal bandwidth allocation is given by:*

$$b_n^* = \frac{B \cdot \sqrt{\kappa_n(1-\rho_n)}}{\sum\limits_{j \in \mathcal{A}_v} \sqrt{\kappa_j(1-\rho_j)}} \tag{12}$$

Proof. Please refer to Appendix C. □

Step 2: Pruning Optimization (Fixed b_n*)*: With fixed bandwidth allocation, the pruning problem becomes:

$$\mathcal{P}_{2-2}: \min_{\{\rho_n, \rho_v\}} \; -Q(\mathcal{A}_v) + \varphi \cdot RD_t \tag{13a}$$

$$\text{s.t.} \quad \rho_n \in [0, 1), \quad \forall n \in \mathcal{A}_v \cup \{v\} \tag{13b}$$

In this problem, the feature quality term penalizes excessive pruning, while the delay term encourages higher pruning to improve system responsiveness. By balancing these opposing effects, the optimization seeks a trade-off tailored to each node's computation and communication capabilities.

Since $Q(\cdot)$ is nonlinear and non-convex, and the max operation RD_t introduces non-smoothness, conventional gradient-based methods are not directly applicable. To overcome this, we adopt a log-sum-exp smoothing technique, which provides a differentiable approximation of the maximum function:

$$RD_t^{\text{smooth}} = \frac{1}{\tau} \log \left(\sum_{i \in \mathcal{A}_v \cup \{v\}} e^{\tau RD_i(\rho_i)} \right) \tag{14}$$

where $\tau > 0$ is the smoothing parameter controlling the approximation accuracy. As $\tau \to \infty$, RD_t^{smooth} converges to the exact $\max$ value. This transformation enables a close approximation of the original delay term. Afther smoothing, the problem (13a) is reformulated as:

$$\min_{\rho \in [0,1)} \mathcal{L}(\rho) = - \sum_{i \in \mathcal{A}_v \cup \{v\}} \omega_i Q_i(\rho_i) + \varphi \cdot RD_t^{\text{smooth}} \tag{15}$$

Given the smooth yet non-convex nature of the objective, we adopt the Projected Gradient Descent (PGD) method to iteratively update the pruning ratios, which has been widely adopted for constrained non-convex optimization [35], [36]. In particular, this gradient-based pruning strategy has also demonstrated effectiveness in similiar neural network compression settings [18], [37]. The gradient of $\mathcal{L}(\cdot)$ with respect to each ρ_i (i.e., $\nabla \mathcal{L}(\rho_i^{(k)})$ is computed as:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \rho_i} &= -w_i \cdot \frac{\partial Q(\rho_i)}{\partial \rho_i} + \varphi \cdot \frac{\partial RD_t^{\text{smooth}}}{\partial \rho_i} \\ &= -w_i \cdot -2\big(\text{erf}^{-1}(\rho_i)\big)^2 + \varphi \cdot \frac{e^{\tau RD_i(\rho_i)} \cdot RD_i'(\rho_i)}{\sum_j e^{\tau RD_j(\rho_j)}} \end{aligned} \tag{16}$$

At each iteration l, each pruning ratio ρ_i is updated via:

$$\rho_i^{(l+1)} = \Pi_{[0,1)}\big(\rho_i^{(l)} - \eta \nabla \mathcal{L}(\rho_i^{(l)})\big) \tag{17}$$

where η is the step size, $\Pi_{[0,1)}(\cdot)$ denotes the Euclidean projection onto the feasible set $[0, 1)$, ensuring that pruning ratios remain within valid bounds. The following remark provides the convergence of PGD in our setting.

Remark 2. ***Convergence of PGD for*** $\mathcal{P}_{2-2}$. *The smoothed objective function $\mathcal{L}(\rho)$ is continuously differentiable, and the feasible region is also convex set i.e., $[0, 1)$. Under the Kurdyka-Łojasiewicz (KL) property, the entire sequence of ρ iterates converges to a critical point [38].*

Building upon the convergence of PGD subroutine, we now establish the overall convergence of our AO procedure for problem $\mathcal{P}_2$, as stated in the following theorem.

Theorem 3. ***Convergence of AO for*** $\mathcal{P}_2$. *The proposed AO strategy for solving $\mathcal{P}_2$ converges to a local stationary point.*

Proof. At each iteration, the algorithm alternately minimizes the objective function $\mathcal{P}_2$ with respect to $\{b_n\}$ and $\{\rho_n, \rho_v\}$. The bandwidth allocation subproblem ($\mathcal{P}_{2-1}$) is convex and solved optimally via KKT conditions, while the pruning optimization subproblem ($\mathcal{P}_{2-2}$) ensures descent through projected gradient updates. Morever, both subproblems lead to a monotonically non-increasing objective (9b) and the function is lower-bounded due to $-Q(\rho) \geq -1$ and $RD_t \geq 0$. Thus, our AO strategy converges to a local stationary point [39]. □

Algorithm 1: TSHO Algorithm for Each Task $t \in \mathcal{T}$

Input: Edge node set $\mathcal{N}$, metadata $< l_n, c_n, \phi_n, \psi_n >$, original model FLOPs F_m, feature datasize d_f, total bandwidth B, selection threshold K.
Output: Selected node set $\mathcal{A}_v$, pruning ratios $\{\rho_n\, \rho_v\}$ bandwidth allocation $\{b_n\}$.

```
// Stage I: Solution of P_1. Coverage-aware Node Selection
```
Obtain perception coverage δ_n for each $n \in \mathcal{N} \cup \{v\}$ via BEV-based grid estimation;
```
// Submodular Coverage Maximization
```
Initialize $\mathcal{A}_v \leftarrow \{\emptyset\}$;
for $k = 1$ *to* K **do**
  $n^* = \arg\max_{n \in \mathcal{N} \setminus \mathcal{A}_v} \big(\mathcal{F}(\mathcal{A}_v \cup \{n\}) - \mathcal{F}(\mathcal{A}_v)\big)$;
  Update $\mathcal{A}_v \leftarrow \mathcal{A}_v \cup \{n^*\}$;
```
// Stage II: Solution of P_2. Heterogeneity-Aware Pruning and Bandwidth Allocation
```
Initialize pruning ratios $\{\rho_n^{(0)}, \rho_v^{(0)}\}$, iteration counter $l = 0$;
repeat
```
// Alternating Optimization (AO)
```
  Fix $\{\rho_n^{(l)}, \rho_v^{(l)}\}$, obtain $\{b_n^{(l+1)}\}$ via Eq. (12);
  Fix $\{b_n^{(l+1)}\}$, update $\{\rho_n^{(l+1)}, \rho_v^{(l+1)}\}$ via Eq. (17);
  Set iteration $l \leftarrow l + 1$;
until *Convergence*;
return $\mathcal{A}_v$, $\{\rho_n, \rho_v\}$, $\{b_n\}$

C. Complexity Analysis

Theorem 4. *The overall computational complexity of TSHO is $\mathcal{O}(K \cdot |\mathcal{N}| + T_{AO} \cdot T_{PGD} \cdot |\mathcal{A}_v|)$, where T_{AO} and T_{PGD} denote the number of outer AO and inner PGD iterations required for convergence.*

Proof. In stage I, the client vehicle selects cooperative edge nodes to maximize the marginal perception coverage with a submodular greedy step. Given a total of $|\mathcal{N}|$ nodes and a selection limit of K, the overall complexity of this stage is $\mathcal{O}(K \cdot |\mathcal{N}|)$. the greedy node selection process aims to maximize marginal perception coverage. Given a candidate set of $|\mathcal{N}|$ edge nodes and a selection budget of K, this submodular maximization incurs a complexity of $\mathcal{O}(K \cdot |\mathcal{N}|)$. Stage II employs an AO strategy, in which each iteration involves two subproblems: (i) for the bandwidth allocation subproblem in $\mathcal{P}_{2-1}$, the computational complexity for solving this subproblem with KKT conditions is linear with respect to the number of selected nodes, i.e, $\mathcal{O}(\mathcal{A}_v)$; (ii) for the pruning optimization subproblem in $\mathcal{P}_{2-2}$, the pruning ratios are updated using PGD

TABLE II
SIMULATION PARAMETERS

Parameter	Definition	Value
$\|\mathcal{T}\|$	Number of CP requests	167
$\|\mathcal{N}\|$	Number of available edge nodes	4
K	Number of selection threshold in Eq. (7c)	3
c_v	Computing capacity of client vehicle	5 GHz
c_n	Computing capacity of edge nodes	[5, 10] GHz
F_m	Computation requirement of model	5 GFLOPs
$d_{\mathbf{f}}$	Datasize of deep features	32 Mbits
ϕ_n	V2X communication coverage	150 m
B	V2X communication bandwidth	20 MHz
P	V2X transmission power	23 dBm
h_{vn}^S	Channel fading coefficient	0.5
f_c	Carrier frequency	5.9 GHz
σ^2	Noise power	-90 dBm
φ	Weight of response delay	0.5
τ	Smoothing parameter in Eq. (14)	10
η	Step size in Eq. (17)	0.05

method. Assuming a maximum of T_{PGD} iterations for convergence, and that each gradient computation is linear in $\mathcal{A}_v$, the computational complexity is $\mathcal{O}(T_{PGD} \cdot |\mathcal{A}_v|)$. Let T_{AO} denote the number of outer alternating optimization iterations required for convergence. Therefore, the total complexity of Stage II is $\mathcal{O}\big(T_{AO} \cdot (\mathcal{A}_v + T_{PGD} \cdot |\mathcal{A}_v|)\big) = \mathcal{O}(T_{AO} \cdot T_{PGD} \cdot |\mathcal{A}_v|)$.

Combining both stages, the total computational complexity of TSHO is $\mathcal{O}(K \cdot |\mathcal{N}| + T_{AO} \cdot T_{PGD} \cdot |\mathcal{A}_v|)$. □

VII. PERFORMANCE EVALUATION

A. Setup

We conduct performance evaluation on OpenCOOD[4], a large-scale cooperative perception platform for autonomous driving with V2X communication. The OPV2V dataset [23] is used as the evaluation benchmark. The dataset includes 73 driving scenes with 2-5 connected vehicles operating across 6 road types in 9 cities. In total, it contains over 12K LiDAR frames and 230K annotated bounding boxes. All vehicles are uniformly distributed with velocities ranging from 0 to 50 km/h. Perceptual data are recorded at 10 Hz and assumed to be well-synchronized. Each vehicle collects LiDAR point clouds within a constrained region of $[-72, 72] \times [-40, 40] \times [-3, 1]$ meters in its local XYZ coordinate.

We regroup all scenes based on the number of connected vehicles and set the 5-vehicle scenarios as the default evaluation setting. In each selected scene, the first-ID vehicle is designated as the client vehicle that initiates the CP request, while the remaining four serve as cooperative edge nodes. The predetermined MCPC region is defined as $[-140.8, 140.8] \times [-40, 40] \times [-3, 1]$ meters in the client vehicle's coordinate. We adopt PointPillars [25] as the default backbone network for feature extraction from LiDAR data. Other system-level parameters, such as computation capacity, V2X bandwidth are configured based on prior works [6], [12], [13] and summarized in Table II.

[4]OpenCOOD: https://github.com/DerrickXuNu/OpenCOOD

TABLE III
ABLATION STUDY ON EDGE NODE SELECTION

Metric	*NS*	*RS*	*CPS*	*CAS (Ours)*
$\overline{U} \uparrow$	0.06	0.1	0.14	**0.15**
$\overline{RD}\,(s) \downarrow$	**1**	1.31	1.24	1.29
$\overline{AP}\,(\%) \uparrow$	56.49	76.84	76.37	**78.86**

For performance evaluation, we first conduct a series of ablation experiments to validate the effectiveness of our selective cooperation and adaptive resource allocation strategies in Section VII-B. We then present both quantitative and qualitative comparisons to assess the scalability, adaptability, and generalization capability of EI-Cooper against representative baseline methods, in Section VII-C. Finally, we implement EI-Cooper on a real-world vehicular testbed to demonstrate its practical feasibility.

The quantitative evaluation metrics are defined as follows:

- *Average Utility* ($\overline{U}$): It is the mean value of our objective function in Eq. (7a) across all CP tasks, reflecting the trade-off between accuracy and response delay.
- *Average Response Delay* ($\overline{RD}$): It represents the average end-to-end response delay in Eq. (4).
- *Average Accuracy* ($\overline{AP}$): It denotes the average CP accuracy with an IoU threshold of 0.7.

B. Ablation Experiment

As mentioned, EI-Cooper involves node selection, model pruning and bandwidth allocation to enable a joint perception-computation-communication coordination. Therefore, we conduct a series of ablation experiments that evaluate the contribution of each component.

1) Edge Node Selection: The following solutions are tested to verify our *Coverage-Aware Selection (CAS)* strategy[5]:

- *No Selection (NS)*: It represents the single-vehicle perception scheme without any CP assistance.
- *Random Selection (RS)*: A fixed number of edge nodes are randomly selected from the available candidates, without considering their spatial coverage or resource profile.
- *Compute Power Selection (CPS)*: The K nodes with the highest computation capacities are selected to minimize processing delay, while ignoring the perception coverage.

Table III summarizes the ablation results of different node selection strategies. Among these, our CAS achieves the highest CP utility $\overline{U}$ and accuracy $\overline{AP}$, significantly outperforming RS and CPS. Although CPS yields the lowest lower response delay $\overline{RD}$ by favoring high-performance nodes, it suffers from degraded accuracy due to limited spatial diversity. These results validate the advantage of CAS in identifying spatially complementary nodes that contribute valuable features to the fusion process. Moreover, the effectiveness of CAS supports our problem decomposition approach, where coverage maximization is treated as a distinct optimization objective.

[5]To isolate the impact of edge node selection, all nodes use the full model without pruning, and equal bandwidth is allocated among selected nodes.

TABLE IV
ABLATION STUDY ON PRUNING AND BANDWIDTH ALLOCATION

Metric	*EBNP*	*EBUP*	*ABUP*	*AO (Ours)*
$\overline{U}\uparrow$	0.13	0.36	0.38	**0.45**
$\overline{RD}$ $(s)\downarrow$	1.31	0.79	0.77	**0.61**
$\overline{AP}$ $(\%)\uparrow$	**78.5**	75.87	75.87	76.15

2) Pruning and Bandwidth Allocation: With a fixed set of selected edge nodes, we evaluate the proposed *Alternating Optimization (AO)* strategy for joint pruning and bandwidth allocation against the following solutions:

- *Equal Bandwidth with No Pruning (EBNP)*: Each selected edge node employs the full model (unpruned), and communication bandwidth is equally allocated among them.
- *Equal Bandwidth with Unified Pruning (EBUP)*:A unified pruning ratio is applied to all nodes irrespective of their resource conditions; bandwidth remains equally allocated.
- *Adaptive Bandwidth with Unified Pruning (ABUP)*: Bandwidth is adaptively allocated using KKT-based optimization, while all nodes use a unified pruning ratio.

Table IV presents the ablation results. AO achieves the best overall performance, with the highest average utility $\overline{U}$ and the lowest response delay $\overline{RD}$, while maintaining competitive detection accuracy $\overline{AP}$. Compared to EBNP, which yields the highest $\overline{AP}$ (78.5%) but suffers from the longest $\overline{RD}$ (1.31 s), AO reduces the response time by 53% with only a marginal accuracy degradation. This demonstrates the effectiveness of introducing model pruning to alleviate computation and communication overhead. While EBUP and ABUP moderately reduce the delay relative to EBNP, they experience a noticeable accuracy drop due to suboptimal pruning. In contrast, AO dynamically adjusts pruning ratios and bandwidth allocations based on each node's resource profile, enabling better trade-offs between efficiency and accuracy. These results validate the importance of joint optimization for pruning and communication scheduling and highlight the advantage of our heterogeneity-aware design.

C. Performance Comparison

The following representative baselines are evaluated against our EI-Cooper framework:

- *Standalone*: This baseline represents the single-vehicle perception without cooperation. The client vehicle performs inference using only its own sensory input and the full (unpruned) model.
- *SmartCooper* [13]: This scheme employs a coverage score-based judger to determine selected vehicles and allocates communication bandwidth equally among them. Notably, model pruning is not applied in their settings.
- *FedMP* [16]: Originally designed for general FL training, FedMP applies adaptive model pruning to reduce communication and computation costs. For fair comparison, we re-implement FedMP within the OPV2V simulation environment, adapting its mechanisms to the CP context.

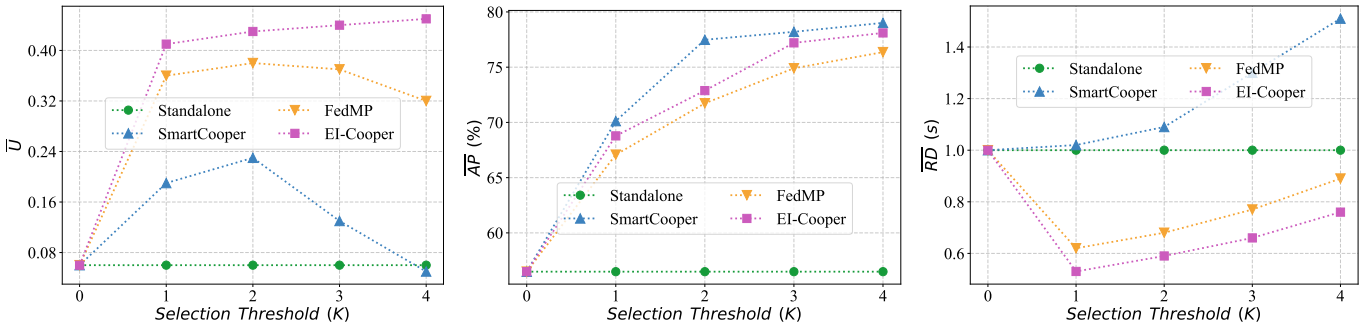
1) Scalability with Varying Selection Threshold: Fig. 7 evaluates the scalability of EI-Cooper by varying the selection threshold K. As shown in Fig. 7(a), all cooperative methods exhibit a notable increase in $\overline{U}$ as K increases, especially from $K = 0$ to $K = 2$, beyond which the marginal gain diminishes. Throughout all configurations, EI-Cooper consistently achieves the highest utility, demonstrating its superior capability to balance perception accuracy and efficiency.

Fig. 7(b) and 7(c) provide further insights into this trend. Specifically, Fig. 7(b) shows that $\overline{AP}$ increases monotonically with larger K, as more cooperative nodes contribute to broader situational awareness. SmartCooper and EI-Cooper yield higher $\overline{AP}$ than FedMP, attributed to their coverage-aware node selection strategies. However, EI-Cooper's use of model pruning introduces a sacrifice in feature quality, leading to a marginally lower $\overline{AP}$ compared to SmartCooper. Fig. 7(c) reveals that $\overline{RD}$ slightly increases with larger K, primarily due to elevated communication overhead under limited bandwidth conditions. Nevertheless, both EI-Cooper and FedMP maintain lower delays than SmartCooper, owing to their use of model pruning for computation and communication efficiency. Notably, EI-Cooper achieves the lowest $\overline{RD}$, benefiting further from its adaptive bandwidth allocation, which effectively mitigates heterogeneous communication bottlenecks.

2) Adaptability to Heterogeneous Resources: Fig. 8 presents a quantitative case study evaluating the adaptability of EI-Cooper under heterogeneous resources. Specifically, we select a representative frame from the OPV2V dataset and examine the selected cooperative nodes, overall accuracy $\overline{AP}$, and per-node computation and communication delays. In this case, all cooperative methods select the same edge node set (V_1, V_2 and V_3), where SmartCooper and EI-Cooper are coverage-based, while FedMP performs random selection. As illustrated in Fig. 8(a), all cooperative schemes significantly outperform the Standalone baseline in terms of $\overline{AP}$, confirming the advantages of multi-agent cooperation. EI-Cooper and FedMP exhibit slightly reduced accuracy compared to SmartCooper, owing to the impact of model pruning, which slightly compromises feature quality in exchange for efficiency.

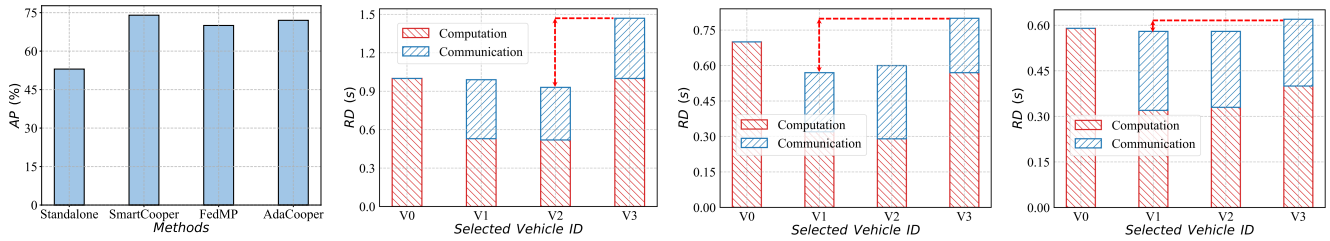
Fig 8(b)∼8(d) provide a breakdown of computation and communication delays across all participating nodes. SmartCooper incurs the highest and most imbalanced delays, particularly for node V_3, which experiences substantial computational latency due to its limited processing capacity. FedMP effectively reduces onboard computational overhead via model pruning but offers limited communication adaptability, resulting in suboptimal latency. EI-Cooper outperforms all baselines by achieving the lowest and most balanced delay, which is attributed to its joint optimization of model pruning and bandwidth allocation, enabling fine-grained adaptation to each node's resource profile.

3) Generalization across Diverse Traffic Scenarios: Fig. 9 and Fig. 10 present qualitative case studies in two representative real-world scenarios: an curvy urban road with vehicle-to-vehicle (V2V) cooperation, and a four-way intersection with infrastructure-to-vehicle (I2V) cooperation. Fig. 9 depicts the curvy road scenario involving five vehicles. As shown in Fig.9(a)∼9(d), the Standalone method suffers from severe



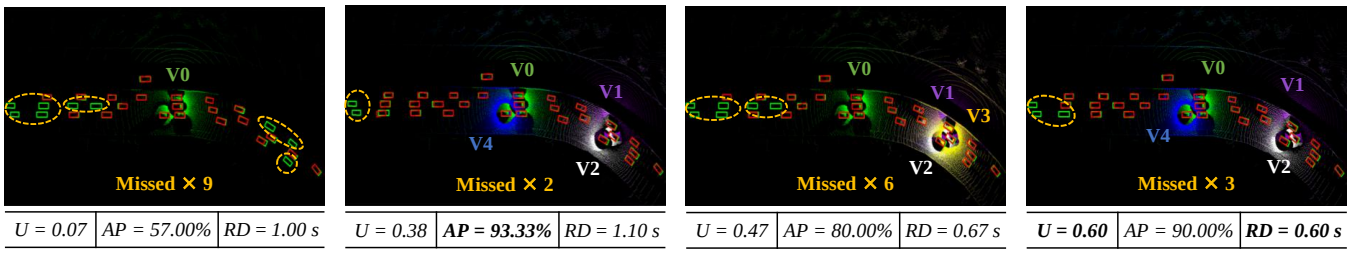
(a) $\overline{U}$ (b) $\overline{AP}$ (c) $\overline{RD}$

Fig. 7. Performance comparison under varying selection threshold.



(a) AP (b) SmartCooper (c) FedMP (d) EI-Cooper

Fig. 8. Quantitative case-study over heterogeneous resources. V_0 is designated as the client vehicle, while $V_1 \sim V_3$ are selected edge nodes.



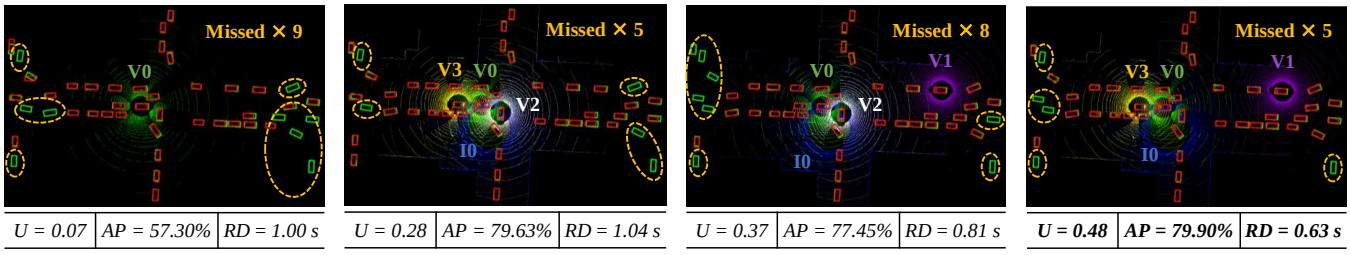
(a) Standalone	(b) SmartCooper	(c) FedMP	(d) EI-Cooper
U = 0.07 | *AP = 57.00%* | *RD = 1.00 s*	*U = 0.38* | ***AP = 93.33%*** | *RD = 1.10 s*	*U = 0.47* | *AP = 80.00%* | *RD = 0.67 s*	***U = 0.60*** | *AP = 90.00%* | ***RD = 0.60 s***

Fig. 9. Qualitative case-study in a curvy road scenario with five vehicles. V_0 denotes the client vehicle, while colored $V_1 \sim V_4$ denote their selected cooperative vehicles. Green and red boxes indicate ground-truth and detection results, respectively. Point clouds from different nodes are rendered using their distinct colors. Note that these raw point clouds are not transmitted to V_0 during CP; they are visualized here solely to aid spatial interpretation.

occlusions and limited sensing, resulting in the most missed detections. SmartCooper and EI-Cooper, both using coverage-based selection, significantly reduce missed detections by incorporating spatial completeness. While EI-Cooper has one additional missed object due to feature degradation from pruning, it achieves the lowest RD and highest U, validating its joint pruning and bandwidth adaptation. In contrast, FedMP, which performs random node selection and applies more aggressive pruning, yields a higher number of missed objects.

Fig. 10 illustrates the complex intersection scenario involving four vehicles and one roadside unit. As shown in Fig.10(a)∼10(d), all cooperative methods benefit from extended perception range provided by infrastructure (I_0), leading to significant improvements in detection accuracy. While SmartCooper and EI-Cooper achieve similar perception performance, they differ notably in their node selections. SmartCooper estimates coverage using static inertial parameters, such as sensing range and geometric location, without considering dynamic occlusions or coverage overlap, thus leading to redundant spatial contributions (e.g., V_2 and V_3). EI-Cooper instead employs submodular optimization with BEV-based dynamic coverage estimation, and incorporates diminishing marginal utility, which enables more diverse node selection (e.g., V_1 and V_3) and better spatial complementarity. Combined with its adaptive pruning and bandwidth allocation, EI-Cooper delivers the lowest RD while preserving competitive AP, achieving an improved efficiency-accuracy trade-off.

4) Practicability with Prototype Implementation: To validate the real-world practicability of EI-Cooper, we further implement a prototype system on a vehicular testbed, as shown in Fig. 11. The testbed includes a client vehicle V_0, a roadside infrastructure I_0 and three surrounding vehicles



U = 0.07	AP = 57.30%	RD = 1.00 s

(a) Standalone

U = 0.28	AP = 79.63%	RD = 1.04 s

(b) SmartCooper

U = 0.37	AP = 77.45%	RD = 0.81 s

(c) FedMP

U = 0.48	**AP = 79.90%**	**RD = 0.63 s**

(d) EI-Cooper

Fig. 10. Qualitative case-study in an intersection scenario with one roadside infrastructure I_0 and four vehicles $V_0 \sim V_3$. V_0 is the client vehicle.

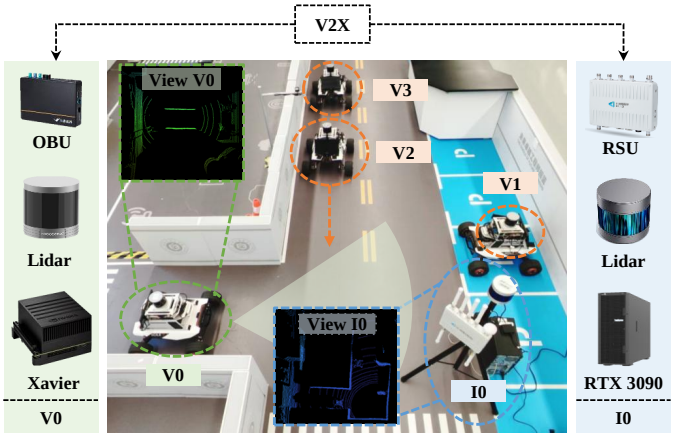


Fig. 11. Prototype case-study in a T-intersection scenario. The client vehicle V_0 cooperates with infrastructure node I_0 to detect vehicles V_1, V_2 and V_3.

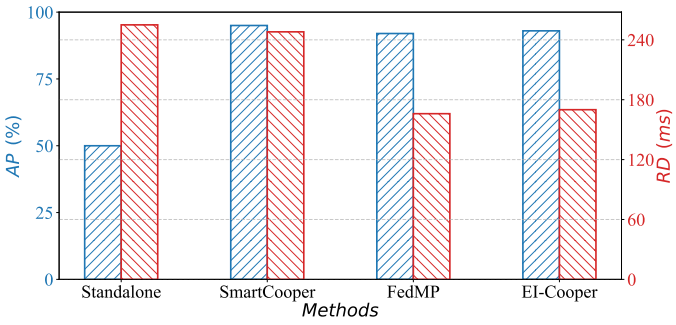
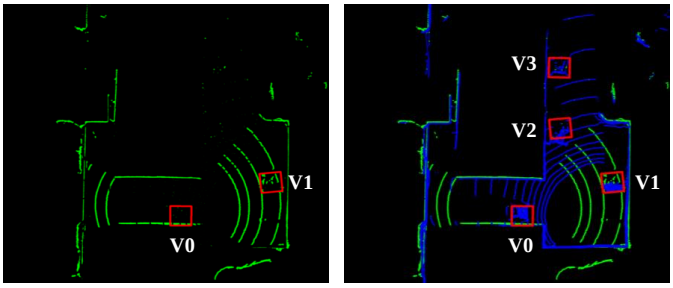


Fig. 12. Quantitative evaluation of CP performance in Fig. 11.

V_1, V_2 and V_3 acting as obstacles. The client V_0 is built on the Agilex Scout MINI Pro platform, equipped with a CICTCI VU4105 Onboard Unit (OBU) for V2X communication, an NVIDIA AGX Xavier computing module, and an RS-Helios-16P LiDAR for local perception. The roadside infrastructure I_0 is equipped with a high-performance NVIDIA RTX 3090 GPU, an RS-Helios-32P LiDAR, for extended perception, and a CICTCI RU5000E Roadside Unit (RSU) to support V2X communication with nearby vehicles.

We first collect raw Lidar point clouds using ROS bag recordings and construct a customized mini-dataset for this prototype scenario. The dataset comprises 3,378 frames captured from both V_0 and I_0, and is organized following the DAIR-V2X format [40] to ensure compatibility and ease of



(a) Standalone (b) EI-Cooper

Fig. 13. Qualitative visualization of CP performance in Fig. 11. Green point clouds are captured from V_0, while blue point clouds are obtained from I_0.

training. Subsequently, we train our CP model using the open-source Heal platform[6], which supports both simulation and real-world datasets. Finally, the well-trained model is deployed on our testbed to evaluate real-world performance.

Fig. 12 presents both AP and RD performance of all methods. The standalone approach yields the lowest AP due to limited sensing range and occlusions. In this simple scenario, all cooperative methods yield similar AP, benefiting from the wide coverage of I_0. On the other hand, due to the computational disparity between V_0 and I_0, the RD of all methods is primarily dominated by the client-side latency. Standalone and SmartCooper apply no optimization and thus exhibit the highest delays, while FedMP and EI-Cooper use adaptive pruning to significantly reduce the computation latency.

Given the similar accuracy performance across these cooperative methods, we only visualize standalone and EI-Cooper results to showcase CP effectiveness. As illustrated in Fig. 13(a), the standalone result shows that V_0 can only detect V_1 within its direct field of view. Fig. 13(b) illustrates that with the assistance of I_0 under EI-Cooper framework, V_0 successfully detects both V_2 and V_3, which were previously occluded. This demonstrate the practical viability of EI-Cooper in real-world deployments, particularly for safety-critical scenarios such as intersections and occluded environments.

VIII. Conclusion

This paper proposed EI-Cooper, an EI-enhanced framework for adaptive and efficient CP deployment in heterogeneous vehicular networks. By jointly optimizing node selection, model

[6]Heal: https://github.com/yifanlu0227/HEAL

pruning, and bandwidth allocation, EI-Cooper enables unified coordination across perception, computation, and communication. We formalized the SECP problem and decomposed it into two tractable subproblems, solved via a TSHO algorithm with theoretical guarantees. Extensive experiments demonstrated the effectiveness of our approach. Future work will extend EI-Cooper to support more general asynchronous fusion, multi-modal sensor alignment, and cross-model cooperation.

REFERENCES

[1] S. Chen, J. Hu, L. Zhao, R. Zhao, J. Fang, Y. Shi, and H. Xu, *Cellular vehicle-to-everything (C-V2X)*. Springer Nature, 2023.

[2] S. Lu and W. Shi, "Vehicle as a mobile computing platform: Opportunities and challenges," *IEEE Network*, vol. 38, no. 6, pp. 493–500, Nov. 2024.

[3] G. Yan, K. Liu, C. Liu, and J. Zhang, "Edge intelligence for internet of vehicles: A survey," *IEEE Trans. Consum. Electron.*, vol. 70, no. 2, pp. 4858–4877, May 2024.

[4] J. Hou, P. Yang, X. Dai, T. Qin, and F. Lyu, "Enhancing cooperative lidar-based perception accuracy in vehicular edge networks," *IEEE Trans. Intell. Transp. Syst.*, 2025, early access, doi:10.1109/TITS.2025.3541265.

[5] G. Luo, C. Shao, N. Cheng, H. Zhou, H. Zhang, Q. Yuan, and J. Li, "Edgecooper: Network-aware cooperative lidar perception for enhanced vehicular awareness," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 1, pp. 207–222, Jan. 2024.

[6] X. Ye, K. Qu, W. Zhuang, and X. Shen, "Accuracy-aware cooperative sensing and computing for connected autonomous vehicles," *IEEE Trans. Mob. Comput.*, vol. 23, no. 8, pp. 8193–8207, Aug. 2024.

[7] Y. Cui, X. Cao, G. Zhu, J. Nie, and J. Xu, "Edge perception: Intelligent wireless sensing at network edge," *IEEE Commun. Mag.*, vol. 63, no. 3, pp. 166–173, Mar. 2025.

[8] S. Yi, H. Zhang, and K. Liu, "V2iviewer: Towards efficient collaborative perception via point cloud data fusion and vehicle-to-infrastructure communications," *IEEE Trans. Network Sci. Eng.*, vol. 11, no. 6, pp. 6219–6230, Nov. 2024.

[9] S. Shi, C. Hu, D. Wang, Y. Zhu, and Z. Han, "Federated HD map updating through overlapping coalition formation game," *IEEE Trans. Mob. Comput.*, vol. 23, no. 2, pp. 1641–1654, Feb. 2024.

[10] J. Shao, T. Li, and J. Zhang, "Task-oriented communication for vehicle-to-infrastructure cooperative perception," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, London, United Kingdom, 2024, pp. 1–6.

[11] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, New Orleans, LA, USA, 2022, pp. 4874–4886.

[12] Z. Fang, S. Hu, H. An, Y. Zhang, J. Wang, H. Cao, X. Chen, and Y. Fang, "PACP: Priority-aware collaborative perception for connected and autonomous vehicles," *IEEE Trans. Mob. Comput.*, vol. 23, no. 12, pp. 15 003–15 018, Dec. 2024.

[13] Y. Zhang, H. An, Z. Fang, G. Xu, Y. Zhou, X. Chen, and Y. Fang, "SmartCooper: Vehicular collaborative perception with adaptive fusion and judger mechanism," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Yokohama, Japan, 2024, pp. 4450–4456.

[14] K. Liu, C. Liu, G. Yan, V. C. S. Lee, and J. Cao, "Accelerating DNN inference with reliability guarantee in vehicular edge computing," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 3238–3253, Dec. 2023.

[15] X. Yang, Z. Xu, Q. Qi, J. Wang, H. Sun, J. Liao, and S. Guo, "PICO: Pipeline inference framework for versatile CNNs on diverse mobile devices," *IEEE Trans. Mob. Comput.*, vol. 23, no. 4, pp. 2712–2730, Apr. 2024.

[16] Z. Jiang, Y. Xu, H. Xu, Z. Wang, J. Liu, Q. Chen, and C. Qiao, "Computation and communication efficient federated learning with adaptive model pruning," *IEEE Trans. Mob. Comput.*, vol. 23, no. 3, pp. 2003–2021, Mar. 2024.

[17] J. Yao, W. Xu, G. Zhu, K. Huang, and S. Cui, "Energy-efficient edge inference in integrated sensing, communication, and computation networks," *IEEE J. Sel. Areas Commun.*, 2025, early access, doi:10.1109/JSAC.2025.3574612.

[18] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10 558–10 578, Dec. 2024.

[19] F. Liang, Q. Yang, R. Liu, J. Wang, K. Sato, and J. Guo, "Semi-synchronous federated learning protocol with dynamic aggregation in internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4677–4691, May 2022.

[20] X. Zhang, Z. Chang, T. Hu, W. Chen, X. Zhang, and G. Min, "Vehicle selection and resource allocation for federated learning-assisted vehicular network," *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 3817–3829, May 2024.

[21] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Adaptive heterogeneous client sampling for federated learning over wireless networks," *IEEE Trans. Mob. Comput.*, vol. 23, no. 10, pp. 9663–9677, Oct. 2024.

[22] Z. Liu, X. Chen, H. Wu, Z. Wang, X. Chen, D. Niyato, and K. Huang, "Integrated sensing and edge AI: Realizing intelligent perception in 6G," *arXiv preprint arXiv:2501.06726*, Jan. 2025.

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

[24] C. Shao, G. Luo, Q. Yuan, Y. Chen, Y. Liu, K. Gong, and J. Li, "Hetecooper: Feature collaboration graph for heterogeneous collaborative perception," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Milan, Italy: Springer, 2024, pp. 162–178.

[25] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 12 697–12 705.

[26] S. Liu, G. Yu, R. Yin, J. Yuan, L. Shen, and C. Liu, "Joint model pruning and device selection for communication-efficient federated edge learning," *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 231–244, Jan. 2022.

[27] W. Anwar, N. Franchi, and G. Fettweis, "Physical layer evaluation of V2X communications technologies: 5G NR-V2X, LTE-V2X, IEEE 802.11bd, and IEEE 802.11p," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Honolulu, HI, USA, 2019, pp. 1–7.

[28] Y. Jia, C. Zhang, Y. Huang, and W. Zhang, "Lyapunov optimization based mobile edge computing for internet of vehicles systems," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7418–7433, Nov. 2022.

[29] 3rd Generation Partnership Project (3GPP), "ATIS 3GPP 37.885 V15.3.0: Technical Specification Group Radio Access Network; Study on NR Vehicle-to-Everything (V2X) (Release 15)," Jun. 2019.

[30] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, UK: Springer, 2020, pp. 605–621.

[31] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, Dec. 1978.

[32] E. W. Weisstein, "Inverse erf," https://mathworld.wolfram.com/InverseErf.html, 2003, online; accessed 2025.

[33] H. Wang, R. Fan, Y. Sun, and M. Liu, "Dynamic fusion module evolves drivable area and road anomaly detection: A benchmark and algorithms," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10 750–10 760, Oct. 2022.

[34] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.

[35] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1, pp. 459–494, Jul. 2014.

[36] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[37] Z. Liu, X. Zhang, Z. Shen, Y. Wei, K.-T. Cheng, and J. Sun, "Joint multi-dimension pruning via numerical gradient update," *IEEE Trans. Image Process.*, vol. 30, pp. 8034–8045, Oct. 2021.

[38] G. Li and T. K. Pong, "Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods," *Found. Comput. Math.*, vol. 18, no. 5, pp. 1199–1232, Aug. 2018.

[39] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1126–1153, Jan. 2013.

[40] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan *et al.*, "DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, 2022, pp. 21 361–21 370.

APPENDIX A
PROOF OF THEOREM 1

We verify each property of $\mathcal{F}(\mathcal{A}_v)$ as follows:

- *Non-negativity*: By definition, $\mathcal{F}(\emptyset) = \text{Area}(\delta_v) \geq 0$. For any $\mathcal{A}_v \subseteq \mathcal{N}$, the union of non-empty perception coverage yields $\mathcal{F}(\mathcal{A}_v) \geq 0$, as perception area is always non-negative.
- *Monotonicity*: Denote two candidate sets $\mathcal{A}_v^1 \subseteq \mathcal{A}_v^2 \subseteq \mathcal{N}$. Since $\bigcup_{n\in\mathcal{A}_v^1} \delta_n \subseteq \bigcup_{n\in\mathcal{A}_v^2} \delta_n$, it follows that: $\mathcal{F}(\mathcal{A}_v^1) = \text{Area}\left(\delta_v \cup \bigcup_{n\in\mathcal{A}_v^1} \delta_n\right) \leq \text{Area}\left(\delta_v \cup \bigcup_{n\in\mathcal{A}_v^2} \delta_n\right) = \mathcal{F}(\mathcal{A}_v^2)$. Thus, $\mathcal{F}$ is monotone non-decreasing, indicating that adding more nodes will not reduce the total perception coverage.
- *Submodularity*: Let $\mathcal{A}_v^1 \subseteq \mathcal{A}_v^2 \subseteq \mathcal{N}$ and $n \in \mathcal{N} \setminus \mathcal{A}_v^2$. Define the marginal coverage gain from adding node n to a set $\mathcal{A}_v$ as:

$$\Delta_n(\mathcal{A}_v) = \mathcal{F}(\mathcal{A}_v \cup \{n\}) - \mathcal{F}(\mathcal{A}_v)$$

We show that $\Delta_n(\mathcal{A}_v^1) \geq \Delta_n(\mathcal{A}_v^2)$ because as the selected set grows, the new region introduced by δ_n has more overlap with the already covered area. Therefore, the additional coverage (i.e., marginal area gain) from adding n is larger when added to the smaller set $\mathcal{A}_v^1$:

$$\mathcal{F}(\mathcal{A}_v^1 \cup \{n\}) - \mathcal{F}(\mathcal{A}_v^1) \geq \mathcal{F}(\mathcal{A}_v^2 \cup \{n\}) - \mathcal{F}(\mathcal{A}_v^2)$$

Thus, $\mathcal{F}(\mathcal{A}_v)$ satisfies the diminishing returns property, and is therefore submodular.

APPENDIX B
PROOF OF LEMMA 1

A. Assumptions and Definitions

We intuitively treat the well-trained model as a closed-box function, and denote its composite weight matrix as $\mathbf{W}^{m\times d} \in \mathbb{R}$, which encompasses all learnable parameters. The feature extraction process is then modeled as $\mathbf{f} = \mathbb{W}(\mathbf{x})$.

Assumption 1. ***Weight Distribution***. *Each neural network weight $W_{ij} \in \mathbf{W}$ is independent and follows a zero-mean Gaussian distribution:*

$$W_{ij} \sim \mathcal{N}(0, \sigma_w^2), \quad \forall i, j \tag{18}$$

where σ_w^2 denotes the weight variance.

Assumption 2. ***Input Statistics***. *Input vectors $\mathbf{x} \in \mathbb{R}^d$ satisfy:*

$$\mathbb{E}[\mathbf{x}] = \mathbf{0}, \quad \mathbb{E}[\mathbf{x}\mathbf{x}^T] = \sigma^2 I \tag{19}$$

where σ^2 represents the input feature variance.

Definition 2. ***Magnitude-based Weight Pruning*** *[18]. Given pruning rate $\rho \in [0, 1)$, we remove weights with smallest absolute values. The pruned weights satisfy:*

$$\widetilde{W}_{ij} = \begin{cases} 0 & \text{if } |W_{ij}| \leq \theta \\ W_{ij} & \text{otherwise} \end{cases} \tag{20}$$

where threshold θ satisfies $\mathbb{P}(|W_{ij}| \leq \theta) = \rho$.

B. Threshold Determination

For Gaussian-distributed weights, the absolute values follow a *folded normal distribution*:

$$\mathbb{P}(|W_{ij}| \leq \theta) = \text{erf}\left(\frac{\theta}{\sigma_w\sqrt{2}}\right) \tag{21}$$

Solving for $\theta(\rho)$ yields:

$$\theta(\rho) = \sigma_w\sqrt{2} \cdot \text{erf}^{-1}(\rho) \tag{22}$$

C. Mean Squared Error Derivation

Lemma 2. ***Single Weight Perturbation***. *For a weight $W_{ij} \sim \mathcal{N}(0, \sigma_w^2)$ pruned with threshold $\theta = \mathbb{P}(|W_{ij}| \leq \theta)$, the expected squared difference is:*

$$\mathbb{E}[(W_{ij} - \widetilde{W}_{ij})^2] = \sigma_w^2\left(\rho - \frac{2}{\sqrt{\pi}}\text{erf}^{-1}(\rho)e^{-(\text{erf}^{-1}(\rho))^2}\right) \tag{23}$$

Proof. $\Delta W_{ij} = W_{ij} - \widetilde{W}_{ij}$, $\mathbb{E}[(\Delta W_{ij})^2] = \mathbb{E}[W_{ij}^2 \mathbf{1}_{\{|W_{ij}|\leq\theta\}}]$.

$$\begin{aligned}\mathbb{E}[W_{ij}^2 \mathbf{1}_{\{|W_{ij}|\leq\theta\}}] &= \frac{2}{\sqrt{2\pi}\sigma_w}\int_0^\theta w^2 e^{-w^2/(2\sigma_w^2)} dw \\ &= \frac{2\sigma_w^2}{\sqrt{\pi}}\int_0^{\text{erf}^{-1}(\rho)} z^2 e^{-z^2} dz \\ &= \sigma_w^2\left(\rho - \frac{2}{\sqrt{\pi}}\text{erf}^{-1}(\rho)e^{-(\text{erf}^{-1}(\rho))^2}\right)\end{aligned}$$

where $\mathbf{1}_{(\cdot)}$ is an indicator function, $z = w/(\sigma_w\sqrt{2})$. □

For a network with N parameters, the feature MSE is:

$$\text{MSE}(\rho) = \sigma^2 N\sigma_w^2\left(\rho - \frac{2}{\sqrt{\pi}}\text{erf}^{-1}(\rho)e^{-(\text{erf}^{-1}(\rho))^2}\right) \tag{24}$$

D. Feature Quality Metric

Definition 3. ***Normalized Quality***. *The retained feature quality metric $Q(\rho) \in [0, 1)$ is defined as:*

$$Q(\rho) = 1 - \frac{MSE(\rho)}{MSE_{max}} = 1 - \left(\rho - \frac{2}{\sqrt{\pi}}\text{erf}^{-1}(\rho)e^{-(\text{erf}^{-1}(\rho))^2}\right) \tag{25}$$

where $MSE_{max} = \sigma^2 N\sigma_w^2$ indicating that all weights are pruned, i.e., $\rho = 1$.

E. Monotonicity

Lemma 3. *The feature quality function $Q(\rho)$ is monotonically decreasing over $\rho \in [0, 1)$.*

Proof. Differentiating $Q(\rho)$:

$$\begin{aligned}\frac{dQ}{d\rho} &= -\left(1 - \frac{2}{\sqrt{\pi}}\frac{d}{d\rho}\left[\text{erf}^{-1}(\rho)e^{-(\text{erf}^{-1}(\rho))^2}\right]\right) \\ &\overset{(a)}{=} -2z^2 < 0, \quad \forall\rho \in [0, 1)\end{aligned}$$

where (a) comes from the product rule and known derivatives, i.e., $z = \text{erf}^{-1}(\rho)$ and $\frac{d}{d\rho}\text{erf}^{-1}(\rho) = \frac{\sqrt{\pi}}{2}e^{z^2}$. □

F. Boundary Conditions

- *Unpruned Case ($\rho = 0$):* $\lim_{\rho \to 0} Q(\rho) = 1$ indicates that when no pruning is applied, the model retains its full capacity and extracts features with the same precision as the original well-trained model.
- *Full Pruning ($\rho = 1$):* $\lim_{\rho \to 1} Q(\rho) = 0$ implies that when the model is entirely pruned, its ability to extract meaningful features is lost with zero feature quality.

Remark 3. *The Gaussian assumption can be relaxed using Chebyshev inequality for general weight distributions:*

$$\mathbb{P}(|W_{ij}| \geq \theta) \geq \frac{\sigma_w^2}{\theta^2 + \sigma_w^2} \tag{26}$$

providing worst-case bounds for non-Gaussian scenarios.

APPENDIX C
PROOF OF THEOREM 2

The Lagrangian function of the convex problem (11) can be constructed as:

$$\mathcal{L}\Big(\{b_n\}, T, \lambda, \{\mu_n\}\Big) = T + \lambda \left(\sum_{n \in \mathcal{A}_v} b_n - B \right) + \sum_{n \in \mathcal{A}_v} \mu_n \left(\frac{F_m(1-\rho_n)}{c_n} + \frac{\kappa_n(1-\rho_n)}{b_n} - T \right) \tag{27}$$

where $\lambda \geq 0$ is the Lagrange multiplier associated with the total bandwidth constraint, and $\mu_n \geq 0$ corresponds to the response delay constraint for each node n. Applying the KKT conditions:

$$\frac{\partial \mathcal{L}}{\partial b_n} = \lambda - \mu_n \cdot \frac{\kappa_n(1-\rho_n)}{b_n^2} = 0 \tag{28}$$

which yields:

$$b_n^* = \sqrt{\frac{\mu_n \cdot \kappa_n(1-\rho_n)}{\lambda}} \tag{29}$$

By enforcing $\sum b_n^* = B$ and assuming uniform normalization of μ_n, we derive the closed-form solution of b_n^*.

$$b_n^* = \frac{B \cdot \sqrt{\kappa_n(1-\rho_n)}}{\sum\limits_{j \in \mathcal{A}_v} \sqrt{\kappa_j(1-\rho_j)}} \tag{30}$$

This concludes the proof.