Attention-Enhanced Hierarchical Reinforcement Learning for Air-Ground Cooperative Perception

Haixia Peng, *Senior Member, IEEE*, Yixin Fan, *Graduate Student Member, IEEE*, Zhou Su, *Senior Member, IEEE*, Tom H. Luan, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Air-ground cooperative perception (AGCP) integrates connected and autonomous vehicles (CAVs), roadside units (RSUs), and uncrewed aerial vehicles (UAVs) to provide wide-area coverage and high-resolution perception by leveraging their complementary perception and communication capabilities. However, the dynamic and heterogeneous characteristics of the air-ground network introduce strong cross-layer coupling across perception, communication, and computation, thereby complicating the coordination of cooperation update intervals and cooperation partner selection. To address these challenges, we develop a unified AGCP framework that jointly models LiDAR-based multi-agent perception, together with its associated communication bandwidth allocation and computation latency models, under dynamic mobility and time-varying resource conditions. Building on this framework, a multi-objective optimization problem is formulated to characterize the interplay between update interval selection and cooperation partner choice, aiming to balance perception accuracy and end-to-end latency. A Tchebycheff distance-based formulation is utilized to normalize and integrate multiple objectives into a unified optimization metric. To efficiently solve this highly coupled problem, an attention-enhanced hierarchical reinforcement learning algorithm is proposed, which leverages a two-level Markov decision process combined with an attention-enhanced actor-critic architecture. Simulation results validate that the proposed algorithm achieves a desirable trade-off between perception performance and end-to-end latency.**

***Index Terms*—Air-ground cooperative perception, connected and autonomous vehicles, uncrewed aerial vehicles, multi-objective optimization, hierarchical reinforcement learning.**

I. Introduction

INTELLIGENT transportation systems are rapidly advancing toward highly connected and autonomous architectures, wherein overall safety and operational efficiency rely critically on accurate and comprehensive environmental perception [1]. However, single-vehicle perception is fundamentally limited by sensor occlusion, restricted perception range, and constrained on-board computational resources, particularly in complex and densely populated urban environments [2]. Consequently, cooperative perception (CP) has gained substantial traction as a paradigm in which multiple agents exchange sensory information to achieve collective situational awareness and enhance perception robustness [3]. For instance, even when a vehicle's sensors are temporarily occluded by a large truck, a pedestrian may still be detected through camera or LiDAR data shared by a neighboring vehicle. Throughout this paper, perception refers to the high-level understanding of the environment, such as object detection, classification, and localization [4], [5], [6], rather than sensing in the integrated sensing and communication context, which mainly concerns signal-level target parameter estimation [7], [8]. Accordingly, this work addresses LiDAR-based cooperative object detection evaluated by detection accuracy.

With the rapid progress of multimodal perception, edge computing, and 5G/6G communications technologies, CP has evolved from traditional vehicle-centric implementations to more sophisticated air-ground integrated systems. A prominent emerging direction is the incorporation of heterogeneous agents, including connected and autonomous vehicles (CAVs), roadside units (RSUs), and uncrewed aerial vehicles (UAVs), into a unified air-ground cooperative perception (AGCP) framework [9], [10], [11]. These agents contribute complementary capabilities: (i) vehicle-to-vehicle (V2V) cooperation enables low-latency information exchange among nearby vehicles, thereby mitigating short-range occlusion effects [12]; (ii) vehicle-to-infrastructure (V2I) cooperation leverages RSUs' stable power supply and enhanced computational capacity to broaden perception coverage and alleviate onboard processing burdens [13]; and (iii) UAV-assisted perception provides elevated viewpoints capable of monitoring line-of-sight-constrained regions and large intersections [14]. By integrating these modalities, an air-ground perception network can simultaneously achieve wide-area situational awareness together with fine-grained local accuracy.

Despite its significant potential, the practical realization of AGCP remains challenging due to the highly dynamic, heterogeneous, and tightly coupled nature of perception, communication, and computation processes. The mobility of vehicles and UAVs, together with rapidly time-varying wireless channels, induces continuous fluctuations in perception visibility and link quality [15], rendering CP inherently time-varying and necessitating frequent strategy adaptation [16]. A central challenge therefore lies in jointly coordinating

Received 5 January 2026; revised 10 June 2026; accepted 14 July 2026. Date of publication 20 July 2026; date of current version 28 July 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62301411 and Grant 62171352, and in part by the Dongguan Strategic Scientist Teams Project under Grant 20231900700022. The associate editor coordinating the review of this article and approving it for publication was Y. Ji. *(Corresponding author: Yixin Fan.)*

Haixia Peng is with the School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523808, China, and also with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: haixia.peng@xjtu.edu.cn).

Yixin Fan is with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: fanyixin@stu.xjtu.edu.cn).

Zhou Su and Tom H. Luan are with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: zhousu@ieee.org; tom.luan@xjtu.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TCCN.2026.3715210

the cooperation update interval and partner selection. The cooperation update interval is defined as the time duration between consecutive updates of the cooperation strategy, determining the frequency at which the system re-evaluates and reconfigures its connections, whereas partner selection is defined as the decision of which neighboring agents are chosen to participate in cooperative perception within each update interval. These two factors are strongly interdependent; namely, frequent updates improve responsiveness but incur higher signaling overhead [17], whereas infrequent updates enhance stability at the cost of adaptability [18]. Moreover, perception, communication, and computation are tightly intertwined. For example, aggressive feature compression mitigates transmission delay but degrades perception fidelity and increases reconstruction and fusion complexity, while transmitting dense features improves accuracy at the expense of bandwidth consumption and communication latency. These cross-layer interdependencies establish a complex coupled trade-off, wherein optimizing one process inevitably impacts the others [19]. Consequently, any static or simplified models fail to capture the evolving balance between perception accuracy and end-to-end latency in dynamic air-ground environments, highlighting the need for holistic modeling and adaptive coordination mechanisms that jointly account for coupled perception, communication, and computation dynamics.

Recent studies have investigated the aforementioned complex problem from multiple perspectives. In communication scheduling and cooperation coordination, mobility-aware frameworks have been proposed to determine cooperation partners and data exchange timing under bandwidth constraints. For instance, Zhu et al. presented an opportunistic V2V relaying strategy to assist V2I perception uploads, improving detection accuracy in dynamic networks [20], while Qu et al. designed an adaptive policy that enables CAV pairs to switch between independent and CP modes in response to network variations [21]. Beyond coordination, several works jointly considered communication, computation, and perception. Hu et al. proposed CodeFilling, a codebook-based message compression and information-filling framework that significantly reduces bandwidth consumption without degrading perception performance [22]. Sarlak et al. jointly optimized CP formation, V2X channel allocation, and transmit power, thereby enhancing end-to-end transmission efficiency and perception quality under LTE/5G NR-V2X conditions [23]. Robustness under imperfect communication has also been investigated, including interruption-aware cooperative frameworks that predict and compensate for delayed or missing features, and asynchronous fusion methods such as CoBEVFlow [24] and LRCP [25], which mitigate latency-induced misalignment via temporal flow estimation. Beyond these model-based designs, deep reinforcement learning (DRL) has been increasingly adopted to handle the dynamic and high-dimensional decision-making of air-ground cooperative networks. In particular, a body of work has studied DRL-based UAV-assisted integrated sensing, computation, and communication, where Liu et al. jointly scheduled tasks and allocated resources to balance sensing, computation, and communication in air-ground cooperative networks [7], and Qin et al. employed soft actor-critic to jointly optimize user association, UAV trajectory, and power allocation in multi-UAV ISAC networks [8]. Another active direction concerns multi-modal trajectory planning for cooperative sensing and communication, exemplified by the emergency-oriented air-ground trajectory planning of Liu et al. for networks comprising CAVs, base stations, and UAVs [26]. Despite these advances in coordination efficiency, resource trade-off optimization, and communication robustness, existing works share a common limitation: the cooperation strategy is implicitly assumed to be reconfigurable at no cost, so the rate at which cooperation is updated is either fixed a priori or left unmodeled. As a result, the temporal dimension of cooperation is treated as a given rather than a decision, and its coupling with partner selection is neglected. Consequently, how to adaptively coordinate the cooperation update interval and partner selection so as to balance perception accuracy and end-to-end latency remains an open problem, which motivates the holistic and adaptive framework proposed in this work.

Motivated by these gaps, this paper proposes a unified AGCP framework that jointly models perception, communication, and computation. In contrast to existing schemes that regard cooperation as instantaneously reconfigurable, this work recognizes that reconfiguring the cooperation strategy incurs switching overhead, giving rise to a fundamental trade-off between the temporal stability and spatial adaptivity of cooperation. Accordingly, the cooperation update interval is modeled as an explicit decision variable and jointly optimized with partner selection, supported by a comprehensive system model that characterizes perception accuracy and end-to-end latency under heterogeneous agents and dynamic network conditions. The main contributions of this paper are summarized as follows:

- We propose an integrated AGCP framework that jointly models CAVs, RSUs, and UAVs within a heterogeneous perception-communication-computation system. This framework captures the intrinsic coupling among perception, communication, and computation under dynamic traffic and network conditions, providing a consistent foundation for system-level optimization.
- We model the cooperation update interval as an explicit decision variable and formulate a multi-objective optimization problem that jointly optimizes it with cooperation partner selection, capturing their interaction across short- and long-term horizons under time-varying communication and computation conditions. A Tchebycheff-based scalarization is further introduced to normalize and balance perception accuracy and end-to-end latency within a unified optimization framework.
- To efficiently solve this coupled problem, we propose an attention-enhanced hierarchical reinforcement learning (AHRL) algorithm based on a two-level Markov decision process (MDP), in which the high-level agent adaptively determines the cooperation update interval and the low-level agent selects cooperation partners. An attention mechanism captures the correlations among candidates, and proximal policy optimization (PPO) is adopted for stable training under the non-stationary dynamics induced by the hierarchical structure.

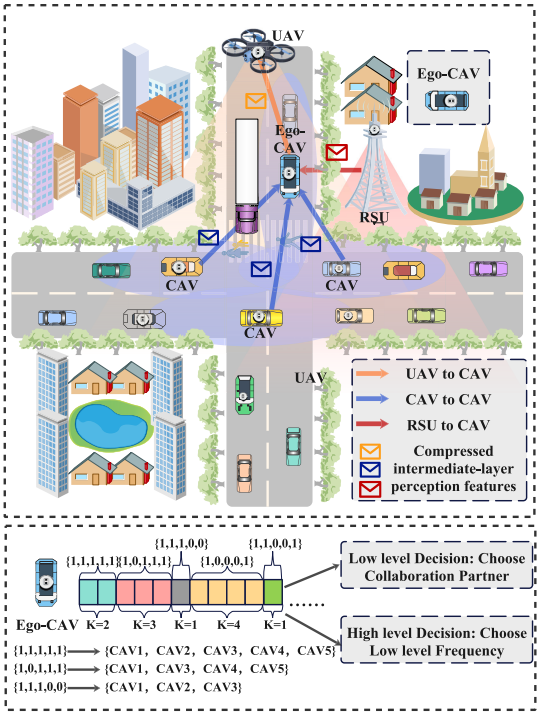


Fig. 1. Illustration of the air-ground cooperative perception scenario.

The remainder of this paper is organized as follows. Section II presents the system models for perception, communication, and computation, and provides the formal problem formulation. Section III introduces the proposed solution, detailing the hierarchical MDP and the AHRL algorithm. Section IV provides comprehensive simulation results and performance evaluations. Finally, Section V concludes the paper.

II. System Model

A. Air-Ground Cooperative Perception Framework

This study considers an AGCP system composed of heterogeneous intelligent agents, including CAVs, RSUs, and UAVs, as shown in Fig. 1. The objective of the system is to improve the perception accuracy of a designated host vehicle, referred to as the ego-CAV, by dynamically establishing cooperative relationships with nearby cooperation partners that exhibit complementary perception and communication capabilities. This work focuses on the cooperative decision-making of a single ego-CAV, which independently selects its own cooperation partners that act solely as feature senders without altering their own perception behavior. To effectively manage the cross-layer coupling, the ego-CAV employs a hierarchical decision-making structure. As illustrated in Fig. 1, the decision process is decomposed into two timescales: a high-level decision determines the cooperation update interval (denoted as K) to govern the frequency of strategy updates, while a low-level decision selects the specific agents that participate in cooperative perception and information sharing during the corresponding interval.

Once the decisions are made, the selected cooperation partners, including the neighboring CAVs, RSUs, and UAVs, transmit compressed features to the ego-CAV. These features correspond to compact, high-level activations extracted from the agents' local perception backbones, enabling the exchange of rich semantic information while avoiding excessive bandwidth costs. UAVs provide top-down aerial coverage to mitigate ground-level occlusions, RSUs offer stable infrastructure-assisted viewpoints with reliable communication, and neighboring CAVs contribute horizontal perspectives to enhance spatial completeness. Within this framework, each RSU is modeled as a fixed roadside unit deployed on existing poles or gantries, equipped with a mechanically stable LiDAR sensor, an edge computing module for local feature extraction, and a C-V2X/DSRC communication interface for feature transmission, which is consistent with typical intelligent roadside infrastructure described in [27] and [28]. Owing to their fixed, elevated installation and grid power supply, RSUs deliver stable wide-area coverage and reliable links. Although the deployment incurs non-negligible installation and maintenance costs, such infrastructure is increasingly shared across multiple intelligent transportation services, so its amortized cost per perception task remains moderate. The UAVs are assumed to follow given flight trajectories, and their trajectory optimization and onboard energy consumption are not jointly optimized in this work, which instead focuses on the cooperative perception decision-making, i.e., the coordination of the cooperation update interval and partner selection. The shared features are transmitted over three distinct communication modalities: UAV-to-CAV links for aerial perception, CAV-to-CAV links for vehicle-level cooperation, and RSU-to-CAV links for infrastructure-assisted perception. As indicated by the colored arrows in Fig. 1, UAV-to-CAV (orange), CAV-to-CAV (blue), and RSU-to-CAV (red) links jointly facilitate efficient feature exchange among heterogeneous cooperation partners.

Upon receiving multi-source features, the ego-CAV integrates them with its local perception backbone to construct a unified feature representation. This fusion process effectively captures complementary spatial and semantic cues across cooperation partners, enabling the ego-CAV to detect objects that would otherwise be occluded or beyond its perception range. As illustrated in Fig. 1, the proposed cooperative perception framework can substantially improve the perception accuracy compared with stand-alone perception.

B. LiDAR-Based Cooperative Perception Pipeline

A LiDAR-based perception system for CAVs is considered, wherein CP is leveraged to enhance perception accuracy [4], [29]. As shown in Fig. 2, the perception pipeline comprises five sequential modules: point cloud acquisition, point cloud preprocessing, feature extraction, feature selection, and multi-agent feature fusion. The design enables each CAV $i \in \mathcal{V}$ to process its local LiDAR observations, selectively share condensed spatial information, and fuse shared features with its own to attain a comprehensive environmental understanding.

Data acquisition provides a point cloud for each CAV i, denoted by $\mathcal{P}_i = \{\mathbf{p}_{i,\ell} \in \mathbb{R}^4\}_{\ell=1}^{N_i}$, where each point $\mathbf{p}_{i,\ell} = (x, y, z, r)$ contains 3D coordinates and reflectance intensity. The raw point cloud undergoes a series of preprocessing operations, yielding a refined point cloud formulated

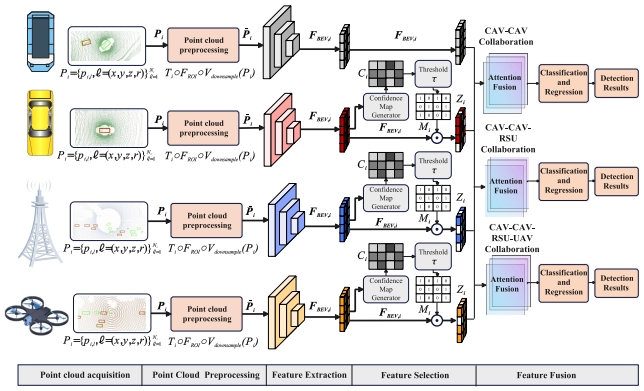


Fig. 2. LiDAR-based cooperative perception pipeline illustrating point cloud preprocessing, feature extraction, confidence-guided feature selection, and multi-agent feature fusion across CAVs, RSUs, and UAVs.

as $\bar{\mathcal{P}}_i = \mathcal{T}_i \circ \mathcal{F}_{\text{ROI}} \circ \mathcal{V}_{\text{down}}(\mathcal{P}_i)$. Here, $\mathcal{V}_{\text{down}}$ performs voxel-based downsampling to control computational complexity, $\mathcal{F}_{\text{ROI}}$ extracts points within a predefined region of interest, $\mathcal{T}_i$ transforms points into the ego-CAV coordinate frame, and $\circ$ denotes the function composition operator. The preprocessed point cloud is subsequently encoded into a high-dimensional bird's-eye view (BEV) feature map via a feature encoder $\mathcal{E}$, given as

$$\mathbf{F}_{\text{BEV},i} = \mathcal{E}(\bar{\mathcal{P}}_i; \boldsymbol{\theta}_{\text{enc}}) \in \mathbb{R}^{H \times W \times D}, \tag{1}$$

where H, W, and D denote the height, width, and channel dimensions, respectively [30].

To enable communication-efficient cooperation, a confidence-guided feature selection mechanism is utilized. A spatial confidence map $\mathbf{C}_i$ is obtained by first computing raw scores via the detection head $\mathcal{D}$ and subsequently normalizing them across channels, expressed by $\mathbf{C}_i = \max_{\text{ch}}\left(\sigma(\mathcal{D}(\mathbf{F}_{\text{BEV},i}))\right)$, where $\sigma(\cdot)$ denotes the sigmoid function and $\max_{\text{ch}}(\cdot)$ represents the channel-wise maximum operator [6]. Optionally, a Gaussian smoothing filter may be applied to improve robustness. For $i \neq$ ego, a binary communication mask $\mathbf{M}_i$ is then obtained by thresholding $\mathbf{C}_i$ with a predefined threshold τ, defined as

$$\mathbf{M}_i(h,w) = \begin{cases} 1, & \mathbf{C}_i(h,w) > \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This mask realizes a device-side filtering step, by which each sender, whether a CAV, RSU, or UAV, encodes its 3D point-cloud data into a compact BEV feature map and locally retains only the high-value regions for transmission. Such device-side filtering, however, is performed independently by each agent based solely on its own confidence map, and therefore captures only the high-value information within a single agent. The selection of the most informative cooperation partners consequently remains a system-level decision that depends on the joint configuration of all candidates.

For the ego-CAV, the mask is set to one, i.e., $\mathbf{M}_{\text{ego}} = 1$. The resulting mask prunes the BEV feature map by element-wise multiplication, yielding a sparse map $\mathbf{Z}_i = \mathbf{F}_{\text{BEV},i} \odot \mathbf{M}_i$ for transmission. As a receiver, CAV i fuses its local features with sparse features from neighbors in $\mathcal{S}_i$. All received maps, $\{\mathbf{Z}_j\}_{j\in\mathcal{S}_i}$, are first warped into the ego-CAV frame using transformation matrices, and the aligned maps are then aggregated with an attention-enhanced fusion operator $\mathcal{F}_{\text{att}}$ [5]. The fusion step is described by

$$\mathbf{F}_{\text{fused},i} = \mathcal{F}_{\text{att}}\left(\mathbf{F}_{\text{BEV},i}, \{\mathcal{W}(\mathbf{Z}_j, \mathbf{T}_{j\to i})\}_{j\in\mathcal{S}_i}\right), \tag{3}$$

where $\mathcal{W}(\cdot)$ denotes the warping operation. $\mathcal{F}_{\text{att}}$ is realized as a multi-head attention operator that, for each BEV spatial location, treats the ego feature as the query and the spatially co-located features from cooperation partners as keys and values, thereby fusing multi-source features in a position-wise manner. The fused features are subsequently fed to the detection head to produce final predictions, denoted as $\{\mathbf{b}_\ell, c_\ell, s_\ell\}_{\ell=1}^{N_{\text{det}}} = \mathcal{D}(\mathbf{F}_{\text{fused},i}; \boldsymbol{\psi})$. Here, $\mathbf{b}_\ell \in \mathbb{R}^7$ encodes $(x, y, z, l, w, h, \theta)$, representing the center coordinates, dimensions, and heading angle, respectively. c_ℓ is the class label, s_ℓ is the confidence score, and N_{det} is the number of detections.

Perception accuracy is quantified by average precision (AP) [31]. For each detection, the intersection over union (IoU) with a ground-truth box $\mathbf{g}_\ell$ is defined as $\text{IoU}(\mathbf{b}_\ell, \mathbf{g}_\ell) = |\mathbf{b}_\ell \cap \mathbf{g}_\ell| / |\mathbf{b}_\ell \cup \mathbf{g}_\ell|$. Detections are classified as true positives (TP) when the IoU exceeds a threshold (e.g., 0.7). Otherwise, they are categorized as false positives (FP). AP is computed as the area under the precision-recall curve, described by

$$\text{AP} = \sum_{m=1}^{M} \big(R(m) - R(m-1)\big)\, P(m), \tag{4}$$

where $R(m)$ and $P(m)$ denote the recall and precision at operating point m, respectively.

C. Computational and Communication Cost Modeling

This section establishes a comprehensive cost model to quantify the end-to-end latency, formulated from the perspective of the ego-CAV whose objective is to select an optimal set of cooperation partners $\mathcal{C}_{\text{AP}}(n)$ through a binary decision vector $\mathbf{x}(n)$ at the decision time slot n. For conciseness, an individual selected cooperation partner is hereafter also referred to as a collaborator, particularly when indexed by a subscript k in the cost expressions. The model adopts a parallel processing paradigm, where the end-to-end latency is determined by the critical path among concurrent computation and communication tasks. Specifically, the framework assumes that: (i) all cooperating senders conduct feature extraction and mask generation in parallel with data transmission; (ii) the ego-CAV performs its initial data preprocessing simultaneously with the senders' operations; and (iii) the final fusion step begins only after both local preprocessing and data reception from all selected cooperation partners are completed.

1) Computation Model: The computation latency L^{comp} of a task is defined as the ratio between the required number of floating point operations (FLOPs), Φ, and the computational capability C_c, described by $L^{\text{comp}} = \frac{\Phi}{C_c}$.

The total workload Φ consists of three principal components in the perception pipeline. The first component represents the computational cost of sensor data preprocessing and the backbone feature extractor (e.g., a 3D voxel-based neural network), denoted as Φ^{preproc}. Two variants are considered: $\Phi^{\text{preproc\_full}}$, referring to the full high-complexity feature extraction performed by a sender to produce rich feature maps; and $\Phi^{\text{preproc\_light}}$, a lightweight version adopted by

the ego-CAV when it expects to enhance perception with cooperation partners' features. The second component, Φ^{mask}, corresponds to the cost of the mask generation module that identifies critical regions for transmission, thereby improving communication efficiency. The third component, $\Phi_{\text{fusion}}(k)$, denotes the cost of the fusion module that integrates feature maps from k cooperation partners, whose complexity grows with the number of sources involved.

Based on these definitions, the computational workload for each role is formulated as follows. Specifically, for the **No Cooperation Role**, an independent ego-CAV executes lightweight preprocessing and its detection head on local sensor data, without performing any feature fusion, formulated by $\Phi_{\text{ego}}^{\text{no\_collab}} = \Phi_{\text{ego}}^{\text{preproc\_light}} + \Phi_{\text{fusion}}(1)$. Although the notation $\Phi_{\text{fusion}}(1)$ is adopted for notational consistency, it represents the computation of the detection head and post-processing on a single input stream, rather than an actual feature fusion operation. In the **Send-Only Role**, a collaborator k performs full-scale feature extraction to produce rich feature maps and then generates a communication mask for transmission. The reason a sender performs full-scale rather than lightweight extraction is dictated by the confidence-guided feature selection pipeline in Section II-B. To generate the communication mask $\mathbf{M}_k$, the sender must first run the complete backbone and detection head to obtain the spatial confidence map $\mathbf{C}_k$, and only then can it threshold $\mathbf{C}_k$ to select the high-value regions for transmission. The full-scale extraction is thus a prerequisite for feature compression rather than an optional choice. By contrast, the ego-CAV acts as a receiver and does not generate any mask, so it only needs lightweight preprocessing before fusion. Since the detection head computation is already encompassed within $\Phi_k^{\text{preproc\_full}}$, no additional fusion-related processing is required; hence, the workload is defined as $\Phi_k^{\text{send}} = \Phi_k^{\text{preproc\_full}} + \Phi^{\text{mask}}$. Finally, for the **Receive-Only Role**, the ego-CAV first performs lightweight preprocessing and then fuses its local features with those received from all cooperation partners in $\mathcal{C}_{\text{AP}}(n)$, expressed as $\Phi_{\text{ego}}^{\text{receive}}(n) = \Phi_{\text{ego}}^{\text{preproc\_light}} + \Phi_{\text{fusion}}(|\mathcal{C}_{\text{AP}}(n)| + 1)$.

2) Communication Model: Communication latency is modeled for three types of links: UAV-CAV, RSU-CAV, and CAV-CAV. The formulation is grounded in large-scale channel statistics, including line-of-sight (LoS) and non-line-of-sight (NLoS) occurrence probabilities as well as log-normal shadowing effects, to quantify the expected end-to-end communication performance. All expectations are taken with respect to large-scale channel states [32].

The unified derivation begins with the expected received power P_r across channel states s, each occurring with probability p_s. Let PL_s denote the path loss for state s (excluding shadowing). The corresponding linear-domain attenuation factor, incorporating log-normal shadowing with standard deviation σ_s, is described as $\zeta_s = 10^{-PL_s/10} \exp\left(\frac{1}{2}\left(\frac{\ln 10}{10}\right)^2 \sigma_s^2\right)$. Accordingly, the expected received power is formulated as

$$P_r = P_t G_t G_r \, 10^{-L_{\text{misc}}/10} \sum_s p_s \zeta_s, \tag{5}$$

where P_t is the transmit power, G_t and G_r are the antenna gains, and L_{misc} denotes the miscellaneous losses in decibels.

The channel bandwidth is dynamically allocated from a finite spectral pool shared among cooperation partners of the same type. For a collaborator k transmitting to the ego-CAV at time slot n, the allocated bandwidth is determined by a fair-sharing rule, expressed as $B_{k\to\text{ego}}(n) = B_{\text{total, type}(k)}/N_{\text{tx, type}(k)}(n)$. For brevity, let $B \triangleq B_{k\to\text{ego}}(n)$. The signal-to-interference-plus-noise ratio (SINR) and the corresponding expected achievable rate R_{exp} are given by

$$\text{SINR} = \frac{P_r}{N_0 B + I}, \quad R_{\text{exp}} = B \log_2\left(1 + \text{SINR}\right), \tag{6}$$

where N_0 denotes the noise power spectral density and I represents the interference power. Consequently, the communication latency required to transmit a payload of size S from collaborator k to the ego-CAV at slot n is calculated as $L_{k\to\text{ego}}^{\text{comm}}(n) = S/R_{\text{exp}}$.

Path-loss characteristics PL_s and probabilities p_s depend on the specific link type. For links exhibiting probabilistic visibility conditions, we have $p_{\text{NLoS}} = 1 - p_{\text{LoS}}$, and the channel state is modeled as $s \in \{\text{LoS}, \text{NLoS}\}$.

a) UAV-CAV links: The LoS probability between a UAV and a CAV depends on the elevation angle θ, geometrically defined by $\theta = \arctan((h_{\text{UAV}} - h_{\text{CAV}})/d_h)$, where h_{UAV} and h_{CAV} denote the antenna heights and d_h is the horizontal separation. The LoS probability follows the Al-Hourani model [33], given as

$$p_{\text{LoS}}^{\text{UAV-CAV}}(\theta) = \frac{1}{1 + a \exp\left[-b(\theta - a)\right]}, \tag{7}$$

where (a, b) are parameters fitted to the specific urban morphology. The expected path losses for LoS and NLoS conditions follow a log-distance model with zero-mean Gaussian shadowing, expressed as

$$PL_{\text{LoS}}^{\text{UAV-CAV}}(d) = PL_0 + 10\, n_{\text{L}} \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma,\text{L}}, \tag{8}$$

$$PL_{\text{NLoS}}^{\text{UAV-CAV}}(d) = PL_0 + 10\, n_{\text{N}} \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma,\text{N}}, \tag{9}$$

where d is the 3D distance, n_{L} and n_{N} denote the path-loss exponents for LoS and NLoS states, respectively, and PL_0 is the free-space reference loss at 1 m.

b) RSU-CAV links: Following the 3GPP urban micro-cell (UMi) street-canyon model, the LoS probability for RSU-CAV links is a piecewise function of the 2D distance d_{2D} [32], given as

$$p_{\text{LoS}}^{\text{RSU-CAV}}(d_{\text{2D}}) = \begin{cases} 1, & d_{\text{2D}} \le 18 \text{ m}, \\ \dfrac{18}{d_{\text{2D}}} + \exp\left(\dfrac{-d_{\text{2D}}}{36}\right) & \\ \quad \times \left(1 - \dfrac{18}{d_{\text{2D}}}\right), & d_{\text{2D}} > 18 \text{ m}. \end{cases} \tag{10}$$

The corresponding LoS and NLoS path losses are empirically modeled as functions of the 3D distance d_{3D} and carrier frequency f_c [32], expressed as

$$\begin{aligned} PL_{\text{LoS}}^{\text{RSU-CAV}}(d_{\text{3D}}) &= 32.4 + 21 \log_{10}(d_{\text{3D}}) \\ &\quad + 20 \log_{10}(f_c) + X_{\sigma,\text{L}}, \\ PL_{\text{NLoS}}^{\text{RSU-CAV}}(d_{\text{3D}}) &= 22.4 + 35.3 \log_{10}(d_{\text{3D}}) \end{aligned} \tag{11}$$

$$+ 21.3 \log_{10}(f_c) - 0.3\,(h_{\text{CAV}} - 1.5) + X_{\sigma,\text{N}}, \quad (12)$$

where h_{CAV} denotes the CAV antenna height in meters.

c) CAV-CAV links: For CAV-CAV, the link condition is deterministically identified based on geometric visibility among vehicles rather than probabilistic channel states. According to the 3GPP TR 37.885 vehicular channel model [34], three visibility conditions are distinguished, namely LoS, vehicle-non-line-of-sight (NLoSv), and NLoS. The LoS condition applies when the direct path between the two CAVs is unobstructed. The NLoSv condition is assigned when the direct path is blocked, specifically by other vehicles, which act as mobile obstacles that introduce an additional but moderate blockage attenuation L_{blk} on top of the LoS path loss. In contrast, the NLoS condition corresponds to obstruction by static environmental structures such as buildings, which causes a more severe attenuation governed by a distinct path-loss exponent. The three conditions are formally expressed as

$$PL_{\text{LoS}}^{\text{CAV-CAV}}(d) = PL_0 + 10\beta_{\text{L}} \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma,\text{L}}, \quad (13)$$

$$PL_{\text{NLoSv}}^{\text{CAV-CAV}}(d) = PL_{\text{LoS}}^{\text{CAV-CAV}}(d) + L_{\text{blk}}, \quad (14)$$

$$PL_{\text{NLoS}}^{\text{CAV-CAV}}(d) = PL_0 + 10\beta_{\text{N}} \log_{10}\left(\frac{d}{d_0}\right) + X_{\sigma,\text{N}}, \quad (15)$$

where β_{L} and β_{N} denote the path-loss exponents for LoS and NLoS states, respectively. L_{blk} represents the additional attenuation caused by vehicle blockage. $X_{\sigma,\bullet}$ captures the large-scale shadowing effects.

3) Overall Latency Synthesis: The end-to-end latency of the cooperative task is synthesized by combining the computation and communication models under a parallel processing paradigm. A sender-side task for collaborator k at slot n consists of local computation followed by data transmission, where the latency is defined as $L_{k,\text{task}}(n) = L_{k,\text{send}}^{\text{comp}}(n) + L_{k\to\text{ego}}^{\text{comm}}(n)$.

The ego-CAV can initiate the final fusion only after its own initial preprocessing is completed and the last message from all selected cooperation partners has arrived. Consequently, the data-ready latency is described by the critical path among parallel events, given as

$$L_{\text{data\_ready}}(n) = \max\left(L_{\text{ego,preproc}}^{\text{comp}}(n),\ \max_{k\in\mathcal{C}_{\text{AP}}(n)}\{L_{k,\text{task}}(n)\}\right). \quad (16)$$

A switching latency is incurred if the set of cooperation partners changes with respect to the previous decision step, capturing network re-association and application-level pipeline reconfiguration. This is formulated as $L_{\text{switch}}(n) = L_{\text{switch}}^{\text{const}} \cdot \mathbb{I}(\mathcal{C}_{\text{AP}}(n) \neq \mathcal{C}_{\text{AP}}(n-1))$, where $\mathbb{I}(\cdot)$ is the indicator function. Finally, the total end-to-end latency at step n accumulates these components, expressed as $L_{\text{total}}(n) = L_{\text{switch}}(n) + L_{\text{data\_ready}}(n) + L_{\text{ego,postproc}}^{\text{comp}}(n)$.

D. Problem Formulation

Building upon the previously described perception pipeline and cost models, this section formally defines the overarching optimization problem of the system.

1) Decision Variables: To characterize the hierarchical control structure, two sets of decision variables are introduced as follows. The problem is defined over a sequence of steps indexed by n, within the mission duration.

- *Cooperation Partner Selection:* For any given time step n, the set of active cooperation partners $\mathcal{C}_{\text{AP}}(n)$ is determined from the candidate pool $\mathcal{V} \setminus \{\text{ego}\}$. This selection status is represented by a binary vector $\mathbf{x}(n)$, where each element $x_k(n)$ indicates whether collaborator k is included in the cooperation set, i.e., $x_k(n) = 1$ if collaborator k is selected and $x_k(n) = 0$ otherwise.
- *Cooperation Update Interval:* The high-level controller governs the cooperation update interval of the partner selection process. At each of its decision points m, it selects an integer k_m from a discrete set of admissible frequencies $\mathcal{K}$, implying that the partner selection vector $\mathbf{x}(n)$ is updated every k_m time steps.

2) Objective Function: The system is designed to balance two competing objectives: maximizing the perception accuracy (quantified by AP) and minimizing the end-to-end latency $L_{\text{total}}(n)$. To achieve a unified scalar objective, the Tchebycheff method is used to minimize the maximum weighted deviation from the ideal point $(\text{AP}_{\max}, L_{\min})$. The reason for using the max operator instead of a weighted sum is that the max formulation always penalizes the worst of the two normalized objectives, thereby enforcing a balanced improvement of both. In contrast, a weighted-sum objective allows a large gain in one objective to compensate for a severe degradation in the other, which may lead to degenerate solutions that, for instance, minimize latency while substantially sacrificing perception accuracy. The max operator avoids such an imbalance and is therefore better suited to the accuracy-latency trade-off targeted in this work. The normalized distances from this ideal point are given as $d_{AP,n} = \frac{\text{AP}_{\max} - \text{AP}(s_n, \mathbf{x}(n))}{\text{AP}_{\max} - \text{AP}_{\min}}$ and $d_{L,n} = \frac{L_{\text{total}}(n) - L_{\min}}{L_{\max} - L_{\min}}$, where s_n represents the global environmental state at step n (including agent positions and occlusion conditions), which determines the baseline information availability for the AP calculation given the partner selection $\mathbf{x}(n)$. And $L_{\text{total}}(n)$ is derived from the previously defined computation and communication models.

Based on these normalized distances, the overall optimization problem is formulated to determine the decision variable sequences that minimize the cumulative Tchebycheff distance over the entire horizon, described by

$$\begin{aligned}
\min_{\{\mathbf{x}(n)\},\{k_m\}}\ & \sum_n \max\{\lambda_{AP} d_{AP,n}, \lambda_L d_{L,n}\} \\
\text{s.t. C1: } & x_k(n) \in \{0,1\}, \quad \forall n, k \in \mathcal{V} \setminus \{\text{ego}\}, \\
\text{C2: } & k_m \in \mathcal{K}, \quad \forall m, \\
\text{C3: } & x_k(n) \le A_n(k), \quad \forall n, k, \\
\text{C4: } & \Phi_i(n) \le F_i k_m \Delta T, \quad \forall n, i \in \{\text{ego}\} \cup \mathcal{C}_{\text{AP}}(n), \\
\text{C5: } & x_k(n)\left(\text{SINR}_{\min} - \overline{\text{SINR}}_{k\to\text{ego}}(n)\right) \le 0, \quad \forall n, k, \\
\text{C6: } & x_k(n)\left(\text{Conf}_{\min} - \text{Conf}_k(n)\right) \le 0, \quad \forall n, k, \\
\text{C7: } & \sum_{k\in\mathcal{V}_{\text{type}}} x_k(n) B_{\min} \le B_{\text{total, type}}, \quad \forall n, \text{type}, \\
\text{C8: } & L_{\text{total}}(n) \le L_{\max}, \quad \forall n.
\end{aligned} \quad (17)$$

where λ_{AP} and λ_L ($\lambda_{AP}, \lambda_L \geq 0$) are weighting factors that balance the trade-off between perception accuracy and end-to-end latency. Specifically, C1 and C2 define the domains of the binary partner-selection and integer interval-selection variables. C3 enforces partner availability, allowing collaborator k to be chosen at step n only if its availability indicator $A_n(k)$ is active. C4 limits the computational workload to ensure per-window feasibility. The workload $\Phi_i(n)$ represents the total operations to be completed within one cooperation-update window of length $k_m \Delta T$ and must be processable by the hardware capability F_i within this duration. C5 guarantees a minimum communication link quality, requiring the average SINR ($\overline{\text{SINR}}_{k\to\text{ego}}(n)$) to exceed $\text{SINR}_{\min}$, while C6 imposes a perception confidence constraint ensuring that the confidence score $\text{Conf}_k(n)$ derived from $\mathbf{C}_k$ remains above $\text{Conf}_{\min}$. C7 addresses shared bandwidth limitations by ensuring that the aggregate bandwidth demand does not exceed $B_{\text{total, type}}$ for each agent type. C8 constrains the end-to-end latency below the maximum permissible limit $L_{\max}$.

III. Hierarchical Markov Decision Process Modeling

The optimization problem introduces a decision-making hierarchy characterized by distinct temporal scales. Specifically, cooperation partner selection determines the optimal set of collaborators for the current decision cycle (short-term execution), whereas the cooperation update interval governs the duration of this cycle to balance responsiveness against switching overhead (long-term planning). To effectively manage this cross-scale coupling, the cooperative decision-making problem is reformulated as a hierarchical Markov decision process (HMDP) comprising two interconnected layers.

A. Markov Decision Process Reformulation

1) Low-Level MDP: The low-level agent operates under a cooperation update interval k specified by the high-level layer and determines the cooperation partners to be maintained for the subsequent k time steps so as to optimize the cumulative reward.

a) Low-level state s_t^l: At each low-level decision time step t, the state s_t^l provides comprehensive information about all C candidates and consists of the following three matrices. 1) Candidate Spatial Matrix $\mathbf{F}_t \in \mathbb{R}^{C\times D_f}$, wherein C denotes the number of candidates and D_f is the spatial feature dimension. Each row $\mathbf{f}_{t,i}$ represents candidate i and is defined as the concatenation $\mathbf{f}_{t,i} = [\mathbf{o}_i, \mathbf{p}_{t,i}, v_{t,i}, \boldsymbol{\theta}_{t,i}]$, where $\mathbf{o}_i \in \{0,1\}^3$ is a one-hot vector for the agent type (CAV, RSU, UAV), $\mathbf{p}_{t,i} \in \mathbb{R}^3$ denotes the 3D spatial coordinates, $v_{t,i} \in \mathbb{R}$ is the scalar speed, and $\boldsymbol{\theta}_{t,i} \in \mathbb{R}^2$ represents the orientation (yaw and roll). Since each low-level decision spans only a short cooperation interval of k steps, within which the relative geometry among candidates varies slightly, this coarse motion representation is sufficient to support reliable partner selection. 2) Candidate Confidence Matrix $\mathbf{G}_t \in \mathbb{R}^{C\times D_g}$, wherein each row $\mathbf{g}_{t,i}$ is derived from the perceptual confidence map of candidate i and reflects its perceptual quality. 3) Candidate Resource Metrics $\mathbf{M}_t \in \mathbb{R}^{C\times D_r}$, wherein each row $\mathbf{m}_{t,i}$ quantifies the resources of candidate i, consisting of compute capacity, LoS, NLoS, NLoSv probability, total bandwidth, the number of cooperation partners, and an availability flag. Consequently, the low-level state is given by the tuple $s_t^l = (\mathbf{F}_t, \mathbf{G}_t, \mathbf{M}_t)$.

b) Low-level action a_t^l: The action is represented by a binary partner-selection vector $\mathbf{x}_t \in \{0,1\}^C$, where $\mathbf{x}_t[i] = 1$ indicates that the i-th candidate is selected as a cooperation partner; this action persists for the next k time steps.

c) Low-level reward r_t^l: The cumulative low-level reward over the interval of length k is defined as $R_t^l = \sum_{\tau=t}^{t+k-1} r_\tau$, where the instantaneous reward r_τ is described by a Tchebycheff distance that balances perception accuracy and end-to-end latency, given as $r_\tau = -\max(d_{AP,\tau}, d_{L,\tau})$ with the normalized distances defined as $d_{AP,\tau} = \frac{\text{AP}_{\max} - \text{AP}(s_\tau, \mathbf{x}_\tau)}{\text{AP}_{\max} - \text{AP}_{\min}}$ and $d_{L,\tau} = \frac{L_{\text{total}}(\tau) - L_{\min}}{L_{\max} - L_{\min}}$. Herein s_τ denotes the system state at time τ, $\text{AP}(\cdot)$ is the perception accuracy measured by AP, and $L_{\text{total}}(\tau)$ is the end-to-end latency synthesized from the previously defined computation and communication models.

2) High-Level MDP: The high-level agent is tasked with configuring the cooperation update interval for the low-level agent. It operates on a longer time scale, typically spanning P low-level decision cycles.

a) High-level state s_t^h: The state s_t^h provides a high-level abstraction of the environment and decision history and consists of three components. 1) Global Environment Summary: Aggregated statistics derived from the low-level state, including the mean spatial matrix $\bar{\mathbf{F}}_t$, the mean confidence matrix $\bar{\mathbf{G}}_t$, and the mean resource metrics $\bar{\mathbf{M}}_t$, all of which are averaged across the C candidates. 2) Historical Decision Information: Including the previously selected interval k_{prev}, the cumulative low-level reward from the last low-level interval R_{prev}^l, and the cumulative high-level reward from the last high-level interval R_{prev}^h. Since the environment summary is obtained by spatially averaging the low-level state over the C candidates, it is only a compressed observation and does not fully capture the underlying high-level state. The high-level decision process is thus partially observable. The historical decision information provides a compact summary of the unobserved low-level evolution during the previous interval, which helps the high-level agent make informed decisions under this partial observability. Consequently, the high-level state is represented as the concatenation $s_t^h = (\bar{\mathbf{F}}_t, \bar{\mathbf{G}}_t, \bar{\mathbf{M}}_t, k_{\text{prev}}, R_{\text{prev}}^l, R_{\text{prev}}^h)$.

b) High-level action a_t^h: The action a_t^h is the selection of a cooperation update interval k from a predefined discrete set $\mathcal{K} = \{k_1, k_2, \ldots, k_N\}$.

c) High-level reward r_t^h: The high-level reward over its decision period is defined as the sum of instantaneous rewards accumulated across P consecutive low-level cycles and is described by $R_t^h = \sum_{\tau=t}^{t+P\,k_{\text{prev}}-1} r_\tau$. Here, k_{prev} denotes the interval chosen by the previous high-level action. Additionally, r_τ follows the previously defined Tchebycheff-type formulation that balances perception accuracy and end-to-end latency via the normalized distances $d_{AP,\tau}$ and $d_{L,\tau}$.

B. Reinforcement Learning Solution

An AHRL algorithm is adopted to solve the formulated HMDP. Specifically, PPO is used to jointly optimize the policies of the high-level and low-level agents under an actor-critic framework. This hierarchical design decouples long-term

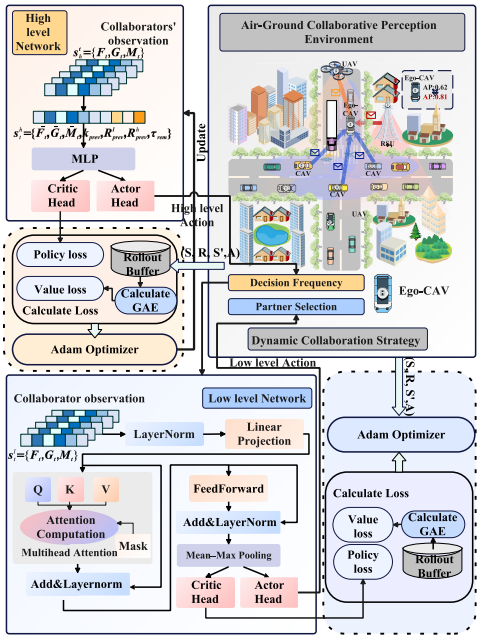


Fig. 3. Overall architecture of the proposed hierarchical reinforcement learning framework for AGCP.

strategic planning from short-term tactical execution, allowing each agent to specialize in its respective decision-making task [35].

As illustrated in Fig. 3, the proposed AHRL algorithm consists of two interconnected decision layers operating at different temporal scales. The high-level agent adjusts the cooperation update interval, whereas the low-level agent dynamically selects cooperation partners by leveraging an attention network to model inter-candidate dependencies. Each layer maintains its own actor-critic pair, and both are optimized jointly using PPO [36] with shared rollout buffers and generalized advantage estimation (GAE) [37]. The training objective follows the instantaneous reward r_τ defined previously via the Tchebycheff distances $d_{AP,\tau}$ and $d_{L,\tau}$, thereby aligning policy optimization with perception accuracy and end-to-end latency.

1) Policy and Value Network Architectures: The high-level agent learns the corresponding policy $\pi_{\theta_h}(k|s_t^h)$ and the value function $V_{\phi_h}(s_t^h)$. Since s_t^h is a compact vector, a two-layer fully connected network (FCNet) is utilized. The transformation is given as $h^{(1)} = \sigma_{\mathrm{r}}(W^{(1)}s_t^h + b^{(1)})$ and $h_t \equiv h^{(2)} = \sigma_{\mathrm{r}}(W^{(2)}h^{(1)} + b^{(2)})$, where $W^{(i)}$ and $b^{(i)}$ denote the weights and biases, and $\sigma_{\mathrm{r}}(\cdot)$ represents the ReLU activation function. The shared representation h_t is then fed into two output heads. The actor head applies a linear layer followed by a softmax to produce a probability distribution over the discrete set of decision frequencies,

$$\pi_{\theta_h}(\cdot \mid s_t^h) = \sigma_{\mathrm{sm}}\left(W_\pi h_t + b_\pi\right), \tag{18}$$

whereas the critic head uses a linear layer to estimate the state value, $V_{\phi_h}(s_t^h) = W_V h_t + b_V$.

The low-level agent learns the policy $\pi_{\theta_l}(\mathbf{x}|s_t^l)$ and the value function $V_{\phi_l}(s_t^l)$. The state s_t^l describes a set of C candidate collaborators, each characterized by its spatial, perceptual, and resource features. These candidates are not independent, as any two of them exhibit latent pairwise relationships, including the spatial relation of their viewpoints, reflected by the position and orientation entries of $\mathbf{F}_t$, and the contention for shared communication resources, reflected by the bandwidth and cooperation-count entries of $\mathbf{M}_t$. Since such relationships are not explicitly available, a multi-head self-attention network is utilized to capture inter-candidate dependencies, in which each candidate attends to all others and the attention weights are learned in a data-driven manner. Unlike the fusion operator $\mathcal{F}_{\mathrm{att}}$ in Section II-B, which attends across agents at each spatial cell to aggregate perceptual feature maps, the attention here operates at the decision level, attending across candidate collaborators to inform the partner-selection action. This makes attention particularly well-suited to the task under consideration. The candidate set is time-varying in size as agents enter and leave the ego-CAV's range, which the permutation-invariant attention handles without retraining. The value of a candidate is relative rather than absolute, depending on its viewpoint complementarity or redundancy with others, which attention explicitly models. Node features $\mathbf{X} \in \mathbb{R}^{C\times(D_{\mathrm{feat}}+D_{\mathrm{grid}})}$ are embedded and refined through L attention blocks [38], described by

$$h_j^{(0)} = \sigma_{\mathrm{r}}\left(\mathrm{LN}\left(W_{\mathrm{in}}x_j + b_{\mathrm{in}}\right)\right), \tag{19}$$

$$\mathbf{H}^{(i)} = \mathrm{LN}\left(\mathbf{H}^{(i-1)} + \mathrm{MHA}\left(\mathbf{H}^{(i-1)}\right)\right), \tag{20}$$

$$\mathbf{H}^{(i)} = \mathrm{LN}\left(\mathbf{H}^{(i)} + \mathrm{FFN}\left(\mathbf{H}^{(i)}\right)\right), \tag{21}$$

where $\mathrm{LN}(\cdot)$ denotes layer normalization, and $\mathbf{H}^{(L)}$ is the final set of node embeddings. A global representation is obtained by mean and max pooling, given as $h_{\mathrm{mean}} = \frac{1}{C}\sum_{j=1}^{C} h_j^{(L)}$ and $h_{\mathrm{max}} = \max_j h_j^{(L)}$, forming $h_{\mathrm{critic}} = [\,h_{\mathrm{mean}} \mid h_{\mathrm{max}}\,]$, while the actor head operates on the flattened node embeddings $h_{\mathrm{actor}} = \mathrm{vec}(\mathbf{H}^{(L)})$.

For the action parameterization, the actor outputs logits for C independent Bernoulli variables, and the partner-selection vector $\mathbf{x} = (x_1, \ldots, x_C)$ follows

$$\pi_{\theta_l}(\mathbf{x} \mid s_t^l) = \prod_{j=1}^{C} p_j^{x_j}(1-p_j)^{1-x_j}, \tag{22}$$

where $p_j = \sigma(f_\pi(h_{\mathrm{actor}}))_j$, with $f_\pi(\cdot)$ denoting the policy head network. The critic head estimates the scalar state value via $V_{\phi_l}(s_t^l) = f_V(h_{\mathrm{critic}})$.

2) Training With Proximal Policy Optimization: PPO is adopted to train both the high-level and low-level agents under an actor-critic paradigm, thereby stabilizing learning by constraining the magnitude of policy updates. For each agent, the value function is fitted, and the policy is improved in alternating steps.

The temporal-difference (TD) residual at time t is given as $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$, where $\gamma \in (0,1]$ denotes the discount factor. The generalized advantage estimator (GAE) is described by an exponentially weighted sum of TD residuals, expressed as $\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}$, where $\lambda \in [0,1]$ controls the bias-variance trade-off. In practice, truncated rollouts are used, and the target value is set to $V_t^{\mathrm{targ}} = \hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} + V_\phi(s_t)$. The value-function loss is then defined by the mean-squared error $L^{\mathrm{VF}}(\phi) = \mathbb{E}_t[(V_\phi(s_t) - V_t^{\mathrm{targ}})^2]$, and the parameters ϕ are updated by minimizing $L^{\mathrm{VF}}(\phi)$.

TABLE I
SIMULATION ENVIRONMENT PARAMETER SETTINGS

Description	Value
Number of CAVs, UAVs, RSUs	5/2/1
Transmission power (C/U/R)	18–23 / 23–28 / 28–33 dBm
Total bandwidth (C/U/R)	40–100 / 40–80 / 80–160 MHz
Computing capability (C/U/R)	6–15 / 6–12 / 30–80 TFLOPS
Time slot duration	200 ms
Switching Latency	10 ms
Carrier Frequency	5.9 GHz
Noise Density	1×10^{-20} W/Hz
Data size	0.976×10^6–72×10^6 bits

TABLE II
HYPERPARAMETER SETTINGS

Hyperparameter	Value	Hyperparameter	Value
Learning Rate	5×10^{-5}	GAE Parameter	0.95
Clip Epsilon	0.1	Entropy Coefficient	0.01
VF Coefficient	0.5	VF Clip Param	5.0
Train Batch Size	4096	SGD Mini-batch Size	128
Epochs per Update	10	Discount Factor	0.99
Rollout Fragment Length	256	AT Multiplier	2
AT Hidden Dim	128	AT Layers	2
Attention Heads	4	AT Dropout	0.05

For policy improvement, the probability ratio between the new policy π_θ and the behavior policy $\pi_{\theta_{\text{old}}}$ is defined as $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$. The PPO clipped surrogate objective is described by

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\Big(r_t(\theta)\hat{A}_t, \text{clip}\big(r_t(\theta), 1-\epsilon, 1+\epsilon\big)\,\hat{A}_t\Big)\Big], \tag{23}$$

where $\hat{A}_t \equiv \hat{A}_t^{\text{GAE}(\gamma,\lambda)}$ and $\epsilon > 0$ specifies the clipping range [36]. The overall loss to be minimized combines the policy surrogate, the value-function loss, and an entropy bonus to encourage exploration, given as $L(\theta,\phi) = \mathbb{E}_t[-L^{\text{CLIP}}(\theta) + c_1 L^{\text{VF}}(\phi) - c_2\,\mathcal{H}(\pi_\theta(\cdot \mid s_t))]$, where $c_1, c_2 \geq 0$ are weighting coefficients and $\mathcal{H}(\cdot)$ denotes the policy entropy. Trajectories are collected at both temporal scales, and the parameters of the two agents are updated jointly using shared rollouts.

IV. SIMULATION RESULTS

In this section, a comprehensive evaluation of the proposed AHRL algorithm is conducted in AGCP scenarios with dynamic networks.

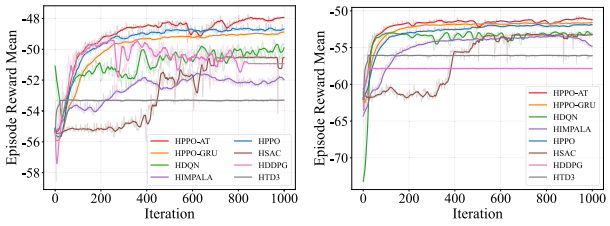
A. Simulation Setup

1) Simulation Environment and Parameters: The simulation scenario is implemented using the CARLA and AirSim simulators to emulate urban expressway conditions, with underlying data sourced from the AirV2X dataset [9]. The setup models a heterogeneous network comprising two UAVs, one RSU, and five CAVs, including one designated ego-CAV. To accurately quantify computational overhead, the total computational load (FLOPs) is obtained by precisely counting constituent atomic operations (additions, multiplications, comparisons, and indexing) across the entire architecture. For the perception configuration, we employ PointPillar as the backbone network for 3D object detection. The detection range is set to $[-140.8, 140.8]$ m along the x-axis, $[-40, 40]$ m along the y-axis, and $[-3, 1]$ m along the z-axis. The voxel size is configured as $(0.4, 0.4, 4)$ m, ensuring a balance between feature resolution and processing efficiency.

To reflect realistic deployment diversity, the computing capabilities of CAVs, UAVs, and the RSU are set from 6 TFLOPS to 15 TFLOPS, from 6 TFLOPS to 12 TFLOPS, and from 30 TFLOPS to 80 TFLOPS, respectively. Communication operates in the 5.9 GHz V2X band, with type-specific bandwidth pools assigned to different cooperation partners: 40 MHz to 100 MHz for CAVs, 40 MHz to 80 MHz for UAVs, and 80 MHz to 160 MHz for the RSU. These bandwidth resources are adaptively allocated among active links according to the current network load. Transmission power is differentiated by the type, set to 18 dBm to 23 dBm for CAVs, 23 dBm to 28 dBm for UAVs, and 28 dBm to 33 dBm for the RSU, while the noise spectral density remains fixed at 1×10^{-20} W/Hz. The transmission data size is set between 0.976×10^6 bits and 72×10^6 bits, depending on the selected threshold value. The time duration of one time slot is set to 200 ms. A switching latency of 10 ms is introduced to capture coordination overhead during partner reconfiguration. The complete simulation parameters are summarized in Table I.

The reinforcement learning (RL) framework adopts a hierarchical PPO architecture, in which a high-level policy determines the cooperation update interval, and a low-level attention-enhanced policy selects optimal cooperation partners. To ensure stable convergence under dynamic vehicular environments, all hyperparameters are tuned through ablation studies. The learning rate is set to 5×10^{-5} with a discount factor of 0.99 and a GAE parameter of 0.95 to balance temporal credit assignment and sample efficiency. The PPO clipping threshold is 0.1, the entropy coefficient is 0.01, and the value-function regularization uses a coefficient of 0.5 with a clipping value of 5.0. These settings jointly stabilize policy updates while maintaining adequate exploration. Each training iteration uses a batch size of 4096, subdivided into 128 mini-batches over 10 stochastic gradient descent (SGD) epochs. The attention-enhanced graph module employs a hidden dimension of 128, two layers, four attention heads, and a dropout rate of 0.05 to enhance the robustness of feature aggregation. The detailed hyperparameter configurations are summarized in Table II.

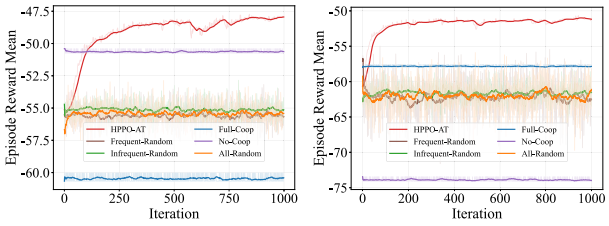
2) Evaluation Metrics: Three metrics are utilized to evaluate the proposed AHRL algorithm and the baseline methods, capturing the trade-off between perception accuracy and system cost at each simulation step. **Tchebycheff Distance** acts as the primary optimization objective by scalarizing the trade-off between perception accuracy and end-to-end latency. It is defined as the maximum of the two dynamically normalized distances, with its negative value serving as the ego-CAV's reward. Additionally, **Average Precision** measures the ego-CAV's perception accuracy at an IoU threshold of 0.5. Finally, **End-to-End Latency** captures the total system delay, including processing-transmitting latency and switching latency.



(a) Performance comparison under high compression threshold.

(b) Performance comparison under medium compression threshold.

Fig. 4. Training performance comparison of HPPO-AT with reinforcement learning baselines.



(a) Performance comparison under high compression threshold.

(b) Performance comparison under medium compression threshold.

Fig. 5. Training performance comparison of HPPO-AT with non-learning baselines.

B. Training and Performance Comparison

To validate the effectiveness of the proposed framework, several hierarchical RL baselines are compared, including hierarchical proximal policy optimization (HPPO) [36], [39], HPPO with a gated recurrent unit (HPPO-GRU) [39], hierarchical deep Q-network (HDQN) [35], hierarchical importance weighted actor-learner architecture (HIMPALA) [40], and three off-policy hierarchical actor-critic baselines, namely hierarchical soft actor-critic (HSAC) [41], hierarchical deep deterministic policy gradient (HDDPG) [42], and hierarchical twin delayed DDPG (HTD3) [43]. To distinguish it from other baselines, the proposed AHRL algorithm is referred to as HPPO-AT in this section. Specifically, **HPPO** serves as an ablation of the attention mechanism where both high- and low-level policies are standard two-layer multilayer perceptrons (MLPs). **HPPO-GRU** is a variant using a GRU-based high-level policy to capture temporal dependencies, while the low-level policy remains an MLP. **HDQN** represents a value-based counterpart replacing PPO with deep Q-network (DQN), where the high-level policy is an MLP and the low-level policy adopts a Factorized DQN to handle multi-discrete collaborator actions. **HIMPALA** is a hierarchical version of IMPALA with V-trace, using feed-forward MLPs for both policy layers. **HSAC**, **HDDPG**, and **HTD3** are off-policy actor-critic variants that adopt the same hierarchical structure as HPPO-AT; since both the persistence-interval and collaborator-selection actions are intrinsically discrete, their actors output relaxed categorical distributions through the Gumbel-Softmax reparameterization, and transitions are reused from a replay buffer for value-function updates.

As shown in Fig. 4(a) and Fig. 4(b), the training convergence of all methods is evaluated under two compression thresholds. A threshold of 0.9818 (98% mask ratio) yields highly compressed feature maps with reduced accuracy but low data payload, whereas a threshold of 0.981 (78% mask ratio) enables richer feature transmission at higher bandwidth cost.

Under the high-compression setting (Fig. 4(a)), HPPO-AT achieves the best performance, converging near -48 with the smallest steady-state fluctuation among all methods. It consistently surpasses HPPO and HPPO-GRU by roughly 1 reward unit and outperforms HDQN by about 2 units. Among the remaining hierarchical baselines, the off-policy actor-critic methods HDDPG and HSAC stabilize around -50.5, HIMPALA converges near -52, and HTD3 remains the lowest at about -53. The advantage of HPPO-AT stems from the attention-enhanced low-level policy, which focuses on partners providing spatially complementary and semantically informative features while down-weighting redundant observations. Meanwhile, the high-level policy adaptively adjusts the persistence interval K to avoid unnecessary reconfigurations and reduce the end-to-end latency $L_{\text{total}}(n)$.

When the compression threshold is relaxed (Fig. 4(b)), all methods exhibit lower episode rewards because the increased data payload leads to higher transmission latency and a larger Tchebycheff distance, despite better perception accuracy. Even so, HPPO-AT remains the top performer, converging around -51 with stable performance, followed by HPPO-GRU and HPPO at about -51.8 and -52, respectively. HDQN and HSAC converge to a comparable level near -53, HIMPALA near -55, HTD3 near -56, and HDDPG performs worst at about -57.8. The off-policy methods again fall behind the PPO-family baselines, confirming that their replay-based updates cannot keep pace with the shifting low-level dynamics induced by the hierarchical structure. By contrast, the attention mechanism of HPPO-AT continues to select the most informative partners under varying bandwidth conditions. At the same time, the high-level policy shortens K during congested intervals to maintain responsiveness. As a result, HPPO-AT consistently achieves the smallest Tchebycheff distance.

In addition to the RL baselines, five non-learning baseline strategies are evaluated to provide a comprehensive performance context. **No-Cooperation (No-Coop)** is a baseline that never selects cooperation partners, fixing the low-level action to an all-zero vector to measure the ego-CAV's performance in isolation, whereas **Full-Cooperation (Full-Coop)** is a greedy baseline that always selects all available cooperation partners, fixing the low-level action to an all-one vector. **Frequent-Random** fixes the high-level decision to the fastest update interval ($k = 1$) and selects a random set of cooperation partners; conversely, **Infrequent-Random** fixes the high-level decision to the slowest update interval ($k = 5$) while also selecting random partners. Finally, **All-Random** is a pure-chance baseline in which both the high-level policy and the low-level policy are sampled uniformly at random.

Fig. 5(a) and Fig. 5(b) compare HPPO-AT with the above non-learning strategies to evaluate the necessity of adaptive coordination between AP and end-to-end latency. Under the high-compression setting, HPPO-AT achieves the highest reward and converges near -48, while all non-learning strategies perform significantly worse. Full-Coop, despite complete feature exchange, suffers from severe latency accumulation and remains near -60. No-Coop avoids communication delay

but forfeits cooperative gains, stabilizing around -51. Among the low-level random baselines, Infrequent-Random performs best at approximately -55, All-Random is intermediate near -55.5, and Frequent-Random is the lowest at -55.5 to -56. This ranking suggests that infrequent high-level updates are advantageous in this environment, as they reduce unnecessary reconfigurations and maintain temporal consistency for the low-level policy.

When the compression threshold is relaxed to the medium level, the rewards of all methods decline, as the larger data payload increases communication delay and enlarges the Tchebycheff distance despite improved perception accuracy. In this setting, Full-Coop achieves higher AP through richer shared features and surpasses No-Coop, yet its accumulated latency still limits its reward, stabilizing near -58. Although the relative ordering of the low-level random strategies remains unchanged, their performance gaps narrow. Infrequent-Random still performs best, while All-Random and Frequent-Random follow more closely, suggesting that with reduced compression and richer exchanged features, sensitivity to the high-level update interval diminishes. HPPO-AT remains the top performer, converging around -51 with a margin of roughly 7 reward units over the best non-learning baseline. Its hierarchical policy balances AP and latency by selecting informative yet cost-efficient partners and adapting the persistence length K to network conditions. These results show that fixed or stochastic strategies cannot accommodate tightly coupled perception-communication-computation dynamics, whereas HPPO-AT maintains robust CP under increasing network load.

TABLE III
COMPREHENSIVE PERFORMANCE OVER A 100-STEP EPISODE UNDER DIFFERENT COMPRESSION THRESHOLDS

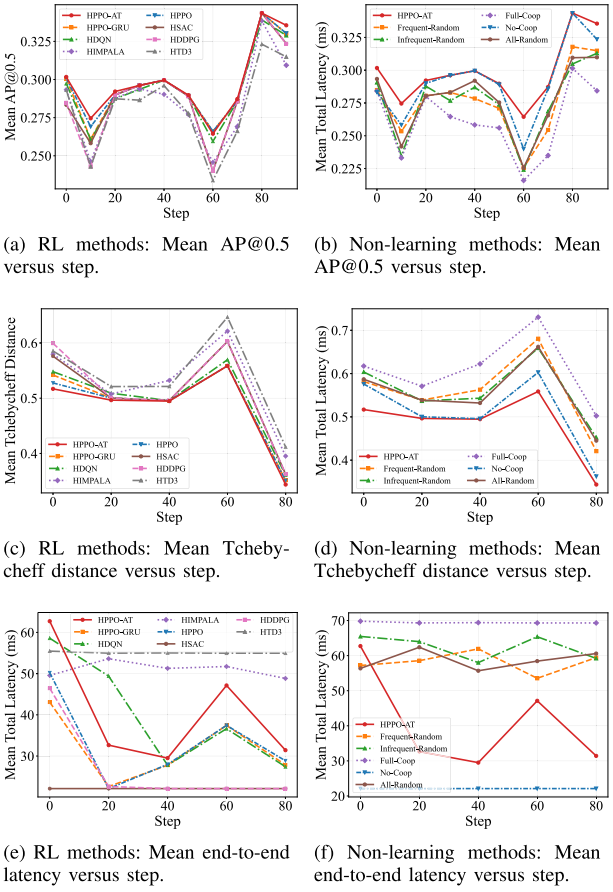
High Compression ($\tau = 0.9818$)			
Method	**AP**	**Latency (ms)**	**Tchebycheff**
HDQN	0.2941	39.96	0.4966
HIMPALA	0.2851	51.00	0.5272
HSAC	0.2909	22.10	0.5075
HDDPG	0.2895	27.07	0.5121
HTD3	0.2822	55.07	0.5373
HPPO-AT (proposed)	**0.2985**	40.68	**0.4821**
HPPO-GRU	0.2962	31.75	0.4895
HPPO	0.2971	33.32	0.4866
Frequent-Random	0.2764	58.13	0.5569
Infrequent-Random	0.2761	62.41	0.5590
Full-Coop	0.2613	69.43	0.6087
No-Coop	0.2909	22.10	0.5075
All-Random	0.2776	58.69	0.5529
Medium Compression ($\tau = 0.981$)			
Method	**AP**	**Latency (ms)**	**Tchebycheff**
HDQN	0.3952	102.38	0.5499
HIMPALA	0.3860	98.47	0.5585
HSAC	0.3925	50.57	0.5343
HDDPG	0.3707	69.75	0.5790
HTD3	0.3795	59.75	0.5609
HPPO-AT (proposed)	**0.4067**	90.09	**0.5249**
HPPO-GRU	0.4038	86.52	0.5322
HPPO	0.4011	94.00	0.5378
Frequent-Random	0.3476	80.41	0.6286
Infrequent-Random	0.3451	91.18	0.6359
Full-Coop	0.3704	112.47	0.5895
No-Coop	0.2914	22.10	0.7409
All-Random	0.3710	88.78	0.5949

C. Comprehensive Performance Evaluation

To enable a comprehensive comparison across all methods, this section presents episode-level results over a full 100-step testing episode. The reported metrics include the average Tchebycheff distance, average AP, and average end-to-end latency.

Table III summarizes the episode-averaged AP, end-to-end latency, and Tchebycheff distance under the two compression thresholds. In the high-compression setting, HPPO-AT achieves the best overall performance, yielding the smallest Tchebycheff distance (0.4821) and the highest AP (0.2985), thus demonstrating the most effective balance between perception accuracy and latency. HPPO-GRU and HPPO follow with competitive accuracy and moderate latency, while HDQN and HIMPALA deliver lower accuracy and larger distances due to weaker policy convergence. The off-policy methods HSAC, HDDPG, and HTD3 tend to converge to policies that over-aggressively prune cooperation partners in order to minimize latency, thereby sacrificing the cooperative perception gain. This tendency is most evident for HSAC, whose converged policy selects almost no collaborators, so that its AP (0.2909), latency (22.10 ms), and Tchebycheff distance (0.5075) nearly coincide with those of the non-cooperative No-Coop baseline. HDDPG exhibits a similar degeneration with low latency but a larger Tchebycheff distance, while HTD3 yields the lowest AP among all RL methods. Among the non-learning strategies, No-Coop achieves the minimum latency (22.10 ms) but suffers from limited perception accuracy. In contrast, Full-Coop incurs the largest communication delay (69.43 ms) while producing the lowest AP, indicating that unselective full feature exchange is inefficient under tight bandwidth constraints. The low-level random strategies lie between these extremes, offering moderate AP and latency but no meaningful improvement in the overall Tchebycheff distance.

When the compression threshold is relaxed to 0.981, all methods exhibit larger Tchebycheff distances, as the increased transmission load raises end-to-end latency despite higher perception accuracy. HPPO-AT remains the best-performing method, achieving both the highest AP (0.4067) and the smallest distance (0.5249), demonstrating strong robustness across compression regimes. HPPO-GRU and HPPO also perform competitively but fall short of HPPO-AT due to their lower AP. Although HSAC attains relatively low latency, HDDPG and HTD3 do not, and all three suffer from reduced AP, leading to larger Tchebycheff distances, with HDDPG reaching 0.5790. This confirms that off-policy actor-critic methods, which rely on stale replay-buffer transitions under the non-stationary hierarchical dynamics, struggle to learn an effective cooperation policy. In contrast, non-learning strategies degrade more noticeably, with No-Coop showing the largest distance (0.7409). Moreover, because the Tchebycheff weights are set to $\lambda_{AP} : \lambda_L = 5 : 1$, the objective is dominated by perception accuracy, making accuracy the primary performance bottleneck once latency optimization saturates.



(a) RL methods: Mean AP@0.5 versus step.

(b) Non-learning methods: Mean AP@0.5 versus step.

(c) RL methods: Mean Tchebycheff distance versus step.

(d) Non-learning methods: Mean Tchebycheff distance versus step.

(e) RL methods: Mean end-to-end latency versus step.

(f) Non-learning methods: Mean end-to-end latency versus step.

Fig. 6. Step-wise performance trends under the high-compression threshold.

Fig. 6 presents the step-wise performance under the high-compression setting, with each marker representing a 10-step average. Within the RL group (Fig. 6(a), (c), and (e)), HPPO-AT consistently attains the highest or joint-highest mean AP across nearly all sampled steps, with HPPO following most closely. The three off-policy baselines, HSAC, HDDPG, and HTD3, remain visibly lower, and the gap widens at the perception troughs around step 10 and step 60, where HTD3 drops to the lowest AP of about 0.235. As shown in Fig. 6(c), HPPO-AT maintains the smallest Tchebycheff distance throughout the episode, whereas the off-policy methods incur the largest distances at the peaks around step 0 and step 60, with HTD3 reaching roughly 0.69 at step 60. These trends confirm the superior perception-latency trade-off of HPPO-AT among the RL methods. The end-to-end latency in Fig. 6(e) reveals two distinct off-policy behaviors. HSAC and HDDPG converge to a near-constant latency of about 22 ms, almost a flat line at the lower bound, indicating that their policies select few or no collaborators and thereby forfeit the cooperative perception gain, consistent with their lower AP in Fig. 6(a). In contrast, HTD3 remains at a persistently high latency of around 55 ms with little variation, yet without any corresponding accuracy advantage, yielding the least efficient trade-off among the RL methods. HPPO-AT instead adapts its latency to the network condition, ranging from about 33 ms in favorable intervals to roughly 47–63 ms when richer cooperation is beneficial, while HPPO-GRU and HPPO follow similar adaptive patterns. This adaptivity, rather than a fixed aggressive or conservative cooperation level, is what enables HPPO-AT to sustain both high AP and a low Tchebycheff distance.

For the non-learning strategies shown in Fig. 6(b), (d), and (f), the perception curves exhibit noticeably stronger fluctuations. Across most sampled steps, HPPO-AT and No-Coop achieve the highest AP among this group, whereas Full-Coop consistently yields the lowest AP over a substantial portion of the episode. All-Random and the two random-interval schemes fall between these extremes but display pronounced variability. In terms of latency, No-Coop maintains the smallest and nearly constant end-to-end delay of approximately 22 ms, while Full-Coop incurs the largest latency throughout, and All-Random generally remains above HPPO-AT across the episode. Consequently, all non-learning strategies result in larger Tchebycheff distances than HPPO-AT. Overall, HPPO-AT sustains the most stable and efficient perception-latency trade-off, maintaining strong coordination even under severe feature compression.

All algorithms exhibit correlated fluctuations in AP, Tchebycheff distance, and end-to-end latency across steps. These variations arise from the dynamic properties of the AGCP environment rather than from learning instability. Each 10-step interval corresponds to a different network condition in which vehicle density, LOS visibility, and link quality evolve over time. Occlusions or dense traffic temporarily reduce perception coverage and thus lower AP, while open scenes improve AP but marginally increase latency due to heavier feature transmission. The consistency of these trends across all methods confirms that the environment imposes shared dynamics. Under these varying conditions, HPPO-AT remains consistently superior, demonstrating strong robustness to network fluctuations.

TABLE IV

SUMMARY OF SENSITIVITY ANALYSIS: AVERAGE EPISODE AP AND END-TO-END LATENCY UNDER DIFFERENT CONFIGURATIONS

Configuration	Mean AP@0.5	Mean Latency (ms)
Objective Weighting $\lambda_{AP} : \lambda_L$		
Weight $5:1$	**0.4067**	90.09
Weight $3:1$	0.4029	92.04
Weight $1:1$	0.3593	68.80
Weight $1:3$	0.2914	**22.10**
Weight $1:5$	0.2914	**22.10**
Collaborator Types		
CAV-CAV	0.2989	40.68
CAV-RSU	0.3033	**39.68**
CAV-UAV	0.4038	91.47
CAV-RSU-UAV	**0.4047**	91.01
Compression Threshold τ		
Low ($\tau = 0.01$)	**0.6710**	235.32
Medium ($\tau = 0.981$)	0.4067	90.09
High ($\tau = 0.9818$)	0.2985	**40.68**

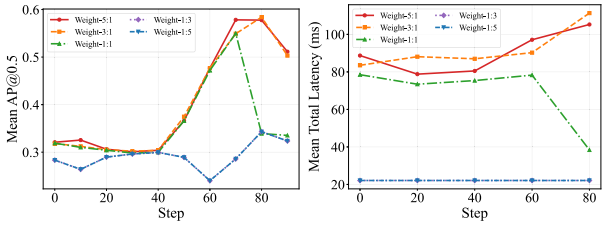
D. Sensitivity Analysis

Table IV provides a concise overview of the sensitivity experiments. As the perception weight λ_{AP} increases relative to λ_L, the ego-CAV tends to maintain cooperation and transmit richer features, thereby achieving higher AP at the expense of

increased end-to-end latency. When $\lambda_{AP} : \lambda_L = 5 : 1$, the system attains the highest accuracy of 0.4067 with a moderate end-to-end latency of 90.09 ms. In contrast, when the end-to-end latency weight dominates, the ego-CAV learns to suppress almost all cooperative actions in order to minimize latency. This behavior leads to a non-cooperative regime, as evidenced by the identical outcomes under $\lambda_{AP} : \lambda_L = 1 : 3$ and $1 : 5$, both of which yield the minimal end-to-end latency of 22.10 ms but the lowest AP of 0.2914. Once cooperation is abandoned, further increasing λ_L no longer changes the performance, which explains the observed saturation.

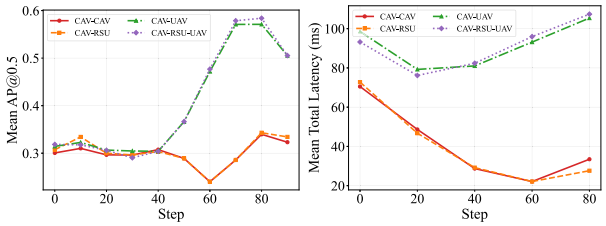
Regarding collaborator composition, the inclusion of aerial cooperation partners provides a noticeable improvement in both perception gain and end-to-end latency. The CAV-UAV and CAV-RSU-UAV configurations achieve AP values around 0.40, corresponding to more than 30% improvement compared with CAV-only cooperation, albeit with end-to-end latency increasing to approximately 91 ms. The CAV-RSU configuration provides a modest perception enhancement (0.3033) with the lowest end-to-end latency, highlighting the low-overhead yet spatially limited role of RSUs. UAVs, by contrast, offer substantial perception benefits owing to their elevated viewpoint and wide coverage, but incur higher communication and fusion costs. These observations underline the complementary advantages of heterogeneous cooperation partners: UAVs contribute depth and coverage diversity, whereas RSUs provide a stable ground-based context with minimal end-to-end latency.

Finally, the compression threshold τ controls the trade-off between feature fidelity and communication efficiency. Recall from Section II-B that τ is applied to the normalized spatial confidence map $\mathbf{C}_i$ to generate the binary transmission mask. A feature cell is transmitted only if its confidence exceeds τ, so a larger τ retains fewer cells and yields a higher mask ratio, i.e., more aggressive compression. The three values are selected from the distribution of confidence scores to represent three distinct operating points, where $\tau = 0.01$ corresponds to a low-compression regime in which nearly all feature cells are transmitted, $\tau = 0.981$ to a medium-compression regime with about a 78% mask ratio, and $\tau = 0.9818$ to a high-compression regime with about a 98% mask ratio. Since the confidence scores are densely concentrated within a narrow high-value range, even a small change in τ within this range translates into a markedly different mask ratio, which is why the medium- and high-compression thresholds appear numerically close yet lead to clearly different compression levels. Lower thresholds enable denser feature sharing and thus yield the highest AP (0.6710), but cause significant end-to-end latency (235.32 ms) due to the larger data volume and associated transmission delays. As τ increases, both AP and end-to-end latency consistently decrease, reflecting the inherent perception-communication trade-off. The difference between the medium threshold ($\tau = 0.981$) and the high-compression setting ($\tau = 0.9818$) illustrates the diminishing benefit of more aggressive compression: although end-to-end latency is further reduced, the accompanying drop in AP becomes disproportionately large, suggesting that excessive compression may remove critical perception details.



(a) Perception accuracy under different weights. (b) End-to-end latency under different weights.

Fig. 7. Step-wise performance of HPPO-AT under various weighting configurations $\lambda_{AP} : \lambda_L$.

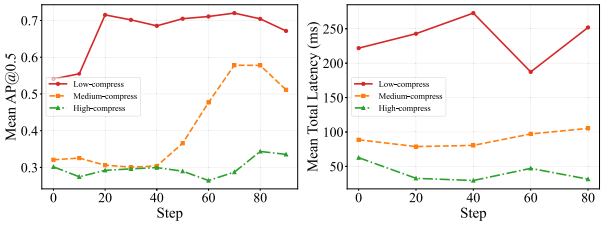


(a) Perception accuracy with different collaborator combinations. (b) End-to-end latency with different collaborator combinations.

Fig. 8. Step-wise performance of HPPO-AT under various collaborator configurations.

1) Impact of Objective Weights λ_{AP} and λ_L: Fig. 7 illustrates how different objective weights shape the step-wise behavior of HPPO-AT over a full episode. When the perception term dominates, that is, for $\lambda_{AP} : \lambda_L = 5{:}1$ and 3:1, all curves start from a similar baseline but the corresponding AP trajectories exhibit a pronounced surge after step 60, with peak values approaching 0.58. This gain is accompanied by a gradual increase in end-to-end latency, which rises from around 80 ms at the beginning to about 110 ms by the end of the episode. For the balanced setting $\lambda_{AP} : \lambda_L = 1{:}1$, the AP curve also improves in the later steps but falls short of the perception-dominant cases, while the end-to-end latency stays in the 70-80 ms range before dropping sharply at the final step, indicating that the ego-CAV eventually adopts a more conservative cooperation pattern. When end-to-end latency is prioritized, namely for $\lambda_{AP} : \lambda_L = 1{:}3$ and 1:5, the end-to-end latency curves remain nearly flat around 22 ms across all steps, showing that the ego-CAV consistently suppresses most cooperative transmissions to reduce communication and processing cost. The corresponding AP trajectories fluctuate only moderately around their initial level. Overall, this experiment demonstrates that HPPO-AT offers clear and interpretable control over the perception-latency trade-off, with behavior evolving coherently over the episode according to the designed objective weights.

2) Impact of Collaborator Types: Fig. 8 illustrates the step-wise evolution of AP and end-to-end latency under different collaborator configurations. The two UAV-assisted settings, CAV-UAV and CAV-RSU-UAV, exhibit nearly identical trajectories. Their AP curves remain close to the ground-only configurations during the first 60 steps and then rise sharply, reaching peak values around 0.58. This pronounced improvement originates from the UAV's elevated viewpoint, which supplies complementary coverage where ground-based



(a) Perception accuracy under different compression thresholds.

(b) End-to-end latency under different compression thresholds.

Fig. 9. Step-wise performance of HPPO-AT under different compression thresholds.

cooperation partners suffer from occlusions. Their end-to-end latency curves follow a similar pattern and increase steadily from approximately 80 ms to more than 110 ms over the episode. The CAV-RSU and CAV-CAV configurations show almost the same behavior throughout the entire episode. Their AP curves overlap closely, including a noticeable drop near step 60 followed by a mild rebound. Across the entire episode, the AP of CAV-RSU remains consistently slightly higher than that of CAV-CAV, indicating the mild but stable perception benefit provided by roadside assistance. The end-to-end latency curves of these two configurations are also tightly aligned, decreasing from about 71 ms toward 22 ms as the episode progresses. Notably, CAV-RSU maintains slightly lower end-to-end latency than CAV-CAV, indicating that RSUs provide stable support with negligible communication latency but do not substantially alter the perception trend. Overall, the results reveal a clear separation between air-ground and ground-only cooperation. Air-ground configurations provide the largest perception gains but incur higher communication and fusion delay. Within the ground-based settings, CAV-RSU consistently outperforms CAV-CAV, offering slightly higher AP and marginally lower end-to-end latency across the entire episode.

3) Impact of Compression Threshold: Fig. 9 illustrates the step-wise evolution of perception accuracy and end-to-end latency under three compression thresholds. The low-compression setting exhibits the highest perception accuracy throughout the episode. Its AP curve undergoes a pronounced increase around step 20, rising from approximately 0.55 to above 0.70, and subsequently remains within the 0.68-0.73 range. This superior AP is accompanied by the largest latency, which increases steadily to nearly 280 ms, drops sharply near step 60, and then rises again toward the end of the episode. Under medium compression, AP stays near 0.30 during the early steps and then experiences a marked growth after step 60, peaking around 0.58 before slightly declining. The corresponding latency remains moderate and shows a gradual upward trend from about 80 ms to slightly above 100 ms. The high-compression setting produces the lowest perception accuracy, with AP maintained in the 0.27-0.33 range and only mild step-wise fluctuations. However, it also achieves the smallest latency, decreasing from roughly 60 ms toward 30 ms, except for a brief rise near step 60. This consistent reduction results from aggressively compressed feature maps that minimize transmission load. Overall, these results indicate that HPPO-AT maintains coherent adaptation behavior under varying compression levels, preserving performance stability while respecting resource constraints.

TABLE V
ABLATION STUDY UNDER DIFFERENT COMPRESSION THRESHOLDS

High Compression ($\tau = 0.9818$)			
Variant	**AP**	**Latency (ms)**	**Tchebycheff**
HPPO-AT (proposed)	**0.2985**	40.68	**0.4821**
HPPO	0.2971	33.32	0.4866
NoFusionAT	0.2882	40.68	0.5169
Fix-Frequency	0.2941	41.87	0.4969
Fix-All	0.2618	69.75	0.6067
Medium Compression ($\tau = 0.981$)			
Variant	**AP**	**Latency (ms)**	**Tchebycheff**
HPPO-AT (proposed)	**0.4067**	90.09	**0.5249**
HPPO	0.4011	94.00	0.5378
NoFusionAT	0.3798	90.09	0.5727
Fix-Frequency	0.4025	89.97	0.5338
Fix-All	0.3707	112.47	0.5889

E. Ablation Study

To justify the key design choices and verify the benefit of jointly optimizing the update frequency and partner selection, an ablation study is conducted with five variants. **HPPO-AT** is the complete proposed method. **HPPO** removes the multi-head attention from the low-level policy, replacing it with a plain MLP, to isolate the contribution of the attention-based policy. **NoFusionAT** retains the attention-based policy but removes the multi-head attention from the perception fusion network, so as to assess the attention-based fusion operator. **Fix-Frequency** disables the high-level policy by fixing the persistence interval to a constant value while keeping the dynamic partner selection, isolating the gain of adaptive frequency control. **Fix-All** further fixes both the update frequency and the partner set, corresponding to a fully static strategy. Note that the dynamic-frequency with fixed-partner case is not included, since once the partner set is fixed, no collaborator switching occurs and the high-level interval becomes inconsequential; this variant therefore degenerates to Fix-All.

Table V reports the ablation results under the two compression thresholds. The complete HPPO-AT consistently attains the highest AP and the smallest Tchebycheff distance in both settings, confirming the effectiveness of the overall design. Removing the attention-based fusion operator (NoFusionAT) causes the most pronounced accuracy drop, with AP falling from 0.2985 to 0.2882 under high compression and from 0.4067 to 0.3798 under medium compression, which shows that the multi-head attention in the fusion network is essential for selectively aggregating informative features from heterogeneous collaborators. Removing the attention from the low-level policy (HPPO) leads to a smaller but still consistent degradation in the Tchebycheff distance, indicating that attention also helps the policy identify spatially complementary and semantically relevant partners.

Regarding the joint optimization, Fix-Frequency, which retains dynamic partner selection but fixes the update interval, achieves a larger Tchebycheff distance than HPPO-AT (0.4969 versus 0.4821, and 0.5338 versus 0.5249), demonstrating that

adaptive frequency control provides an additional gain on top of partner selection. The fully static Fix-All performs the worst by a clear margin, yielding the lowest AP and the largest latency and Tchebycheff distance in both regimes, as it neither prunes redundant collaborators nor adapts the reconfiguration frequency to network conditions. These results jointly verify that the multi-head attention, the attention-based fusion operator, and the joint optimization of update frequency and partner selection each contribute to the performance of the proposed framework.

V. Conclusion

This paper proposed a unified AGCP framework and an attention-enhanced hierarchical reinforcement learning algorithm to address the intricate coupling of perception, communication, and computation in heterogeneous networks. Simulation results confirmed that the proposed approach effectively balanced perception accuracy with end-to-end latency, significantly outperforming existing baselines in dynamic environments. These results validated the effectiveness of the proposed framework in tackling the intrinsic challenges of dynamic decision-making in AGCP systems, including the tight coupling between cooperation update intervals and partner selection across multiple time scales and time-varying environments. Future work will focus on real-world UAV-CAV testbed validation and on extending the framework to large-scale, non-stationary scenarios. In particular, since the current design considers a single ego-CAV, an important direction is to support multiple ego-CAVs deciding concurrently, where conflicting partner-selection policies and shared-bandwidth contention can be resolved via multi-agent reinforcement learning under a centralized-training with decentralized-execution paradigm. Another important direction is to jointly optimize the UAV flight trajectories and their onboard energy consumption together with the cooperative perception decisions, so that UAVs can actively adjust their positions to improve perception coverage while respecting their limited battery budget.

References

[1] C. Sun et al., "Toward ensuring safety for autonomous driving perception: Standardization progress, research advances, and perspectives," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 3286–3304, May 2024.

[2] W. Feng et al., "Radio map-based cognitive satellite-UAV networks towards 6G on-demand coverage," *IEEE Trans. Cognit. Commun. Netw.*, vol. 10, no. 3, pp. 1075–1089, Jun. 2024.

[3] J. Li et al., "Learning for vehicle-to-vehicle cooperative perception under lossy communication," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 4, pp. 2650–2660, Apr. 2023.

[4] 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.*, Springer, 2020, pp. 605–621.

[5] 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. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2583–2589.

[6] 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.* 35, 2022, pp. 4874–4886.

[7] Y. Liu et al., "Joint task scheduling and resource allocation for UAV-assisted air-ground collaborative integrated sensing, computation, and communication," *IEEE Trans. Commun.*, vol. 73, no. 11, pp. 10929–10945, Nov. 2025.

[8] Y. Qin, Z. Zhang, X. Li, W. Huangfu, and H. Zhang, "Deep reinforcement learning based resource allocation and trajectory planning in integrated sensing and communications UAV network," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8158–8169, Nov. 2023.

[9] X. Gao et al., "AirV2X: Unified air-ground vehicle-to-everything collaboration," 2025, *arXiv:2506.19283*.

[10] Y. Zhang et al., "Integrated sensing, communication, and computation in SAGIN: Joint beamforming and resource allocation," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 5, pp. 3128–3143, Oct. 2025.

[11] Q. Guo, F. Tang, and N. Kato, "Routing for space-air-ground integrated network with GAN-powered deep reinforcement learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 2, pp. 914–922, Apr. 2025.

[12] H. Yin et al., "V2VFormer++: Multi-modal vehicle-to-vehicle cooperative perception via global-local transformer," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 2153–2166, Feb. 2024.

[13] Y. Lin, H. Xu, Z. Yin, and G. Tan, "Edge assisted low-latency cooperative BEV perception with progressive state estimation," *IEEE Trans. Mobile Comput.*, vol. 24, no. 4, pp. 3346–3358, Apr. 2025.

[14] Q. Wang, R. Chai, R. Sun, R. Pu, and Q. Chen, "ISAC-enabled multi-UAV cooperative perception and trajectory optimization," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 40982–40995, Dec. 2024.

[15] S. Ren et al., "Interruption-aware cooperative perception for V2X communication-aided autonomous driving," 2023, *arXiv:2304.11821*.

[16] Z. Song, T. Xie, F. Wen, and J. Li, "Wireless communication as an information sensor for multi-agent cooperative perception: A survey," 2025, *arXiv:2505.00747*.

[17] D. Yu, J. You, X. Pei, A. Qu, D. Wang, and S. Jia, "Which2comm: An efficient collaborative perception framework for 3D object detection," 2025, *arXiv:2503.17175*.

[18] M. K. Abdel-Aziz, C. Perfecto, S. Samarakoon, M. Bennis, and W. Saad, "Vehicular cooperative perception through action branching and federated reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 891–903, Feb. 2022.

[19] A. M. Zaki, S. A. Elsayed, K. Elgazzar, and H. S. Hassanein, "Quality-aware task offloading for cooperative perception in vehicular edge computing," 2024, *arXiv:2405.20587*.

[20] R. Zhu et al., "Boosting collaborative vehicular perception on the edge with vehicle-to-vehicle communication," in *Proc. 22nd ACM Conf. Embedded Networked Sensor Syst*. New York, NY, USA: Association for Computing Machinery, Nov. 2024, pp. 141–154.

[21] K. Qu, W. Zhuang, Q. Ye, W. Wu, and X. Shen, "Model-assisted learning for adaptive cooperative perception of connected autonomous vehicles," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8820–8835, Aug. 2024.

[22] Y. Hu, J. Peng, S. Liu, J. Ge, S. Liu, and S. Chen, "Communication-efficient collaborative perception via information filling with codebook," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2024, pp. 15481–15490.

[23] A. Sarlak, R. Amin, and A. Razi, "Extended visibility of autonomous vehicles via optimized cooperative perception under imperfect communication," 2025, *arXiv:2503.18192*.

[24] S. Wei et al., "Asynchrony-robust collaborative perception via bird's eye view flow," 2023, *arXiv:2309.16940*.

[25] J. Wang and T. Nordström, "Latency robust cooperative perception using asynchronous feature fusion," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Tucson, AZ, USA, Feb. 2025, pp. 1–10.

[26] Y. Liu et al., "Multi-modal trajectory planning for emergency-oriented air-ground collaborative sensing and communication," *IEEE Trans. Cognit. Commun. Netw.*, vol. 11, no. 5, pp. 3094–3111, Oct. 2025.

[27] J. Hu, S. Luo, J. Lai, and C. Liu, "Enhanced infrastructure-enabled perception system based on edge computing," *IEEE Internet Things J.*, vol. 12, no. 11, pp. 16493–16510, Jun. 2025.

[28] Y. Cui, H. Xu, J. Wu, Y. Sun, and J. Zhao, "Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system," *IEEE Intell. Syst.*, vol. 34, no. 3, pp. 44–51, May 2019.

[29] 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 *Proc. 4th ACM/IEEE Symp. Edge Comput*. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 88–100, doi: 10.1145/3318216.3363300.

[30] 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)*, Jun. 2019, pp. 12697–12705.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[32] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, document TR 38.901, 3GPP, 2025.
[33] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2898–2904.
[34] *Study on Evaluation Methodology of New Vehicle-to-Everything (V2X) Use Cases for LTE and NR*, document TR 37.885, 3GPP, 2018.
[35] T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, and J. B. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2016, pp. 3675–3683.
[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[37] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. Int. Conf. Learn. Represent.*, 2016.
[38] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.
[39] A. C. Li, C. Florensa, I. Clavera, and P. Abbeel, "Sub-policy adaptation for hierarchical reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[40] L. Espeholt et al., "IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 1406–1415.
[41] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.
[42] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016.
[43] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Jul. 2018, pp. 1587–1596.

Haixia Peng (Senior Member, IEEE) received the dual Ph.D. degrees in computer science and electrical and computer engineering from Northeastern University, Shenyang, China, in 2017, and the University of Waterloo, Waterloo, Canada, in 2021. She is currently a Full Professor with the School of Information and Communications Engineering, Xi'an Jiaotong University, China. Her research interests include satellite-terrestrial vehicular networks, multi-access edge computing, resource management, artificial intelligence, and reinforcement learning.

Yixin Fan (Graduate Student Member, IEEE) received the M.S. degree in transportation engineering from Xidian University, Xi'an, China. She is currently pursuing the Ph.D. degree in communication engineering with Xi'an Jiaotong University, Xi'an. Her current research interests include cooperative perception and the integration of sensing, communication, and computing in vehicular networks.

Zhou Su (Senior Member, IEEE) is currently a Professor with Xi'an Jiaotong University, China. His research interests include multimedia communication, wireless communication, network security, and network traffic. He has published numerous technical papers in top journals and leading conferences, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), IEEE/ACM TRANSACTIONS ON NETWORKING (ToN), IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), and IEEE INFOCOM.

Tom H. Luan (Fellow, IEEE) received the B.E. degree in electrical and computer engineering from Xi'an Jiaotong University, China, the M.S. degree in electrical and computer engineering from The Hong Kong University of Science and Technology, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada. From 2013 to 2017, he was a Lecturer in Mobile and Apps with Deakin University, Australia. From 2017 to 2022, he was a Professor with the School of Cyber Engineering, Xidian University, Xi'an, China.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include network resource management, wireless network security, the Internet of Things, 5G and beyond, and vehicular networks. He is a Registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and an Engineering Academy of Japan International Fellow. He received the West Lake Friendship Award from Zhejiang Province in 2023, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) from the Province of Ontario, Canada, in 2003. He serves/served as the General Chair for the 6G Global Conference'23, and ACM Mobihoc'15, Technical Program Committee Chair/Co-Chair for IEEE Globecom'24, '16 and '07, IEEE Infocom'14, and IEEE VTC'10 Fall. He is the Past President of the IEEE ComSoc, the Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*.