Energy Efficient Orchestration in Multiple-Access Vehicular Aerial-Terrestrial 6G Networks

Mohammad Farhoudi[1], Hamidreza Mazandarani[2], Masoud Shokrnezhad[3], Tarik Taleb[2], and Ignacio Lacalle[4]
[1] *Oulu University, Finland*; mohammad.farhoudi@oulu.fi
[2] *Ruhr University Bochum (RUB), Germany*; {hamidreza.mazandarani, tarik.taleb}@rub.de
[3] *ICTFICIAL Oy, Espoo, Finland*; masoud.shokrnezhad@ictficial.com
[4] *Universitat Politècnica de València, Spain*; iglaub@upv.es

***Abstract*—The proliferation of users, devices, and novel vehicular applications–propelled by advancements in autonomous systems and connected technologies–is precipitating an unprecedented surge in novel services. These emerging services require substantial bandwidth allocation, adherence to stringent Quality of Service (QoS) parameters, and energy-efficient implementations, particularly within highly dynamic vehicular environments. The complexity of these requirements necessitates a fundamental paradigm shift in service orchestration methodologies to facilitate seamless and robust service delivery. This paper addresses this challenge by presenting a novel framework for service orchestration in Unmanned Aerial Vehicles (UAV)-assisted 6G aerial-terrestrial networks. The proposed framework synergistically integrates UAV trajectory planning, Multiple-Access Control (MAC), and service placement to facilitate energy-efficient service coverage while maintaining ultra-low latency communication for vehicular user service requests. We first present a non-linear programming model that formulates the optimization problem. Next, to address the problem, we employ a Hierarchical Deep Reinforcement Learning (HDRL) algorithm that dynamically predicts service requests, user mobility, and channel conditions, addressing the challenges of interference, resource scarcity, and mobility in heterogeneous networks. Simulation results demonstrate that the proposed framework outperforms state-of-the-art solutions in request acceptance, energy efficiency, and latency minimization, showcasing its potential to support the high demands of next-generation vehicular networks.**

***Index Terms*—Service orchestration, Service placement, Predictive resource allocation, Hierarchical DRL, Multi time-scale optimization, and 6G aerial-terrestrial networks.**

I. Introduction

The rapid proliferation of connected autonomous vehicles and associated devices, such as on-board units and short-range communication transceivers, is driving unprecedented growth in both technological sophistication and deployment volume. Technological advancements have enabled vehicular User Equipment (UEs) to integrate with diverse vehicular services, including object detection, traffic analysis, and high-precision cartographic updates [1], playing a pivotal role in augmenting vehicular functionality and enhancing it in various aspects like road safety protocols [2]. However, the exponential growth in deployment density has precipitated unprecedented traffic demands [3], a phenomenon amplified by the cumulative effect of each additional vehicle transmitting increasingly data-intensive sensor readings, high-definition video streams, and telemetry information simultaneously [4]. This multifaceted data proliferation renders the provision of continuous service access for UEs a critical technical challenge [5].

Vehicular services are architected through the integration of multiple fundamental functions, each executing a discrete task. For instance, real-time traffic monitoring represents a paradigmatic composed service, synthesizing distinct functional components including vehicle velocity monitoring, safety-critical message dissemination, and traffic density quantification. These constituent functions operate in parallel and synergistically to deliver comprehensive service functionality. Such composition necessitates adherence to stringent Quality of Service (QoS) parameters, with ultra-low End-to-End (E2E) latency emerging as the predominant requirement for real-time communications [6], [7]. These exacting performance criteria present significant implementation challenges, as contemporary service orchestration approaches demonstrate inadequate capability to consistently maintain uninterrupted connectivity while satisfying the requisite E2E latency thresholds [8].

The evolution of networks has enabled advanced solutions tailored to emerging vehicular services, among which the vehicular edge-cloud continuum has emerged as a promising paradigm for real-time vehicular services. By leveraging resource-constrained edge nodes such as Roadside Units (RSUs) as servers to deliver services to users [9], easing computational loads for users and enhancing their experience [10]. One of the primary challenges in effective orchestration within the continuum is optimizing service placement, which involves selecting the most suitable functions for UE requests while jointly allocating computing and networking resources. This approach promotes resource sharing and maintains a deterministic system to ensure requests are met according to their latency requirements [11]. However, meeting stringent QoS requirements during high-velocity vehicular mobility remains challenging [12]. Also, mobility induces spatiotemporal heterogeneity in service demand, creating localized congestion where UE demands exceed edge node capacities. These challenges necessitate innovative strategies that address both the stochastic nature of vehicular traffic patterns and the limitations of conventional terrestrial infrastructure.

Unmanned Aerial Vehicles (UAVs) have significant potential to enhance the edge-cloud continuum's capabilities in ser-

vice delivery. Conventional terrestrial networks, characterized by sparse distribution, often struggle to maintain consistent connections, especially on busy roads and during peak traffic hours. In this context, aerial-terrestrial networks, which are cost-effective and flexible, can be employed to provide prompt responses in demanding environments [13]. UAVs serve as aerial base stations and edge servers to deliver high-bandwidth services to ground-based UEs. Also, UAVs are considered integral components of the upcoming 6G landscape, playing a crucial role in the envisioned ubiquitous connectivity that supports bandwidth-intensive and real-time vehicular applications. However, as UAVs traverse diverse routes and engage with multiple UEs while managing a variety of computing requests, trajectory planning optimization becomes essential to ensure service coverage in UAV-assisted vehicular networks [14], [15]. Furthermore, the variability of time-dependent channel dynamics presents a challenge for maintaining E2E latency in continuous service delivery, resulting from vehicles' high mobility [16]. This necessitates the development of efficient trajectory and resource planning, as well as Multiple-Access Control (MAC) schemes, to mitigate mutual interference in a shared spectrum environment [17], [18].

Extant literature has advanced UAV-assisted vehicular networks; however, these approaches predominantly employ reactive mechanisms, artificially decouple the optimization of resource planning and MAC from trajectory planning, and insufficiently address composed service orchestration—collectively constraining system scalability and adaptability in dynamic environments. To fill in this gap, we propose a novel service orchestration framework for vehicular aerial-terrestrial 6G networks that integrates MAC, UAV trajectory optimization, and composed service placement within a heterogeneous edge-cloud continuum where both RSUs and UAVs function as communication interfaces for UEs. The main contributions and novelties of this paper are outlined as follows:

- A multi-time-scale Mixed Integer Non-Linear Programming (MINLP) formulation to optimize coverage and energy consumption of vehicular composed requests under E2E latency requirements in aerial-terrestrial networks.
- A decomposition of the problem into multi-UAV trajectory planning, MAC, and composed service placement for complexity reduction. To the best of our knowledge, this is the first work to consider these interconnected aspects while managing user interference on shared channels.
- A predictive Hierarchical Deep Reinforcement Learning (HDRL) framework that combines DRL with a Bayesian algorithm to enhance requests and channel quality prediction accuracy. The HDRL framework balances long-term and short-term objectives by facilitating interactions between the trajectory planning, MAC, and service placement modules, considering their distinct dynamics and time scales to achieve a globally optimal solution.

The upcoming sections of the paper are organized as follows. Section II provides a detailed overview of existing works and their limitations. Section III details the system model, presenting the fundamental elements and their interactions. The problem formulation is introduced in Section IV.Section V elaborates on the proposed method, with a detailed explanation of its design and implementation. Subsequently, Section VI presents the simulation settings, analyzes the convergence, and compares the proposed method's performance against baseline approaches. Finally, Section VII encapsulates the study's key insights and future research directions.

II. Related Works

The field of service orchestration in the vehicular edge-cloud continuum, vital for enabling next-generation vehicular applications, has witnessed advancements in recent years. Research in this domain addresses diverse dimensions such as the transition from single-UAV to multi-UAV scenarios, the evolution from heuristic approaches to adaptive and learning algorithms, and the shift from isolated challenges to complex, integrated problems, including joint trajectory planning and service provisioning [19]. Table I provides a summary of the literature, offering a comparative analysis of existing works.

Given UAV mobility, the literature has focused on trajectory planning [20] or assuming deterministic, predefined mobility patterns [21]. There is increasing attention to trajectory planning solutions that leverage their agility for rapid deployment, reliable Line-of-Sight (LoS) connectivity, and the flexibility to adapt their coverage areas [22]. In this regard, Santos *et al.* [23] proposed deploying multiple UAVs in underserved regions to ensure low-latency service delivery for mobile users. Similarly, Wei *et al.* [20] tackled UAV trajectory planning while accounting for physical and environmental obstacles.

Advancements in UAV mobility have spurred research on integrating joint trajectory planning and channel selection to optimize aerial-terrestrial networks with limited interfaces. As evidenced by Nabi *et al.* [24], which highlighted key challenges in aerial edge computing, including real-time adaptability and connectivity management for reliable communication. Due to this, some studies addressed these challenges by incorporating non-orthogonal multiple access in edge-cloud environments [25]. Pervez *et al.* [26] proposed an iterative algorithm for user association, channel power allocation, and segment-based UAV trajectories in integrated aerial-terrestrial networks for smart vehicular services. Qin *et al.* [27] introduced a cluster-based air-ground integrated network with UAVs for access and high-altitude platforms for backhaul, optimizing UAV trajectories and subchannel selection to enhance energy and spectrum efficiency. Further, other works focused on joint scheduling and channel selection in the edge-cloud environment, accounting for dynamic channel variations to minimize latency and energy consumption [28]. Huang *et al.* [29] addressed the integration of satellite communications and aerial platforms through a DRL-based approach, optimizing both channel selection and trajectory planning. Qi *et al.* [30] extended this line of research by proposing an energy-efficient framework combining content placement, spectrum allocation, co-channel pairing, and power control, improving channel selection and system performance.

Some studies have been carried out in the literature to study service provisioning and trajectory planning together for non-terrestrial networks. He *et al.* [31] used an online DRL

Table I
SUMMARY OF EXISTING SCHEMES AND COMPARISON BASED ON CONSIDERATIONS IN TRAJECTORY PLANNING, MAC, AND SERVICE PLACEMENT.

Reference	Objective Function	Algorithm	Multi-UAV	UEs Mobility	Trajectory Planning	Multiple Access	Service Placement
He *et al.* [31]	Maximize acceptance & enhance energy	Actor-Critic & Q-learning	✓	Random	✓	–	✓
SAC-TORA [32]	Minimize energy consumption	Soft Actor-Critic	✓	–	✓	–	✓
Gupta *et al.* [33]	Max-min aggregate throughput	SCA optimization	✓	–	✓	–	–
Qin *et al.* [34]	Minimize energy consumption	PMADDPG	✓	–	✓	✓	✓
Li *et al.* [35]	Provisioning rate	GNN-DRL	–	–	✓	–	–
Muto *et al.* [15]	Minimize computational costs	Multi-agent DRL	∼	–	✓	–	✓
Dutriez *et al.* [36]	Maximize energy efficiency	Deep Q-Network	–	–	–	∼	–
FL-SNTD3 [37]	Provisioning rate & latency	Deep federated learning	✓	–	✓	–	–
DM-SAC-H [29]	Minimize energy consumption & latency	Soft Actor-Critic	✓	–	✓	–	✓
HaDDQN [30]	Energy efficiency	HaDDQN	✓	Random	∼	✓	✓
Wei *et al.* [20]	Service execution success rate	Deep Q-Network	–	–	✓	–	✓
SCOFT [38]	Minimize energy consumption	Hierarchical DRL (HDRL)	✓	Random	✓	–	∼
Proposed Solution	Maximize coverage & optimize energy	Hierarchical DRL (HDRL)	✓	Predictive	✓	✓	✓

SCA: Successive Convex Approximation; PMADDPG: Probabilistic Multi-Agent Deep Deterministic Policy Gradients.

approach to investigate the interplay between continuous UAV trajectory planning and discrete service deployment actions. Le *et al.* [35] addressed optimization challenges in UAV-assisted edge networks, focusing on UAV trajectory and service provisioning in dynamic environments, using Graph Neural Networks (GNN) to optimize UAV speed, heading, and service deployment. Ning *et al.* [15] developed a framework for UAV trajectory design that considers users' computational tasks and probabilistic service preferences. By facilitating decentralized trajectory optimization, they tried to minimize computational costs while maximizing service efficiency. Additionally, Li *et al.* [32] explored a multi-UAV-enabled orchestration scheme for heterogeneous services, utilizing collaborative capabilities to minimize overall energy consumption in the system.

Despite innovative strategies, existing research on service orchestration in UAV-assisted networks still exhibits noteworthy limitations. Approaches assuming static users often fail to provide timely responses in highly dynamic environments [20], [33], [34]. Several studies overlook the challenges of orchestrating composed services with diverse functions and shared resources across multiple users, which constrains scalability and flexibility [33], [35]–[37]. Current solutions are predominantly reactive, adapting UAV trajectories and deployment only after receiving feedback, underscoring the need for proactive orchestration of future conditions like user mobility and demand. For instance, SCOFT [38] optimized UAV trajectory and service placement for energy efficiency; however, its decisions remain reactive and are based solely on instantaneous system states, while we explicitly integrate predictive models of user mobility and service demand. While some works consider communication links between network elements [30], [34], the complexities of dynamically sharing spectrum or considering channel qualities for UAV-assisted service orchestration remain unexplored. For example, HaDDQN does not predict channel dynamics and instead models them as stochastic processes, resulting in a reactive orchestration strategy that cannot anticipate future conditions. Likewise, Qin *et al.* [27] optimized decisions at a single control layer using only the current network state, without explicitly forecasting future demand. Recent advances in wireless techniques [39], [40] improve spectral efficiency but are limited to communication-layer offloading and do not jointly address trajectory planning and composed service orchestration Finally, the coupled optimization of trajectory planning, MAC, and service placement, each significantly influencing the others, has yet to be holistically addressed, which is essential for next-generation vehicular 6G service orchestration.

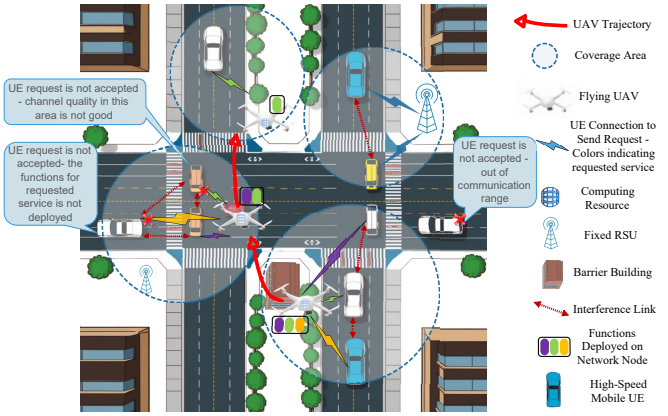

Figure 1. System Model: Supporting UEs through RSUs and UAVs with service coverage, quality channel access, and deployed function availability.

III. SYSTEM MODEL

This section describes the system's detailed structures: network architecture, services, and interactions between the UEs, RSUs, and UAVs across the network, as shown in Fig. 1.

A. Vehicular Network Architecture

In this paper, a tiered aerial-terrestrial network is investigated, denoted by $\mathcal{G}(\mathcal{N}, \mathcal{L}, \mathcal{P})$, which encompasses the coexistence of vehicular networks and the edge-cloud continuum. The network integrates $\mathbb{N}$ computing nodes, such as core nodes, RSUs, and UAVs, each equipped with networking capabilities. They connected through links to deliver services over designated areas during uniform time frames, indexed by t. Nodes close to UEs provide limited computation capabilities at a high cost, whereas core nodes possess significant computing power with lower resource expenses [41]. Each node n is characterized by its processing capability $\hat{\mathcal{C}}_n$ and its energy budget $\overline{\mathcal{E}}_n$. Wired and wireless links connect the different network elements, the set of which is denoted by

$\mathbb{L}^t \subset \{l : (n, n') | n, n' \in \mathbb{N}\}$[1]. Each link l is associated with a bandwidth capacity $\widehat{\mathcal{L}}_l$ and transmission energy consumption $\overline{\xi}_l$. Packets associated with request r that traverse link l experience latencies at time frame t determined using a function denoted as $\mathcal{D}^t_{r,l}$, which is deterministically computed as a function of the current network state, link length, and link load. Based on the links available at each time frame, a set of paths, $\mathbb{P}^t$, is available for packet transmissions. Nodes forming a particular path p are represented as $\mathbb{N}_p$, where $\mathcal{H}^t_p$ and $\mathcal{T}^t_p$ are head and tail nodes respectively, with $\mathcal{J}^t_{p,l}$ denoting the inclusion of link l in path p in time fame t.

B. Services

Services are characterized by outlining essential aspects, including their functions, data model, and data graph [42]. Each composed service $s \in \mathbb{S}=\{1, 2, ..., \mathcal{S}\}$ is segmented in functions, denoted as $\mathbb{F}_s=\{1, 2, ..., \mathcal{F}_s\}$, where each function implemented by virtual instances. The data model accounts for the complex interdependencies between these functions, ensuring that service execution starts with the initial function and proceeds with sequential or parallel execution of subsequent functions. The data graph, $\mathcal{G}_s$, outlines the structure of the composed service, including inputs, outputs, preconditions, and results, along with the total service duration time $\overrightarrow{\mathcal{T}_s}$.

C. Vehicular User Equipment

The set of vehicular UEs is defined as $\mathbb{U} = \{1, 2, \ldots, \mathcal{U}\}$. UEs generate requests at various time intervals for vehicle-to-everything communication, enabling them to transmit requests via appropriate wireless communication protocols in heterogeneous networks. The first node a UE connects to is known as the Point of Attachment (PoA), through which UE requests are handled to reach desired services. Requests $r \in \{1, \ldots, \mathcal{R}\}$ are sent by UEs, and u_r identifies who generates the request r. Each request r enters the system at time $\mathcal{T}'_r$, requesting a composed service $\mathcal{S}_r$ over a predefined duration, which means the request should be processed within $\Delta_r = [\mathcal{T}'_r, \mathcal{T}'_r + \overrightarrow{\mathcal{T}_s}]$. In each time frame, the number of active requests may vary due to factors like mobility patterns and bandwidth-saving strategies. Requests come with specific requirements, including network bandwidth $\breve{\mathcal{L}}_r$, each atomic function's minimum capacity $\breve{\mathcal{I}}_{r,f}$, and E2E latency $\breve{\mathcal{D}}_r$ [43]. Successful service delivery entails meeting the capacity and QoS requirements.

D. Network Areas

The network is divided into areas denoted by $\mathbb{A} = \{1, 2, ..., \mathcal{A}\}$ to ensure comprehensive coverage. The dimensions of areas vary based on geographical factors like obstacles. The assumption is adopted for analytical and computational tractability, where UAV movement is represented at the area level rather than a fully continuous flight trajectory, within which UAVs reposition or hover within areas to serve users. This abstraction is known to capture the dominant mobility effects with negligible loss of accuracy at the considered time scale [44]. UEs exhibit dynamic behavior by frequently moving between areas, while being assumed to remain within a single area during each time frame to simplify movement modeling. The area of UEs who send a request u_r at time frame t is called $\mathcal{A}^t_{u_r,a}$. Computing nodes are located in various parts of the network, with core nodes and RSUs placed in fixed areas and UAVs moving in different areas.

The energy consumption of UAVs traveling between the areas a_1 and a_2 is given by $\Lambda(a_1, a_2)$ (1), which captures hovering and propelling energy [45]. The hovering power includes a_1 and a_2 travel duration Δ_{a_1,a_2}, induced power coefficient I, UAV's total weight W_n, air density φ, and rotor disk area υ_r. The movement power accounts for aerodynamic drag, where ς and υ_f represent the drag coefficient and frontal area, and $V_w(t)$ is the UAV's instantaneous velocity.

$$\Lambda(a_1, a_2) = \int_0^{\Delta_{a_1,a_2}} \left(\frac{I \cdot W_n^{3/2}}{\sqrt{2 \cdot \varphi \cdot \upsilon_r}} + \frac{1}{2} \cdot \varsigma \cdot \varphi \cdot \upsilon_f \cdot V_w(t)^3 \right) dt \quad (1)$$

E. Wireless Channel Model

The wireless channel model is defined with a focus on uplink transmissions, wherein UE requests are dynamically scheduled to minimize the collision probability in each time frame through effective channel assignment. Each time frame t is subdivided into $\mathfrak{T}_t$ smaller time slots to enable fine-grained MAC. The set of available channels is denoted by $\mathbb{C}=\{1, 2, ..., \mathcal{C}\}$, and each channel c is associated with a specific energy consumption $\overline{\mathcal{M}}_c$ and channel quality $\breve{\mathcal{Q}}^\tau_{c,a}$ in area a, as energy requirements vary with operating frequency and path loss. To mitigate collisions resulting from simultaneous transmissions over the same channel within different locations, our model accommodates multiple uplink channels and allows channel reuse in neighboring areas. Meanwhile, we assume that the downlink channels used for service responses to UEs are collision-free, ensuring reliable response transmission.

UE and UAV movements lead to time-varying channel conditions, where the absence of a LoS link can degrade E2E transmission quality. To model this, θ^{LoS}_a is defined, which represents the likelihood of establishing a LoS connection in area a, determined by environmental density, UAV altitude, and weather conditions [22], [46]. When a LoS connection exists, transmission quality is assumed to be ideal; otherwise, under Non-LoS (NLoS) conditions occurring with probability $1-\theta^{\text{LoS}}_a$, signal quality is governed by an instantaneous channel gain $H^\tau_{c,a}$ that incorporates path loss, small-scale fading, and weather-dependent absorption Ω_τ, expressed as (2). In this expression, $R^\tau_{c,a}$ and $\chi^\tau_{c,a}$ denote the Rayleigh and lognormal components, D_0 is the reference distance and $d^\tau_{c,a}$ denotes the transmitter-receiver separation on channel c in area a at time slot t. Also, the parameters $(\nu_s, \eta_s(\Omega_\tau))$ correspond to the path loss exponent and weather-dependent shadowing factor under LoS/NLoS propagation, and $\zeta(\Omega_\tau)$ models atmospheric attenuation such as fog or rain. The received signal-to-noise ratio (SNR) for the channel is given by (3), where P_{tx} represents the transmit power, $\mathcal{N}_0$ is the thermal noise spectral density, and Δf is the subcarrier spacing. Following real-world 6G multiple-access modeling [36], a transmission is

[1]Mobility of both UEs and UAVs leads to a dynamically changing network topology, and UEs should stay within the coverage area of an RSU or UAV to maintain a connection; otherwise, links between them become unavailable.

considered successful if $\gamma^{\tau}_{c,a}$ exceeds a predefined quality threshold $\widehat{\mathcal{Q}}$ (determined by QoS requirements). Accordingly, the resulting binary channel-quality indicator $\mathcal{Q}^{\tau}_{c,a}$ equals 1 for successful transmissions (either through LoS or when NLoS SNR is acceptable) and 0 when signal degradation becomes excessive[2], thereby capturing the essential dynamics of UAV communication reliability under variable weather conditions.

$$H^{\tau}_{c,a} = R^{\tau}_{c,a} \cdot 10^{\chi^{\tau}_{c,a} \cdot \eta_s(\Omega_\tau)/10} \cdot (D_0/d^{\tau}_{c,a})^{\nu_s} \cdot 10^{-\zeta(\Omega_\tau)/10} \quad (2)$$

$$\breve{\mathcal{Q}}^{\tau}_{c,a} = \begin{cases} 1, & \text{w.p. } \theta^{\text{LoS}}_a, \\ \mathbb{1}\Big(\gamma^{\tau}_{c,a} = \frac{P_{tx} \cdot H^{\tau}_{c,a}}{\mathcal{N}_0 \cdot \Delta f} \geq \widehat{\mathcal{Q}}\Big), & \text{w.p. } 1 - \theta^{\text{LoS}}_a \end{cases} \quad (3)$$

IV. Problem Formulation

In this section, the formulation of a MINLP optimization problem, termed energy-Aware muLtipLe-access service Orchestration for vehicular Aerial-TErrestrial networks (ALLOCATE) is presented. The problem pertains to optimizing energy-efficient service coverage through the placement of requested service functions on network nodes, the allocation of a specific set of functions to each request based on its service graph, the assignment of a channel and path for each request to facilitate data delivery from its PoA to the corresponding functions, and the subsequent return of data to the originating PoA. Table II shows the notations used in ALLOCATE.

A. Objective Function

The objective function is formulated to maximize request acceptance - ensuring comprehensive service coverage - while minimizing energy consumption (OF). A scaling factor (α) is introduced to balance the trade-off between energy consumption and request acceptance, adjusting the relative importance of the two factors to ensure an optimal solution. It not only aligns with theoretical optimization goals but also reflects the practical constraints and behaviors of 6G aerial-terrestrial networks [46]. The total energy consumption $\overline{\mathcal{W}}$ (C1) accounts for the prioritization of nodes, from edge to cloud, with varying computational capabilities, communication channels, and links, as well as the energy required for UAVs to traverse between candidate areas (incorporates propulsion dynamics derived from the aerodynamic energy model (Eq. (1))). For a request to be satisfied, all required functions during the request's duration should be deployed (C2).

The selection of key components within the system is governed by binary decision variables: $\tilde{\mathcal{X}}^{t}_{r,f}$ indicates whether request r is served by function f, $\tilde{\mathcal{Y}}^{t}_{f,n}$ denotes the hosting node n for the function f, and $\tilde{\mathcal{Z}}^{\tau}_{r,c}$ represents the selection of channel c for request r. Also, $\tilde{\mathcal{S}}^{t}_{n,a}$ specifies the candidate area a for network node n while RSU areas are always fixed, $\tilde{\mathcal{B}}^{t}_{u,n}$ represents the PoA of UE u at time frame t while they are moving, and $\overrightarrow{\mathcal{R}}^{t}_{r,p}$ determines if path p is selected to send request r packets to deployed nodes and receive the response.

[2]Intuitively, $\mathbb{1}(C)$ equals one if the condition C is satisfied.

Table II
List of notations used in the problem formulation.

Notation	Description
$\mathcal{G}(\mathcal{N},\mathcal{L},\mathcal{P})$	Vehicular aerial-terrestrial edge-cloud network
$\mathcal{G}_s$	Service s data graph
$t \in \mathbb{T}$	Time frame (of Total service time)
$\tau \in \mathfrak{T}_t$	Time slot (of time frame t)
$\overrightarrow{\mathcal{T}_s}$	Total time for delivering service s
$\mathcal{T}'_r$	Entry time of UE request r
$\breve{\mathcal{T}}_r$	Minimum required time slots to send request r
$\mathbb{N}/\mathbb{A}$	Set of network nodes / predefined areas
$\mathbb{L}^t/\mathbb{P}^t$	Set of (wireless links / active paths) at time t
$\mathbb{U}/\mathbb{R}/\mathbb{C}$	Set of (active UEs / Requests / uplink channels)
$\widehat{\mathcal{C}}_n/\overline{\mathcal{E}}_n$	(Processing / Energy consumption) of node n
$\widehat{\mathcal{L}}_l/\overline{\xi}_l$	(Bandwidth capacity / Transmission energy) of link l
$\mathcal{H}^t_p/\mathcal{T}^t_p$	Head/Tail node of path p at time frame t
$\mathcal{J}^t_{p,l}$	The inclusion of link l in path p at time frame t
$\mathcal{D}^t_{r,l}$	Latency experienced by request r over link l at time t
$\mathcal{F}_f \in \mathbb{F}_s$	Atomic function f (of functions set)
$u_r/\mathcal{S}_r$	(UE who send / Composed service) of request r
$\mathcal{A}^t_{u,a}$	Indicator for UE u located in area a at time t
$\overline{\mathcal{M}}_c$	Energy consumption for using uplink channel c
$\breve{\mathcal{Q}}^{\tau}_{c,a}$	Quality of uplink channel c in area a at time slot τ
$\breve{\mathcal{L}}_r$	Network bandwidth required for request r
$\breve{\mathcal{I}}_{r,f}$	Minimum capacity required for function f of request r
$\breve{\mathcal{D}}_r$	Latency requirement for request r
$\tilde{\mathcal{X}}^t_{r,f}$	if function f of request r is selected at time frame t
$\tilde{\mathcal{Y}}^t_{f,n}$	if function f is placed on node n at time frame t
$\tilde{\mathcal{Z}}^{\tau}_{r,c}$	if channel c is selected for request r at time slot τ
$\tilde{\mathcal{S}}^t_{n,a}$	if node n is deployed in area a at time frame t
$\tilde{\mathcal{B}}^t_{u,n}$	if UE u is connected (binned) to node n at time frame t
$\overrightarrow{\mathcal{R}}^t_{r,p}$	if path p is selected for request r at time frame t

$$\text{ALLOCATE: } \max \sum_{\mathbb{R}} \Big(\tilde{\mathcal{X}}_r\Big) - \alpha \cdot \overline{\mathcal{W}} \qquad \textit{s.t.} \quad \text{C1 - C12.} \quad \text{(OF)}$$

$$\overline{\mathcal{W}} \triangleq \sum_{\mathbb{F}_s,\mathbf{N},\mathbb{T}} \tilde{\mathcal{Y}}^t_{f,n} \cdot \overline{\mathcal{E}}_n + \sum_{\mathbf{N},\mathbf{A},\mathbb{T}} \Lambda(\tilde{\mathcal{S}}^{t+1}_{n,a_1} - \tilde{\mathcal{S}}^t_{n,a_2}) + \sum_{\mathbb{L}^t,\mathbb{P}^t,\mathbb{R},\Delta_r} \overline{\xi}_l \cdot \mathcal{J}^t_{p,l} \cdot \overrightarrow{\mathcal{R}}^t_{r,p} + \sum_{\mathbb{R},\mathbb{C},\mathbb{T},\mathfrak{T}_t} \tilde{\mathcal{Z}}^{\tau}_{r,c} \cdot \overline{\mathcal{M}}_c \quad \text{(C1)}$$

$$\tilde{\mathcal{X}}_r = \prod_{\mathbb{F}_{s_r},\Delta_r} \tilde{\mathcal{X}}^t_{r,f} \qquad \forall r \in \mathbb{R} \quad \text{(C2)}$$

B. Constraints

Constraints ensure: (1) appropriate channel allocation within network areas (MAC protocol); (2) optimal service deployment on network nodes with efficient packet routing from PoAs to service nodes (service placement); and (3) dynamic UAV adjustment to accommodate active service requests (trajectory planning). All optimization processes simultaneously satisfy capacity limitations and QoS requirements. Notably, the system operates on multi time-scales: time frames (denoted by t), and time slots (denoted by τ) with each time frame comprising $\mathfrak{T}_t$ time slots. All resource allocation tasks, except for channel selection, operate on time frame granularity, while channel selection functions at a higher frequency of time slots.

Channel Selection: Efficient service delivery in a multiple-access environment necessitates avoiding simultaneous transmissions over the same channel in an area to prevent collisions. For each UE, no more than one channel should be selected for transmitting its request (C3). Other UEs within the same

area should be prevented from using the same channel at the same time slot (C4). This is vital due to the limited available channels of sufficient quality ($\breve{\mathcal{Q}}^{\tau}_{c,a}$), derived from the wireless channel model for request transmission Eq. (3). It indicates UAVs' inability to handle multiple channels simultaneously to maintain real-world consistency between link admission and channel quality. Requests are sent through the channels of nodes to which UEs are directly connected, affecting energy consumption $\overline{\mathcal{W}}$ with the selected transmission channel.

$$\sum_{\mathbb{C}} \tilde{\mathcal{Z}}^{\tau}_{r,c} \leq 1 \qquad \forall r, \tau \in \mathbb{R}, \bigcup_{t \in \Delta_r} \mathfrak{T}_t \qquad \text{(C3)}$$

$$\sum_{\mathbb{R}} \tilde{\mathcal{Z}}^{\tau}_{r,c} \cdot \mathcal{A}^{t}_{u_r,a} \leq 1 \qquad \forall c, a, \tau \in \mathbb{C}, \mathbb{A}, \bigcup_{t \in \mathbb{T}} \mathfrak{T}_t \qquad \text{(C4)}$$

Function Placement: When dealing with composed services, it is necessary to consider the deployment of various functions of services. Thus, each function targeted by at least one request should be deployed on an available network node for the duration of the request (C5). Moreover, each request r should be assigned to appropriate functions based on its required service $\mathbb{F}_{s_r}$, only if they are transmitted on a quality channel not used by other UEs during (C6). This constraint, along with C4, assesses whether a UE's transmissions over quality channels meet the minimum required time slots ($\breve{\mathcal{T}}_r$). If they do, the service could be provided; otherwise, the variable $\tilde{\mathcal{X}}^{t}_{r,f}$ is set to zero (request is not accepted).

$$\sum_{\mathbb{N}} \tilde{\mathcal{Y}}^{t}_{f,n} \geq \left(\sum_{\mathbb{R}} \tilde{\mathcal{X}}^{t}_{r,f} \right) / \mathcal{R} \qquad \forall f, t \in \mathbb{F}_s, \mathbb{T} \qquad \text{(C5)}$$

$$\tilde{\mathcal{X}}^{t}_{r,f} \leq \Big(\sum_{\mathbb{C}, \mathbb{A}, \mathfrak{T}_t} \tilde{\mathcal{Z}}^{\tau}_{r,c} \cdot \breve{\mathcal{Q}}^{\tau}_{c,a} \cdot \mathcal{A}^{t}_{u_r,a} \Big) / \breve{\mathcal{T}}_r \qquad \forall r, f, t \in \mathbb{R}, \mathbb{F}_{s_r}, \Delta_r \quad \text{(C6)}$$

Path Selection: For the effective transmission of inquiry traffic from a UE to its designated nodes, where the requested service is deployed, and the subsequent return of the response, feasible E2E routes should be provided. A unique inquiry path $\overrightarrow{\mathcal{R}}^{t}_{r,p}$ is established for each request, originating at the UE's PoA ($\tilde{\mathcal{B}}^{t}_{u,n}$). Considering UE mobility, the response will be directed to the PoA corresponding to the location where the UE will be present when the request duration concludes, thereby addressing its mobility. The requested service's functions are interconnected based on $\mathcal{G}_{s_r}$ to reach their final destination, with the last function sending the response to u_r (C7).

$$\sum_{\substack{p \in \mathbb{P}^t,\ \mathbb{N}_p,\ \mathbb{F}_{s_r} \\ \mathcal{H}^t_p = \sum_{\mathbb{N}} n \cdot \tilde{\mathcal{B}}^{\mathcal{T}'_r}_{u_r,n} \\ \mathcal{T}^t_p = \sum_{\mathbb{N}} n \cdot \tilde{\mathcal{B}}^{\mathcal{T}'_r + \overrightarrow{\mathcal{T}_{sr}}}_{u_r,n}}} \overrightarrow{\mathcal{R}}^{t}_{r,p} \cdot \mathbb{1}\left(\tilde{\mathcal{Y}}^{t}_{f,n} = 1 \right) = 1 \qquad \forall r,\ t \in \mathbb{R},\ \Delta_r \qquad \text{(C7)}$$

Capacity: Given the network's limited capabilities, it is essential to manage nodes' and links' capacities to maintain system stability. So, the total number of requests allocated to any node does not exceed its processing capacity (C8). Additionally, each link's capacity should not be exceeded during request and response transmission (C9).

$$\sum_{\mathbb{R}, \mathbb{F}_{s_r}} \tilde{\mathcal{X}}^{t}_{r,f} \cdot \tilde{\mathcal{Y}}^{t}_{f,n} \cdot \breve{\mathcal{I}}_{r,f} \leq \hat{\mathcal{C}}_n \qquad \forall n, t \in \mathbb{N}, \mathbb{T} \quad \text{(C8)}$$

$$\sum_{\mathbb{R}, \mathbb{P}^t} \mathcal{J}^{t}_{p,l} \cdot \overrightarrow{\mathcal{R}}^{t}_{r,p} \cdot \breve{\mathcal{L}}_r \leq \hat{\mathcal{L}}_l \qquad \forall l, t \in \mathbb{L}^t, \mathbb{T} \quad \text{(C9)}$$

UAV Trajectory Planning: Trajectory planning is essential for covering UEs and meeting latency requirements while minimizing energy consumption. As the objective is to optimize energy consumption and changes in UAV locations affect energy usage in $\overline{\mathcal{W}}$, the optimization problem prompts to limit UAV mobility while ensuring vehicular UE connectivity. Each network node should remain within exactly one area during each time frame (C10) and each UE should be associated with one PoA within its area at a specified time frame (C11).

$$\sum_{\mathbb{A}} \tilde{\mathcal{S}}^{t}_{n,a} = 1 \qquad \forall n, t \in \mathbb{N}, \mathbb{T} \quad \text{(C10)}$$

$$\sum_{\mathbb{N}, \mathbb{A}} \tilde{\mathcal{B}}^{t}_{u_r,n} \cdot \tilde{\mathcal{S}}^{t}_{n,a} \cdot \mathcal{A}^{t}_{u_r,a} \leq 1 \qquad \forall r, t \in \mathbb{R}, \Delta_r \quad \text{(C11)}$$

QoS Requirements: Ensuring timely and consistent service delivery is paramount to meeting UEs' stringent QoS expectations. Constraint (C12) sets a maximum acceptable latency for request handling, verifying that if a request is accepted, its latency requirements are met. This constraint prevents UEs from monopolizing acceptance based solely on low energy consumption, ensuring requests' QoS within the specified requirements. The latency threshold $\breve{\mathcal{D}}_r$ aggregates transmission, propagation, and processing delays from $\mathcal{D}^{t}_{r,l}$ to capture realistic E2E latency. Coupled with the channel-quality indicator $\breve{\mathcal{Q}}^{\tau}_{c,a}$, only transmissions meeting the required SNR threshold are accepted, ensuring compliance with latency and reliability standards in UAV-assisted vehicular networks [47].

$$\sum_{\mathbb{P}^t, \mathbb{L}^t, \Delta_r} \mathcal{J}^{t}_{p,l} \cdot \mathcal{D}^{t}_{r,l} \cdot \overrightarrow{\mathcal{R}}^{t}_{r,p} \leq \breve{\mathcal{D}}_r \qquad \forall r \in \mathbb{R} \quad \text{(C12)}$$

C. Complexity Analysis

The ALLOCATE problem is classified as NP-hard, reducible from the multidimensional knapsack problem presented in [48]. This classification implies a worst-case computational complexity proportional to the solution space size [49]. Determining the optimal solution for a set of requests requires interdependent evaluations: analyzing each UAV node in every area (trajectory planning), assessing each channel in each time slot (channel selection), and considering every node, function, and path (placement). As any allocation for any request in a given time frame impacts and is influenced by allocations made for other requests, complexity arises. Consequently, all permutations of UAVs, requests, and times should be examined, creating an exponentially large solution space $\mathbb{T}!(\mathbb{N}_{\text{UAV}}!\mathbb{A}) \cdot (\mathbb{C}\mathfrak{T}_t) \cdot (\mathbb{N}\mathbb{F}_s\mathbb{P}\mathbb{U}!)$.

In addition to the inherent complexity, in dynamic networks characterized by UAV and UE mobility, several key parameters remain uncertain. Without prior knowledge of UE areas ($\mathcal{A}^{t}_{u,a}$) and their request arrivals, it is impossible to determine the appropriate channels to send requests ($\tilde{\mathcal{Z}}^{\tau}_{r,c}$) or assess channel qualities ($\breve{\mathcal{Q}}^{\tau}_{c,a}$). Consequently, UAV trajectory planning ($\tilde{\mathcal{S}}^{t}_{n,a}$) is impossible ahead of time, as the UE's locations dictate UAV movement. The uncertainty also extends to function placement ($\tilde{\mathcal{Y}}^{t}_{f,n}$) and path planning ($\overrightarrow{\mathcal{R}}^{t}_{r,p}$), as the links connecting UAVs and network nodes are not predetermined. Therefore, tackling ALLOCATE requires a novel method that accommodates the dynamic nature and inherent uncertainties in the network.

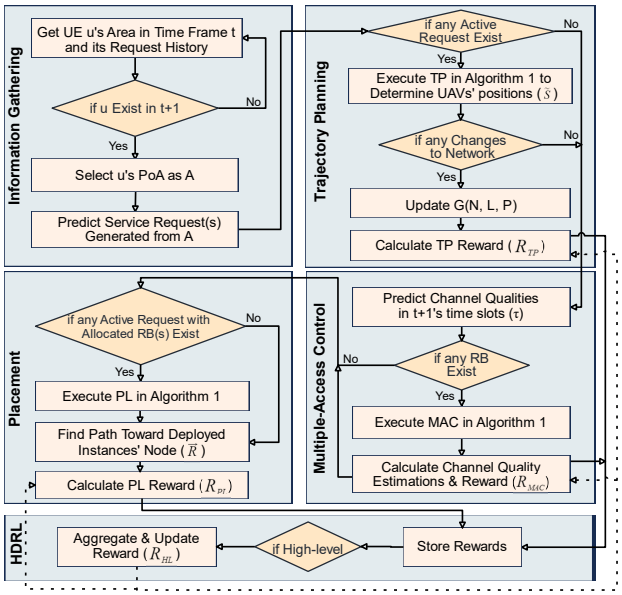

Figure 2. The proposed method's process for time frame t, including Information Gathering and Service Orchestration phases.

V. Proposed Method

We propose predictive energy efficient service orchestration (PERFECT) to tackle ALLOCATE, which operates in two phases: Information Gathering and Service Orchestration. As illustrated in Fig. 2, the PERFECT workflow is presented as a flowchart depicting the sequential execution and interconnection of its key components. The Information Gathering phase employs predictive learning to capture the temporal evolution of UE mobility and service request dynamics. Specifically, the process begins with the collection of request histories and network states, followed by the generation of predicted mobility and service demands. Using information retrieved from the former, the latter allocates resources. To further manage the complexity of large-scale problems, the Service Orchestration phase is divided into three sub-problems: Trajectory Planning (TP), MAC, and Placement (PL) modules. The TP module determines UAV trajectories and PoA updates based on predicted UE distributions; its output defines feasible communication links and coverage areas for the next time slot. The MAC module subsequently manages channel access to mitigate interference and updates the channel quality estimations for the next time frame. Finally, the PL module decides where network functions should be deployed by evaluating node capacities, latency constraints, and energy efficiency. The outputs of these modules are looped back into the environment, updating the state for the next orchestration cycle.

Fig. 3 outlines a complementary perspective by detailing technical underpinnings and addressing challenges. The Information Gathering phase, implemented via a Double Deep Q-Network (DDQN), resolves imperfect knowledge limitations. The DRL-based TP ensures energy-efficient UAV movement while maximizing coverage; the MAC's heuristic channel allocation achieves collision-aware transmission by evaluating channel qualities; and the action masking-enhanced Dueling Double Deep Q-Learning (D3QL) PL enables energy-efficient, latency-optimized function deployment and path selection.

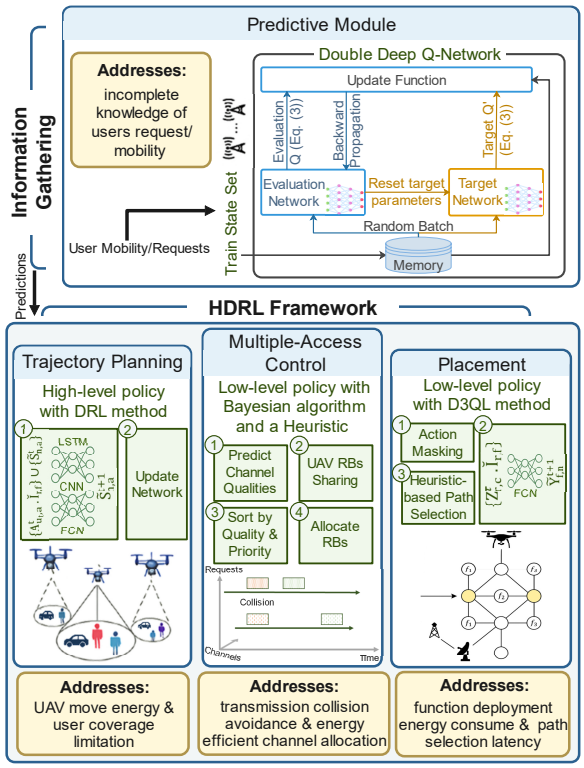

Figure 3. PERFECT's technologies and addressed challenges. The predictive module forecasts UE mobility and request patterns, while the HDRL framework integrates TP, MAC, and PL modules.

A. Motivation for Learning Algorithms

To facilitate adaptive decision-making in dynamic and uncertain scenarios, DRL demonstrates promise. Classical value-based approaches like DQL, a DRL's potential techniques, approximate action-value functions through deep Neural Networks (NNs). Double DQL enhances the stability of DQL by decoupling action selection and evaluation, utilizing the target value defined in (4) [50]. This target incorporates reward R, state O, real action a, and predicted action a' to update DQL weights ($\mathcal{W}$) for each observation-action of time t (O^t, a^t). Here, $a' = \text{argmax}_{a \in \boldsymbol{\mathcal{A}}} Q(O^{t+1}, a; \mathcal{W}^t)$ with $\mathcal{W}$ representing evaluation weights updated at each step, and $\mathcal{W}^-$ as target weights, synchronized every $\hat{t} \gg 0$ step. D3QL enhances DDQL by integrating Wang *et al.*'s dueling approach [51]. In D3QL, separate estimators calculate state values ($\mathcal{V}$) and action advantages (ρ), combining them to compute Q-values (5) and weights (6). This approach improves training stability, accelerates convergence, and mitigates overestimation issues.

$$Y^t = R^t + \Gamma \cdot Q(O^{t+1}, a'; \mathcal{W}^{t-}) \tag{4}$$

$$Q(O^t,a^t;\mathcal{W}^t) = \mathcal{V}(O^t;\mathcal{W}^t) + \rho(O^t,a^t;\mathcal{W}^t) - \frac{1}{|\boldsymbol{\mathcal{A}}|}\sum_{a' \in \boldsymbol{\mathcal{A}}} \rho(O^t,a';\mathcal{W}^t) \tag{5}$$

$$\mathcal{W}^{t+1} \leftarrow \mathcal{W}^t + \sigma \cdot [Y^t - Q(O^t, a^t; \mathcal{W}^t)] \nabla_{\mathcal{W}^t} Q(O^t, a^t; \mathcal{W}^t) \tag{6}$$

Although effective in small and moderate-sized decision spaces, these non-hierarchical methods operate on a single action space and uniform time scale. Consequently, when applied to heterogeneous processes with multi-node orchestration, they exhibit degraded convergence, become sensitive to state dimensionality, and require large exploration budgets. Also,

policy-gradient methods such as Proximal Policy Optimization (PPO) alleviate some instability issues through clipped surrogate objectives, yet they also suffer from slow convergence under combinatorial action spaces and lack mechanisms for decomposing multi-time-scale decisions.

To handle these limitations in large state and action spaces, HDRL employs action-space and temporal abstraction, decomposing orchestration into coordinated high- and low-level modules. This design enhances training efficiency by enabling module interaction across distinct dynamics and timescales: high-level policies manage long-term strategy, while low-level policies handle short-term control. Each module operates within a tailored state-action domain, reducing learning complexity compared to monolithic DQN/PPO models. Hierarchical coupling allows high-level policies to integrate low-level outcomes, improving stability, sample efficiency, and scalability with increasing UAVs, UEs, and functions. To achieve a globally optimal solution and resolve the limited observation capability, careful coordination to harmonize the modules' behaviors is needed. By designing well-structured state representations and employing action masking in low-level policies, the proposed HDRL architecture avoids convergence issues common in flat DRL. Thus, PERFECT achieves faster convergence, high service quality, and superior performance under multi-timescale constraints, establishing HDRL as an effective solution for complex large-scale problems.

B. Information Gathering Phase

The Information Gathering phase constructs a dynamic network graph $\mathcal{G}(\mathcal{N}, \mathcal{L}, \mathcal{P})$, representing UE areas and anticipated requests for the next time frame. Following the strategy by Farhoudi *et al.* [43], we adopt an online model to handle requests' sporadic and dynamic nature, as offline machine learning models cannot adapt to rapid changes in request patterns and render them insufficient for accurate predictions.

The proposed framework inherently addresses real-time demand fluctuations and unpredictable mobility patterns through its learning-based approach. Specifically, each PoA is equipped with a D3QL agent, continuously updating its policy based on newly observed transitions, enabling rapid adaptation to sudden traffic or request changes. These agents predict the probability of each request r being issued in the next time frame, identifying the presence of u_r in the area a ($\mathcal{A}^{t+1}_{u_r,a}$). During the prediction process, the agent returns a prioritized list of requests with the highest likelihood as well, with a reward based on the prediction accuracy and the state consisting of the received requests' history and the previous UE areas. Unlike static predictors, our online DRL method exploits Short-Term Memory (LSTM) layers to capture temporal variations in request arrivals caused by high UE mobility, while Convolutional Neural Network (CNN) layers extract spatial correlations between adjacent areas. A memory bank stores observed transitions to enable efficient NN updates through random sampling, as illustrated in Fig. 3.

C. Service Orchestration Phase

Following centralized aggregation of UE area predictions and anticipated service requests for the subsequent time frame, this phase deploys functions and allocates resources to meet predicted requests and requirements. To tackle the multi-faceted challenges of decision-making in dynamic, resource-constrained environments, we employ an HDRL framework. It facilitates decision-making by decomposing complex problems into manageable subproblems, employing high-level policies for strategic, long-term decisions and low-level policies for operational, short-term ones. The hierarchical interaction perfectly fits the proposed decomposition approach and ensures immediate action adaptation to real-time network dynamics without deviating from long-term objectives. Reward-sharing and feedback mechanisms enable continuous synchronization of short-term adaptations with global optimization goals.

Algorithm 1: Service Orchestration Phase

Input: $\mathbb{T}$, ϵ', and $\widetilde{\epsilon}$
Output: $\tilde{\mathcal{S}}$, $\tilde{\mathcal{B}}$, $\tilde{\mathcal{Z}}$, $\tilde{\mathcal{Y}}$, $\overrightarrow{\mathcal{R}}$

$\mathcal{W}_{\text{TP, PL}} \leftarrow \mathbf{0}$, $\mathcal{W}^{-}{}_{\text{TP, PL}} \leftarrow \mathbf{0}$, $\epsilon_{\text{TP, PL}} \leftarrow 1$, $\psi_{\text{TP, PL}} \leftarrow \{\}$
for t *in* $[1 : \mathbb{T}]$ **do**
  ⋆ Trajectory Planning (high-level, frame-scale) ⋆
  $\tilde{\mathcal{S}}^{t+1} \leftarrow \textit{EpsilonGreedy}\Big(Q(O^t_{\text{TP}}, A^t_{\text{TP}}; \mathcal{W}_{\text{TP}}), \epsilon_{\text{TP}}\Big)$
  $\mathcal{H} \leftarrow \{t - \mathcal{H}, \ldots, t\}$
  Calculate O^{t+1}_{TP} according to $\mathcal{H}$ and (7)
  $\epsilon_{\text{TP}} \leftarrow \max(\epsilon_{\text{TP}} - \epsilon', \widetilde{\epsilon})$
  Update $\mathcal{G}(\mathcal{N}, \mathcal{L}^t, \mathcal{P}^t)$ based on $\tilde{\mathcal{S}}^{t+1}$
  Select $\tilde{\mathcal{B}}^{t+1}$ for each request
  ⋆ Multiple-Access Control (low-level, slot-scale) ⋆
  Calculate O^{t+1}_{MAC} according to (10)
  Calculate $\overline{\mathcal{Q}}^{t+1}$ based on O^{t+1}_{MAC} (11)
  $\tilde{\mathcal{Z}} \leftarrow \textit{MAC}\Big(\tilde{\mathcal{B}}^{t+1}, \tilde{\mathcal{S}}^{t+1}, \omega^t, \overline{\mathcal{Q}}^{t+1}, \mathfrak{T}_t\Big)$
  Compute R^t_{MAC} based on channel qualities
  ⋆ Placement (low-level, frame-scale) ⋆
  $\tilde{\mathcal{Y}}^{t+1} \leftarrow \textit{EpsilonGreedy}\Big(Q(O^t_{\text{PL}}, A^t_{\text{PL}}; \mathcal{W}_{\text{PL}}), \epsilon_{\text{PL}}\Big)$
  Calculate O^{t+1}_{PL} according to (12)
  $\epsilon_{\text{PL}} \leftarrow \max(\epsilon_{\text{PL}} - \epsilon', \widetilde{\epsilon})$
  Select $\overrightarrow{\mathcal{R}}^{t+1}$ for each request
  ⋆ Training and Reward Propagation ⋆
  Calculate R^t_{TP} and R^t_{PL} based on (9) and (14)
  if *high level* **then**
    $R^t_{\text{HL}} = R^t_{\text{TP}} + \chi \cdot R^t_{\text{MAC}} + \kappa \cdot R^t_{\text{PL}}$
    $R^t_{\text{TP}}, R^t_{\text{PL}} = R^t_{\text{HL}}$
    Update global state to sync TP → MAC/PL
  $\psi_{\text{TP}} \leftarrow \psi_{\text{TP}} \cup \{(O^t_{\text{TP}}, \tilde{\mathcal{S}}^{t+1}, R^t_{\text{TP}}, O^{t+1}_{\text{TP}})\}$
  Train $\mathcal{W}_{\text{TP}}$ on batch of samples from ψ_{TP} (6)
  $\psi_{\text{PL}} \leftarrow \psi_{\text{PL}} \cup \{(O^t_{\text{PL}}, \tilde{\mathcal{Y}}^{t+1}, R^t_{\text{PL}}, O^{t+1}_{\text{PL}})\}$
  Train $\mathcal{W}_{\text{PL}}$ on batch of samples from ψ_{PL} (6)

The TP module serves as the high-level policy, responsible for long-term UAV movement decisions that optimize service coverage, feasible communication links, and the structural conditions under which low-level modules operate. The MAC and PL modules operate as low-level policies, responsible for real-time, fine-grained decisions; such as channel allocation, Resource Block (RB) management, function deployment, and path selection–based on rapidly changing environmental con-

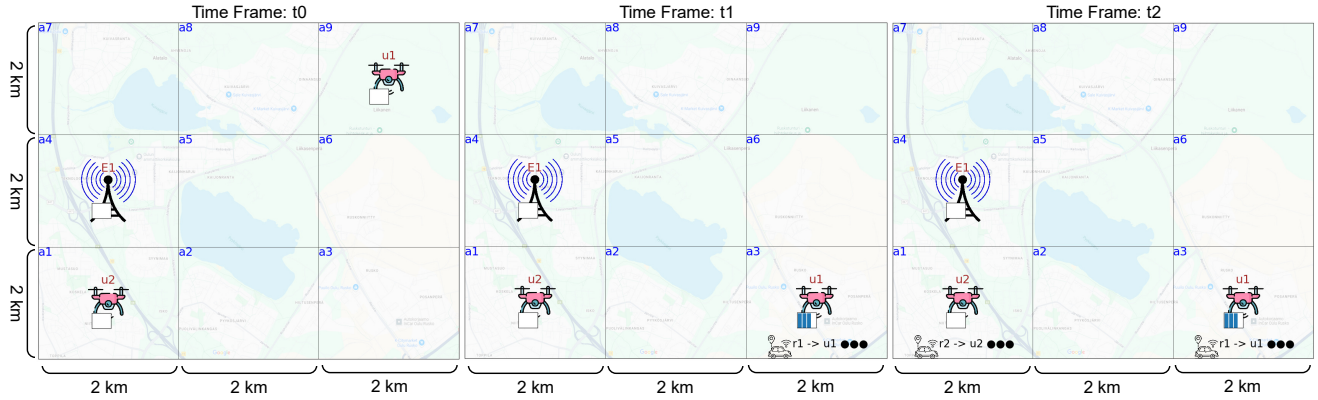

Figure 4. A possible scenario where UE r_1 enters from area a_3 in time frame t_1. Although u_2 requires less energy to reach a_3 than u_1, TP learning algorithm decides to move u_1, predicting that r_2 will enter from a_1. Each grid cell in this deployment across Oulu city represents approximately 2km × 2km.

Algorithm 2: Learning's *EpsilonGreedy* Process

Input: $Q(S, A; W)$, ϵ
Output: Actions
$\zeta \leftarrow$ randomly generate a number from $[0:1]$
if $\zeta > \epsilon$ **then**
  $Actions \leftarrow \text{argmax}_{a \in A} Q(O, a; \mathcal{W})$
else
  $Actions \leftarrow$ select random actions

ditions. Each RB represents a time slot within a specific channel, determining the allocation of communication resources for transmitting UE requests and enabling the MAC to react at a higher temporal resolution than the TP. Using a simultaneous learning approach (Algorithm 1), the low-level policies are first stabilized independently to ensure reliable short-term behavior. MAC learns efficient channel allocation across time slots, while PL optimizes function deployment and path selection. Their decisions shape the environment observed by TP, and their instantaneous rewards form the components of the high-level reward. Once stabilized, their operational outcomes (aggregated rewards) are integrated into the training of high-level policies, forming a closed feedback loop where updated short-term results influence long-term optimization. Hence, R^t_{HL} is considered the total reward, which combines rewards from TP, MAC, and PL modules, scaled by factors χ and κ to balance their contributions. Through this dynamic interaction across distinct time scales, low-level policies optimize within TP-defined constraints while TP learns to anticipate downstream effects, maintaining coherence between times and achieving real-time adaptability without sacrificing efficiency.

Trajectory Planning: The TP module determines optimal UAV areas for the upcoming time frame, aiming to optimize total UAV movement energy consumption while maximizing request coverage. The purpose of using a learning algorithm rather than a heuristic approach lies in its ability to achieve long-term optimization, thereby reducing UAVs' overall energy consumption during their movement. Using a D3QL algorithm, detailed in Algorithm 1 (steps 5–10), it predicts UAV movements based on past observations, prioritizing long-term energy optimization. A scenario is illustrated in Fig. 4 that represents the Oulu city area, where each grid cell represents approximately 2km × 2km. In this scenario, a UE request r_1 arises in area a_3 in time frame t_1, with UAVs u_1 existing at a_9 and u_2 at a_1 in t_0. Although u_1 requires more energy to reach a_3 (based on Λ), moving u_1 is more efficient, predicting another request at a_1 and avoiding unnecessary movements for u_2. In another scenario, if r_1 moves from a_3 at t_1 to a_2 at t_2, the algorithm pre-positions u_3 at a_2 from t_0, minimizing movement energy while ensuring timely delivery.

The module implementation approach is to design the state and action spaces in a scalable manner. The state O^t_{TP} incorporates total requested capacities per area and UAV locations, encoded in the one-hot format (7), based on the last $\mathcal{H}$ observations. This representation is request number independent, making it scalable for networks with varying UE numbers. The state serves as input to a NN architecture consisting of LSTM, CNNs, and linear layers. The action A^t_{TP} is then generated that represents areas assigned to UAVs for the next time frame, categorized as $\tilde{\mathcal{S}}^{t+1}$ (8). The action results in changes to the network graph, as UAVs would be relocated across different areas, leading to alterations in the links and paths. Hereafter, $\tilde{\mathcal{B}}^{t+1}$ (PoAs) are determined based on $\mathcal{A}^{t+1}_{u,a}$ (UE areas) and $\tilde{\mathcal{S}}^{t+1}$ (network node areas). The reward function R^t_{TP} maximizes coverage while optimizing UAV energy consumption, aligning with ALLOCATE's objective (9). This approach balances UE demand, service coverage, and energy efficiency, using predictive insights.

$$o^t_{\text{TP}} = \Big\{ \sum_{\mathbb{R},\mathbb{F}_{s_r}} \mathcal{A}^t_{u_r,a} \cdot \breve{\mathcal{I}}_{r,f} \;|a \in \mathbb{A}\Big\} \cup \Big\{\tilde{\mathcal{S}}^t_{n,a} | n, a \in \mathbb{N}, \mathbb{A}\Big\} \tag{7}$$

$$O^t_{\text{TP}} = \Big\{o^h_{\text{TP}} | h \in \{t-\mathcal{H}, \ldots, t\}\Big\}$$
$$A^t_{\text{TP}} = \Big\{\tilde{\mathcal{S}}^{t+1}_{n,a} | n, a \in \mathbb{N}, \mathbb{A}\Big\} \tag{8}$$

$$R^t_{\text{TP}} = \sum_{\mathbb{R},\mathbb{N}} \tilde{\mathcal{B}}^{t+1}_{u_r,n} - \alpha \cdot \sum_{\mathbb{N},\mathbb{A}} \Lambda(\tilde{\mathcal{S}}^{t+1}_{n,a_1} - \tilde{\mathcal{S}}^t_{n,a_2}) \tag{9}$$

Multiple-Access Control: The MAC module allocates energy-efficient channels ($\overline{\mathcal{M}}_c$) to UEs for request transmission. It functions at the time slot level, with each time frame containing $\mathfrak{T}_t$ time slots. Efficient allocation requires predicting channel qualities ($\breve{\mathcal{Q}}^{\tau}_{c,a}$) in each area and mapping them to requests. Thus, the module is divided into two parts: channel quality prediction and heuristic-based channel assignment. Since channel predictions should update dynamically during HDRL training periods, this component is included in the MAC module rather than the Information Gathering phase.

Table III
MAPPING OF PERFECT SUBPROBLEMS TO SYSTEM ROLES, DECISION LAYERS, AND APPLIED TECHNOLOGIES.

Subproblem	Physical Aspect	Decision Layer	Applied Technology (Rationale)
Info Gathering	UE mobility & demand prediction	Sensing & Prediction	D3QL+CNN+LSTM (spatiotemporal feature extraction)
TP	UAV mobility, coverage & energy efficiency	Mobility management	D3QL (long-term control, stable convergence)
MAC	Channel allocation & latency control	Communication/MAC	Bayesian + heuristic (probabilistic–deterministic)
PL	Function deploy, routing, & QoS assure	Resource allocation	D3QL + masking + heuristic route (energy-efficient placement)
HDRL	Cross-layer coordination across multi-time-scale	Hierarchical control	Policy integration (temporal abstraction, scalability)

This hybrid design prioritizes predictive elements for critical tasks while employing deterministic algorithms for simpler processes, ensuring energy-efficient channel allocation.

To predict channel quality, the MAC module utilizes a Bayesian algorithm that models and updates beliefs under uncertainty. The Bayesian algorithm assigns an initial quality value of 0.5 to each channel in every area. As UAVs traverse across areas, these estimates (posterior beliefs) are revised incrementally based on updated observations O^t_{MAC} (10) and weighted by λ, which governs observation impact. This process, outlined in steps 12–13 of Algorithm 1, continuously refines channel quality estimates $\overline{\mathcal{Q}}^t_{c,a}$, calculated using (11).

$$O^t_{\text{MAC}} = \Big\{ \frac{1}{|\mathfrak{T}_t|} \cdot \sum_{\mathfrak{T}_t} \breve{\mathcal{Q}}^{\tau}_{c,a} | c, a \in \mathbb{C}, \mathbb{A} \Big\} \tag{10}$$

$$\overline{\mathcal{Q}}^t_{c,a} = \Big\{ \lambda \cdot O^t_{\text{MAC}} + (1-\lambda) \cdot \overline{\mathcal{Q}}^{t-1}_{c,a} \Big\} \tag{11}$$

Following channel quality prediction, the MAC module allocates them to requests, as detailed in Algorithm 3. The allocation strategy prioritizes proximity to request deadlines to meet E2E latency requirements $\breve{\mathcal{D}}_r$. At the beginning of each time frame, co-located UAVs share available RBs. UAVs then allocate their RBs to connected UEs using predicted qualities $\overline{\mathcal{Q}}^t_{c,a}$, priorities ω^t, and required time slots $\breve{\mathcal{T}}_r$. Channels and requests are sorted by their respective quality and priority, and RBs are assigned iteratively until either RBs are exhausted or request requirements are fulfilled. Specifically, the algorithm calculates the minimum between remaining and required time slots ($\acute{\tau}$) for each request. It allocates all slots within the interval $\tau : \acute{\tau}$ if no prior allocation exists, adhering to the single-channel transmission constraint (C3). This procedure ensures high-priority UEs access superior-quality channels, optimizing resource usage while complying with QoS constraints.

Placement: This module determines the optimal deployment of functions on network nodes $\tilde{\mathcal{Y}}^t_{f,n}$ and the paths to reach them $\overrightarrow{\mathcal{R}}^t_{r,p}$, relying on predicted requests' UE areas and calculated UAV locations. It aims at minimizing energy consumption for function deployment ($\overline{\mathcal{E}}_n$) while adhering to node processing and link bandwidth constraints ($\widehat{\mathcal{C}}_n$, $\widehat{\mathcal{L}}_l$), reducing transmission energy ($\overline{\xi}_l$), and meeting latency requirements ($\breve{\mathcal{D}}_r$). Given the impact of UE and UAV mobility on placement dynamics, the PL module uses a D3QL learning algorithm (Algorithm 1, steps 16–19) to strategically deploy functions on nodes, ensuring that the total energy consumption is efficient and request requirements are in compliance over time.

PL module implementation provides a highly efficient and deployable state space, designed for scalability and independence from request volume. Its state O^t_{PL} includes total requested function capacities across all nodes with channel access as well as nodes' available capacities and associated energy consumption (12). Additionally, the reward function R^t_{PL} focuses on maximizing accepted requests (coverage) while optimizing energy consumption, considering latency constraints (14). Furthermore, directly considering all network nodes, functions, links, and paths in the action space poses scalability and convergence challenges due to the large action space. Two key strategies are employed to overcome this challenge. First, path selection is decoupled from learning via a heuristic integrated into reward calculation. Specifically, after selecting nodes for function deployment, the algorithm prioritizes requests by latency requirements and identifies feasible paths that meet the requirements with minimal energy consumption ($\overline{\xi}_l$). The algorithm penalizes invalid deployments with negative rewards, while otherwise rewarding the total energy consumed to establish the connection. Second, action masking reduces the effective action space without sacrificing optimality by eliminating infeasible actions by assigning large negative values. The infeasible actions include deploying functions with no predicted requests or exceeding thresholds. The PL module action A^t_{PL} is defined as in (13).

Algorithm 3: Channel Allocation Method

Input: $\tilde{\mathcal{B}}^{t+1}, \tilde{\mathcal{S}}^{t+1}, \omega^t, \overline{\mathcal{Q}}^{t+1}, \mathfrak{T}_t$
Output: $\{\tilde{\mathcal{Z}}^{\tau} | \tau \in \mathfrak{T}_t\}$

$\tilde{\mathcal{Z}}^{\tau}_{r,c} \leftarrow 0 \quad \forall\, r, c, \tau \in \mathbb{R}, \mathbb{C}, \mathfrak{T}_t$
foreach n *in* $\mathbb{N}$ **do**
  $\mathbb{C}_{sorted} \leftarrow SORT_c\big\{\mathbb{C}, key = \overline{\mathcal{Q}}^{t+1}_{c,a} | \tilde{\mathcal{S}}^{t+1}_{n,a} = 1\big\}$
  $\mathbb{R}_{sorted} \leftarrow SORT_r\big\{\mathbb{R}, key = \omega^t | \tilde{\mathcal{B}}^{t+1}_{u_r,n} = 1\big\}$
  foreach c *in* $\mathbb{C}_{sorted}$ **do**
    $\tau \leftarrow 0$
    while $\tau \leq |\mathfrak{T}_t|$ **do**
      foreach r *in* $\mathbb{R}_{sorted}$ **do**
        $\acute{\tau} \leftarrow \tau + min(\breve{\mathcal{T}}_r, |\mathfrak{T}_t| - \tau)$
        if $\sum_{c,\tau} \tilde{\mathcal{Z}}^{\tau:\acute{\tau}}_{r,c} == 0$ **then**
          $\tilde{\mathcal{Z}}^{\tau:\acute{\tau}}_{r,c} \leftarrow 1$
          $\tau \leftarrow \acute{\tau}$

$$O^t_{PL} = \Big\{ \sum_{\mathbb{R},\mathbb{C},\mathfrak{T}_t} \tilde{\mathcal{Z}}^{\tau}_{r,c} \cdot \breve{\mathcal{I}}_{r,f} \;|f \in \mathbb{F}_s \Big\} \cup \Big\{ (\widehat{\mathcal{C}}_n, \overline{\mathcal{E}}_n) | n \in \mathbb{N} \Big\} \tag{12}$$

$$A^t_{\text{PL}} = \Big\{ \tilde{\mathcal{Y}}^{t+1}_{f,n} | \text{ action is not masked}, f, n \in \mathbb{F}_s, \mathbb{N} \Big\} = \Big\{ \tilde{\mathcal{Y}}^{t+1}_{f,n} \;\Big|\; \begin{matrix} \sum_{\mathbb{R}} \breve{\mathcal{I}}_{r,f} \leq \sum_{\mathbb{N}} (\tilde{\mathcal{Y}}^{t+1}_{f,n} \cdot \hat{\mathcal{C}}_n) \\ \sum_{\mathbb{R}} (\tilde{\mathcal{X}}^{t+1}_{r,f} \cdot \tilde{\mathcal{Y}}^{t+1}_{f,n}) > 0 \end{matrix} \quad f, n \in \mathbb{F}_s, \mathbb{N} \Big\} \tag{13}$$

$$R^t_{\text{PL}} = \sum_{\mathbb{R}} \Big(\prod_{\mathbb{F}_{s_r}} \tilde{\mathcal{X}}^{t+1}_{r,f} \Big) - \alpha \Big(\sum_{\mathbb{F}_s, \mathbb{N}, \mathbb{T}} \tilde{\mathcal{Y}}^{t+1}_{f,n} \overline{\mathcal{E}}_n + \sum_{\mathbb{L}^{t+1}, \mathbb{P}^{t+1}, \mathbb{R}, \Delta_r} \overline{\xi}_l \mathcal{J}^{t+1}_{p,l} \overrightarrow{\mathcal{R}}^{t+1}_{r,p} \Big) \tag{14}$$

Table III details the subproblems addressed in PERFECT, highlighting their physical aspects, corresponding decision layers, and underlying technologies. The modular decomposition enables hierarchical decision-making across multiple time scales, effectively enhancing adaptability, energy efficiency, and QoS assurance in aerial-terrestrial vehicular 6G networks.

VI. Performance Evaluation

This section evaluates our proposed method's efficiency. It begins with a convergence analysis, examining the impact of hyperparameters on the algorithm's performance. Subsequently, PERFECT is benchmarked against baseline methods using diverse metrics, demonstrating its superiority.

A. Simulation settings

The simulations are conducted within a vehicular edge–cloud continuum, where UAVs initially fly at a fixed height and are randomly distributed across the network area. Key simulation parameters are summarized in Table IV, where parameters follow a uniform distribution $\mathcal{U}$. The parameters ensure that energy consumption is physically modeled following the aerodynamic formulation in, while the wireless channel reflects environment-dependent attenuation following. This configuration enables a fair and realistic comparison between PERFECT and baseline frameworks under varying conditions. To model UE mobility and reflect realistic user dynamics, we employ Simulation of Urban Mobility (SUMO) [52], a microscopic vehicular mobility simulator designed for large-scale network environments. Specifically, we consider a grid environment with bidirectional roads, while SUMO provides realistic trajectories across different urban zones of Oulu City in Finland, including both city-center and suburban areas, allowing us to reflect diverse densities and mobility patterns. UEs follow the Manhattan mobility model, moving straight with a 50% probability and turning left or right with a 25% probability at intersections. The mobility model is inspired by the Manhattan-like urban cities and implemented with additional stochasticity in vehicle interactions and departure processes, providing diverse and generalized dynamics. During the simulation, there are 50 UEs in the network, each capable of generating various types of service requests with heterogeneous data rate requirements, providing a practical context for assessing our proposed method.

Table IV
Simulation Parameters.

Domain	Parameter	Value
Network	Node Processing Capacity ($\widehat{\mathcal{C}}_n$)	$\sim \mathcal{U}(25, 70)$ Mbps
	Node Energy Capability ($\overline{\mathcal{E}}_n$)	$\sim \mathcal{U}(12, 36)$ Hz
	Link Bandwidth Capacity ($\widehat{\mathcal{L}}_l$)	$\sim \mathcal{U}(10, 30)$ Mbps
	Link Latency ($\mathcal{D}^t_{r,l}$)	$\sim \mathcal{U}(4, 16)$ Mbps
	Link Energy Consumption ($\overline{\xi}_l$)	$\sim \mathcal{U}(5, 8)$ Hz
	UAV Weight, velocity (W_n, $V_w(t)$)	$\mathcal{U}(4, 6)$kg,$\mathcal{U}(8, 12)$m/s
Area	Areas ($\mathbb{A}$)	4*4 Grid
	Induced Power, Drag Coefficient (I,ς)	0.08, 0.05
	Air Density (φ)	1.225 $kg.m^{-3}$
	Rotor Disk, Frontal Area (υ_r, υ_f)	0.6, 0.25 m^2
Service	Number of Composed Services ($\mathbb{S}$)	20
	Number of Atomic Functions ($\mathbb{F}_s$)	32
	Service Duration ($\overrightarrow{\mathcal{T}_s}$)	$\sim \mathcal{U}(3, 10)$ frames
	Service Required Time Slots ($\breve{\mathcal{T}}_r$)	$\sim \mathcal{U}(3, 9)$ slots
UE	Vehicular UEs ($\mathbb{U}$)	50
	Bandwidth Requirement ($\breve{\mathcal{L}}_r$)	$\sim \mathcal{U}(2, 8)$ Mbps
	Capacity Requirement ($\breve{\mathcal{I}}_{r,f}$)	$\sim \mathcal{U}(8, 20)$ Mbps
	Latency Requirement ($\breve{\mathcal{D}}_r$)	$\sim \mathcal{U}(50, 100)$ ms
Channel	Time slots per time frame ($\mathfrak{T}_t$)	10
	LoS Probability (θ_a^{LoS})	$\sim \mathcal{U}(0.2, 0.8)$
	Rayleigh coefficient ($R^{\tau}_{c,a}$)	$\sim \mathcal{R}(\sigma \sim \mathcal{U}(0.2, 0.8))$
	Quality Threshold ($\widehat{\mathcal{Q}}$)	1
	Channel Energy Consumption ($\overline{\mathcal{M}}_c$)	$\sim \mathcal{U}(2, 8)$ Hz
	Weather Attenuation ($\zeta(\Omega_\tau)$)	$\{0, 2.5, 5.0\}$ dB
	Shadowing Std. Deviation ($\eta_s(\Omega_\tau)$)	$\{2.0, 3.0, 4.5\}$ dB
HDRL	Running episodes	100,000
	Replay Memory ($\psi_{\text{TP}}, \psi_{\text{PL}}$)	2000, 1000
	Discount Factor (Γ)	0.8
	Scaling Factors (α, χ, κ)	0.001, 0.5, 0.8
	EpsilonGreedy process (ϵ, ϵ', $\widetilde{\epsilon}$)	1, 0.00005, 0.0001
TP	NN Layers (LSTM / Two CNN (kernel size, stride, and pooling size) / Three Fully Connected)	128 units / (3, 2, 2) / (256, 128, 64 units)
	Activation Function	Hyperbolic Tangent
PL	NN Layers (Three Fully Connected)	(512, 256, 128 units)
	Activation Function	Leaky ReLU

$\mathcal{U}$: Uniform distribution, $\mathcal{R}$: Rayleigh distribution, Mbps: Megabits per second, ms: millisecond, Hz: Hertz, ReLU: Rectified Linear Units

B. Convergence Analysis

We conduct experiments with varying hyperparameters to assess the proposed method's convergence behavior. The parameters are critical for training efficiency and stability, with the learning rate controlling the magnitude of NN weight updates and batch size defining sample numbers per training episode. Small learning rates result in prolonged training durations or even non-convergence because of ineffective loss minimization. Conversely, large rates lead to unstable training dynamics and prevent convergence, where rapid weight updates may overshoot optimal values or become trapped in local optima. Fig. 5 demonstrates that a learning rate of 0.001 achieves optimal convergence, offering high rewards and stability across training episodes. Also, smaller batch sizes introduce variability and instability due to random sampling, while larger batch sizes increase overhead and slow convergence. As shown in Fig. 6, a batch size of 32 strikes the right balance, ensuring faster and smoother convergence, compared to the more variable results with a batch size of 8 and slower convergence with a batch size of 64. A noteworthy feature of our HDRL framework is that it considers two high- and low-level policies, leading to reward escalation once we start a high-level one (episode 25,000).

C. Comparison experiments

We compare our proposed method against baseline approaches to demonstrate its effectiveness. The compared approaches include the optimal solution of ALLOCATE derived via CPLEX (with complete knowledge); a random selection strategy for UAV trajectory planning, channel selection, and

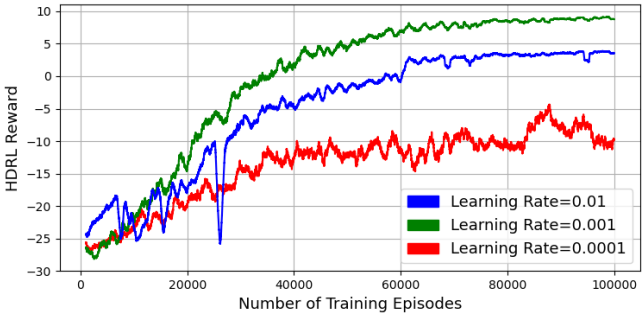

Figure 5. Convergence performance of the PERFECT algorithm for different learning rates, highlighting its stability and reward optimization.

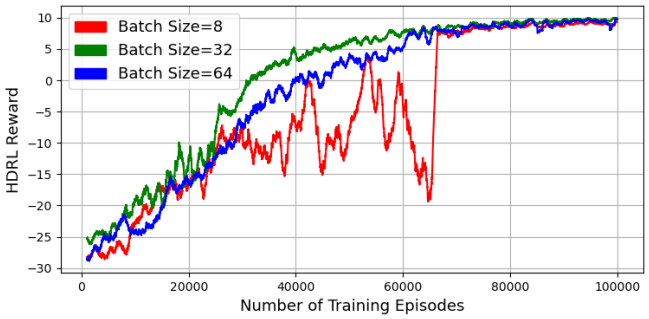

Figure 6. Convergence performance of the PERFECT algorithm for different batch sizes, showing the trade-offs between stability and convergence speed.

service deployment; the Successive Convex Approximation-based (SCA) method [33] that considers multi-UAV trajectory and resource planning; and the Hungarian and DDQN-based (HaDDQN) method [30], which uses a DDQN approach for service placement and a Hungarian algorithm for RB allocation in co-channel settings. The SCA method is modified to include latency constraints instead of UAV recharging as considered in the original study. Likewise, the HaDDQN method is adapted to align with our framework by embracing the Hungarian algorithm for UAV trajectory decisions.

Three quantitative metrics critical to vehicular networks are considered to compare methods. First, the number of accepted requests (request coverage) indicates scalability in dynamic environments, essential for connectivity. Second, energy consumption quantifies the energy required for UAV movement, communication through channels, and resource utilization, addressing sustainability goals and operability extension. Third, E2E latency assesses the ability to ensure seamless user experiences, which is crucial for futuristic latency-sensitive applications like autonomous driving. These metrics collectively provide a comprehensive assessment of the method's effectiveness, highlighting the potential to balance service delivery, energy use, and E2E latency requirements.

To evaluate the different aspects of the problem and their implications for service orchestration, we increase the number of requests, network nodes, and communication channels that are varied across simulation scenarios. The first one assesses the response to varying UE and active requests to verify the ability to maintain high request acceptance under dynamic workload conditions. It simulates real-world request fluctuations, as seen in futuristic applications like intelligent transportation systems [53], where vehicles generate variable real-time data during peak hours or emergencies. Second, varying network node and areas through SUMO[3] evaluate the effect of network and infrastructure scale. The scenario reflects expected growth in 6G infrastructure size [54] to ensure effective performance across diverse environments, from small-scale edge-cloud systems to expansive, distributed networks. In futuristic networks, densely populated urban areas with high vehicular density and connectivity demand can strain channel availability [55]. Thus, we consider varying communication channel $\mathbb{C}$ as the third scenario that determines adaptability to constrained channel availability across diverse conditions.

[3]The transition from sparse suburban to dense urban traffic is captured.

Fig. 7 illustrates PERFECT and baseline methods comparisons in terms of accepted requests percentage (1) as well as energy consumption (2) and E2E latency (3) incurred by each request under increasing requests (a), nodes (b), and channels (c) scenarios. Notably, requests failing latency requirements are considered unacceptable, reflecting their inability to provide a satisfactory user experience. To ensure statistical robustness in a setting containing uniform distributions, we consider average values from multiple system runs, each using identical seed values for all methods. This approach ensures that each method operates under the same conditions in each iteration. In this regard, even in optimal solutions, energy consumption and E2E latency fluctuate due to dynamic factors like node/link capacities and evolving bandwidth requirements. Besides, the shaded regions around the trend lines represent standard deviations, indicating performance variability. The observed differences in shaded regions stem from the varying robustness of the methods to network dynamics, with the Random method showing the highest variability due to uninformed decisions, while PERFECT and ALLOCATE achieve more stable performance through adaptive allocation.

Scenario I: This scenario begins with increasing request numbers from 7 (minimal load) to 30 (extreme load) per time frame while keeping network size (10 nodes) and channel availability (10 channels) constant. Fig. 7(a) demonstrates the scalability of the proposed framework under different request numbers. All methods exhibit a slight decline in accepted requests as the number of requests increases, as the number of network nodes remains fixed. Higher request volumes lead to increased E2E latencies due to diverse requests with varying requirements across networks with high-capacity nodes located far from PoAs. Also, to accommodate high demand, methods are compelled to deploy additional functions and utilize more links for transmitting requests and responses, thereby increasing energy consumption.

PERFECT maintains high request coverage and relatively stable energy consumption, outperforming HaDDQN, SCA, and random methods. Regarding E2E latency, both HaDDQN and PERFECT perform well, as they prioritize maintaining latency within acceptable requirements. The random method's poor request handling leads to service interruptions, latency infringement, and high energy consumption, rendering it unsuitable for real-world applications. SCA and HaDDQN's limited ability to predict requests contributes to their reduced performance, particularly in larger request volumes. Also, their

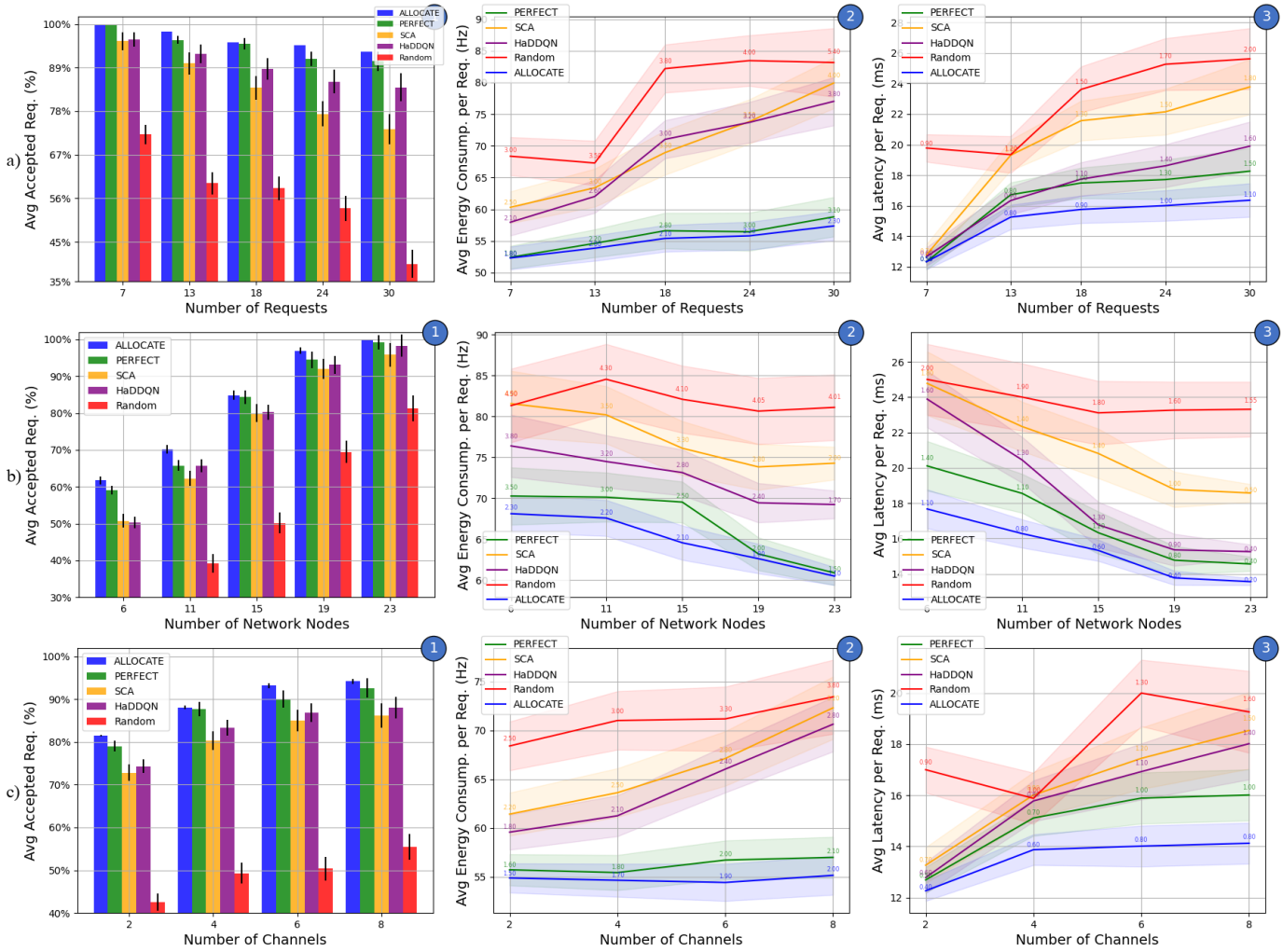

Figure 7. (1) supported requests percentage, (2) energy consumption, and (3) E2E latency are compared between the ALLOCATE, PERFECT, SCA, HaDDQL, and random methods as (a) request set, (b) network size, and (c) channel size expand. The shaded regions indicate the standard deviation across multiple runs.

TP methods overlook total energy minimization (discussed in Section V-C), incurring higher energy consumption. SCA's lack of shared channel support further diminishes its acceptance range compared to HaDDQN, despite its comparable performance under lighter loads. HaDDQN, benefiting from its learning PL, achieves superior energy efficiency than SCA in function deployment, though it remains less efficient than PERFECT. PERFECT's predictive capabilities and Bayesian-based channel sharing, combined with efficient TP and PL modules, ensure E2E latency compliance. It strategically deploys functions farther from UEs to reduce energy consumption, resulting in slightly higher, but within latency tolerances in some cases. The gap between ALLOCATE and PERFECT is minimal and stems from occasional prediction errors that necessitate deploying slightly more functions. However, these increases are negligible in practical scenarios and underscore PERFECT's efficient prediction algorithm. Such consistency demonstrates the predictive module's adaptability to unexpected mobility or traffic patterns, as it dynamically updates its knowledge base to reflect new behavior, preventing service degradation and sustaining stable orchestration efficiency.

Scenario II: The second scenario scales the network infrastructure from 6 to 23, with fixed requests (15) and channels (10). As the network size increases, the number of areas also expands (from 4×4 to 6×6). Fig. 7(b) shows the scalability and flexibility of the proposed framework in varying environments and under different infrastructure. Initially, all methods show lower acceptance rates due to an imbalance between requests and limited UAVs. As network density increases, acceptance rates improve with additional network resources. Energy consumption starts high due to reliance on high energy-intensive edge nodes but gradually decreases as UAVs are deployed, requiring fewer links for service delivery. Similarly, E2E latency declines as UAVs are positioned closer to UEs, reducing transmission latencies.

The proposed method exhibits strong scalability and efficient energy consumption while performing practically similarly to HaDDQN regarding latency. The random approach's energy-intensive node selection leads to high energy consumption and prolonged latency despite minor improvement due to increased nodes. As seen in the first scenario, SCA and HaDDQN struggle with request prediction, which affects their performance in smaller networks. However, as the network size increases, the acceptance of both methods increases since

there are sufficient nodes in each area. SCA struggles with high latency in limited networks due to distant function deployments, occasionally violating latency requirements. As the network size increases, SCA's latency improves, reflecting its ability to better utilize the expanded infrastructure. Energy consumption in PERFECT is notably efficient, with lower UAV movement and deployment energy compared to SCA and HaDDQN, owing to its HDRL algorithm that optimally selects proper nodes for service coverage and delivery. The HDRL algorithm enhances this efficiency through high-level policies that become increasingly impactful as the network size grows. While HaDDQN and PERFECT exhibit similar energy usage in terms of deployment, PERFECT's predictive capabilities offer minor improvements. As UAV numbers grow, PERFECT further reduces energy consumption by accurately predicting user behavior and optimizing unnecessary UAV movements. The information gathering DRL agents continuously capture mobility-induced variations, enabling PERFECT to anticipate real-time mobility and adjust TP decisions. This optimization also lowers E2E latency, contributing to the method's overall effectiveness in handling scalability. ALLOCATE performs comparably, especially in the presence of sufficient resources in the network, with minor variations due to slight prediction errors during information gathering.

Scenario III: In this scenario, the number of channels is increased from 2 to 8 while keeping the number of requests (10) and network nodes (10) fixed. Channel availability influences system performance, with noticeable improvements observed when the number of channels exceeds two. This enhancement is attributed to the requirement of three or more time slots for most requests. With more channels, accepted requests increase across all methods due to additional available RBs for transmission. However, this growth raises energy consumption because of the unique energy demands of each channel and the complexity of managing shared channels. Similarly, the rise in accepted requests slightly increases E2E latency, reflecting the higher system load.

In this scenario, PERFECT achieves higher request acceptance rates and superior energy efficiency than SCA and HaDDQN due to its Bayesian algorithm for channel quality prediction that optimizes channel allocation. HaDDQN lacks channel prediction, leading to inefficient utilization and high energy consumption. Furthermore, the absence of support for shared channels led to an increase in channel energy consumption and E2E latency for SCA. The difference in accepted requests between ALLOCATE and PERFECT stems from channel quality prediction errors and the proposed greedy channel selection. E2E latency remains stable for PERFECT across channel configurations, whereas other methods, especially SCA and random, experience rising latency with increasing channel availability, highlighting their limitations in managing higher channel counts.

D. Discussion

A comparison of PERFECT with alternative approaches highlights its superior timely response. On average, PERFECT produces results 88% faster than the optimal approach. This speed advantage is supported by its complexity analysis: $\mathbb{T}\big((\mathbb{U}+\mathbb{R})\mathbb{A}\big)+(\mathbb{N}_{\text{UAV}}\mathbb{A})+(\mathbb{N}\mathbb{U}\mathbb{C}\mathfrak{T}_t)+(\mathbb{N}\mathbb{F}_s+\mathbb{P}\mathbb{U})\big)$. This complexity arises from (i) evaluating $\mathbb{U}$ user trajectories and $\mathbb{R}$ requests over areas (Prediction); (ii) evaluating each UAV in each area (TP); (iii) assessing nodes, UEs, channels, and RBs (MAC); and (iv) analyzing nodes and functions for function deployment, and determining optimal paths for requests (PL). The increased number of nodes and requests affects the runtime slightly, but this marginal cost is offset by performance gains in acceptance, latency, and energy, as demonstrated in Fig. 7. Notably, PERFECT's complexity is dominated by lightweight online inference, while expensive offline training is amortized over long-term system usage. The HDRL training stage is computationally demanding due to iterative exploration, reward evaluation, and parameter updates within multiple simulated episodes; however, this cost is incurred once before deployment. In contrast, the online inference phase requires sub-second execution times on standard edge hardware. Hence, PERFECT achieves an effective balance between computational overhead and performance metrics, confirming scalability and HDRL design efficiency in next-generation vehicular networks.

To benchmark complexity, PERFECT was compared with SCA and HaDDQN. In SCA, the non-convex problem is decomposed into sequential convex subproblems, each solved via an interior-point method over all nodes, areas, users, and channels, yielding $\mathbb{T}(\mathbb{N}\mathbb{A}\mathbb{U}\mathbb{P}\mathbb{C}\mathfrak{T}_t)^{3.5}$, complexity. While SCA converges to a stationary point, its runtime scales poorly with network size and channel numbers. HaDDQN incurs a per-frame cost of $\mathbb{T}\big(((\mathbb{U}+\mathbb{R})\mathbb{P}\mathbb{C}\mathfrak{T}_t)^3+(\mathbb{N}_{\text{UAV}}\mathbb{A})\big)$, dominated by the cubic complexity of the assignment operations, which relies on the Hungarian algorithm for user-resource and UAV-area association, followed by a DDQN-based decision. Thus, PERFECT maintains lower practical complexity and faster execution. As shown in Fig. 8.a, PERFECT consistently surpasses SCA and HaDDQN in the oracle's objective, owing to its prediction-aware decisions, Bayesian channel adaptation, and hierarchical coordination. Each radar in this figure is generated by first normalizing (min-max) the averaged values of accepted requests, energy consumption, and E2E latency collected across multiple scenarios at different densities, and then plotting the aggregated triplet for each algorithm to visualize its overall performance balance in each scenario.

Fig. 8.a) presents heat maps of accepted requests and energy consumption, highlighting balanced behavior across different metrics. PERFECT achieves over 92% of ALLOCATE's optimal rates for varying requests, significantly outperforming the random approach–which drops to 24% at higher request levels. In the network scaling scenario, PERFECT's performance rates rise from 51% to 99%, achieving 93% of ALLOCATE's performance. Similarly, for channel variations, PERFECT excels with a 92% of ALLOCATE, outperforming SCA and HaDDQN, which plateau at 71-74%. Besides, PERFECT's performance metrics remain stable when the number of users, networks, and channels increases, as depicted in Fig. 8.b). Each heatmap is constructed by computing the ratio of total accepted requests to total energy consumption based on averaged simulation statistics for every method that provides a compact representation of the objective trade-off. The illustrations of

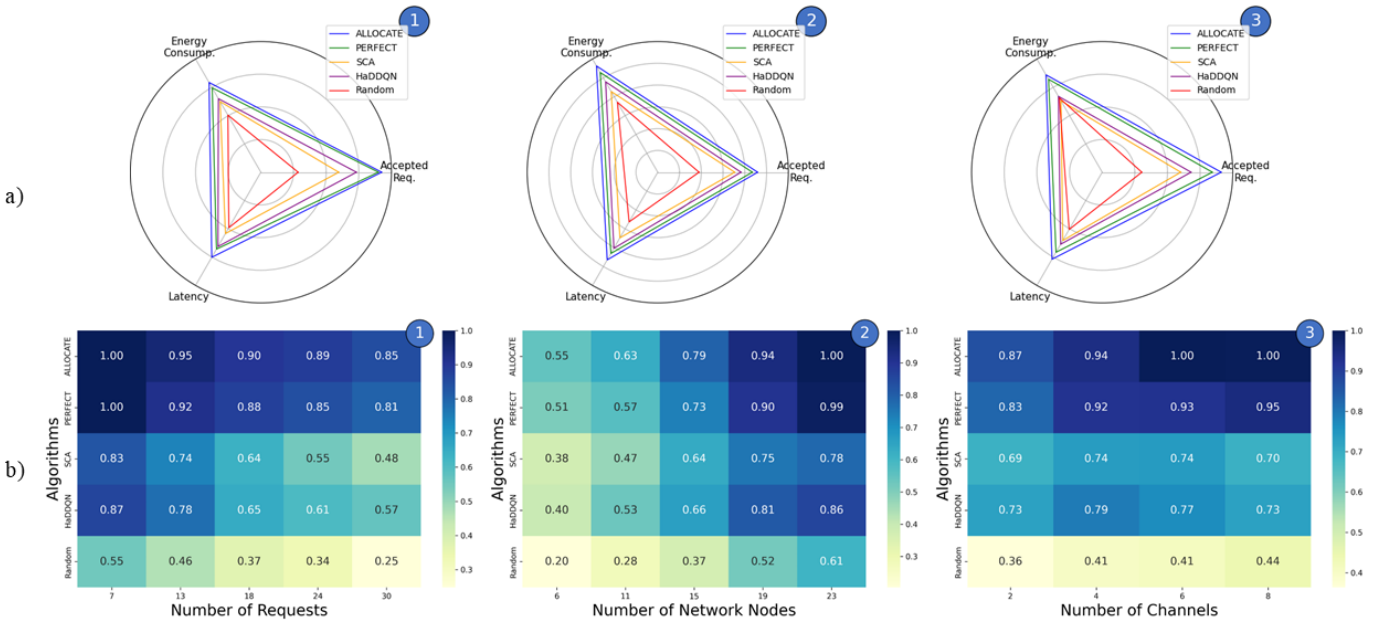

Figure 8. Performance metrics comparison of the baseline methods as (1) request set, (2) network size, and (3) channel size expand. In (a), radar plots present the normalized trade-offs among accepted requests, latency, and energy consumption, demonstrating how each algorithm behaves under varying scenarios. In (b), the heat maps depict the ratio of accepted requests to energy consumption (as objectives) under the same conditions.

the PERFECT evaluations underscore the proposed method's scalability and ability to (i) balance competing objectives effectively, (ii) sustain efficient operation, and (iii) effectively adapt to varying network conditions.

VII. Conclusion

In this study, we proposed a comprehensive framework for composed service orchestration in 6G aerial-terrestrial networks, addressing the intertwined challenges of mobility, resource planning, and service coverage. Our orchestration approach ensures that we are able to meet the diverse requirements of modern vehicular applications, resulting in efficient resource allocation and high-quality service provisioning. An MINLP problem of service orchestration in an integrated aerial-terrestrial network, while accounting for capacity constraints, changing user behavior, and E2E latencies, was first formulated to maximize service coverage while optimizing energy consumption. To solve the NP-hard problem, the integration of HDRL with predictive modeling enabled efficient UAV trajectory planning and resource-efficient service placement, ensuring QoS compliance and enhanced system performance. Our simulation results highlighted significant improvements in request acceptance, energy efficiency, and latency minimization, outperforming traditional and state-of-the-art methods. This framework underscores the transformative potential of HDRL-driven solutions for managing the complexity and scalability of next-generation vehicular networks.

Future works focus on further enhancing the scalability of the proposed method. Promising directions include integrating federated learning to enable decentralized training and decision-making, and exploring multi-agent RL to improve coordination among UAVs and network nodes in heterogeneous, multi-tiered environments involving satellite, optical wireless, and radio frequency communication nodes. Also, we plan to explore opportunities for using Large Language Models [56] for high-level decisions to assist in reasoning about resource allocation challenges in emerging quantum internet settings, such as qubit routing and link-level scheduling [57].

Acknowledgment

The research work is supported in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2411; and in part by the European Union's Horizon Europe research and innovation programme under the 6G-Path project (Grant No. 101139172).

References

[1] Z. Chen and X. Wang, "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, p. 188, 2020.

[2] I. Lee and D. K. Kim, "Decentralized multi-agent DQN-based resource allocation for heterogeneous traffic in V2X communications," *IEEE Access*, vol. 12, pp. 3070–3084, 2024.

[3] M. Shokrnezhad *et al.*, "Semantic revolution from communications to orchestration for 6G: Challenges, enablers, and research directions," *IEEE Netw.*, vol. 38, no. 6, pp. 63–71, 2024.

[4] V. S. Hapanchak, A. Costa, J. Pereira, and M. J. Nicolau, "An intelligent path management in heterogeneous vehicular networks," *Veh. Commun.*, vol. 45, p. 100690, 2024.

[5] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary V2X technologies toward the internet of vehicles: Challenges and opportunities," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 308–323, 2020.

[6] S. Wright, "Autonomous cars generate more than 300 tb of data per year," Tech Blog, Tuxera, 2021. [Online]. Available: https://www.tuxera.com/blog/autonomous-cars-300-tb-of-data-per-year/

[7] X. Li *et al.*, "Federated multi-agent deep reinforcement learning for resource allocation of Vehicle-to-Vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 8, pp. 8810–8824, 2022.

[8] M. N. Avcil, M. Soyturk, and B. Kantarci, "Fair and efficient resource allocation via vehicle-edge cooperation in 5G-V2X networks," *Veh. Commun.*, vol. 48, p. 100773, 2024.

[9] Q. Wu *et al.*, "Mobility-aware cooperative caching in vehicular edge computing based on asynchronous federated and deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 17, no. 1, pp. 66–81, 2023.

[10] F. Busacca, C. Grasso, S. Palazzo, and G. Schembra, "A smart road side unit in a microeolic box to provide edge computing for vehicular applications," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 194–210, 2023.

[11] M. Shokrnezhad *et al.*, "Toward a dynamic future with adaptable computing and network convergence (ACNC)," *IEEE Netw.*, vol. 39, no. 2, pp. 268–277, 2025.

[12] C. R. Storck and F. Duarte-Figueiredo, "A survey of 5G technology evolution, standards, and infrastructure associated with Vehicle-to-Everything communications by internet of vehicles," *IEEE Access*, vol. 8, pp. 117 593–117 614, 2020.

[13] H. Mazandarani, M. Shokrnezhad, and T. Taleb, "Semantic-aware dynamic and distributed power allocation: a multi-UAV area coverage use case," 2025.

[14] F. Zhou, R. Q. Hu, Z. Li, and Y. Wang, "Mobile edge computing in unmanned aerial vehicle networks," *IEEE Trans. Wireless Commun.*, vol. 27, no. 1, pp. 140–146, 2020.

[15] Z. Ning *et al.*, "Multi-agent deep reinforcement learning based UAV trajectory optimization for differentiated services," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5818–5834, 2024.

[16] M. Z. Alam and A. Jamalipour, "Multi-agent DRL-based hungarian algorithm (MADRLHA) for task offloading in multi-access edge computing internet of vehicles (IoVs)," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7641–7652, 2022.

[17] H. Mazandarani, M. Shokrnezhad, and T. Taleb, "A novel multiple access scheme for heterogeneous wireless communications using symmetry-aware continual deep reinforcement learning," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 3, pp. 353–368, 2025.

[18] H. Mazandarani *et al.*, "A semantic-aware multiple access scheme for distributed, dynamic 6G-based applications," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2024, pp. 1–6.

[19] N. I. Sarkar and S. Gul, "Artificial intelligence-based autonomous UAV networks: A survey," *Drones*, vol. 7, no. 5, p. 322, 2023.

[20] X. Wei *et al.*, "Joint UAV trajectory planning, DAG task scheduling, and service function deployment based on DRL in UAV-empowered edge computing," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12 826–12 838, 2023.

[21] Z. Md. Fadlullah and N. Kato, "HCP: Heterogeneous computing platform for federated learning based collaborative content caching towards 6G networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 1, pp. 112–123, 2022.

[22] X. Liu, Y. Liu, and Y. Chen, "Reinforcement learning in multiple-UAV networks: Deployment and movement design," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, 2019.

[23] H. Santos *et al.*, "A mobility-aware flying edge computing service orchestration with quality of service support," in *Proc. IEEE World Forum on Internet of Things (WF-IoT)*, 2023, pp. 01–06.

[24] A. Nabi, T. Baidya, and S. Moh, "Comprehensive survey on reinforcement learning-based task offloading techniques in aerial edge computing," *Internet of Things*, p. 101342, 2024.

[25] S. Han *et al.*, "DRL-assisted energy minimization for NOMA-based dynamic multi-user multi-access MEC networks," *IEEE Internet Things J.*, 2024.

[26] F. Pervez, L. Zhao, and C. Yang, "Joint user association, power optimization and trajectory control in an integrated satellite-aerial-terrestrial network," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3279–3290, 2022.

[27] P. Qin *et al.*, "Joint trajectory plan and resource allocation for UAV-enabled C-NOMA in air-ground integrated 6G heterogeneous network," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3421–3434, 2023.

[28] J. Gao, Z. Kuang, J. Gao, and L. Zhao, "Joint offloading scheduling and resource allocation in vehicular edge computing: A two layer solution," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3999–4009, 2023.

[29] C. Huang, G. Chen, P. Xiao, Y. Xiao, Z. Han, and J. A. Chambers, "Joint offloading and resource allocation for hybrid cloud and edge computing in SAGINs: A decision assisted hybrid action space deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, 2024.

[30] W. Qi, Q. Song, L. Guo, and A. Jamalipour, "Energy-efficient resource allocation for UAV-assisted vehicular networks with spectrum sharing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7691–7702, 2022.

[31] Q. He and J. Liang, "Online joint optimization of virtual network function deployment and trajectory planning for virtualized service provision in multiple-unmanned-aerial-vehicle mobile-edge networks," *Electronics*, vol. 13, no. 5, p. 938, 2024.

[32] B. Li, R. Yang, L. Liu, and C. Wu, "Service placement and trajectory design for heterogeneous tasks in multi-UAV edge computing networks," *IEEE Internet Things J.*, 2024.

[33] N. Gupta, S. Agarwal, D. Mishra, and B. Kumbhani, "Trajectory and resource allocation for UAV replacement to provide uninterrupted service," *IEEE Trans. Commun.*, 2023.

[34] P. Qin, J. Li, J. Zhang, and Y. Fu, "Joint task allocation and trajectory optimization for multi-UAV collaborative air-ground edge computing," *IEEE Trans. Netw. Sci. Eng.*, 2024.

[35] K. Li, W. Ni, X. Yuan, A. Noor, and A. Jamalipour, "Exploring graph neural networks for joint cruise control and task offloading in UAV-enabled mobile edge computing," in *Proc. IEEE Veh. Technol. Conf.*, 2023, pp. 1–6.

[36] D. Clément *et al.*, "Energy efficiency relaying election mechanism for 5G Internet of Things: A deep reinforcement learning technique," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2024, pp. 1–6.

[37] F. Li *et al.*, "Multi-UAV hierarchical intelligent traffic offloading network optimization based on deep federated learning," *IEEE Internet Things J.*, 2024.

[38] Z. Chen, F. Wang, and J. Wang, "Joint optimization for service-caching, computation-offloading, and UAVs flight trajectories over rechargeable UAV-aided MEC using hierarchical multi-agent deep reinforcement learning," *Veh. Commun.*, vol. 50, p. 100844, 2024.

[39] Z. Lin *et al.*, "Hybridrdn: delay-optimal computation offloading for autonomous vehicle fleets based on rsma," *IEEE Trans. Mobile Comput.*, vol. 24, no. 11, pp. 12 456–12 470, 2025.

[40] ——, "Sma-assisted distributed computation offloading in vehicular networks based on stochastic geometry," *IEEE Trans. Veh. Technol.*, vol. 74, no. 6, pp. 10 047–10 051, 2025.

[41] M. Farhoudi, M. Shokrnezhad, T. Taleb, R. Li, and J. Song, "Discovery of 6G services and resources in edge-cloud-continuum," *IEEE Netw.*, vol. 39, no. 3, pp. 223–232, 2024.

[42] M. Farhoudi, M. Shokrnezhad, S. Kianpisheh, and T. Taleb, "Deep learning based service composition in integrated aerial-terrestrial networks," in *International Conf. on Net. Softwarization*, 2025, pp. 204–208.

[43] M. Farhoudi, M. Shokrnezhad, and T. Taleb, "QoS-aware service prediction and orchestration in cloud-network integrated beyond 5G," in *Proc. IEEE Global Telecommun. Conf.*, 2023, pp. 369–374.

[44] Y. Guo, C. You, C. Yin, and R. Zhang, "UAV trajectory and communication co-design: Flexible path discretization and path compression," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3506–3523, 2021.

[45] Oubbati, Omar Sami *et al.*, "A UAV-UGV cooperative system: Patrolling and energy management for urban monitoring," *IEEE Trans. Veh. Technol.*, vol. 74, no. 9, pp. 13 521–13 536, 2025.

[46] Jamal Alotaibi *et al.*, "Optimizing disaster response with UAV-mounted RIS and HAP-enabled edge computing in 6G networks," *Journal of Network and Computer Applications*, vol. 241, pp. 104–213, 2025.

[47] 3rd Generation Partnership Project (3GPP), "Study on Channel Model for Frequencies from 0.5 to 100GHz," ETSI, Tech. Rep. TR 38.901 V16.1.0, Nov 2020.

[48] F. Faticanti *et al.*, "Cutting throughput with the edge: App-aware placement in fog computing," in *IEEE International Conf. on Cyber Security and Cloud Comput.*, 2019, pp. 196–203.

[49] G. Pataki, M. Tural, and E. B. Wong, "Basis reduction and the complexity of branch-and-bound," in *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, ser. Proceedings. Society for Industrial and Applied Mathematics, Jan. 2010, pp. 1254–1261.

[50] H. v. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016.

[51] Z. Wang *et al.*, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 1995–2003.

[52] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *International Conference on Intell. Transp. Syst. (ITSC)*, 2018, pp. 2575–2582.

[53] D. Oladimeji, K. Gupta, N. A. Kose, K. Gundogan, L. Ge, and F. Liang, "Smart transportation: An overview of technologies and applications," *Sensors*, vol. 23, no. 8, 2023.

[54] T. Taleb *et al.*, "6G system architecture: A service of services vision," *ITU J. on future and Evol. Technol.*, vol. 3, no. 3, pp. 710–743, 2022.

[55] K. Deng, Z. He, H. Lin, H. Zhang, and D. Wang, "A novel channel-constrained model for 6G vehicular networks with traffic spikes," in *Proc. IEEE Wireless Commun. and Networking Conf.*, 2024, pp. 1–6.

[56] M. Shokrnezhad and T. Taleb, "An autonomous network orchestration framework integrating large language models with continual reinforcement learning," *IEEE Commun. Mag.*, vol. 63, no. 8, pp. 78–84, 2025.

[57] J. Prados-Garzon *et al.*, "Deterministic 6GB-assisted quantum networks with slicing support: A new 6GB use case," *IEEE Netw.*, vol. 38, no. 1, pp. 87–95, 2024.