Machine Learning-Based Resource Allocation in 6G Integrated Space and Terrestrial Networks-Aided Intelligent Autonomous Transportation

Sasinda C. Prabhashana, *Student Member, IEEE*, Dang Van Huynh, *Member, IEEE*,
Keshav Singh, *Member, IEEE*, Hans-Jürgen Zepernick, *Senior Member, IEEE*,
Octavia A. Dobre, *Fellow, IEEE*, Hyundong Shin, *Fellow, IEEE*,
and Trung Q. Duong, *Fellow, IEEE*

Abstract—**The integration of terrestrial and non-terrestrial networks with mobile edge computing (MEC) and orbital edge computing (OEC) technologies is essential for advancing 6G communication networks. This paper introduces a network architecture that combines terrestrial and non-terrestrial networks by integrating drones (also known as UAV)-carried reconfigurable intelligent surfaces (RIS) and satellite-based MEC to optimize resource allocation in intelligent autonomous transportation systems (IATS). The primary objective is to minimize total system utility costs through the optimal allocation of bandwidth, computational power at the base station and low Earth orbit (LEO) satellite, and offloading decisions, all while adhering to strict performance and delay constraints. We address the complex resource optimization challenge by formulating a nonlinear programming (NLP) problem. To solve this problem, we employ long short-term memory (LSTM)-enhanced deep deterministic policy gradient (DDPG) and LSTM-enhanced twin delayed deep deterministic policy gradient (TD3) algorithms, which enable dynamic and adaptive resource management. These LSTM-enhanced algorithms improve convergence speed by 44.44% and 73.81%, respectively, compared to their conventional counterparts, while significantly enhancing cost efficiency. Our simulation results demonstrate substantial improvements in system performance, with effective resource allocation and minimal utility costs, providing a robust solution for ensuring high-quality, low-latency communication in diverse 6G IATS environments.**

Index Terms—**6G networks, deep reinforcement learning, mobile edge computing, orbital edge computing, intelligent autonomous transportation systems.**

I. Introduction

The sixth-generation (6G) networks, expected to be deployed by 2030, are set to revolutionize global connectivity with comprehensive coverage, enhanced spectral efficiency, higher data transmission rates, and reduced energy consumption and latency [1]. A major innovation in 6G is the integration of artificial intelligence, which will enable smarter, more efficient handling of the large amounts of data and devices in communication networks [2]. Furthermore, 6G will seamlessly merge terrestrial and non-terrestrial networks, such as satellites and drones (also known as UAV), to provide fully utilized coverage [3]. Currently, the fifth-generation (5G) networks cover only a small fraction of the world's land area and an even smaller portion of the Earth's surface, revealing significant limitations [4]. To overcome these challenges, both novel and existing technologies must be enhanced to meet the growing demands.

Mobile edge computing (MEC) is one such technology, offering substantial computational resources at the network edge, close to end users. This proximity helps minimize energy consumption in mobile devices, extend battery life, and maintain low latency by offloading computationally intensive tasks to high-performance edge servers [5]. Moreover, the integration of terrestrial and non-terrestrial networks has led to a paradigm shift in edge-computing-enabled communication services. Terrestrial edge computing is transitioning to non-terrestrial and orbital-edge computing (OEC), finding widespread applications in remote areas [6]. These integrated networks provide ubiquitous connectivity, supporting diverse services such as remote area monitoring, high-speed Internet access, and disaster relief, while operating independently [7]. In disaster scenarios, where terrestrial communication infrastructure may be compromised, UAVs can

The work of Keshav Singh was supported by the National Science and Technology Council of Taiwan under Grant NSTC 112-2221-E-110-038-MY3 and Grant NSTC 113-2218-E-110-008. The work of Octavia A. Dobre was supported by Canada Research Chairs Program under Grant CRC-2022-00187. The work of Hyundong Shin was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) under Grant RS-2025-00556064. The work of Trung Q. Duong was supported by Canada Excellence Research Chairs (CERC) Program under Grant CERC-2022-00109. An earlier version of this paper was presented in part at the IEEE International Conference on Communications (ICC), Montreal, Canada, in June 2025.

Sasinda C. Prabhashana, Dang Van Huynh, and Octavia A. Dobre are with Memorial University, St. John's, NL A1C 5S7, Canada (e-mail: cwelhengodag@mun.ca; vdhuynh@mun.ca; odobre@mun.ca).

Keshav Singh is with National Sun Yat-sen University, Kaohsiung 804, Taiwan (e-mail: keshav.singh@mail.nsysu.edu.tw).

Hans-Jürgen Zepernick is with Blekinge Institute of Technology, 37179 Karlskrona, Sweden (e-mail: hans-jurgen.zepernick@bth.se).

Hyundong Shin is with Kyung Hee University, Republic of Korea (e-mail: hshin@khu.ac.kr).

Trung Q. Duong is with Memorial University, St. John's, NL A1C 5S7, Canada, also with Queen's University Belfast, BT7 1NN Belfast, U.K., and also with Kyung Hee University, South Korea (e-mail: tduong@mun.ca).

reestablish connections between users and the nearest communication systems [8]. Their high altitudes enable line-of-sight (LoS) communication with ground base stations, reducing issues of shadowing and signal blockage. Additionally, their maneuverability allows for real-time repositioning to meet dynamic communication needs, acting as aerial relays between transmitters and receivers [9], [10]. Reconfigurable intelligent surfaces (RISs) are also revolutionizing future communication systems. These arrays of controllable elements precisely adjust signal phase, enhancing communication efficiency [11]. However, most existing RIS implementations are fixed in positions such as on walls or roofs, which creates problems when obstacles block these surfaces, leading to reduced system performance [12], [13].

By integrating RIS with UAVs, communication system performance can be significantly improved. This leverages UAV mobility and RIS properties to enhance communication in obstructed environments, such as urban areas or disaster zones, by dynamically establishing LoS links. These systems mitigate interference and optimize phase shifts to improve coverage and spectral efficiency, making them effective for dense networks. Additionally, active RIS on UAVs amplifies signals, reducing fading effects and improving security-reliability trade offs in cooperative networks [14], [15]. Furthermore, LEO satellites play a pivotal role in enabling seamless global connectivity for next-generation 6G networks, particularly in remote and underserved areas. Their proximity to Earth ensures low-latency, high-speed communication, making them ideal for supporting real-time and computation-intensive applications. By integrating LEO satellites with UAV-carried RIS, these systems can dynamically enhance signal propagation and improve coverage in challenging environments [4], [6], [16].

To fully leverage MEC in these integrated networks, various approaches have been proposed to optimize resource usage. One approach involves making binary decisions on task offloading [17], [18], which reduces edge server's idle time and ensures timely responses to user requests. However, rapid changes in system parameters create a high demand for fast offloading decision-making and resource allocation. As the number of users and tasks increases, conventional and heuristic task offloading techniques struggle to execute decisions efficiently and solve complex computation problems [19]. Although traditional approaches can achieve stable management and scheduling decisions, large-scale MEC networks experience higher delays, making them impractical for real-world applications [20].

Deep reinforcement learning (DRL), a sub-field of machine learning that integrates reinforcement learning with deep neural networks, has proven effective for system optimization and real-time decision-making in wireless networks [21], [22]. Specifically, DRL agents can solve complex problems in dynamic and stochastic environments with large state spaces by accurately learning the optimal policy and long-term rewards without prior knowledge of the system. Recent studies in the realm of MEC task offloading have increasingly leveraged DRL to enhance decision-making processes, demonstrating substantial promise [23]. The integration of existing technologies with DRL not only optimizes resource allocation across the network but also significantly improves the efficiency of computational task distribution. This optimization is crucial as it addresses the growing demands on network resources by intelligently managing the where and when of task offloading. Such advancements in DRL algorithms warrant further exploration to fully harness their potential in complex network environments, ensuring optimal performance and resource utilization in real-time scenarios.

A. Related Works

Recently, MEC has emerged as a transformative force in communication systems, providing critical computational resources directly to edge users. However, the management of MEC systems in wireless networks remains a crucial challenge, prompting researchers to propose various computational algorithms [5], [6], [12], [24], [25], [26], [27]. In [24], an energy-efficient resource allocation and task offloading approach for multi-UAV-assisted edge computing systems was proposed, introducing a block successive upper-bound minimization algorithm to minimize the total energy consumption of mobile devices and UAVs. Similarly, the proposed framework in [25], employing a matching-optimization method to minimize execution latency, enhance resource utilization, and ensure efficient bandwidth allocation in device-to-device enabled MEC networks. Furthermore, a heuristic algorithm jointly optimized task offloading and scheduling for a multi-user, multi-server MEC system introduced in [26]. More recently, an innovative task offloading approach for mission-critical applications using UAVs as mobile edge servers was proposed in [27], introducing a low-complexity algorithm to minimize latency by optimizing bandwidth allocation and task offloading probability, while ensuring quality of service and energy efficiency for users.

However, due to the dynamic and complex nature of communication systems, the research on MEC optimization has shifted towards DRL-based approaches [7], [11], [17], [18], [19], [20], [21], [22], [23], [28], [29]. In [21] and [28], the primary focus was on minimizing computation delay in MEC networks. These approaches leveraged the capabilities of DRL to optimize task offloading decisions, which is crucial for latency-sensitive applications. However, the strong emphasis on delay minimization often comes at the expense of energy efficiency, a critical consideration in resource-constrained environments, particularly for Internet of Things (IoT) and mobile devices. In contrast, the work in [29] emphasized the energy efficiency, employing a multi-agent DRL algorithm to reduce energy consumption while ensuring timely task execution. When considering the scalability and computational complexity of MEC systems, DRL can efficiently handle these challenges. Moreover, the scalability of these MEC systems can be improved using DRL with dynamic network conditions and varying task requirements [22], [30]. Specifically, partial task offloading was the focus in [22], permitting more flexible resource allocation that could enhance scalability in larger networks. Furthermore, the approach was extended by [30], which introduced a distributed task offloading framework that distributed the load between edge and cloud resources, thus increasing the system's ability to manage a greater number of tasks and devices.

In recent years, several studies have explored the use of deep deterministic policy gradient (DDPG) and its variants, such as

TD3, to address resource allocation and task offloading challenges in MEC environments [4], [11], [21], [22], [28], [31], [32], [33], [34], [35]. In [32], DDPG was applied to multiuser industrial Internet of Things (IIoT) edge computing networks, focusing on intelligent delay-aware partial task offloading and dynamic resource allocation. The study demonstrated that DDPG could significantly reduce system delay and energy consumption, outperforming traditional methods in managing varying channel conditions, diverse user requirements, and real-time decision-making in dynamic, resource-constrained environments. In [33], TD3 was utilized to optimize the allocation of computational resources and the configuration of RIS-enabled MEC systems, which highlighted the algorithm's effectiveness in managing complex interactions between RIS and MEC servers. On the other hand, an improved centralized dual-actor DDPG algorithm was proposed in [31] to jointly manage long-term service caching and short-term task offloading, computing, and bandwidth resource allocation in multi-access edge computing networks, which effectively minimized system delay and cache costs while enhancing overall network efficiency and ensuring more stable and faster convergence compared to traditional methods. Moreover, the integration of long short-term memory (LSTM) networks with the DDPG algorithm to enhance UAV-assisted MEC systems was adeptly executed in [34] and [35]. Specifically, in [34], LSTM was utilized to predict service content demands from users, enabling the dynamic optimization of UAV trajectories, caching strategies, and task offloading by DDPG.

The integration of terrestrial and non-terrestrial networks, particularly with MEC systems, can be extended towards OEC [4], [6], [16], [36], [37], [38]. In [16], the integration of MEC into satellite networks was explored as a means to significantly enhance the performance of 6G IoT applications, particularly in challenging environments where terrestrial networks are inadequate. By placing MEC servers on satellites or related infrastructure, this integration aims to reduce latency and improve data processing capabilities, addressing the limitations of traditional satellite communications. Moreover, complex challenges of multi-task offloading and resource allocation in mobile edge computing systems within the satellite-IoT context was addressed in [36]. Furthermore, energy-aware task offloading and resource allocation challenges within intelligent LEO satellite networks was analyzed in [37]. Authors proposed a joint task offloading and resource allocation strategy which designed to optimize satellite energy consumption while meeting task delay requirements. Additionally, the deployment of MEC servers on LEO satellites has significantly enhanced the potential for providing computational services to remote areas, with DRL emerging as a key enabler of this advancement. It was demonstrated that DRL could effectively optimize task offloading and resource allocation in LEO satellite networks, resulting in substantial reductions in latency and energy consumption in [38]. This study showcased DRL's power in managing complex, multi-user scenarios, making it a critical component in advancing OEC.

Moreover, DRL has emerged as a key enabler in addressing the complexities of vehicular networks and intelligent autonomous transportation systems (IATS), particularly in optimizing resource allocation, enhancing decision-making, and ensuring real-time adaptability. As vehicular environments grow increasingly dynamic with the advent of 6G networks, DRL offers a versatile framework to support both efficiency and scalability [18], [39], [40], [41], [42], [43], [44], [45], [46], [47]. In vehicular edge computing (VEC), DRL facilitates efficient task offloading, as demonstrated in [43], where shared task processing reduced energy consumption and improved system response times. Similarly, advanced DRL algorithms such as SAC and TD3 were employed in [41] to allocate resources effectively in dynamic internet of vehicles networks, ensuring scalability and task prioritization. In cloud-edge cooperation, [42] proposed a DRL-driven framework that optimizes task offloading and resource allocation while reducing redundant content delivery in vehicular networks. Additionally, [39] utilized DRL to enhance channel allocation and task processing in UAV-assisted VEC, improving connectivity in remote or congested areas. For autonomous vehicles (AVs), [44] developed a transferable DRL framework for joint radar-data communication, achieving significant adaptability and enhanced detection accuracy under dynamic conditions. In addressing critical safety concerns, [45] proposed an edge learning-aided offloading framework to optimize inference accuracy under strict latency constraints. In urban environments, [40] introduced the MESON framework, a mobility-aware DRL scheme that prioritizes tasks with dependencies, reducing response times and enhancing system stability in high-density traffic. A broader perspective in [46] surveyed the integration of DRL with edge intelligence in IATS, emphasizing its potential to reduce latency, enhance privacy, and enable scalable analytics. In addition, [47] presented a DRL-based framework to optimize communication and computing resources, improving traffic throughput and safety in connected autonomous driving systems. Collectively, these studies demonstrate DRL's pivotal role in adapting to network dynamics, leading to notable gains in energy efficiency, task execution speed, and overall system performance.

B. *Motivation and Contributions*

The future of IATS will fully leverage advanced communication technologies, with 6G-integrated space and terrestrial networks playing a crucial role. As illustrated in Fig. 1, autonomous vehicles can communicate with each other through various wireless links, including connections from other vehicles, base stations, and UAV-carried RIS (UCR). This ubiquitous connectivity is enabled by integrated satellite-UAV networks, which provide a resilient communications infrastructure for vehicles in urban areas as well as trains on railways. Inspired by the envisioned future of IATS and the aforementioned related works, our study focuses on resource optimization within a MEC and OEC-enabled integrated terrestrial and non-terrestrial network. We tackle this challenge by formulating a nonlinear programming (NLP) problem aimed at achieving optimal resource allocation. Specifically, our approach includes the allocation of bandwidth to users, the distribution of computational resources, and the optimization of offloading decisions based on the overall system utility cost. We employ advanced DRL techniques, specifically DDPG and TD3, each enhanced with an LSTM layer in the actor network. These enhancements enable DDPG and TD3 to provide a robust and adaptive solution that dynamically adjusts

Fig. 1. An illustration of 6G integrated space and terrestrial networks-aided intelligent autonomous transportation systems.

resource allocation and offloading decisions in real time. To the best of our knowledge, our proposed network architecture uniquely employs LSTM-enhanced DDPG and TD3 for optimizing resource allocation to minimize system costs within an integrated terrestrial and non-terrestrial network. The major contributions of our work can be summarized as follows:

- We propose a MEC and OEC-enabled integrated terrestrial and non-terrestrial network architecture designed to minimize total system costs by optimally allocating bandwidth to users, assigning computational resources, and determining offloading fractions.
- We formulate a joint computation offloading and resource allocation problem as a nonlinear programming (NLP) problem, constrained by the available resources at the BS and LEO satellites and the maximum tolerable delay for each heterogeneous task generated by the users.
- We employ LSTM-enhanced DDPG and TD3 algorithms to solve the NLP problem by optimally allocating resources and minimizing system costs within an integrated terrestrial and non-terrestrial network.
- Our simulation results show significant improvements in system performance and cost efficiency using LSTM-enhanced DDPG and TD3 when comparing with conventional DDPG and TD3, underscoring the efficacy of our proposed architecture and optimization strategies within the integrated network.

C. Paper Structure and Notations

The rest of this paper is organized as follows. Section II presents the system model and problem formulation, including the channel model, task model, and the formulation of the addressed optimization problem. Section III discusses the DRL-based solution, including definition of key elements of LSTM-enhanced DDPG and TD3. Numerical results and discussions are provided in Section IV. Finally, Section V concludes the paper by providing promising directions for future works.

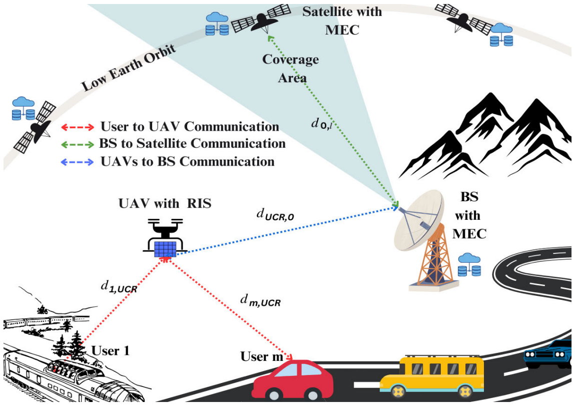

Fig. 2. System model of 6G integrated terrestrial and non-terrestrial networks with mobile and orbital edge computing for intelligent autonomous transportation systems.

Notations: Throughout this paper, lowercase letters represent scalars, while bold uppercase and lowercase letters denote matrices and vectors, respectively. The notation $\mathbf{x} \sim \mathcal{CN}(\cdot,\cdot)$ indicates that $\mathbf{x}$ follows a complex circularly symmetric Gaussian distribution. The symbol $|\cdot|$ represents the Euclidean norm of a vector, and $\mathbb{C}$ denotes the set of complex numbers. We use $x_{m,r}(t)$ to refer to a variable x associated with the m-th transmitter and r-th receiver at time t.

II. System Model and Problem Formulation

In this paper, we consider the system architecture for a task offloading strategy aimed at accommodating the resource allocation requirements of end-users through an integrated terrestrial and non-terrestrial network framework supporting IATS, as illustrated in Fig. 2. This model includes a set of M users denoted by $\mathcal{M} = \{1,\ldots,m,\ldots,M\}$, which are registered with a base station (BS). To overcome the limitations posed by non-line-of-sight (NLoS) communication, these users utilize an UAV equipped with a passive RIS with N passive reflecting elements for facilitating make a LoS signal reflection towards the BS. The process of signal reflection is mathematically expressed through the diagonal matrix $\boldsymbol{\phi}$, where $\boldsymbol{\phi} = diag(e^{j\phi_1}, e^{j\phi_2}, \ldots, e^{j\phi_N})$ represents the phase shifts induced by each reflecting element. The BS, equipped with K antennas, incorporates a MEC node to enhance edge computing services for users. Despite these provisions, the BS faces significant challenges in managing the overflow in task requests during peak times, primarily due to the rigorous latency demands of users. To mitigate these challenges, the BS hires $\mathcal{L} = \{1,\ldots,l,\ldots,L\}$ LEO satellites which are in the same circular orbit. Each satellite consists of a single antenna and is capable of delivering MEC services and guaranteeing the delivery of seamless and dependable services. Moreover, we assume that whenever offloading occurs, a LEO satellite is always in the coverage area and the total coverage time is sufficient to handle and execute the offloaded tasks.

A. Channel Modeling

1) User to BS via UAV-Carried RIS: In this study, we quantify the channel vector of the link between the m-th user and the UCR as $\mathbf{h}_{m,ucr}(t) \in \mathbb{C}^{N\times 1}$ and the channel matrix between the UCR and the BS as $\mathbf{H}_{ucr,0}(t) \in \mathbb{C}^{K\times N}$. We utilize the Rician fading model along with large-scale path loss for channel behavior analysis. Given the dynamic nature of UCR, the effects of the NLoS components are negligible. Consequently, this allows for a simplified expression of the channel gain vectors. Therefore, at time t, $\mathbf{h}_{m,ucr}(t)$ and $\mathbf{H}_{ucr,0}(t)$ can be expressed as in [11]:

$$\mathbf{h}_{m,ucr}(t) = \sqrt{\epsilon_0} d_{m,ucr}^{-\delta^{(1)}}(t) \left(\Psi_1^{\mathrm{LoS}} \mathbf{h}_{m,ucr}^{\mathrm{LoS}}(t)\right), \tag{1}$$

$$\mathbf{H}_{ucr,0}(t) = \sqrt{\epsilon_0} d_{ucr,0}^{-\delta^{(1)}}(t) \left(\Psi_1^{\mathrm{LoS}} \mathbf{H}_{ucr,0}^{\mathrm{LoS}}(t)\right), \tag{2}$$

where ϵ_0 represents the path loss at the reference distance. The terms $d_{m,ucr}(t)$ and $d_{ucr,0}(t)$ denote the distance between the m-th user and the UCR, and the distance between the UCR and the BS, respectively. The path loss exponent is given by $\delta^{(1)}$, and $\Psi_1^{\mathrm{LoS}} = \sqrt{\frac{\beta 1}{\beta_1+1}}$, where β_1 is the Rician fading factor. At time t, $\mathbf{h}_{m,u}^{LoS}(t) \in \mathbb{C}^{N\times 1}$ is calculated as $\mathbf{h}_{m,ucr}^{LoS}(t) = \left[1, e^{-j\frac{2\pi}{\lambda} d_u \cos(\phi_{\mathrm{AoA}}(t))}, \ldots, e^{-j\frac{2\pi}{\lambda}(N-1) d_u \cos(\phi_{\mathrm{AoA}}(t))}\right]^T$, where λ is the wavelength of the transmission signal, d_u is the uniform spacing between the RIS elements, and $\phi_{\mathrm{AoA}}(t)$ is the angle of arrival (AoA) at UCR. Furthermore, $\mathbf{H}_{u,bs}^{LoS}(t) \in \mathbb{C}^{K\times N}$ is given by $\mathbf{H}_{u,bs}^{LoS}(t) = \mathbf{a}_{bs}(\phi_{AoA}(t))\mathbf{a}_u^H(\phi_{AoD}(t))$. The steering vector for the BS, $\mathbf{a}_{bs}(\phi_{AoA}(t)) \in \mathbb{C}^{K\times 1}$, is calculated as: $\mathbf{a}_{bs}(\phi_{AoA}(t)) = \left[1, e^{-j\frac{2\pi}{\lambda} d_{bs} \cos(\phi_{AoA}(t))}, \ldots, e^{-j\frac{2\pi}{\lambda}(K-1) d_{bs} \cos(\phi_{AoA}(t))}\right]^T$, where d_{bs} is the spacing between the BS antennas and $\phi_{AoA}(t)$ is the AoA at BS. Similarly, the steering vector for the UCR, $\mathbf{a}_u(\phi_{AoD}(t)) \in \mathbb{C}^{N\times 1}$, represents the phase shifts introduced by the N elements of the RIS as the signal is reflected towards the BS. It is calculated as: $\mathbf{a}_u(\phi_{AoD}(t)) = \left[1, e^{-j\frac{2\pi}{\lambda} d_u \cos(\phi_{AoD}(t))}, \ldots, e^{-j\frac{2\pi}{\lambda}(N-1) d_u \cos(\phi_{AoD}(t))}\right]^T$ [11].

2) BS to LEO Satellite: The link between the BS and a LEO satellite is modeled as a ground-to-air channel, where we consider free space path loss as the path loss model. Therefore, the channel vector $\mathbf{h}_{k,l}(t) \in \mathbb{C}^{1\times K}$ between the k-th antenna and the l-th LEO satellite can be formulated as follows [10]:

$$\mathbf{h}_{k,l}(t) = \left(\frac{4\pi f_c d_{0,l}(t)}{c}\right)^{\frac{-\delta^{(2)}}{2}} \times \left(\Psi_2^{\mathrm{LoS}} \mathbf{h}_{k,l}^{\mathrm{LoS}}(t) + \Psi_2^{\mathrm{NLoS}} \mathbf{h}_{k,l}^{\mathrm{NLoS}}(t)\right). \tag{3}$$

In (3), the carrier frequency of the transmission signal is denoted by f_c, and c denotes the speed of light. The path loss exponent is given by $\delta^{(2)}$. The terms Ψ_2^{LoS} and Ψ_2^{NLoS} are defined as $\sqrt{\frac{\beta_2}{\beta_2+1}}$ and $\sqrt{\frac{1}{\beta_2+1}}$, respectively, where β_2 is the Rician factor for this link. The vectors $\mathbf{h}_{k,l}^{\mathrm{LoS}}(t)$ and $\mathbf{h}_{k,l}^{\mathrm{NLoS}}(t)$ represent the LoS and NLoS components, respectively [11]. The distance $d_{0,l}(t)$ between the BS and the l-th satellite varies relative to the BS [6] and can be calculated as

$$d_{0,l}(t) = \sqrt{R^2 + (R+r)^2 - 2R(R+r)\cos(\mu(t))}, \tag{4}$$

where R represents the Earth radius and r denotes the height from the BS to the LEO satellite, $\mu(t)$ is the geocentric angle, which can be formulated as $\mu(t) = \cos^{-1}\left(\frac{R}{R+r}\cos\alpha(t)\right) - \alpha(t)$ [6]. Here, $\alpha(t)$ is the elevation angle between the BS and the l-th LEO satellite.

B. Communication Model

Users are enabled to offload their computationally heavy tasks to the base station through the UCR. In BS, the instantaneous signal-to-interference-plus-noise ratio (SINR) for the m-th user can be expressed as [18]:

$$\Gamma_m^{\mathrm{bs}}(t) = \frac{p_m^{\mathrm{u}} \left|\mathbf{H}_{ucr,0}(t)\boldsymbol{\phi}(t)\mathbf{h}_{m,ucr}(t)\right|^2}{\sum_{j=1,j\neq m}^{M} p_j^{\mathrm{u}} \left|\mathbf{H}_{ucr,0}(t)\boldsymbol{\phi}(t)\mathbf{h}_{j,ucr}(t)\right|^2 + z^2(t)}, \tag{5}$$

where p_m^{u} represents the total transmit power, and $z(t)$ is the instantaneous noise power characterized by the Gaussian complex normal distribution $\sim \mathcal{CN}\left(0, \sigma^2\right)$. Therefore, the achievable data rate of the m-th user can be calculated as [18]:

$$R_m^{\mathrm{bs}}(t) = b(t) B \log_2\left(1 + \Gamma_m^{\mathrm{bs}}(t)\right), \tag{6}$$

where $b_m(t) \in [0, 1]$ is the allocated bandwidth coefficient of the m-th user, and B is the total system bandwidth. We assume that for offloading tasks to a LEO satellite, all K antennas jointly transmit the task. This approach is necessary due to the long distance to the LEO satellite, which requires more power to transmit the tasks. Therefore, the signal-to-noise ratio (SNR) at the l-th LEO satellite for an offloaded task of the m-th user can be formulated as [37]:

$$\Gamma^{\mathrm{l}}(t) = \frac{p_0 |\sum_{k=1}^{K} \mathbf{h}_{k,l}(t)|^2}{B N_0}, \tag{7}$$

where p_0 is the total transmit power of the BS, and N_0 is the single-sided noise spectral density. Therefore, the data rate at the l-th LEO satellite for an offloaded task of the m-th user can be expressed as [22]:

$$R_m^{\mathrm{l}}(t) = B \log_2\left(1 + \Gamma^{\mathrm{l}}(t)\right). \tag{8}$$

C. Task Offloading Model

We propose the following task offloading model to handle task overflow at the BS during peak times. Let the task from the m-th user be denoted as a 3-tuple $x_m = \{g_m, q_m, T_m^{\max}\}$, where g_m is the size of the task in bits, q_m represents the computational requirement, and $T_m^{\max}$ is the maximum threshold delay. The transmission delay $T_{m,tx}^{\mathrm{bs}}$ and transmission energy $E_{m,tx}^{\mathrm{bs}}$ of the m-th user can be formulated as [12]:

$$T_{m,tx}^{\mathrm{bs}} = \frac{g_m}{R_m^{\mathrm{bs}}(t)}, \quad E_{m,tx}^{\mathrm{bs}}(t) = p_m^{\mathrm{u}} T_{m,tx}^{\mathrm{bs}}. \tag{9}$$

Therefore, the total transmission utility cost $U_{m,tx}^{\mathrm{bs}}$ of the m-th user can be expressed as [18]:

$$U_{m,tx}^{\mathrm{bs}}(t) = \psi T_{m,tx}^{\mathrm{bs}} + (1-\psi) E_{m,tx}^{\mathrm{bs}}(t), \tag{10}$$

TABLE I

Summary of Key Notations

Notation	Definition	Notation	Definition
M	Number of users	N	Number of RIS elements
K	Number of BS antennas	R	Earth radius
r	Distance from BS to low Earth orbit	$d_{m,ucr}$	Distance from BS to low Earth orbit
$d_{ucr,0}$	Distance from user to UCR	$\delta^{(1)}$, $\delta^{(2)}$	Path loss exponents
F^{bs}	Computational power at the BS	F^{l}	Computational power at LEO satellite
p_m^{u}	Transmission power of m-th user	g_m	Task size
q_m	Task complexity	p_0	Total transmission power of BS to LEO satellite
B	System bandwidth	N_0	Noise power
$T_m^{\max}$	Maximum delay for user m	ω^{bs}, ω^{l}	Energy coefficients of the processor at BS and LEO satellite
f_c	Carrier frequency	ψ	Weighting factor between delay and energy
$\mathbf{h}_{m,ucr}(t)$	Channel vector between user m and UCR	$\mathbf{H}_{ucr,0}$	Channel vector between UCR and BS
$\mathbf{h}_{k,l}(t)$	Channel vector between BS and LEO satellite	β_1, β_2	Rician factor
$\mu(t)$	Geocentric angle	$d_{0,l}(t)$	Distance from BS to LEO satellite
c	Speed of light	$\alpha(t)$	Elevation angle between the BS and the LEO satellite
$T_{m,tx}^{\mathrm{bs}}$	Transmission delay from user m to BS	$E_{m,tx}^{\mathrm{bs}}$	Energy consumption due to transmission data from user m to BS
$U_{m,tx}^{\mathrm{bs}}$	Transmission utility cost of user m at BS	$T_{m,pr}^{\mathrm{bs}}$	Processing delay of task from user m at BS
$E_{m,pr}^{\mathrm{bs}}$	Energy for processing the task from user m at BS	$U_{m,pr}^{\mathrm{bs}}$	Processing utility cost for task from user m
$\eta_m^{(1)}$	Computational power coefficient at the BS	$\eta_m^{(2)}$	Computational power coefficient at LEO satellite.
$T_{m,tx}^{\mathrm{l}}$	Transmission delay from BS to LEO satellite	$E_{m,tx}^{\mathrm{l}}$	Energy consumption due to transmission from BS to LEO satellite
$U_{m,tx}^{\mathrm{l}}$	Transmission utility cost for BS to LEO satellite	$T_{m,pr}^{\mathrm{l}}$	Processing delay for the task at LEO satellite
$E_{m,pr}^{\mathrm{l}}$	Energy for processing the task at LEO satellite	$U_{m,pr}^{\mathrm{l}}$	Utility cost for processing the task at LEO satellite
$b_m(t)$	Bandwidth allocation for user m	$\chi_m(t)$	Task fraction for user m at BS

where $\psi \in [0, 1]$ is the weighting factor between delay and energy. When a task arrives at the BS, the processing time $T_{m,pr}^{\mathrm{bs}}$ at the BS can be denoted as [12]:

$$T_{m,pr}^{\mathrm{bs}} = \frac{q_m}{\eta_m^{(1)}(t) F^{\mathrm{bs}}}, \tag{11}$$

where $\eta_m^{(1)}(t) \in [0, 1]$ is the allocated computational power coefficient at the BS for the m-th user's task, and F^{bs} is the total computational power at the BS. Moreover, the energy consumption $E_{m,pr}^{\mathrm{bs}}$ at the BS for the m-th user's task can be formulated as [12]:

$$E_{m,pr}^{\mathrm{bs}}(t) = \omega^{\mathrm{bs}} q_m (\eta_m^{(1)}(t) F^{\mathrm{bs}})^2, \tag{12}$$

where ω^{bs} is the energy coefficient of the BS processor, which depends on the capacitance of the integrated chip. Consequently, the total utility cost $U_{m,pr}^{\mathrm{bs}}$ for processing the m-th user's task at the BS is expressed as

$$U_{m,pr}^{\mathrm{bs}}(t) = \psi T_{m,pr}^{\mathrm{bs}} + (1 - \psi) E_{m,pr}^{\mathrm{bs}}(t). \tag{13}$$

When offloading the entire tasks to the l-th LEO satellite without processing at the BS, the total transmission time $T_{m,tx}^{\mathrm{l}}$ and energy consumption $E_{m,tx}^{\mathrm{l}}$ of the m-th user's task can be denoted as

$$T_{m,tx}^{\mathrm{l}} = \frac{g_m}{R_m^l(t)}, E_{m,tx}^{\mathrm{l}}(t) = p_0 T_{m,tx}^{\mathrm{l}}. \tag{14}$$

Therefore, total utility cost $U_{m,tx}^{\mathrm{l}}$ for offloading the m-th user's task can be expressed as

$$U_{m,tx}^{\mathrm{l}}(t) = \psi T_{m,tx}^{\mathrm{l}} + (1 - \psi) E_{m,tx}^{\mathrm{l}}(t). \tag{15}$$

When the task from the m-th user arrives at the l-th LEO satellite for processing, the total processing time $T_{m,pr}^{\mathrm{l}}$ at the l-th LEO satellite can be expressed as

$$T_{m,pr}^{\mathrm{l}} = \frac{q_m}{\eta_m^{(2)}(t) F^{\mathrm{l}}}, \tag{16}$$

where $\eta_m^{(2)}(t) \in [0, 1]$ is the allocated computational power coefficient at l-th LEO satellite for the m-th user's task. F^{l} is the total computation power of the l-th LEO satellite. We consider that all the L satellites have the same computational powers. Moreover, the energy consumption for the execution of the m-th user's task at the l-th satellite can be expressed as

$$E_{m,pr}^{\mathrm{l}}(t) = \omega^{\mathrm{l}} q_m (\eta_m^{(2)}(t) F^{\mathrm{l}})^2, \tag{17}$$

where ω^{l} is the energy coefficient at the l-th LEO satellite processor, which depends on the capacitance of the integrated chip. We consider ω^{l} to be the same for the processors at each LEO satellite. Consequently, the total utility cost for processing the m-th user's task at the l-th satellite can be expressed as

$$U_{m,pr}^{\mathrm{l}}(t) = \psi T_{m,pr}^{\mathrm{l}} + (1 - \psi) E_{m,pr}^{\mathrm{l}}(t). \tag{18}$$

The offloading decision from the BS depends on the total utility cost for each user's task. Therefore, the total utility cost for the m-th user's task can be calculated as

$$\begin{aligned} U_m^{\mathrm{tot}}(t) = {} & \psi T_{m,tx}^{\mathrm{bs}} + (1 - \psi) E_{m,tx}^{\mathrm{bs}}(t) \\ & + \chi(t) \left(\psi T_{m,pr}^{\mathrm{bs}} + (1 - \psi) E_{m,pr}^{\mathrm{bs}}(t) \right) \\ & + (1 - \chi(t)) \Big(\psi T_{m,tx}^{\mathrm{l}} + (1 - \psi) E_{m,tx}^{\mathrm{l}}(t) \\ & + \psi T_{m,pr}^{\mathrm{l}} + (1 - \psi) E_{m,pr}^{\mathrm{l}}(t) \Big), \end{aligned} \tag{19}$$

where $\chi(t) \in [0, 1]$ is the task offloading fraction determined by the BS based on the utility cost U_m^{tot} of each user's task.

Moreover, the total delay for the m-th user's task can be expressed as

$$T_m^{\mathrm{tot}} = T_{m,tx}^{\mathrm{bs}} + \chi(t) T_{m,pr}^{\mathrm{bs}} + (1 - \chi(t)) \left(T_{m,tx}^{\mathrm{l}} + T_{m,pr}^{\mathrm{l}} \right), \tag{20}$$

D. Problem Formulation

In this paper, we aim to minimize the total utility cost for all $\mathcal{M}$ users during task offloading, which is expressed as

$$\begin{aligned} \Omega(\mathbf{b}, \boldsymbol{\chi}, \boldsymbol{\eta}) = \sum_{m=1}^{M} & U_{m,tx}^{\mathrm{bs}}(t) + \chi_m(t) \left(U_{m,pr}^{\mathrm{bs}}(t) \right) \\ & + (1 - \chi_m(t)) \left(U_{m,tx}^{\mathrm{l}}(t) + U_{m,pr}^{\mathrm{l}}(t) \right), \end{aligned} \tag{21}$$

where $\mathbf{b} \triangleq \{b_m(t)\}_{\forall m}$, $\boldsymbol{\chi} \triangleq \{\chi_m(t)\}_{\forall m}$, $\boldsymbol{\eta} \triangleq \{\eta_m^{(1)}(t), \eta^{(2)}(t)\}_{\forall m}$ are the optimization variables.

Then, the optimization problem is formulated as follows:

$$\textbf{\textit{(P1):}} \quad \min_{\mathbf{b}, \boldsymbol{\chi}, \boldsymbol{\eta}} \quad \Omega(\mathbf{b}, \boldsymbol{\chi}, \boldsymbol{\eta}), \tag{22a}$$

$$\text{s.t.} \quad 0 \leq b_m(t) \leq 1, \quad \forall m, \sum_{m=1}^{M} b_m(t) \leq 1, \tag{22b}$$

$$0 \leq \chi_m(t) \leq 1, \quad \forall m, \tag{22c}$$

$$0 \leq \eta_m^{(1)}(t) \leq 1, \quad \forall m, \tag{22d}$$

$$0 \leq \eta_m^{(2)}(t) \leq 1, \quad \forall m \tag{22e}$$

$$T_m^{\mathrm{tot}} \leq T_m^{\mathrm{max}}, \quad \forall m, \tag{22f}$$

$$F^{\mathrm{bs}} \geq \sum_{m=1}^{M} \eta_m^{(1)}(t) F^{\mathrm{bs}}, \; F^{l} \geq \sum_{m=1}^{M} \eta_m^{(2)}(t) F^{\mathrm{l}}, \tag{22g}$$

As outlined in (22), constraint (22b) ensures that the bandwidth allocation coefficient for each user $b_m(t)$ lies between 0 and 1 for all $m \in \mathcal{M}$ and and total bandwidth allocation does not exceed 1. Constraint (22c) requires the offloading fraction $\chi_m(t)$ to also be between 0 and 1 for all $m \in \mathcal{M}$. Constraints (22d) and (22e) mandate that the computation power allocation coefficients $\eta_m^{(1)}(t)$ at the BS and $\eta_m^{(2)}(t)$ at the satellite, respectively, must be between 0 and 1 for all $m \in \mathcal{M}$. Furthermore, constraint (22f) ensures that the delay for each user's task does not exceed the maximum tolerable delay. Constraint (22g) ensures that the computational resources at the BS and the l-th LEO satellite meet the required computational power allocation.

III. Deep Reinforcement Learning Based Solution

The optimization problem formulated in (22) involves optimizing continuous variables such as bandwidth allocation $b(t)$, offloading fraction $\chi(t)$, and computational power at both the BS $\eta^{(1)}(t)$ and the l-th LEO satellite $\eta^{(2)}(t)$. These variables interact in a complex, non-linear manner, creating a challenging optimization landscape. Traditional methods struggle to address such problems due to the dynamic and continuous nature of the decision variables. The complexity is further compounded by the need for real-time decisions, making it difficult to obtain an optimal solution using conventional approaches. Given these challenges, the problem cannot be efficiently solved using standard optimization techniques, which may not adapt well to the time-varying nature of the system. Therefore, instead of relying on conventional approaches, we propose employing a DRL approach. DRL is well-suited for handling changes in the environment, enabling it to effectively minimize the total utility cost in this context. This approach allows for dynamic optimization, providing a practical and scalable solution to the problem described.

Therefore, we propose DRL-based solutions to address the formulated optimization problem (22). In (22), the optimization variables $b_m(t)$, $\chi_m(t)$, $\eta_m^{(1)}(t)$, and $\eta_m^{(2)}(t)$ are continuous variables, making the DDPG and TD3 frameworks suitable choices due to their support for continuous action spaces. To apply the DRL framework for solving (22), we first need to formulate it as a Markov decision process (MDP), characterized by a 3-tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{R}\}$. The state space $\mathcal{S}$ includes all possible states of the system, while the action space $\mathcal{A}$ encompasses all possible actions, and the reward function $\mathcal{R}$ specifies the immediate reward received after transitioning from one state to another due to an action. At each time step t, the agent observes the state $s(t)$, executes the action $a(t)$, and receives the reward $r(t)$.

A. MDP Formulation

1) State Space: The state space $s(t)$ at time t consists of the current total utility costs for all users, represented as $s(t) = \{U_1^{\mathrm{tot}}(t), U_2^{\mathrm{tot}}(t), \ldots, U_M^{\mathrm{tot}}(t)\}$. The utility cost $U_m^{\mathrm{tot}}(t)$ for each user m is calculated based on the current system conditions and the immediate action decisions, without dependence on previous actions. It reflects the current transmission delays, processing delays, and energy consumption resulting from the current resource allocations and offloading decisions. By focusing on the present utility costs in the state space, the agent can optimize resource allocation effectively in real-time, adhering to the Markov property where the next state depends only on the current state and action.

2) Action Space: The action space $a(t)$ consists of key decision variables that the agent controls to optimize system performance. Specifically, it includes bandwidth allocation for each user, task fraction, and computational resource allocation at BS and LEO satellites. This action space is represented as $a(t) = \{b_m(t), \chi_m(t), \eta_m^{(1)}(t), \eta_m^{(2)}(t)\}$, where $b_m(t)$ denotes the bandwidth allocation for each user, ensuring efficient communication, $\chi_m(t)$ represents the task fraction, determining how tasks are distributed among available resources, and $\eta_m^{(1)}(t)$ and $\eta_m^{(2)}(t)$ correspond to the computational resource allocations at the BS and LEO satellites, respectively. These elements are continuous variables that lie between 0 and 1, allowing for fine-grained control over resource distribution to optimize system efficiency.

3) Reward: The reward function $r(t)$ plays a pivotal role in aligning the DRL model's objectives with the optimization goals of the MDP. In reinforcement learning, the reward function is designed to quantify the immediate reward received by the agent following an action $a(t)$ taken at state $s(t)$. The primary objective is to maximize the cumulative rewards over time. Consequently, the reward function can be formulated as the inverse of the system's total utility cost, as expressed in the following equation:

$$r(t) = \left(\sum_{m=1}^{M} \left(U^{\text{bs}}_{m,tx} + \chi(t) U^{\text{bs}}_{m,pr} \right.\right.$$
$$\left.\left. + (1-\chi(t)) \left(U^{1}_{m,tx} + U^{1}_{m,pr} \right) \right) \right)^{-1}. \tag{23}$$

The utility cost for each user m is adjusted by incorporating a penalty ϵ which is denoted as $\epsilon = w(T^{\text{tot}}_m - T^{\text{max}}_m)$ where w is scaling factor. This adjustment is represented by $U^{\text{tot}}_m = U^{\text{tot}}_m + \epsilon$. This formulation ensures that a reduction in the total utility cost of the system, while satisfying the delay constraint, results in an increase in the reward $r(t)$. Conversely, if the delay constraint is violated, the penalty increases the utility cost, thereby reducing the reward. This approach encourages the DRL model to minimize the overall system costs while satisfying the required constraints, ultimately guiding the agent toward optimal behavior. It is important to note that other constraints represent physical limitations and are enforced directly within the environment, ensuring they cannot be violated during learning.

B. LSTM-Enhanced DDPG Algorithm

To enhance the optimization capabilities within the formulated MDP framework, we employ the DDPG algorithm, a state-of-the-art model-free, actor-critic DRL technique. It involves two primary networks: the actor network $\mu(s|\theta^{\mu})$, which generates actions given states, and the critic network $Q(s,a|\theta^{Q})$, which evaluates these actions. The actor network includes an LSTM layer that processes each state individually. Although the LSTM is typically used for sequence processing, in this implementation, it operates on a single time step, utilizing its internal hidden state dynamics for potentially enhanced learning. This integration ensures that the formulated MDP is not affected. To stabilize training, target networks μ' and Q' are used, initialized with the parameters $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{Q'} \leftarrow \theta^{Q}$. A replay buffer R stores experiences $\{s(t), a(t), r(t), s(t+1)\}$, allowing random mini-batch sampling to break correlation between experiences. During each step, actions $a(t)$ are generated with added noise to encourage exploration which can be formulated as [28]:

$$a(t) = \mu(s(t)|\theta^{\mu}) + Z(t), \tag{24}$$

where $Z(t)$ represents the exploration noise added to the action. These actions are executed to receive rewards $r(t)$ and next states $s(t+1)$, which are stored in the replay buffer R. The added noise $Z(t)$ helps in exploring the state space more thoroughly, preventing the policy from getting stuck in local optima. For training, target values are computed using the target networks y_i which can be denoted as [28]:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}), \tag{25}$$

where y_i represents the target Q-value, r_i is the reward, and γ is the discount factor. This equation provides the expected return for a given state-action pair, accounting for future rewards as estimated by the target critic network Q'. The critic network parameters θ^{Q} are updated by minimizing the loss L which can be expressed as [28]:

$$L = \frac{1}{S} \sum_{i} \left(y_i - Q(s_i, a_i|\theta^{Q}) \right)^2, \tag{26}$$

where L is the loss function, S is the mini-batch size, and $Q(s_i, a_i|\theta^{Q})$ is the predicted Q-value. Minimizing this loss aligns the critic's predictions with the more stable target values, improving the critic's accuracy in estimating the value of actions. The actor network parameters θ^{μ} are updated using the policy gradient $\nabla_{\theta^{\mu}} J$ which can be denoted as [28]:

$$\nabla_{\theta^{\mu}} J = \frac{1}{S} \sum_{i} \left(\nabla_a Q(s_i, a_i|\theta^{Q})\big|_{a_i=\mu(s_i|\theta^{\mu})} \nabla_{\theta^{\mu}} \mu(s_i|\theta^{\mu}) \right), \tag{27}$$

where $\nabla_{\theta^{\mu}} J$ is the gradient of the objective function with respect to the actor network parameters. This update aims to maximize the expected return by adjusting the policy μ in the direction that increases the value estimated by the critic. Finally, the target networks are softly updated to ensure stable learning [28]:

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}, \tag{28}$$
$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}, \tag{29}$$

where τ is a small coefficient that determines the rate of the soft update. This method gradually updates the target networks, ensuring they slowly track the learned networks, thereby providing stability to the learning process. The detailed algorithm of the proposed LSTM-enhanced DDPG solution is given in Algorithm 1.

C. LSTM-Enhanced TD3 Algorithm

TD3 is a variant of the DDPG algorithm developed to overcome overestimation bias. Unlike DDPG, which uses a single critic network, TD3 employs two critic networks (Q_1 and Q_2) to provide more robust value estimates. For target value computation, TD3 uses the minimum value from the two target critic networks [33]:

$$y_i = r_i + \gamma \min(Q'_1(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'_1})$$
$$Q'_2(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'_2})). \tag{30}$$

Both critic networks are updated by minimizing their respective losses, L, which can be denoted as [33]:

$$L = \frac{1}{S} \sum_{i} \left(y_i - Q_j(s_i, a_i|\theta^{Q_j}) \right)^2 \quad \text{for } j = 1, 2. \tag{31}$$

The actor network updates are delayed and based on the first critic network. An LSTM layer has been added to the actor network, similar to the LSTM-enhanced-DDPG implementation. The gradient $\nabla_{\theta^{\mu}} J$ used to update the actor network parameters is given by [33]:

$$\nabla_{\theta^{\mu}} J = \frac{1}{S} \sum_{i} \left(\nabla_a Q_1(s_i, a_i|\theta^{Q_1})\big|_{a_i=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s_i|\theta^{\mu}) \right). \tag{32}$$

Additionally, the target networks for both critics are softly updated [33]:

$$\theta^{Q'_1} \leftarrow \tau\theta^{Q_1} + (1-\tau)\theta^{Q'_1}, \tag{33}$$
$$\theta^{Q'_2} \leftarrow \tau\theta^{Q_2} + (1-\tau)\theta^{Q'_2}. \tag{34}$$

The detailed algorithm of the proposed LSTM-enhanced TD3 solution is presented in Algorithm 2.

Algorithm 1 : Proposed LSTM-Enhanced DDPG Algorithm for Solving (22)

```
Initialize the environment with its specified parameters.
Initialize the actor network $\mu(s|\theta^{\mu})$ with an LSTM layer and parameters $\theta^{\mu}$ and the critic network $Q(s, a|\theta^{Q})$ with parameters $\theta^{Q}$.
Initialize the target networks $\mu'$ and $Q'$ with $\theta^{\mu'} \leftarrow \theta^{\mu}$ and $\theta^{Q'} \leftarrow \theta^{Q}$.
Set up a replay buffer $\mathcal{R}$.
for each episode do
    for $t = 1, 2, \ldots, T$ do
        Process the state $s(t)$ through the LSTM layer in the actor network to generate the action $a(t)$;
        Generate an action with added noise: $a(t) = \mu(s(t)|\theta^{\mu}) + Z(t)$;
        Execute the action $a(t)$, then receive reward $r(t)$ and the next state $s(t+1)$;
        Store $\{s(t), a(t), r(t), s(t+1)\}$ in the replay buffer;
        Randomly sample a batch $S$ from the replay buffer;
        Compute target values using target networks and store in $y_i$ using equation (25);
        Update the critic network parameters $\theta^{Q}$ by minimizing the loss $L$ using equation (26);
        Compute policy gradients and update the actor network parameters $\theta^{\mu}$ using gradient $\nabla_{\theta^{\mu}} J$ from equation (27);
        Softly update the target networks using the soft update rule with coefficient $\tau$ according to equations (28) and (29);
    end for
end for
```

Algorithm 2 : Proposed LSTM-Enhanced TD3 Algorithm for Solving (22)

```
Initialize the environment with its specified parameters.
Initialize the actor network $\mu(s|\theta^{\mu})$, incorporating an LSTM layer, and the critic networks $Q_1(s, a|\theta^{Q_1})$ and $Q_2(s, a|\theta^{Q_2})$ with parameters $\theta^{\mu}$, $\theta^{Q_1}$, and $\theta^{Q_2}$.
Initialize the target networks $\mu'$, $Q'_1$, and $Q'_2$ with $\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{Q'_1} \leftarrow \theta^{Q_1}$, and $\theta^{Q'_2} \leftarrow \theta^{Q_2}$.
Set up a replay buffer $\mathcal{R}$.
for each episode do
    for $t = 1, 2, \ldots, T$ do
        Generate an action with noise $a(t) = \mu(s(t)|\theta^{\mu}) + Z(t)$;
        Execute the action $a(t)$, then receive reward $r(t)$ and the next state $s(t+1)$;
        Store $\{s(t), a(t), r(t), s(t+1)\}$ in the replay buffer;
        Randomly sample a batch $S$ from the replay buffer;
        Compute target value $y_i$ using (30);
        Update the critic network's parameters $\theta^{Q_1}$ and $\theta^{Q_2}$ by minimizing the loss $L$ (31);
        if update actor network every $d$ steps then
            Update the actor network parameters $\theta^{\mu}$ using the gradient (32);
            Update the target networks using the soft update rule with coefficient $\tau$ as given in (33) and (34);
        end if
    end for
end for
```

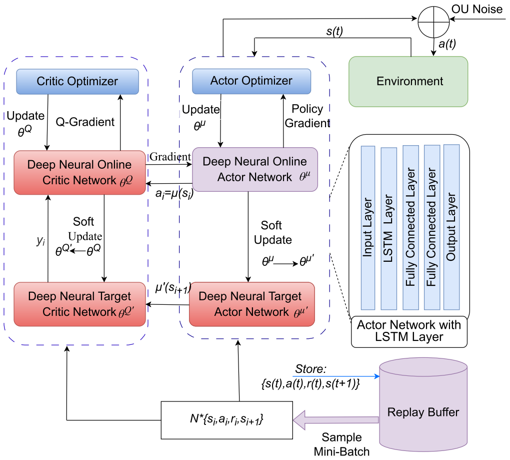

Fig. 3. LSTM-enhanced DDPG framework.

The implementation of the proposed DRL based solution is illustrated in Fig. 3. We evaluate system performance using two DRL frameworks: LSTM-enhanced DDPG and TD3. Both frameworks have been modified by incorporating an LSTM layer in their actor networks. This LSTM layer operates on a single time step, maintaining the integrity of the MDP formulation by utilizing its internal hidden state dynamics for potentially enhanced learning. While not leveraging the full capabilities of LSTM, the layer functions as a more complex feed-forward layer without introducing dependencies on past or future states. As shown in the Fig. 3, the actor network of the LSTM-enhanced DDPG and TD3 consists of one LSTM layer and two fully connected hidden layers, each with 256 neurons. The LSTM layer processes the input state, and its output is passed through the fully connected layers, where ReLU activation functions introduce non-linearity. The final output layer generates the action, scaled by a sigmoid activation function to ensure it falls within the desired range. In both the LSTM-enhanced DDPG and TD3 frameworks, the critic network architecture is similar, consisting of two fully connected hidden layers with 256 neurons each, followed by a final output layer. The critic network takes the state and action as inputs, concatenates them, and passes the result through the hidden layers with ReLU activation functions.

IV. Numerical Results and Discussions

A. Simulation Settings

This subsection presents the settings of parameters for the implementation of the proposed DRL-based solutions and simulations. Firstly, the replay buffer of the DDPG algorithm can store up to 100,000 experience transitions and 32 mini-batches at a time. The actor learning rate is set to 0.0001 and the critic learning rate to 0.001. The discount factor γ is 0.99, and the soft update parameter τ is 0.001. The temporal correlated noise adopts the Ornstein Uhlenbeck process with mean reversion rate of 0.15 and volatility of 0.2. We compare the total rewards

TABLE II

SIMULATION PARAMETERS [18], [27]

Parameter	Value
Number of users, M	10
Number of RIS elements, N	16 (4x4)
Number of antennas at BS, K	8
Earth radius	6371 km
Distance to LEO satellites orbit, r	500 km
Distance from m-th user to UAV , $d_{m,ucr}$	100 m
Distance from UAV to BS, $d_{ucr,0}$	100 m
Path loss exponent, $\delta^{(1)}$, [min, max], $\delta^{(2)}$	[3.65, 3.75] , [2]
BS MEC computation power, F^{bs}	4 GHz
Satellite MEC computation power, F^{l}	2 GHz
Transmission power of each user, p_m^{u}	20 dBm
Task size, g_m	[5, 6] Mbits
Task complexity, q_m	[500, 600] Mcycles
Transmit power of BS, p_0	40 dBm
System bandwidth, B	50 MHz
Noise power	-110 dBm/Hz
Maximum delays for each task, T_m^{max}	[5-6] s
Energy coefficient (BS and satellite), ω^{bs} and ω^{l}	1e-27
Carrier frequency, f_c	2 GHz
Balancing factor, ψ	0.5

over 1,000 testing episodes, with each episode consisting of 100 time slots. Thus, the total reward earned in an episode is the sum of the rewards earned across all 100 time slots [28], [31]. Moreover, the other system parameters are summarized in TABLE II.

Moreover, the weighting factor ψ can be selected between 0 and 1, where a higher ψ prioritizes delay minimization and a lower ψ focuses on energy efficiency. In our simulations, we choose $\psi = 0.5$ to balance both aspects equally [18].

B. Numerical Results

We evaluate the effectiveness of the proposed solutions in various aspects, including the convergence performance, the impacts of the system parameters, such as system bandwidth, number of users, task size, and task complexity.

1) Convergence Performance: We first analyse the convergence behavior of the proposed LSTM-enhanced DDPG and TD3 when compared with conventional DDPG and TD3 DRL algorithms, as shown in Fig. 4. The results clearly demonstrate that the LSTM-enhanced frameworks significantly outperform their conventional counterparts in terms of convergence speed. Specifically, the LSTM-enhanced DDPG begins to converge around episode 100, which is approximately 44.44% faster than the conventional DDPG, which starts converging around episode 180. This improvement highlights the efficiency gained by incorporating LSTM, which appears to enhance the learning process even within the context of single time step decisions.

In the TD3 framework, the LSTM-enhanced TD3 exhibits an even more pronounced acceleration, showing a 73.81% faster convergence compared to the conventional version, with convergence starting around episode 110 instead of 420. This substantial reduction in convergence time suggests that the combination of LSTM with TD3 not only addresses the instability commonly observed in standard TD3 but also significantly reduces the delay in learning effective policies.

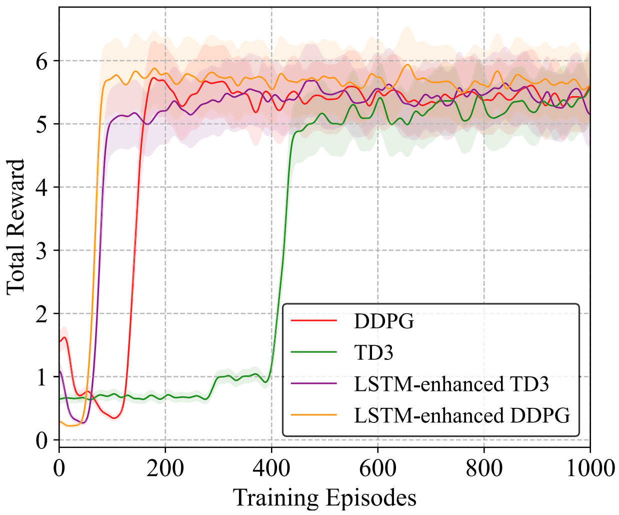

Fig. 4. Convergence Performance.

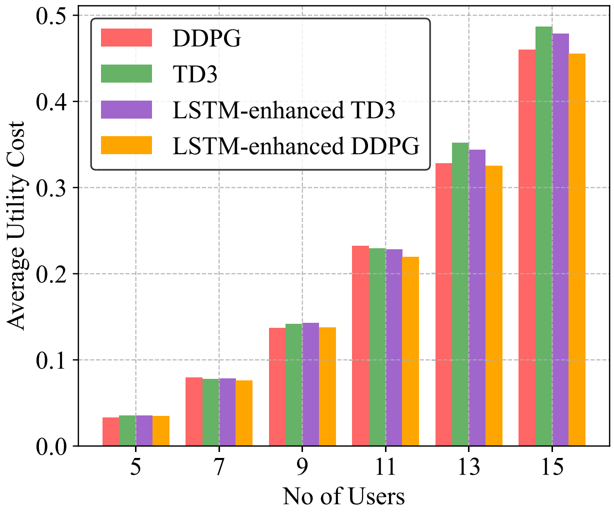

Fig. 5. Average utility cost for different number of users.

These results indicate that the LSTM integration enhances both the speed and stability of convergence, making the proposed methods highly effective for this proposed MDP.

2) Performance of the Proposed Algorithms With Varying Number of Users: We next compare the effect of varying the number of users on the average utility cost of the system. Fig. 5 shows the average utility cost for different numbers of users across all algorithms (DDPG, TD3, LSTM-enhanced DDPG, and LSTM-enhanced TD3). As the number of users increases, the utility cost rises, reflecting the growing resource demands. For smaller user counts, the differences between the algorithms are minimal, with LSTM-enhanced DDPG showing a slight advantage by achieving the lowest cost. As the number of users grows, performance differences become more pronounced. LSTM-enhanced DDPG consistently outperforms the other methods, maintaining the lowest utility costs. LSTM-enhanced TD3 also outperforms conventional TD3, though the improvement is less substantial compared to LSTM-enhanced DDPG. Overall, the results indicate that LSTM-enhanced algorithms, especially LSTM-enhanced DDPG, provide significant benefits in managing utility costs as user numbers increase.

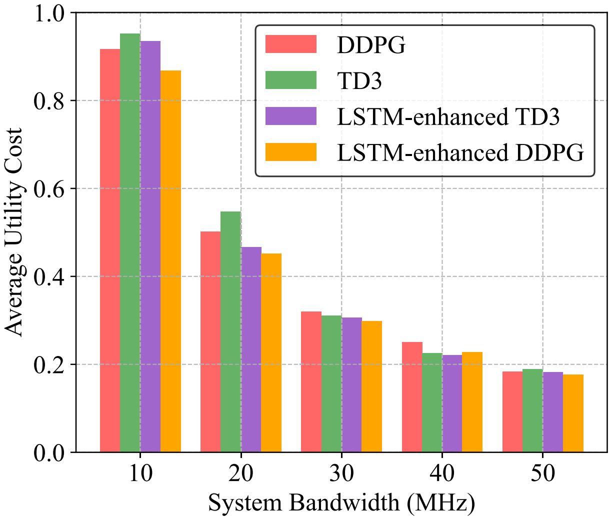

Fig. 6. Average utility cost for different system bandwidth.

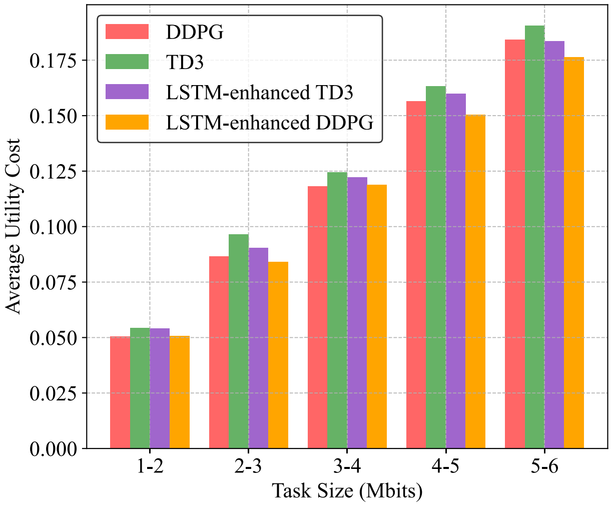

Fig. 7. Average utility cost for different task sizes.

This makes them particularly effective in scenarios with higher demand.

3) Performance of the Proposed Algorithms With Varying System Bandwidth: Next, the results depicted in Fig. 6 illustrates how the average utility cost responds to varying system bandwidths. As expected, an increase in system bandwidth generally leads to a decrease in utility costs, reflecting more efficient resource management. Initially, at lower bandwidths, all algorithms incur relatively high costs, with LSTM-enhanced DDPG demonstrating the most efficient performance. As bandwidth expands, a notable reduction in utility costs is observed, with LSTM-enhanced DDPG consistently leading in efficiency. Interestingly, while LSTM-enhanced TD3 also shows improvements over the conventional TD3, the difference between the enhanced and conventional algorithms becomes less distinct at intermediate bandwidth levels. However, as the bandwidth further increases, the utility costs for all models continue to decline, although with a narrowing performance gap between the LSTM-enhanced and conventional algorithms. Despite this convergence at higher bandwidths, the LSTM-enhanced models, particularly LSTM-enhanced DDPG perform well.

4) Performance of the Algorithms With Varying Task Sizes: In this subsection, we evaluate how the average utility cost varies with task sizes. Fig. 7 shows the average utility cost across different task sizes. A clear trend emerges: utility costs generally increase as task size grows. This pattern is consistent across all algorithms, highlighting the increased resource demand of larger tasks. For smaller task sizes, utility costs remain low. LSTM-enhanced DDPG demonstrates slightly better efficiency compared to other methods. As task sizes increase, differences between the algorithms become more noticeable. LSTM-enhanced DDPG consistently maintains lower utility costs, indicating better resource allocation as task demands grow. LSTM-enhanced TD3 also performs better than conventional TD3, though the improvement is modest. When tasks reach the largest sizes, all algorithms show a noticeable rise in average utility costs. However, LSTM-enhanced DDPG continues to perform most efficiently, demonstrating robustness in handling larger and more resource-intensive tasks

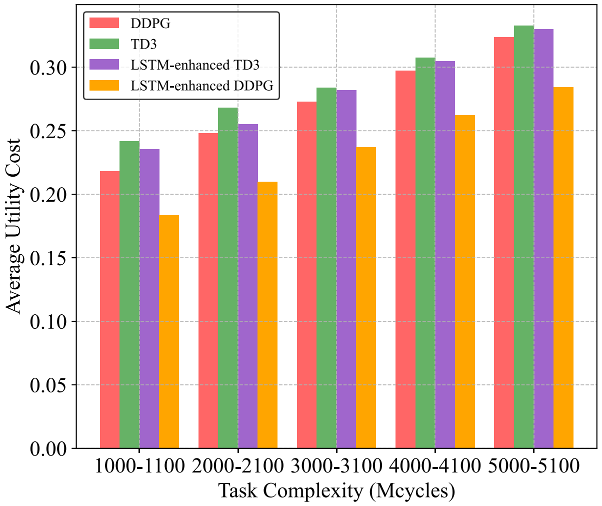

Fig. 8. Average utility cost for different task complexities.

5) Performance of the Algorithms With Varying Task Complexities: Next, we evaluate the average utility cost with task's complexities. Fig. 8 shows a consistent upward trend as task complexity increases. This trend is observed across all algorithms, which is expected as more complex tasks generally demand more resources, resulting in higher utility costs. At lower levels of task complexity, LSTM-enhanced DDPG demonstrates a clear advantage, achieving the lowest utility costs among all methods. As the task complexity increases, the gap between LSTM-enhanced DDPG and the other algorithms remains evident, with LSTM-enhanced DDPG consistently maintaining lower utility costs. This indicates that it manages the increased computational demands more efficiently than the other approaches. In contrast, LSTM-enhanced TD3, while performing better than conventional TD3, shows less of an improvement relative to LSTM-enhanced DDPG. As the task complexity reaches its highest levels, all algorithms experience a noticeable rise in utility costs, however LSTM-enhanced DDPG continues to outperform the others, suggesting its robustness in managing highly complex tasks. This analysis highlights the varying effectiveness of the algorithms as task

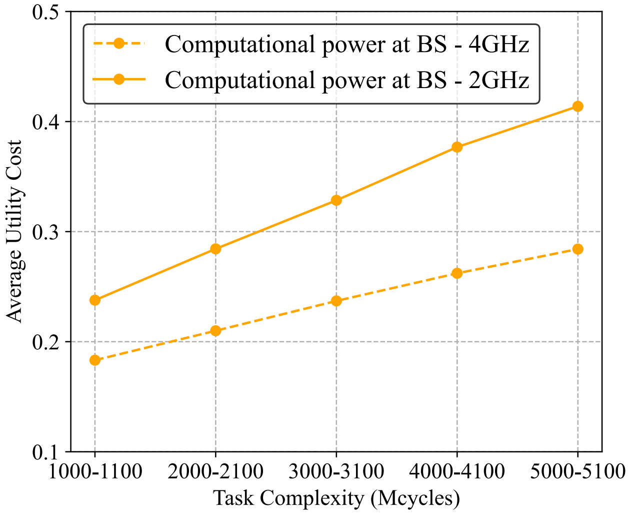

Fig. 9. Average utility cost for varying task complexities at different computational power levels at the BS with LSTM-enhanced DDPG.

complexity increases, with LSTM-enhanced DDPG showing superior performance in handling more complex tasks.

6) Performance of the Proposed Algorithms With Varying Task Complexities at Different Computational Power Levels at BS: Furthermore, the average utility cost as a function of task complexity under different computational power settings at the BS, as depicted in Fig. 9, highlights the impact of computational power on the efficiency of the LSTM-enhanced DDPG algorithm. The results clearly show that, for both 2 GHz and 4 GHz computational power at the BS, the average utility cost increases with task complexity. This is consistent with the expectation that more complex tasks require more resources, leading to higher costs. However, a key observation is the difference in utility costs between the two computational power settings. At every level of task complexity, the average utility cost is consistently lower when the computational power is 4 GHz compared to 2 GHz. This indicates that higher computational power at the BS allows for more efficient processing, thereby reducing the overall average utility cost associated with completing the tasks. The gap between the average utility costs at 2 GHz and 4 GHz widens as task complexity increases, suggesting that the advantage of higher computational power becomes more pronounced for more complex tasks. This analysis demonstrates the importance of computational power at the BS in managing utility costs, particularly as task complexity increases. The LSTM-enhanced DDPG algorithm benefits significantly from increased computational power, as it is able to handle more complex tasks more efficiently.

V. Conclusion and Future Work

This paper introduced a novel architecture for integrated terrestrial and non-terrestrial networks, leveraging MEC and OEC with UAV-carried RIS in an intelligent autonomous transportation system. The proposed architecture was designed to minimize average total system utility costs through the optimal allocation of bandwidth to users, allocation of computational powers at the BS and satellites, and determination of offloading fractions. To address the complex challenges of resource management within this architecture, we developed a comprehensive optimization strategy that leverages the strengths of LSTM-enhanced DDPG and TD3 algorithms. These algorithms are employed to effectively manage the distribution of computational tasks between terrestrial and non-terrestrial components, ensuring efficient utilization of available resources while adhering to stringent performance requirements. The simulation results clearly showed that the LSTM-enhanced algorithms significantly outperform conventional DDPG and TD3, and LSTM-enhanced DDPG performs best out of all algorithms, leading to substantial improvements in system performance and average utility cost efficiency. These findings underscore the effectiveness of our proposed architecture and optimization strategies in managing the complexities of an integrated terrestrial and non-terrestrial network environment.

Future extensions of this work could explore several avenues to further enhance the proposed framework. One potential direction involves optimizing the selection of satellites for task offloading, which could lead to significant improvements in overall system efficiency. Another area worth investigating is the integration of non-orthogonal multiple access into the network architecture. This integration could increase spectral efficiency and accommodate a higher density of users, thereby enhancing the network's overall performance. Additionally, further exploration of different DRL algorithms, particularly through the expansion into multi-agent systems where multiple learning agents operate within a shared environment, could result in more advanced and efficient resource management strategies.

References

[1] K. Trichias, A. Kaloxylos, and C. Willcock, "6G global landscape: A comparative analysis of 6G targets and technological trends," in *Proc. Joint Eur. Conf. Netw. Commun.; 6G Summit (EuCNC/6G Summit)*, Gothenburg, Sweden, Jun. 2024, pp. 1–6.

[2] S. Yrjola, M. Matinmikko-Blue, and P. Ahokangas, "Developing 6G visions with stakeholder analysis of 6G ecosystem," in *Proc. Joint Eur. Conf. Net. Commun. 6G Summit (EuCNC/6G Summit)*, Gothenburg, Sweden, Jun. 2023.

[3] T. T. Bui, A. Masaracchia, V. Sharma, O. Dobre, and T. Q. Duong, "Impact of 6G space-air-ground integrated networks on hard-to-reach areas: Tourism, agriculture, education, and indigenous communities," *EAI Endorsed Trans. Tour. Technol. Intell.*, vol. 1, no. 1, pp. 1–8, Sep. 2024.

[4] Q. Dong, X. Xu, S. Han, R. Liu, and X. Zhang, "DDPG-based task offloading in satellite-terrestrial collaborative edge computing networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Rome, Italy, May 2023, pp. 1541–1546.

[5] Y. Qian, J. Xu, S. Zhu, W. Xu, L. Fan, and G. K. Karagiannidis, "Learning to optimize resource assignment for task offloading in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 26, no. 6, pp. 1303–1307, Jun. 2022.

[6] K. Wei, Q. Tang, J. Guo, M. Zeng, Z. Fei, and Q. Cui, "Resource scheduling and offloading strategy based on LEO satellite edge computing," in *Proc. IEEE 94th Veh. Technol. Conf. (VTC-Fall)*, Norman, OK, USA, Sep. 2021, pp. 1–6.

[7] A. Umar, S. A. Hassan, and H. Jung, "Computation offloading and resource allocation in NOMA-MEC enabled aerial-terrestrial networks exploiting mmWave capabilities for 6G," in *Proc. ICC-IEEE Int. Conf. Commun.*, Denver, CO, USA, Jun. 2024, pp. 3256–3261.

[8] G. Sun et al., "Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8582–8600, Sep. 2024.

[9] Y. Gao, F. Lu, P. Wang, W. Lu, Y. Ding, and J. Cao, "Resource optimization of secure data transmission for UAV-relay assisted maritime MEC system," in *Proc. ICC-IEEE Int. Conf. Commun.*, Rome, Italy, May 2023, pp. 3345–3350.

[10] M. T. Nguyen et al., "Real-time optimized clustering and caching for 6G satellite-UAV-terrestrial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 3009–3019, Jun. 2024.

[11] K. K. Nguyen, S. R. Khosravirad, D. B. da Costa, L. D. Nguyen, and T. Q. Duong, "Reconfigurable intelligent surface-assisted multi-UAV networks: Efficient resource allocation with deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 358–368, Apr. 2022.

[12] M. Zhang, Z. Su, Q. Xu, Y. Qi, and D. Fang, "Energy-efficient task offloading in UAV-RIS-assisted mobile edge computing with NOMA," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Vancouver, BC, Canada, May 2024, pp. 1–6.

[13] H. Yu, H. D. Tuan, A. A. Nasir, T. Q. Duong, and H. V. Poor, "Joint design of reconfigurable intelligent surfaces and transmit beamforming under proper and improper Gaussian signaling," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2589–2603, Nov. 2020.

[14] J. Yuan, G. Chen, M. Wen, D. Wan, and K. Cumanan, "Security-reliability tradeoff in UAV-carried active RIS-assisted cooperative networks," *IEEE Commun. Lett.*, vol. 28, no. 2, pp. 437–441, Feb. 2024.

[15] A. S. Abdalla, T. F. Rahman, and V. Marojevic, "UAVs with reconfigurable intelligent surfaces: Applications, challenges, and opportunities," 2020, *arXiv:2012.04775*.

[16] Y. Lin et al., "Satellite-MEC integration for 6G Internet of Things: Minimal structures, advances, and prospects," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 3886–3903, 2024.

[17] M. Mukherjee, V. Kumar, A. Lat, M. Guo, R. Matam, and Y. Lv, "Distributed deep learning-based task offloading for UAV-enabled mobile edge computing," *IEEE Commun. Lett.*, vol. 28, no. 2, pp. 437–441, Feb. 2024.

[18] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, "Computation offloading and resource allocation in MEC-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21478–21491, Nov. 2022.

[19] N. Sharma, A. Ghosh, R. Misra, and S. K. Das, "Deep meta Q-learning based multi-task offloading in edge-cloud systems," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2583–2597, Apr. 2024.

[20] F. Jiang, L. Dong, K. Wang, K. Yang, and C. Pan, "Distributed resource scheduling for large-scale MEC systems: A multiagent ensemble deep reinforcement learning with imitation acceleration," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6597–6610, May 2022.

[21] K. Zheng, G. Jiang, X. Liu, K. Chi, X. Yao, and J. Liu, "DRL-based offloading for computation delay minimization in wireless-powered multi-access edge computing," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1755–1770, Mar. 2023.

[22] D.-B. Ha, V.-T. Truong, and Y. Lee, "Performance analysis for RF energy harvesting mobile edge computing networks with SIMO/MISO-NOMA schemes," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 8, no. 27, Jun. 2021, Art. no. 169425.

[23] F. Zhang, G. Han, L. Liu, M. Martinez-Garcia, and Y. Peng, "Deep reinforcement learning based cooperative partial task offloading and resource allocation for IIoT applications," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2991–3004, Sep. 2023.

[24] N. N. Ei, M. Alsenwi, Y. K. Tun, Z. Han, and C. S. Hong, "Energy-efficient resource allocation in multi-UAV-assisted two-stage edge computing for beyond 5G networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16421–16432, Sep. 2022.

[25] H. Zeng, X. Li, S. Bi, and X. Lin, "Delay-sensitive task offloading with D2D service-sharing in mobile edge computing networks," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 607–611, Mar. 2022.

[26] T. Yang, R. Chai, and L. Zhang, "Latency optimization-based joint task offloading and scheduling for multi-user MEC system," in *Proc. 29th Wireless Opt. Commun. Conf. (WOCC)*, Newark, NJ, USA, May 2020, pp. 1–6.

[27] D. Van Huynh, Y. Li, A. Masaracchia, T. Hoang, and T. Q. Duong, "Optimal resource allocation for 6G UAV-enabled mobile edge computing with mission-critical applications," in *Proc. IEEE Int. Conf. Metaverse Comput., Netw. Appl. (MetaCom)*, Kyoto, Japan, Jun. 2023, pp. 720–723.

[28] J. Wang, Y. Wang, P. Cheng, K. Yu, and W. Xiang, "DDPG-based joint resource management for latency minimization in NOMA-MEC networks," *IEEE Commun. Lett.*, vol. 27, no. 7, pp. 1814–1818, Jul. 2023.

[29] S. Chouikhi, M. Esseghir, and L. Merghem-Boulahia, "Energy-efficient computation offloading based on multiagent deep reinforcement learning for industrial Internet of Things systems," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 12228–12239, Apr. 2024.

[30] H. Li, K. D. R. Assis, S. Yan, and D. Simeonidou, "DRL-based long-term resource planning for task offloading policies in multiserver edge computing networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 4151–4164, Dec. 2022.

[31] Q. Liu, H. Zhang, X. Zhang, and D. Yuan, "Improved DDPG based two-timescale multi-dimensional resource allocation for multi-access edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 73, no. 6, pp. 9153–9158, Jun. 2024.

[32] X. Deng, J. Yin, P. Guan, N. N. Xiong, L. Zhang, and S. Mumtaz, "Intelligent delay-aware partial computing task offloading for multiuser industrial Internet of Things through edge computing," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 2954–2966, Feb. 2023.

[33] J. Xu, B. Ai, L. Wu, Y. Zhang, W. Wang, and H. Li, "Deep reinforcement learning for computation rate maximization in RIS-enabled mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 73, no. 7, pp. 10862–10866, Jul. 2024.

[34] X. Ren, X. Chen, L. Jiao, X. Dai, and Z. Dong, "Joint optimization of trajectory, caching and task offloading for multi-tier UAV MEC networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2024, pp. 1–6.

[35] Y. Gao, C. Li, and Z. Li, "Deep reinforcement learning-driven adaptive task offloading and resource allocation for UAV-assisted mobile edge computing," in *Proc. 27th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, Tianjin, China, May 2024, pp. 1004–1009.

[36] F. Chai, Q. Zhang, H. Yao, X. Xin, R. Gao, and M. Guizani, "Joint multi-task offloading and resource allocation for mobile edge computing systems in satellite IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 7783–7795, Feb. 2023.

[37] Y. Song, X. Li, H. Ji, and H. Zhang, "Energy-aware task offloading and resource allocation in the intelligent LEO satellite network," in *Proc. IEEE 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Kyoto, Japan, Sep. 2022, pp. 481–486.

[38] J. Wu, M. Jia, Q. Guo, and X. Gu, "Efficient resource management based on DQN in LEO satellite edge computing system," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 135–140.

[39] C. Yang, B. Liu, H. Li, B. Li, K. Xie, and S. Xie, "Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9884–9895, Sep. 2022.

[40] L. Zhao et al., "MESON: A mobility-aware dependent task offloading scheme for urban vehicular edge computing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4259–4271, May 2024.

[41] B. Hazarika, K. Singh, S. Biswas, and C.-P. Li, "DRL-based resource allocation for computation offloading in IoV networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8027–8038, Nov. 2022.

[42] C. Fang et al., "DRL-driven joint task offloading and resource allocation for energy-efficient content delivery in cloud-edge cooperation networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16195–16207, Dec. 2023.

[43] X. Peng et al., "Deep reinforcement learning for shared offloading strategy in vehicle edge computing," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2089–2099, Jun. 2023.

[44] N. Q. Hieu, D. T. Hoang, D. Niyato, P. Wang, D. I. Kim, and C. Yuen, "Transferable deep reinforcement learning framework for autonomous vehicles with joint radar-data communications," *IEEE Trans. Commun.*, vol. 70, no. 8, pp. 5164–5180, Aug. 2022.

[45] B. Yang, X. Cao, X. Li, C. Yuen, and L. Qian, "Lessons learned from accident of autonomous vehicle testing: An edge learning-aided offloading framework," *IEEE Wireless Commun. Lett.*, vol. 9, no. 8, pp. 1182–1186, Aug. 2020.

[46] T. Gong, L. Zhu, F. R. Yu, and T. Tang, "Edge intelligence in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 8919–8936, Sep. 2023.

[47] K. Xiong, S. Leng, X. Chen, C. Huang, C. Yuen, and Y. L. Guan, "Communication and computing resource optimization for connected autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12652–12663, Nov. 2020.